

Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”  
Завідувач кафедри кібербезпеки  
та програмного забезпечення  
д.т.н., професор  
\_\_\_\_\_ Олексій СМІРНОВ  
“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за другим (магістерським) рівнем вищої освіти**  
на тему  
**“Дослідження та програмна реалізація системи**  
**автоматизованого розподілу аудиторного фонду розкладу**  
**занять для факультетів”**

Виконав здобувач вищої освіти  
II курсу, групи КН-22М-2  
ОПП «Комп’ютерні науки»  
спеціальності 122 «Комп’ютерні науки»  
\_\_\_\_\_ Горшков В.О.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

Керівник проекту  
кандидат технічних наук, доцент  
\_\_\_\_\_ Коваленко А.С.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.  
Рецензент \_\_\_\_\_  
\_\_\_\_\_

Центральноукраїнський національний технічний університет  
Факультет Механіко-технологічний  
Кафедра Кібербезпеки та програмного забезпечення  
Рівень вищої освіти магістр  
Галузь знань 12 "Інформаційні технології"  
Спеціальність 122 "Комп'ютерні науки"  
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерні науки"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

## ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Горшкову Владиславу Олеговичу

(прізвище, ім'я, по батькові)

- |  |  |  |                            |   |   |  |  |  |                     |  |  |
|--|--|--|----------------------------|---|---|--|--|--|---------------------|--|--|
| 1. Тема роботи   | <i>Дослідження та програмна реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів</i>  |  |                            |   |   |  |  |  |                     |  |  |
| 2. Керівник роботи   | <i>Коваленко Анна Степанівна, канд. техн. наук, доцент</i><br>(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)<br>затверджені наказом вищого навчального закладу № 33-13 від 04.08.2023 року  |  |                            |   |   |  |  |  |                     |  |  |
| 3. Строк подання студентом роботи до захисту   | <i>10.12.2023 р.</i>   |  |                            |   |   |  |  |  |                     |  |  |
| 4. Мета та завдання випускної кваліфікаційної роботи:                                | <i>Метою розробки є дослідження та програмна реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів</i>   |  |                            |   |   |  |  |  |                     |  |  |
| 5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) | <table border="1"><tr><td><i>1. Призначення та область використання.</i></td><td><i>6. Наукова новизна.</i></td></tr><tr><td><i>2. Перегляд аналогічних існуючих систем.</i></td><td><i>7. Економічна ефективність розробленої програми.</i></td></tr><tr><td><i>3. Опис і обґрунтування проектних рішень.</i></td><td><i>8. Заходи з охорони праці та техніки безпеки.</i></td></tr><tr><td><i>4. Етапи програмування системи.</i></td><td><i>9. Висновки.</i></td></tr><tr><td><i>5. Впровадження системи в промислову експлуатацію</i></td><td></td></tr></table> | <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> | <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Економічна ефективність розробленої програми.</i> | <i>3. Опис і обґрунтування проектних рішень.</i> | <i>8. Заходи з охорони праці та техніки безпеки.</i> | <i>4. Етапи програмування системи.</i> | <i>9. Висновки.</i> | <i>5. Впровадження системи в промислову експлуатацію</i> |  |
| <i>1. Призначення та область використання.</i>                                       | <i>6. Наукова новизна.</i>   |  |                            |   |   |  |  |  |                     |  |  |
| <i>2. Перегляд аналогічних існуючих систем.</i>                                      | <i>7. Економічна ефективність розробленої програми.</i>  |  |                            |   |   |  |  |  |                     |  |  |
| <i>3. Опис і обґрунтування проектних рішень.</i>                                     | <i>8. Заходи з охорони праці та техніки безпеки.</i>   |  |                            |   |   |  |  |  |                     |  |  |
| <i>4. Етапи програмування системи.</i>   | <i>9. Висновки.</i>  |  |                            |   |   |  |  |  |                     |  |  |
| <i>5. Впровадження системи в промислову експлуатацію</i>                             |  |  |                            |   |   |  |  |  |                     |  |  |
| 6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)         |  |  |                            |   |   |  |  |  |                     |  |  |
| <i>Наукова новизна</i>   | <i>1 аркуш</i>   |  |                            |   |   |  |  |  |                     |  |  |
| <i>Структурна схема системи</i>  | <i>1 аркуш</i>   |  |                            |   |   |  |  |  |                     |  |  |
| <i>Функціональна схема системи</i>   | <i>1 аркуш</i>   |  |                            |   |   |  |  |  |                     |  |  |
| <i>Діаграма процесів</i>   | <i>1 аркуш</i>   |  |                            |   |   |  |  |  |                     |  |  |
| <i>Блок-схема алгоритму роботи додатку</i>   | <i>2 аркуша</i>  |  |                            |   |   |  |  |  |                     |  |  |
| <i>Показники економічної ефективності</i>  | <i>1 аркуш</i>   |  |                            |   |   |  |  |  |                     |  |  |

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2023	14.11.2023
Охорона праці	Оришака О.В.	06.10.2023	16.11.2023

7. Дата видачі завдання « 6 » вересня 2023 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2023 р.	
3.	Розробка моделі компонента	20.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2023 р.	
6.	Програмування алгоритмів	10.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2023 р.	
9.	Оформлення ПЗ	17.11.2023 р.	
10.	Попередній захист роботи	10.12.2023 р.	

Дата видачі завдання  
« 6 » вересня 2023 р.

Підпис керівника

\_\_\_\_\_  
(прізвище та ініціали)Завдання прийнято до виконання  
« 6 » вересня 2023 р.

Підпис здобувача

\_\_\_\_\_  
(прізвище та ініціали)

## АНОТАЦІЯ

**Горшков В.О. Дослідження та програмна реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.**

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

Метою розробки є дослідження та програмна реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

Об'єктом дослідження є процес автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

Предметом дослідження є методи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

Методи дослідження базуються на методах теорії оптимізації, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі Delphi 10.4.1.

**Ключові слова:** комп'ютерні науки, розклад занять

## ABSTRACT

**Horshkov V.O. Research and software implementation of the system of automated distribution of the classroom fund of the class schedule for faculties. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.**

In this graduation thesis for the second (master's) level of higher education, software was developed, which is intended for the system of automated distribution of the classroom fund of the class schedule for faculties.

The purpose of the development is the research and software implementation of the system of automated distribution of the classroom fund of the class schedule for faculties.

The object of the study is the process of automated distribution of the classroom fund of the class schedule for faculties.

The subject of the study is the methods of automated distribution of the classroom fund of the class schedule for faculties.

Research methods are based on optimization theory methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the system for the automated distribution of the classroom fund of the class schedule for the faculties.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on PCs of IBM PC architecture with Windows 10/11 OS.

The program was developed in the Delphi 10.4.1 environment.

**Keywords:** computer science, class schedule

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	11
1.1 Призначення системи.....	11
1.2 Область застосування.....	11
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	21
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	21
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	28
2.3 Розгорнута постановка завдання .....	34
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	36
3.1 Опис функціонування системи .....	36
3.2 Розробка структурної схеми.....	44
3.3 Розробка функціональної схеми .....	53
3.4 Розробка діаграми процесів.....	60
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	62
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	62
4.2 Захист розробленого програмного забезпечення.....	76
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	81
6 НАУКОВА НОВИЗНА .....	83

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>			
<b>Вим.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підп.</b>	<b>Дата</b>	Дослідження та програмна реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів	<b>Літ.</b>	<b>Аркуш</b>	<b>Аркушів</b>
<b>Розроб.</b>	Гориков В.О.					<b>М</b>	1	123
<b>Перев.</b>	Коваленко А.С.							
<b>Н.контр.</b>	Коваленко А.С.					ЦНТУ КН-22М-2		
<b>Затв.</b>	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	84
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	84
7.2 Розрахунок трудомісткості розробки програмної продукції.....	86
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	88
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	92
7.5 Визначення собівартості розробки та ціни програмної продукції.....	96
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	100
7.7 Визначення експлуатаційних витрат.....	100
7.8 Визначення економічної ефективності програмної продукції.....	102
7.9 Висновок.....	104
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	105
8.1 Вступ .....	105
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	105
8.3 Забезпечення електробезпеки .....	106
8.4 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста .	108
8.5 Розробка заходів з умов поліпшення охорони праці.....	111
8.6 Розрахункова частина .....	112
9 ОСНОВНІ ВИСНОВКИ.....	115
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	117

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АСУ	–	автоматизовані системи управління
БД	–	база даних
ЗВО	–	заклад вищої освіти
ПЗ	–	програмне забезпечення
СУБД	–	система управління базами даних

КБПЗ-2023

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

**Актуальність теми.** Завдання синтезу розкладів є предметом широких наукових досліджень із середини ХХ століття. Область їхнього застосування містить у собі такі різні сфери людської діяльності, як промисловість, транспортні перевезення, утворення, масове обслуговування й т.д. Практика висуває безліч завдань, які неможливо ефективно вирішувати шляхом повного перебору. Для більшості моделей теорії розкладів знаходження оптимального розкладу є важкорозв'язним завданням, а рішення наближених до реальних умов завдань має ще більшу складність, так як дані рішення повинні задовольняти численним, таким що найчастіше конфліктує між собою, обмеженням виробничого, організаційного й психофізіологічного характеру. Головною проблемою тут є велика багатоваріантність, що не дозволяє за прийнятний час одержувати оптимальне рішення. Виходом з даного положення є відмова від підходу, коли придатним вважається тільки найкраще рішення. Цей напрямок, одержало в останні десятиліття широкий розвиток у дискретній оптимізації – напрямок побудови ефективних алгоритмів наближеного рішення NP-складних завдань, детерміністичні методи рішення, для яких, неприйнятні або не забезпечують необхідного ступеня точності. У даній роботі розглянуто завдання подібного класу, що виникають у конкретних областях управлінської діяльності – складання розкладів навчальних занять у ЗВО. Дане завдання є, завданнями на розміщення вимог в обмежені ресурси, з нелінійними обмеженнями й критеріями, у вигляді формул вирахування предикатів, що приймають дискретні значення. У силу NP-складнощів ці завдання, для свого рішення вимагають розробки ефективних наближених методів.

Кількісний і якісний ріст вищої школи вимагає нового підходу до рішення завдань керування навчальною, науковою й господарською діяльністю вузів. Цей підхід в останні роки знаходить своє втілення в застосуванні сучасних засобів

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

обчислювальної техніки й математичних методів у керуванні вищими навчальними закладами. У сучасному світі все більше впровадження одержують різного роду системи автоматизації тих процесів, які завжди виконувалися вручну. Наприклад, системи ухвалення рішення в маркетингу, експертні системи які замінюють досвідчених фахівців, що прогнозують системи у всіляких галузях науки й техніки. До таких же процесів відноситься й складання розкладу, що дотепер у багатьох навчальних закладах складається вручну, на основі багаторічного досвіду. Сучасна наука має у своєму розпорядженні засоби, що дозволяють щонайкраще організувати будь-який процес, у тому числі й навчальний.

Завдання планування розкладу навчальних занять – це завдання на складання розкладу комбінаторного типу, характерною рисою, який є величезна розмірність і наявність великої кількості обмежень складної форми. Фактично, у теперішній час, не існує універсальних методів рішення таких завдань. Математична (класична) теорія розкладів охоплює лише вузьке коло добре формалізуємих проблем, які звичайно зводяться до завдань комівояжера, транспортної й т.п. Пряме застосування яких-небудь даних методів до завдання складання розкладу навчальних занять не представляється можливим. Проте, є ряд евристичних і переборних методів, які цілком піддаються програмуванню.

Існує думка, що досвідчений диспетчер може скласти розклад так, що воно буде більшою мірою відповідати інтересам навчального процесу й громадського життя освітньої установи. Однак із цим не можна погодитися. Ручне рішення завдання складання розкладу занять вимагає наявності великих витрат часу, кваліфікованих фахівців, у той час як результат такого рішення в багатьох випадках виходить далеко не оптимальним. Мова йде не про те хто краще складе розклад комп'ютер або людина, так як в реальності існує не автоматичне, а автоматизоване складання розкладу. Після уведення вихідної інформації потрібно її узгодження, так як неможливість одержання необхідного розкладу може бути визначена ще на етапі аналізу. Під час складання розкладу

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

можливе виникнення тупикових ситуацій. Все це вимагає зміни вихідних даних і ослаблення обмежень, тут без людини не обійтися. Без внесення даних змін розклад не буде мати практичної цінності. Тут також варто врахувати той момент, що розклад може змінюватися й під час його використання, тобто після його складання, і тут досить важливий людський фактор. У цьому плані важлива підтримка даного процесу, шляхом застосування автоматизованих методів і процедур. Основна перевага полягає в тому, що автоматизоване складання забирає масу рутинної роботи, такий як пошук можливих варіантів внесення чергових елементів у розклад, перевірки виконання вимог, пошуку випадкових помилок у готовому розкладі, оформлення розкладу на папері у вигляді різних таблиць (для викладачів, груп, покабінетного), залишаючи людині більше часу на більше інтелектуальні дії. Комп'ютер, у цьому випадку, також, є інструментом, що істотно підсилює здатності людини, так як людина не в змозі перебрати й проаналізувати таку ж кількість варіантів розкладів як комп'ютер.

Вивчення досвіду створення подібних систем, показує, що в останні роки вживали множинні спроби вдосконалювання планування навчального процесу, шляхом побудови алгоритмів оптимізації завдань планування навчальної роботи ЗВО і наступною їхньою реалізацією на обчислювальній техніці. У першу чергу це спроби рішення завдання складання розкладів занять, що є, як відомо, самої складної в підсистемі планування. Такі дослідження в різний час проводилися й тривають у деяких ЗВО. Однак практичне впровадження планування навчального процесу з використанням комп'ютерної техніки має місце лише в деяких ЗВО.

Аналіз стану цих розробок дозволяє зробити наступні висновки:

– Розробка й впровадження вузами завдань створення розкладу занять здійснюється в ініціативному порядку й ці роботи, як правило, спрямовані на рішення окремих завдань. Роз'єднаність груп дослідників і розроблювачів привела до створення безлічі систем, спрямованих на розробку алгоритмів і програм, розрахованих на обслуговування тільки конкретного ЗВО.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

– Існуючі системи не враховують специфіку складання розкладів у університеті (обмежений аудиторний фонд, неможливість завдання необхідних обмежень).

– Багато систем покладають на розроблювача розкладу всю відповідальність за облік реальних вимог. Зокрема, облік вимог викладачів, обмежень на кількість проведених занять у день, у тиждень, необхідні переноси занять у випадку нестачі ресурсів – всі ці й багато інших рутинних завдань у таких системах доводиться вирішувати людині, найчастіше наважання, методами перебору.

– Наявні програми не припускають багатокористувальницький режим роботи й не підтримують весь необхідний електронний документообіг. Вихідні документи не відповідають за формою застосовуваним у нашій ЗВО. Відсутні можливості налаштування шаблонів документів.

– При плануванні не забезпечується єдність інформаційного й програмного забезпечення й технічної бази.

– Відсутній системний підхід до використання комп'ютерної техніки в керуванні ЗВО, тобто завдання розробляються при відсутності технічних завдань і технічних проектів.

– Майже не виконуються роботи з розробки типових уніфікованих елементів для створення єдиної автоматизованої системи для керування вищою школою.

– Наявні програми мають досить незручний інтерфейс для введення вихідних даних і редагування отриманого розкладу.

У зв'язку з розширенням робіт з удосконалювання системи керування вищою школою шляхом створення й впровадження у ЗВО різних автоматизованих систем керування виникла необхідність в уніфікації засобів складання навчального розкладу на обчислювальній техніці. Для цього необхідно чітко формалізувати вимоги до розкладу й розробити відповідне алгоритмічне забезпечення.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

При розробці алгоритмів автоматизованого складання розкладу занять гостро коштує проблема створення універсальних алгоритмів, що враховують специфіку умов кожного конкретного завдання. Такі алгоритми повинні бути досить "гнучкими" у тому розумінні, щоб без істотної їхньої зміни можна було включати й виключати вимоги із системи вимог до розкладу. Однак спроба вирішувати завдання яким-небудь одним єдиним універсальним алгоритмом, на даний момент не представляється можливим. Алгоритми що дозволяють вирішувати широкий клас завдань не дають тої ефективності, як більше конкретні, адаптовані з урахуванням конкретних умов, алгоритми.

Для систем складання розкладу занять характерна сильна залежність від специфіки конкретних навчальних закладів уже на рівні математичних моделей і подання даних, що утрудняє використання типових систем. Це збільшується роз'єднаністю груп дослідників і розроблювачів. Систему створену в одному ЗВО звичайно без зміни й доробки неможливо ефективно використовувати в іншому. До того багато які з них створювалися досить давно й з їхньою допомогою неможливо ефективно вирішувати поставлене завдання.

Для рішення поставлених проблем, потрібна побудова гнучкої й легка адаптуємої системи на основі нових принципів, з використанням сучасних комп'ютерних технологій. Необхідна система, що становить розклад відповідно до обраних критеріїв і заданих вимог, тобто, що бере на себе якнайбільше функцій людини, щоб розклад доводилося менше доводити вручну. Дані можливості повинні здійснюватися також, без зміни вихідного коду системи, щоб використовуватися користувачами-непрограмістами. Для покриття найбільш типових випадків, необхідне створення декількох типових алгоритмів, що реалізують складання розкладів. Потрібна система підтримки прийняття рішень, що має можливість налаштування, потенційно враховуючого всілякого обмеження й побажання. Дана система повинна мати можливість доповнення й зміни існуючої бази даних і користувальницького інтерфейсу, а також забезпечувати можливість самостійно підключення додаткових самостійно розроблених алгоритмів, можливо з викликом деяких надаваних системою

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

типових алгоритмів. Все це давало б можливість задавати в кожному ЗВО вимоги, що відповідають його умовам, і, за допомогою підбора й налаштування підходящого алгоритму, одержувати необхідний розклад.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

– Дослідження системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

– Програмна реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

*Об'єктом дослідження* є процес автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

*Предметом дослідження* є методи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

*Методи дослідження* базуються на методах теорії оптимізації, методах математичної статистики, методах розробки програмного забезпечення.

**Наукова новизна отриманих результатів.** У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

– Розроблено вітчизняний продукт автоматизованого розподілу аудиторного фонду розкладу занять для факультетів, який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

**Достовірність наукових результатів** підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2023, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №14.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБГІЗ-2023

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>10</b>

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Система призначена для реалізації програмного забезпечення автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

З погляду кінцевих цілей навчального процесу близьким до ідеалу можна вважати такий розклад занять, у якому максимально ефективно використані всі п'ять основних ресурсів ЗВО. Це:

- календарний час;
- набір досліджуваних дисциплін;
- контингент студентів;
- аудиторний фонд;
- штат викладачів.

На жаль, людині властиво концентруватися на одних ресурсах на шкоду іншим. До чого веде ця типова помилка, добре відомо методистам ЗВО. Приміром, неефективне використання такого ресурсу як час викладачів – обмежує їхні можливості по підготовці до занять, проведенню факультативів, участі в науково-дослідних проектах ЗВО і т.д. А це, в остаточному підсумку, позначається на якості знань студентів. Аналогічним образом «розбазарювання» аудиторного фонду ЗВО створює штучний дефіцит приміщень для занять навчальних груп, організації наукових конференцій, роботи студентських кружків, громадських організацій і т.д. Ці й інші проблеми можна легко запобігти за допомогою інструментів обмеження ресурсів, убудованих у систему, що розробляється в магістерській роботі. Її користувач, зокрема, може заборонити використання в певний час окремих аудиторій і проведення певних навчальних пар, обмежувати зміну графіку роботи викладачів і занять навчальних груп, коректувати календар неробочих днів. Обмеження на використання аудиторій для вивчення певних дисциплін можуть накладатися з урахуванням видів робіт. У той

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

же час спеціальний інструментарій дозволяє оперативно бронювати аудиторії по запитах викладачів. Система, що розробляється в магістерській роботі, дозволяє накладати обмеження як по інтервалі часу, так і по номері навчальної пари, як в обов'язковому, так і в бажаному режимі. Для швидкого й зручного створення пакетних обмежень розроблені спеціальні форми введення даних. Потужні інструмент візуалізації даних полегшують оперативний аналіз і контроль за використанням ресурсів ЗВО – завантаженістю аудиторій у певний період, завантаженням професорсько-викладацького складу й ін.

Завдяки унікальному сполученню убудованих функцій обмеження ресурсів, конструктора розкладу й технологічної карти вивчення дисциплін система, що розробляється в магістерській роботі, являє собою не рішення, що має аналогів, для вітчизняних ЗВО.

## 1.2 Область застосування

Областю застосування є вищі навчальні заклади. Вищий навчальний заклад – освітній, освітньо-науковий заклад, який заснований і діє відповідно до законодавства про освіту, має один з чотирьох рівнів акредитації, реалізує відповідно до наданої ліцензії освітньо-професійні програми вищої освіти за певними освітніми та освітньо-кваліфікаційними рівнями, забезпечує навчання, виховання та професійну підготовку осіб відповідно до їх покликання, інтересів, здібностей та нормативних вимог у галузі вищої освіти, а також здійснює наукову та науково-технічну діяльність.

### **Класифікація вищих навчальних закладів України за формою власності**

Заклад вищої освіти державної форми власності – заклад вищої освіти, заснований державою, що фінансується з державного бюджету і підпорядковується відповідному центральному органу виконавчої влади.

Заклад вищої освіти, що перебуває у власності Автономної Республіки Крим – заклад вищої освіти, заснований органами влади Автономної Республіки

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Крим, що фінансується з бюджету Автономної Республіки Крим і підпорядкований органам влади Автономної Республіки Крим.

Заклад вищої освіти комунальної форми власності – заклад вищої освіти, заснований місцевими органами влади, що фінансується з місцевого бюджету і підпорядкований місцевим органам влади.

Заклад вищої освіти приватної форми власності – заклад вищої освіти, заснований на приватній власності і підпорядкований власнику чи власникам;

### **Типи вищих навчальних закладів**

Станом на жовтень 2011 серед вищих навчальних закладів України I-IV рівнів акредитації:

- 197 університетів;
- 62 академії;
- 109 інститутів;
- 234 коледжі;
- 130 технікумів;
- 121 училище;
- 1 консерваторія [4].

Університет – багатoproфiльний заклад вищої освіти четвертого рівня акредитації, який провадить освітню діяльність, пов'язану із здобуттям певної вищої освіти та кваліфікації широкого спектра природничих, гуманітарних, технічних, економічних та інших напрямів науки, техніки, культури і мистецтв, проводить фундаментальні та прикладні наукові дослідження, є провідним науково-методичним центром, має розвинуту інфраструктуру навчальних, наукових і науково-виробничих підрозділів, відповідний рівень кадрового і матеріально-технічного забезпечення, сприяє поширенню наукових знань та здійснює культурно-просвітницьку діяльність.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13



кількома спорідненими спеціальностями і має відповідний рівень кадрового та матеріально-технічного забезпечення;

Технікум (училище) – заклад вищої освіти першого рівня акредитації або структурний підрозділ закладу вищої освіти третього або четвертого рівня акредитації, який провадить освітню діяльність, пов'язану із здобуттям певної вищої освіти та кваліфікації за кількома спорідненими спеціальностями, і має відповідний рівень кадрового та матеріально-технічного забезпечення.

### **Національний заклад вищої освіти**

Державному закладу вищої освіти четвертого рівня акредитації відповідно до законодавства може бути надано статус національного.

Національному закладу вищої освіти за рішенням Кабінету Міністрів України може бути надано повноваження:

– укладати державні контракти з виконавцями державного замовлення для потреб закладу вищої освіти;

– приймати рішення про створення, реорганізацію, ліквідацію підприємств, установ, організацій, структурних підрозділів закладу вищої освіти;

– вносити пропозиції щодо передачі об'єктів закладу вищої освіти до сфери управління інших органів, уповноважених управляти державним майном, у комунальну власність та передачі об'єктів комунальної власності у державну власність і віднесення їх до майна закладу вищої освіти;

– виступати орендодавцем нерухомого майна, що належить закладу вищої освіти;

– встановлювати і присвоювати вчені звання доцента чи професора закладу вищої освіти;

– визначати та встановлювати власні форми морального та матеріального заохочення працівників закладу вищої освіти.

### **Структурні підрозділи закладу вищої освіти**

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Структурними підрозділами закладу вищої освіти третього і четвертого рівнів акредитації є кафедри, факультети, інститути, філії, бібліотека тощо.

### **Кафедра**

Кафедра – базовий структурний підрозділ закладу вищої освіти (його філій, інститутів, факультетів), що проводить навчально-виховну і методичну діяльність з однієї або кількох споріднених спеціальностей, спеціалізацій чи навчальних дисциплін і здійснює наукову, науково-дослідну та науково-технічну діяльність за певним напрямом. Кафедра створюється рішенням Вченої ради закладу вищої освіти за умови, якщо до її складу входить не менше ніж п'ять науково-педагогічних працівників, для яких кафедра є основним місцем роботи, і не менше ніж три з яких мають науковий ступінь або вчене звання.

Керівництво кафедрою здійснює завідувач кафедри, який обирається на цю посаду за конкурсом Вченою радою закладу вищої освіти строком на п'ять років (для національного закладу вищої освіти – строком на сім років). Із завідувачем кафедрою укладається контракт.

### **Факультет**

Факультет – основний організаційний і навчально-науковий структурний підрозділ закладу вищої освіти третього та четвертого рівнів акредитації, що об'єднує відповідні кафедри і лабораторії. Факультет створюється рішенням Вченої ради закладу вищої освіти за умови, якщо до його складу входить не менше ніж три кафедри і на ньому навчається не менше ніж 200 студентів денної (очної) форми навчання.

Рішення про створення факультету поза місцем розташування закладу вищої освіти державної форми власності приймається спеціально уповноваженим центральним органом виконавчої влади у галузі освіти і науки, іншими центральними органами виконавчої влади, які мають у своєму підпорядкуванні вищі навчальні заклади.

Структурними підрозділами закладу вищої освіти першого та другого рівнів акредитації є відділення і предметні (циклові) комісії

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Відділення – структурний підрозділ, що об'єднує навчальні групи з однієї або кількох спеціальностей, методичні, навчально-виробничі та інші підрозділи. Відділення створюється рішенням керівника закладу вищої освіти, якщо на ньому навчається не менше ніж 150 студентів.

Предметна (циклова) комісія – структурний навчально-методичний підрозділ, що проводить виховну, навчальну та методичну роботу з однієї або кількох споріднених навчальних дисциплін. Предметна (циклова) комісія створюється рішенням керівника закладу вищої освіти за умови, якщо до її складу входить не менше ніж три педагогічних працівники.

Філія – відокремлений структурний підрозділ, що створюється з метою забезпечення потреб у фахівцях місцевого ринку праці та наближення місця навчання студентів до їх місця проживання.

Філію очолює директор, який призначається у порядку, встановленому статутом закладу вищої освіти.

Структурними підрозділами закладу вищої освіти четвертого рівня акредитації можуть бути наукові, навчально-наукові, науково-дослідні та науково-виробничі інститути. Керівництво інститутом здійснює директор, який обирається на посаду за конкурсом строком на п'ять років.

Заклад вищої освіти повинен мати у своєму складі бібліотеку, яку очолює директор бібліотеки закладу вищої освіти третього і четвертого рівнів акредитації обирається Вченою радою закладу вищої освіти строком на п'ять років (для національного закладу вищої освіти – строком на сім років).

Заклад вищої освіти може мати у своєму складі підготовчі відділення (підрозділи), підрозділи перепідготовки та підвищення кваліфікації кадрів, лабораторії, навчально-методичні кабінети, комп'ютерні та інформаційні центри, навчально-виробничі та творчі майстерні, навчально-дослідні господарства, виробничі структури, видавництва, спортивні комплекси, заклади культурно-побутового призначення та інші підрозділи, діяльність яких не заборонена законодавством.

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

## **Форми навчання у вищих навчальних закладах**

Навчання у вищих навчальних закладах здійснюється за такими формами:

- денна;
- вечірня;
- заочна, дистанційна;
- суботня;
- екстернатна.

Форми навчання можуть бути поєднані. Терміни навчання за відповідними формами визначаються можливостями виконання освітньо-професійних програм підготовки фахівців певного освітньо-кваліфікаційного рівня.

Екстернатна форма навчання – особлива форма навчання, що передбачає самостійне вивчення навчальних дисциплін, складання у закладі вищої освіти заліків, екзаменів та проходження інших форм підсумкового контролю, передбачених навчальним планом.

## **Форми організації навчального процесу**

Навчальний процес у вищих навчальних закладах здійснюється у таких формах:

- навчальні заняття;
- самостійна робота;
- практична підготовка;
- контрольні заходи.

Основними видами навчальних занять у вищих навчальних закладах є:

- лекція;
- лабораторне, практичне, семінарське, індивідуальне заняття, майстер-класове заняття;
- консультація.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Вищим навчальним закладом може бути встановлено інші види навчальних занять.

### **Управління вищим навчальним закладом**

Безпосереднє управління діяльністю закладу вищої освіти здійснює його керівник – ректор (президент), начальник, директор тощо.

Керівник закладу вищої освіти в межах наданих йому повноважень:

– вирішує питання діяльності закладу вищої освіти, затверджує його структуру і штатний розпис;

– видає накази і розпорядження, обов'язкові для виконання всіма працівниками і структурними підрозділами закладу вищої освіти;

– представляє заклад вищої освіти у державних та інших органах, відповідає за результати його діяльності перед органом управління, у підпорядкуванні якого перебуває заклад вищої освіти;

– є розпорядником майна і коштів;

– виконує кошторис, укладає угоди, дає доручення, відкриває банківські рахунки;

– приймає на роботу та звільняє з роботи працівників;

– забезпечує охорону праці, дотримання законності та порядку;

– визначає функціональні обов'язки працівників;

– формує контингент осіб, які навчаються у закладі вищої освіти;

– відраховує та поновлює на навчання осіб, які навчаються у закладі вищої освіти;

– контролює виконання навчальних планів і програм;

– контролює дотримання всіма підрозділами штатно-фінансової дисципліни;

– забезпечує дотримання службової та державної таємниці;

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

– здійснює контроль за якістю роботи викладачів, організацією навчально-виховної та культурно-масової роботи, станом фізичного виховання і здоров'я, організовує побутове обслуговування учасників навчально-виховного процесу та інших працівників закладу вищої освіти;

– разом із профспілковими організаціями подає на затвердження вищому колегіальному органу громадського самоврядування закладу вищої освіти правила внутрішнього розпорядку та колективний договір і після затвердження підписує його.

Керівник закладу вищої освіти відповідає за провадження освітньої діяльності у закладі вищої освіти, за результати фінансово-господарської діяльності, стан і збереження будівель та іншого майна.

Для вирішення основних питань діяльності відповідно до статуту керівник закладу вищої освіти створює робочі та дорадчі органи, а також визначає їх повноваження.

Керівник закладу вищої освіти щорічно звітує перед власником (власниками) або уповноваженим ним (ними) органом (особою) та вищим колегіальним органом громадського самоврядування закладу вищої освіти.

Керівник закладу вищої освіти відповідно до статуту може делегувати частину своїх повноважень своїм заступникам та керівникам структурних підрозділів.

При виході на пенсію з посади керівника закладу вищої освіти керівник, який пропрацював на цій посаді не менше двох термінів підряд, може бути призначений на посаду почесного ректора (президента), начальника, директора тощо закладу вищої освіти із виплатою грошового утримання за рахунок навчального закладу в розмірі заробітної плати, яку він отримував перед виходом на пенсію.

Призначення почесного ректора (президента), начальника, директора тощо закладу вищої освіти здійснюється у порядку, встановленому Кабінетом

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Міністрів України. Умови призначення та виплати грошового утримання визначаються у статуті закладу вищої освіти.

Таким чином, виходячи з вищеперерахованого, розробка програмного забезпечення автоматизованої системи розподілу аудиторного фонду розкладу занять для факультетів, є актуальною задачею, яка потребує вирішення у даній магістерській роботі.

КБГПЗ - 2023

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

## 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

### 2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень по профілю теми магістерської роботи

#### Експрес-розклад

Програма «Експрес-розклад» розроблена спеціально для автоматизації роботи завучів і диспетчерів інститутів, університетів, академій, а завдяки побажанням користувачів постійно вдосконалюється й «обростає» масою додаткових можливостей. Програма зареєстрована в Міністерстві освіти, впроваджена більш ніж в 450 навчальних закладах країни.

У програмі можна вести розклад як очного, так і заочного відділення, облік практик, консультацій, іспитів, облік заміщення викладачів, передачу навантаження іншому викладачеві протягом семестру, призначення занять без основного розкладу (у поточному). При автоматичному складанні основного розкладу враховуються заборони, тобто графіки викладачів сумісників, методичні дні й т.п. Отриманий розклад можна коректувати вручну. Також можна призначити деякі заняття вручну, а потім доскладати розклад автоматично. У програмі зроблена зручна можливість копіювання інформації на флеш-пам'ять, щоб можна було переносити дані й працювати із програмою, наприклад, дома. Установка програми на домашній комп'ютер диспетчера безкоштовна. Для самостійного вивчення програми був розроблений навчальний фільм, довідкова інформація (допомога) і посібник користувача. Будучи споконвічно простою й зручною, програма здобула популярність серед завучів і диспетчерів. Рекомендуємо використовувати ліцензійну версію програми «Експрес-розклад» у Вашому навчальному закладі. Це дозволить полегшити й удосконалити процес складання навчального розкладу, поліпшити його якість і оперативність.

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22



- Можливість установки нульових пар.
- Повний облік виконання навантаження.
- Планування відсутності й зайнятості викладачів, груп, аудиторій.
- Друк основного й поточного розкладу, аркуша заміни.
- Можливість друку таблиць й інших звітів (передаються в Excel), коректування шаблонів звітів.
- Повна інформованість диспетчера в процесі складання розкладу (виділення кольором, підказки).
- Вибірка занять для ведення журналу.

### **Ректор-ЗВО**

Програма призначена для складання розкладу занять у ЗВО. Програма складається із чотирьох розділів: «Списки», «Навантаження», «Розклад» і «Заміни».

Розділ «Списки» служить для уведення, редагування й друку списків кафедр, спеціальностей, груп, дисциплін, аудиторій, викладачів і видів занять. Розділ «Навантаження» використовується для уведення, редагування й друку навчальних планів по кожній спеціальності, навантажень викладачів, графіків розподілу годин по тижнях у межах семестру, звітів по завантаженню викладачів, кафедр і ЗВО в цілому. Розділ «Розклад» призначений для складання розкладу по групах, викладачам, аудиторіями й ЗВО в цілому. Розділ «Заміни» дозволяє оперувати замінами викладачів.

Розклад занять можна встановлювати в автоматичному, ручному або комбінованому режимі; переходити від одного режиму до іншого можна в будь-який момент часу. При складанні розкладу в автоматичному режимі програма враховує всі сформульовані вимоги до розкладу. При складанні розкладу в ручному режимі програма підказує можливі варіанти розміщення уроків обраного викладача, можливі варіанти заповнення порожніх кліток у розкладі групи, стежить за кількістю місць в аудиторіях.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Готовий розклад занять однієї, всіх або деяких груп і викладачів можна зберегти у форматах Microsoft Word, Excel або HTML.

## Розклад ПРО

Програма проста й зручна у використанні. Вона дозволяє значно скоротити час складання шкільних або інститутських розкладів при мінімальних витратах зусиль.

Основна технологія роботи представлена в деревоподібній структурі "Керування" – необхідно пройти послідовно по "вітках" цього дерева зверху долилиць. Кожний варіант розкладу розглядається як окремий проект. Увівши один раз вихідні дані, можна надалі робити копію проекту й працювати (редагувати або експериментувати) з нею.

Програма підтримує два режими керування даними: ручний і автоматичний.

The screenshot shows the 'Расписание ПРО' software interface. The main window title is 'Расписание ПРО - Школа №123 (Пример.сш)'. The menu includes 'Файл', 'Расписание', 'Вид', and 'Помощь'. The left sidebar shows a tree view of groups: 'Исходные данные', 'Расписание', 'Группы' (with sub-items like 'Классы 2', 'Общнй', '2а', '2б', 'Классы 8', 'Гуманитарные', '8а', '8б', 'Физико-мет...', '8в'). The main table displays the schedule for groups 'Пг' and 'Вг'. The right sidebar shows a summary table for group '2а'.

Группа	№	Уч	Уч	Уч	Уч
Пг	-	РусЯзык Кас2а	ОБЖ Кас2б		РусЛит КабЛит
	2	Рисов Студия	РусЯзык КабЛит	УкрЛит КабЛит	
	3	ОБЖ Кас2а	РусЛит Кас2б		УкрЯзык КабЛит
	4	Музыка Студия	Труд КабТруд		ИнПЗык КабЛит
	5				Информ КабЛит
	6				Матем КабМат
Вг	-	Матем Кас2а	РусЯзык Кас2б	УкрЛит КабЛит	
	2	РусЛит Кас2а	Музыка Студия	ИнЯзык КабЛит	РусЛит КабЛит
	3	Рисов Студия	Матем КабТруд	УкрЛит КабЛит	История КабЛит
	4	Физкульт	Физкульт	Физика	Информ

Предметы	Занятий...
Матем	10
Музыка	11
ОБЖ	0
Рисов	0
РусЛит	10
РусЯзык	0
Труд	0
Физкульт	11

Занятия всего: (20)  
 В семестре: 40  
 В расписании: 4  
 Не распределено занятий:  
 В семестре: 10  
 В расписании: 1

## Рисунок 2.2 – Інтерфейс користувача Розклад ПРО

### Експрес-розклад ЗВО

Програма «Експрес-розклад ЗВО» розроблена спеціально для автоматизації роботи завучів і диспетчерів академій, інститутів, університетів. Програма зареєстрована в Міністерстві освіти.

У програмі можна вести розклад як очного, так і заочного відділення, облік практик, іспитів, облік заміщення викладачів і ін.

При автоматичному складанні основного розкладу враховуються заборони.

У програмі зроблена зручна можливість копіювання інформації на флеш-пам'ять, щоб можна було переносити дані й працювати із програмою, наприклад, будинку. Установка програми на домашній комп'ютер диспетчера безкоштовна.

Для самостійного вивчення програми був розроблений навчальний фільм, довідкова інформація (допомога) і посібник користувача.

Будучи споконвічно простою й зручною, програма здобула популярність серед завучів і диспетчерів.

Рекомендуємо використовувати ліцензійну версію програми «Експрес-розклад» у навчальному закладі. Це дозволить полегшити й удосконалити процес складання навчального розкладу, поліпшити його якість і оперативність.

#### Можливості:

- Складання розкладу для декількох змін, а також потоків з різними строками початку й закінчення навчання.
- Ведення поточного розкладу й можливість щоденної заміни уроків, учителів, аудиторій.
- Можливість об'єднання класів у потік.
- Можливість установки нульових уроків.
- Робота з підгрупами (розподіл до 10 підгруп).
- Повний облік виконання навантаження за тиждень і за рік.
- Друк основного й поточного розкладу, аркуша заміни.

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

- Планування відсутності й зайнятості вчителів, класів, аудиторій.
- Публікація основного й поточного розкладу на сайті навчального закладу.
- Повна інформованість диспетчера в процесі складання розкладу.
- Можливість друку таблиць й інших звітів, коректування шаблонів звітів.
- Вибірка уроків для ведення журналу.
- Ведення журналу замінь для формування довідки на додаткову оплату.
- Облік норм СанПіН.

Microsoft Excel - Расписание на 22.09.2009.xls

УТВЕРЖДАЮ  
 Директор МДОУ СОШ №1 им. Н.Ф. Зюнова  
 (Г.Н. Онучина)  
 " " " 2009 г.

**Расписание занятий  
 на 22.09.2009 г. (вторник).**

Урок	Время	Предмет		Ауд.		Предмет		Ауд.		Предмет		Ауд.	
		Учитель	Учитель	Учитель	Учитель	Учитель	Учитель	Учитель	Учитель				
№	Подгруппа:	1	2	1	2	1	2	1	2	1	2	1	2
	Класс:	5 а		5 б		6 а		6 б					
1	8.00-8.40	Физкультура Перминов С.И.	АкЗл	Русский язык Новоселова Г.П.	15	Математика Скурихина С.А.	12	Математика Бехтерева И.П.	11				
2	8.50-9.30	Труд Кропотов С.Г.	М	Русский язык Новоселова Г.П.	Ф	Литература Боброва Г.А.	24	Русский язык Харламова Н.А.	20				
3	9.45-10.25	Математика Онучина Г.Н.	15	Физкультура Перминов С.И.	АкЗл	Русский язык Боброва Г.А.	24	История Леденцова Н.Н.	6				
4	10.40-11.20	Музыка Зобнина С.А.	АкЗл	Математика Онучина Г.Н.	15	Труд Кропотов С.Г.	М	Ин. язык Торопова И.В.	23				
5	11.35-12.15	ИЗО Обухова К.В.	5			Музыка Зобнина С.А.	АкЗл	Труд Кропотов С.Г.	М				
6	12.25-13.05					ИЗО Бехтерева А.В.	5						
7	13.15-13.55							ИЗО Обухова К.В.	5				

Рисунок 2.3 – Интерфейс користувача Експрес-розклад ЗВО

## Програма складання розкладу занять «НІКА»

Програма складання розкладу занять «НІКА» повністю автоматизує процес складання розкладу занять в освітніх установах різного статусу й профілю (школи, гімназії, коледжі, училища), з огляду на особливості навчальних планів і організації навчального процесу, а також вимоги методичного, санітарно-гігієнічного й індивідуально-психологічного характеру. Програма надзвичайно легка в освоєнні й зручна в роботі.

Результатом роботи програми є розклад занять, складене з мінімальною кількістю «вікон» для викладачів.

Програма «НІКА» ураховує:

- наявність другої зміни, п'яти- або шестиденної форм навчання;
- особливості кабінетного фонду;
- методичні дні й небажані годинники роботи вчителів;
- дні й годинники навантаження поза розкладом;
- граничні рівні денного навчального навантаження з урахуванням коефіцієнтів складності предметів.

Програма «НІКА» забезпечує:

- поділ класів (груп) на навчаються по п'яти- або шестиденці;
- «плаваючий» початок занять і другої зміни;
- розбивка класів (груп) на підгрупи (до трьох підгруп), у тому числі, по різних предметах;
- здвоювання, утроювання й т.д. заданої кількості годин;
- об'єднання класів у потоки (у т.ч. з подальшим розподілом на групи);
- викладання одним учителем двох і більше предметів;
- дотримання «паралелей» учителів;
- обмеження максимально припустимого числа годин по предметі в день

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

– складання розкладу для вчителя, що веде заняття у двох групах одного класу (формування пара з іншими предметами або розміщення на перший/останній уроки);

– відображення й друк розкладу занять у трьох формах;

– зручний інструмент для погодинного контролю розкладу окремих учителів, а також «дружній напівавтомат» для ручної редакції розрахованого розкладу;

– складання розкладу з урахуванням особливостей наявного аудиторного фонду.

У програмі «НІКА» реалізований ефективний механізм контролю за коректністю введення вихідних даних.

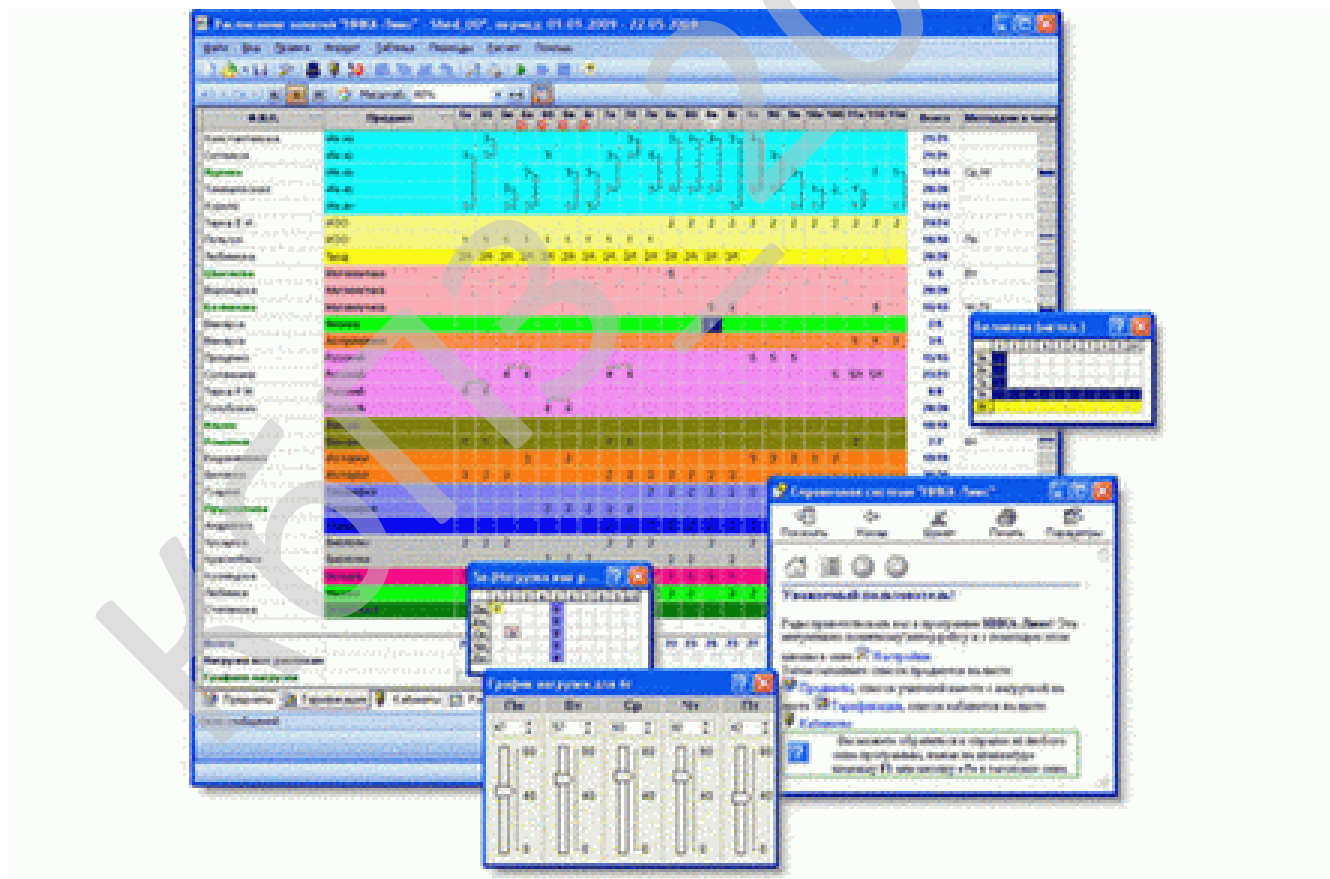


Рисунок 2.4 – Интерфейс користувача програми складання розкладу занять «НІКА»

## 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

### Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

#### Основні можливості Delphi 10.4.1:

– Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

– Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

– Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

– Тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання.

– Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

– Відладник Win 64 (на LLDB) і збирач для C++.

– Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.

– Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

– Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

– У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4k моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з підвищеною швидкодією. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису custom managed records. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

### **Істотне поліпшення Delphi Code Insight**

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

### **Delphi Custom Managed Records**

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільняються з допомогу вашого коду, який буде виконуватися у відповідний момент.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

Це розширює потужність конструкцій records в Delphi, які використовуються щоб одержати більшу ефективність у порівнянні із класами.

### **Єдине керування пам'яттю**

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існуючим кодом і спрощує написання компонентів, бібліотек і застосунків.

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

### **Розширена підтримка бібліотек C++**

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

### **Win 64-відладник і збирач для C++**

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

### **Підвищення якості й швидкодії інструментів**

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Snake.
- Велика кількість виправлень для підвищення стабільності і якості.
- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.
- Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

### **Змінені стилі VCL для High DPI**

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично масштабуються під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

Кожний графічний елемент UI може бути обраний з наборів різних масштабів і масштабований до потрібного DPI, що дає чітке зображення елементів UI на всіх моніторах.

### **Нові High DPI стилі й стилізація окремих VCL компонент**

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів.

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентах на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

### **Поліпшена кроссплатформеність**

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

### **Оновлений менеджер пакетів Getit**

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

### **Універсальний інсталятор для установки Online і Offline**

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

## **2.3 Розгорнута постановка завдання**

Згідно з технічним завданням на випуск кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи автоматизованого розподілу

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>35</b>

аудиторного фонду розкладу занять для факультетів.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі.

Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Однією з найважливіших проблем якісної організації навчального процесу у закладі вищої освіти є завдання створення автоматизованого навчального розкладу. Правильно й точно складений розклад забезпечує рівномірне завантаження студентських груп і професорсько-викладацького складу.

У теперішній час використання інформаційних систем у вищих освітніх установах не є рідкістю. Спектр їхнього застосування широкий і варіюється від автоматизації окремо взятих робочих місць до повної автоматизації діяльності ЗВО.

Поза залежністю від об'єкта автоматизації, будь то викладацький склад або адміністрація університету, в освітній установі такі системи впроваджують, переслідуючи кінцеву мету – підвищення якості утворення.

ЗВО, як і будь-яке підприємство, неодмінно проходить процес автоматизації й, незважаючи на те, що поняття освітньої діяльності єдино для всіх освітніх установ, у кожному ЗВО цей процес проходить по-різному. Значний вплив на процеси автоматизації робить як наявність коштів, так і готовність використання пропонованих ринком інформаційних послуг програмних продуктів.

На даний момент у закладі вищої освіти можна використовувати інформаційний простір, що включає компоненти по різних напрямках: кадровий облік співробітників, облік студентів, служба безпеки, рейтинг викладачів, контроль успішності студентів і інші.

У зв'язку з функціонуванням ЗВО у рамках єдиного інформаційного простору, використання сторонніх програмних продуктів робиться неможливим через специфіку роботи наявних систем або ж через дорожнечу впровадження,

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

що тягне значну доробку як наявних, так і інформаційних систем, що здобуваються.

З метою автоматизації планування було розроблене рішення, що спрощує процес створення електронного розкладу на основі аналізу наявних навчальних планів спеціальностей, що дозволяє аналізувати структуру навантаження, а також планувати структурну доробку й деяку уніфікацію наявних навчальних планів.

Загальні вимоги, пропоновані до автоматизованої системи, впливають наступні:

- використання норм часу для розрахунку обсягів навчального навантаження;
- використання інформації з навчальних планів спеціальностей;
- швидкість обробки інформації за рахунок автоматизації можливих операцій користувача системи;
- формування звітних форм;
- розширюваність системи (можливість її доробки у випадку підвищення вимог до автоматизованої системи);
- зручний користувальницький інтерфейс.

Сформулюємо вимоги до розроблюваної системи, визначимо критерії вибору СУБД такі як:

- критерій моделювання даних;
- критерій, що визначає особливості архітектури й функціональні можливості;
- критерій контролю роботи системи;
- критерій продуктивності;
- критерій надійності й ін.

Стрімкий прогрес засобів обчислювальної техніки, зміна умов утворення, зміна засобів і форм навчання, розширення спектра технічних засобів, а так само використання більших обсягів інформації диктують необхідність впровадження інформаційних технологій в утворення.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

Питань для чого потрібна автоматизація тих або інших процесів, як правило, не виникає. Але от питання – як і із чого почати автоматизацію виникають у багатьох. Скільки буде потрібно коштів? Що й у кого здобувати? Як впроваджувати? Хто буде займатися супроводом програмного продукту? Хто буде фінансувати? Ці й багато інших питань обов'язково виникають на початку автоматизації будь-яких процесів.

Варто почати з визначення поняття «автоматизація». Під автоматизацією звичайно розуміють застосування технічних і програмних засобів, частково або повністю звільняючої людини від особистої участі в процесах одержання, перетворення, передачі й використання матеріалів або інформації. Процесу автоматизації передують його формалізація, тобто одержання повного набору однозначно трактуємих інструкцій, керуючись якими досягається результат реалізації процесу.

Переваги автоматизації очевидні – це прискорення виконання операцій і зниження помилок при їхньому виконанні, зниження витрат на реалізацію операцій і підвищення якості. Успішною може вважатися автоматизація, у результаті впровадження й використання якої вдалося повернути інвестовані в неї кошти.

Виділяють наступні етапи процесу автоматизації, застосовувані в цілому до діяльності тієї або іншої установи. Кожний з етапів вимагає осмисленого й послідовного виконання:

– Етап 1: постановка проблеми, оцінка необхідності автоматизації й можливостей підприємства.

– Етап 2: формування вимог до програмно-апаратного комплексу, вибір або реалізація програмного продукту й технічного забезпечення.

– Етап 3: впровадження програмного продукту.

– Етап 4: післягарантійне обслуговування програмно-апаратного комплексу.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

Перш ніж почати автоматизацію необхідно ясно й чітко сформулювати свої вимоги до неї. Потрібно визначити якої саме функції необхідно автоматизувати. Варто враховувати, що найчастіше впровадження автоматизованих систем знижує ступінь впливу людського фактора на виконання тих або інших операцій.

Із числа керівників вищої ланки необхідно виділити співробітника, відповідального за проведення даного процесу й уповноваженого ухвалювати рішення щодо питань автоматизації, ступеня й ролі участі співробітників, що мають відношення до автоматизації.

Виділення кваліфікованих фахівців, відповідальних за розробку або впровадження програмного продукту, є обов'язковою умовою автоматизації. Кожний окремий фахівець у рамках своїх обов'язків повинен уміти оперативно усувати виникаючі в системі відмови, виявляти й усувати порушення умов експлуатації, вирішувати ті або інші проблеми користувачів.

Вибір програмного продукту або середовища його розробки не слід відокремлювати від вибору технічного забезпечення, на якому має бути надалі працювати. Не слід забувати й про наявну інформаційну систему ЗВО. Інтеграція в єдиний інформаційний простір дозволяє використовувати раніше реалізовані компоненти і єдину базу даних, що підвищує гнучкість інформаційної системи в цілому й знижує дублювання вже наявної інформації.

Описаний процес аналізу діяльності ЗВО звичайно називають «передпроектним обстеженням установи». У ході обстеженням будується повна модель організації, що описує не тільки взаємодію структурних одиниць, але й реалізовані ними операції й інформаційних потоків. Побудова подібних документованих моделей дозволяє не тільки оптимізувати поточну роботу, але й зробити діяльність підприємства менш залежною від конкретних людей, а також допомогти в навчанні нових співробітників.

Після побудови моделі установи й визначення вимог до програмного продукту, необхідно визначитися з вибором програми. Функціональна повнота (достатність) майбутньої програми є базовою вимогою до вибору програмного продукту.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

У рамках даної магістерської роботи розглядається автоматизація розкладу занять факультетів закладу вищої освіти. При виборі програмного продукту, особлива увага варто звернути на можливість гнучкого настроювання програмного продукту під специфіку конкретної освітньої установи (використовувані норми часу, реалізовані навчальні плани).

Варто звернути увагу на формування звітності. Важливо вирішити – кому будуть адресовані звіти, і для яких цілей, що дозволить визначити зміст і вид звітів.

Можна придбати єдиний програмний комплекс для автоматизації різних процесів ЗВО, але подібне рішення неминуче приведе до більших грошових і трудових витрат, зв'язаним як із придбанням нового програмного забезпечення, так і з переходом і адаптацією до нього. Можна придбати або розробити програму повністю адаптовану під специфіку конкретного ЗВО і яка автоматизує конкретний процес. Такі програми легкі в освоєнні й містять набір функцій і операцій, що набагато полегшують діяльність користувача системи. До достоїнств подібних програм варто віднести більше низьку вартість, простоту впровадження й адаптації в порівнянні з великими програмними комплексами, спрямованими на великої кількості завдань.

Вибір варіанта – купувати або розробляти самим – повністю залежить від кожної освітньої установи окремо. Тут відіграє більшу роль готовність у фінансуванні проекту, професіоналізм розроблювачів – співробітників освітньої установи, готових створити й впровадити програму, а також кваліфікація майбутніх користувачів.

Завдання складання розкладів є предметом наукових досліджень із середини минулого століття. Область їхнього застосування містить у собі різні сфери людської діяльності, такі як: транспортні перевезення, масове обслуговування, промисловість, утворення й т.д. Практика висуває безліч завдань, які неможливо ефективно вирішити шляхом повного перебору. Для більшості моделей теорії розкладів знаходження оптимального розкладу є

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

важкорозв'язним завданням, а рішення наближених до реальних умов завдань має ще більшу складність, так як дані рішення повинні задовольняти численним, що найчастіше конфліктує між собою обмеженням виробничого, організаційного й психофізіологічного характеру. Виходом з даного положення є відмова від підходу, коли придатним вважається тільки найкраще рішення. Розглянемо завдання подібного класу, що виникає в конкретній області управлінської діяльності, – складання розкладів навчальних занять у ЗВО.

Кількісний і якісний ріст вищої школи вимагає нового підходу до рішення завдань керування навчальною, науковою й господарською діяльністю ЗВО. Цей підхід в останні роки знаходить своє втілення в застосуванні сучасних засобів обчислювальної техніки й математичних методів у керуванні вищими навчальними закладами. У сучасному світі все більше поширення одержують різного роду системи автоматизації технічних процесів, які завжди виконувалися вручну. Наприклад, системи ухвалення рішення в маркетингу, експертні системи, що заміняють досвідчених фахівців, що прогнозують системи у всіляких галузях науки й техніки. До таких же процесів відноситься й складання розкладу, що дотепер у багатьох навчальних закладах створюється вручну на основі багаторічного досвіду. Сучасні ІТ-технології мають у своєму розпорядженні засоби, що дозволяють щонайкраще організувати будь-який процес, у тому числі й навчальний.

Завдання планування розкладу навчальних занять – це завдання на складання розкладу комбінаторного типу, характерною рисою якої є величезна розмірність і наявність великої кількості обмежень складної форми. Фактично, у теперішній час, не існує універсальних методів рішення таких завдань. Пряме застосування математичної (класичної) теорії розкладу до завдання складання навчальних занять не представляється можливим. Проте, є ряд евристичних і переборних методів, які цілком піддаються програмуванню.

Є думка, що досвідчений диспетчер зможе скласти розклад так, що воно буде відповідати інтересам навчального процесу й громадського життя освітньої

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

установи. Однак із цим не можна погодитися. Ручне рішення завдання складання розкладу занять вимагає більших витрат часу, кваліфікованих фахівців, у той же час результат такого рішення часто виходить далеко не оптимальним. Після уведення вихідної інформації потрібно її узгодження, у той час як неможливість одержання необхідного розкладу може бути визначена ще на етапі аналізу. Під час складання розкладу можливе виникнення тупикових ситуацій. Все це вимагає зміни вихідних даних і ослаблення обмежень, і тут без людини не обійтись. Без внесення даних змін розклад не буде мати практичної цінності. Також варто врахувати той момент, що розклад може мінятися й під час його використання, тобто після складання, і тут досить важливий людський фактор. У цьому плані важлива підтримка даного процесу автоматизованими методами й процедурами. Основна перевага полягає в тому, що автоматизоване складання усуває масу рутинної роботи, такий як: пошук можливих варіантів внесення чергових елементів у розклад, перевірку виконання вимог, пошук випадкових помилок у готовому розкладі, оформлення розкладу на папері у вигляді різних таблиць (для викладачів, груп, покабінетного), залишаючи людині більше часу на більше інтелектуальні дії. Комп'ютер у цьому випадку також є інструментом, що істотно підсилює здатності людини, так як людина не в змозі перебрати й проаналізувати таку ж кількість варіантів розкладів, як комп'ютер.

В останні роки вживають множинні спроби вдосконалювання планування навчального процесу шляхом побудови алгоритмів оптимізації завдань планування навчальної роботи ЗВО з використанням обчислювальної техніки й програмного забезпечення Microsoft Excel. Практичне впровадження планування навчального процесу з використанням веб-технологій має місце лише в деяких ЗВО. Аналіз стану цих розробок дозволяє зробити наступні висновки:

– розробка й впровадження вузами завдань АСУ здійснюється в ініціативному порядку й ці роботи, як правило, спрямовані на рішення окремих проблем. Роз'єднаність груп дослідників і розроблювачів привела до створення безлічі систем, спрямованих на розробку алгоритмів і програм, розрахованих на обслуговування тільки конкретного ЗВО;

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

– багато систем покладають на розроблювача розкладу всю відповідальність за облік реальних вимог. Зокрема, облік вимог викладачів, обмежень на кількість проведених занять у день, у тиждень – всі ці й багато інших рутинних завдань у таких системах доводиться вирішувати людині найчастіше методами перебору;

– наявні програми не припускають багатокористувальницький режим роботи й не підтримують весь необхідний електронний документообіг;

– не впроваджується розробка типових уніфікованих елементів для створення єдиної автоматизованої системи керування вищою школою;

– наявні програми мають досить незручний інтерфейс для уведення вихідних даних і редагування отриманого розкладу.

У зв'язку з розширенням робіт з удосконалювання системи керування вищою школою шляхом створення й впровадження у ЗВО різних автоматизованих систем керування виникає необхідність в уніфікуванні засобів складання навчального розкладу на обчислювальній техніці. Для цього необхідно чітко формалізувати вимоги до розкладу й розробити відповідне алгоритмічне забезпечення.

При розробці алгоритмів автоматизованого складання розкладу занять гостро стоїть проблема створення універсальних алгоритмів, що враховують специфіку умов кожного конкретного завдання. Такі алгоритми повинні бути досить "гнучкими", тобто без істотної їхньої зміни можна було б включати й виключати вимоги із системи вимог до розкладу. Однак спроба вирішувати завдання яким-небудь одним єдиним універсальним алгоритмом на даний момент не представляється можливою. Алгоритми, що дозволяють вирішувати широкий клас завдань, не дають тої ефективності, що забезпечують більше конкретні, адаптовані з урахуванням конкретних умов алгоритми.

Для систем складання розкладу занять характерна сильна залежність від специфіки конкретних навчальних закладів уже на рівні математичних моделей і подання даних, що утрудняє використання типових систем. Систему, створену в одному ЗВО, звичайно без зміни й доробки неможливо ефективно

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

використовувати в іншому. До того ж багато які з них створювалися досить давно й з їхньою допомогою неможливо ефективно вирішувати поставлене завдання.

Для рішення існуючих проблем потрібна побудова гнучкої й легкої адаптуємої системи на основі нових принципів, з використанням сучасних веб-технологій. Необхідна система, що становить розклад відповідно до обраних критеріїв і заданих вимог. Дані можливості повинні здійснюватися також без зміни вихідного коду системи. Для покриття найбільш типових випадків необхідне створення декількох типових алгоритмів, що реалізують складання розкладів. Дана система повинна мати можливість доповнення й зміни існуючої бази даних і користувальницького інтерфейсу. Все це давало б можливість задавати в кожному ЗВО вимоги, що відповідають його умовам, і за допомогою підбора й налаштування підходящого алгоритму одержувати необхідний розклад.

### 3.2 Розробка структурної схеми

У загальній постановці завдання складання розкладу являє собою процес розподілу деякого кінцевого набору подій у часі в умовах ресурсних і інших обмежень. У якості вихідних даних для складання розкладу виступають перелік спеціальностей, списки груп студентів, дисциплін, а також навчальне навантаження на групи. Наявний ресурс виражений як список професорсько-викладацького складу, які можуть читати деякі предмети в заданих групах, списку аудиторій, кількості навчальних днів у тиждень, максимальної кількості занять у день. Необхідно скласти розклад занять для кожної групи по днях тижня, розподілити навчальні аудиторії в кожний із днів тижня, розподілити навантаження на викладачів по днях з урахуванням їх побажань. Тобто найбільш зручним образом розподілити наявні ресурси відповідно до запитів і виділити час для занять кожної групи. Існування вдало складених розкладів говорить про те, що завдання складання оптимального розкладу розв'язні, або хоча б про те, що для неї існують припустимі рішення. Складання розкладу відноситься до завдань цілочисельного програмування, складність рішення яких росте експоненційно з

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

ростом числа й можливих варіюємих значень. Крім того, для неї характерна наявність великого обсягу різної по своєму составі вихідної інформації й великого числа складноформалізуємих вимог. Зазначені складності перешкоджають автоматизації процедури складання розкладу, незважаючи на наявність широкого спектра методів цілочисельного програмування.

### **Алгоритм складання розкладу занять**

Алгоритм складання розкладу завдань включає в себе наступні етапи:

1.1. Надаються вихідні дані для усіх курсів:

- Кількість навчальних дисциплін для усіх курсів, розбиті по курсам.
- Список навчальних груп по курсам.
- Прізвища викладачів, які ведуть той, або інший курс.
- Аудиторний фонд.

1.2. На основі розкладу за попередній рік, формується таблиця у якій введений розклад за попередній навчальний рік.

1.3. З пункту 1.2 формується аудиторний фонд, тобто вказуються зайняті аудиторії (номера аудиторій, назви груп, начальних предметів, ПБ викладача).

1.4 Складається новий розклад методом зсуву груп на 1 курс вище. Тобто дисципліни, які були наприклад у групі КІ-11-2, передаються групі КІ-12-2.

Таким чином формується розклад занять на поточний рік. У цей розклад можливо вносити правки щодо назв дисциплін та ПБ викладачів.

2. За закінченням навчального року, проводиться архівування у базу даних поточного розкладу занять.

За допомогою точних (класичних) методів і алгоритмів цілочисельного програмування можливе одержання точної математичної моделі, що відповідає всім обмеженням, але в силу NP-складного характеру завдання складання розкладу ця модель буде громіздкою й складною. Щоб уникнути цього, застосовується імітаційне моделювання. У цьому випадку алгоритм оперує безпосередньо розкладом і списком занять, які необхідно включити в розклад (навчальним планом). Процес складання починається з порожнього розкладу, коли всі заняття перебувають у списку неврахованих занять.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46



кількість ітерацій. При реалізації алгоритму, заснованого на принципах імітаційного моделювання, особлива увага приділяється розробці евристичних правил вибору чергового заняття зі списку, визначення найкращої для нього позиції в розкладі й оцінці одержуваного розкладу.

До позитивних рис такого підходу можна віднести можливість детального обліку специфіки розв'язуваного завдання у випадку складання розкладу для конкретного ЗВО. Однак при цьому сильно обмежується можливість застосування розробленої системи в інших навчальних закладах. Крім того, очевидно, знадобиться вносити істотні зміни в алгоритм при незначних внутрішніх змінах у ЗВО.

Так як алгоритм заснований на діях, чинених диспетчером при складанні розкладу, то існує можливість організації діяльного діалогу між користувачем і системою при пошуку оптимального розкладу.

Підходи до автоматизації складання розкладу. У багатьох університетах розклад формується в ручному режимі, складається воно на картонних планшетах, дрібним почерком олівцем на них наноситься сітка розкладу. У такому ж виді вони поширюються по факультетах і кафедрам. Знайти потрібну інформацію або зробити певну вибірку в такій ситуації вкрай важко. Тому як мінімум необхідно автоматизувати введення й зберігання інформації про розклад занять.

Програмні продукти дозволяють досить успішно формувати розклад занять в автоматичному режимі, але при цьому губляться досвід і стійка звична структура розкладу, отримані при ручному складанні.

Крім того, такі програми найчастіше використовують локальний підхід, тобто автоматизацію тільки одного відділу, відповідального за складання розкладу. Співробітникам цього відділу потрібно провести трудомісткий процес введення вихідної інформації в єдину базу даних.

Співвідношення обсягу вхідних і вихідних даних настільки велике, що мають місце необґрунтовані втрати часу, що виникають при підготовці значного

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

обсягу вхідних даних тільки для складання розкладу. Можливість же використання бази даних для рішення інших завдань відсутня.

Для рішення цієї проблеми необхідно поставити в обов'язку уведення кожного блоку інформації відповідним службам, безпосередньо відповідальним за ці дані, які будуть контролювати, підтримувати актуальність і відповідати за інформацію, що вводиться. Відомості про викладачів – відділ кадрів; навчальні плани й дисципліни – учбово-методичний відділ; групи й студенти – деканати факультетів. Крім того, необхідно зацікавити ці служби в уведенні даних в інформаційну систему, наприклад, можливістю автоматизувати їхній бізнес-процеси, налагодити пошук і усунення критичних помилок, а також можливістю формувати різні вибірки, використовуючи консолідовані дані з різних блоків інформації.

Таким чином, структурна схема системи прийме вид, представлений на рисунку 3.1. Це і є принципова схема системного підходу до автоматизації складання розкладів ЗВО, тобто автоматизації всіх бізнес-процесів, пов'язаних із предметною областю розкладу. І саме це дозволить автоматизувати рутини, а остаточний вибір при ухваленні рішення залишити за користувачем.

При локальному підході процес автоматизації складання розкладу досить трудомісткий. До позитивних моментів можна віднести, що розробити інформаційну систему, що реалізує даний підхід, вуз може самотужки, у досить короткий строк і з мінімальними фінансовими витратами.

База даних містить 3 довідники, що зберігають необхідний для складання розкладу мінімум інформації. Головні завдання при розробці інформаційної системи:

- автоматизація уведення розкладів занять із урахуванням специфіки ЗВО;
- автоматизація пошуку при складанні розкладу;
- надання розкладів занять користувачам за допомогою зручного web-інтерфейсу.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

Дана інформаційна система являє приклад окремої частини всього навчального процесу. При цьому, за допомогою убудованих засобів експорту-імпорту існує можливість взаємозв'язку між іншими об'єктами діяльності університету.

У результаті аналізу існуючих методів і підходів складання розкладу можна зробити вивід про економічну доцільність застосування повністю автоматизованих систем складання розкладів у середніх і великих ЗВО при побудові точних математичних моделей. Найбільш ефективним рішенням завдання є застосування системи діалогового процесу складання розкладу, побудованого з використанням системного підходу і який є частиною корпоративної управлінської інформаційної системи. Цей підхід забезпечить зберігання всієї необхідної й актуальної інформації для складання розкладу, а остаточне рішення буде приймати співробітник відділу по складанню розкладів.

Однією із цілей переслідуваних при проведенні автоматизації ЗВО є створення єдиної інформаційної системи. Наявність єдиного центра зберігання інформації (бази даних) з метою мінімізувати функції окремих користувачів – не менш важливе завдання при створенні автоматизованих систем. У зв'язку з наявністю у вищих навчальних закладах єдиної бази даних, одним з головних вимог до розроблювальної системи є можливість інтеграції в дану базу, для використання інформації, що зберігається в ній, про студентів, кафедри й навчальні плани.

Короткий опис призначень основних об'єктів:

– «База даних» – об'єкт, що містить інформацію про спеціальності, групи, викладачів, навчальні плани, додаткову довідкову інформацію.

– «Користувальницький інтерфейс» – об'єкт, що забезпечує високу інформативність виведеної на екран інформації, що організує зручність її виводу й обробки користувачем автоматизованої системи.

Також однією з вимог, пропонованих до розроблювальної системи, є створення й налаштування зручного користувальницького інтерфейсу, що

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

забезпечує легке сприйняття й обробку інформації, а також мінімізуючого користувальницькі операції.

Розроблювальна система при розрахунку навантаження й штатної чисельності кафедр університетів урахує прийняті норми часу у ЗВО.

Звітні форми надають адміністратором докладну інформацію:

- навантаження спеціальностей і кафедр за рік;
- нормативній кількості штатних одиниць по кафедрах за певний проміжок часу;

Таким чином, методами досягнення поставлених завдань можна вважати:

- створення єдиної бази даних автоматизованої системи закладу вищої освіти;
- створення й налаштування користувальницького інтерфейсу;
- генерація звітних форм.

Вибір системи керування базами даних є одним з важливих етапів при розробці автоматизованої системи розкладу занять. Обраний програмний продукт повинен задовольняти як поточним, так і майбутнім потребам навчального закладу, при цьому варто враховувати витрати розробку й налаштування необхідного програмного забезпечення, а також навчання персоналу.

Найбільш правильний підхід при виборі СУБД заснований на оцінці того, якою мірою існуючі системи задовольняють основним вимогам створюваного проекту інформаційної системи.

Існує кілька критеріїв вибору системи керування базами даних:

- моделювання даних;
- особливості архітектури й функціональні можливості;
- контроль роботи системи;
- особливості розробки додатків;
- продуктивність;

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

- надійність;
- вимоги до робітничого середовища;
- змішані критерії.

При розгляді критерію моделювання даних звичайно звертають увагу на використовувану модель даних. Існує безліч моделей, найпоширеніші з них – ієрархічна, мережна, реляційна, об'єктно-реляційна й об'єктна. Питання про використання тієї або іншої моделі повинно зважуватися на початковому етапі проектування інформаційної системи. Важливим критерієм є засоби пошуку, підтримувані типи даних, закладені в систему, а також наявність і можливість розширення типів.

Критерій, що визначає особливості архітектури й функціональні особливості, припускає оцінку:

- мобільності системи, її незалежність від середовища, у якій вона працює;
- масштабованості, можливості відповідати росту інформаційної системи;
- розподіленості, можливості керування розподіленими базами даних;
- мережних можливостей, можливостей використання системами широкого діапазону мережних протоколів і служб для роботи й адміністрування.

Критерій контролю роботи має на увазі контроль використання пам'яті комп'ютера, коли система може управляти використанням, як оперативної пам'яті, так і дискового простору, що може виражатися, наприклад, у стиску баз даних, або видаленні надлишкових файлів. Багато сучасних систем містять у собі можливості самоконфігурування, як правило, що опираються на результати роботи сервісів самодіагностики продуктивності. Дана можливість дозволяє виявити слабкі місця конфігурації системи й автоматично настроїти її на максимальну продуктивність.

Багато виробників СУБД випускають також засоби розробки додатків для своїх систем. Як правило, ці засоби дозволяють щонайкраще реалізувати всі

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

можливості сервера, тому при аналізі СУБД варто розглянути також і можливості засобів розробки додатків. Деякі системи мають засоби автоматичного проектування, як баз даних, так і прикладних програм.

Підтримка великої кількості національних мов і можливості розробки web-інтерфейсу розширюють область застосування системи й додатків, побудованих на їх основі, а наявність широкого спектра використовуваних мов програмування підвищує доступність системи для розроблювачів і істотно впливає на швидкодію й функціональність створюваних додатків.

Для тестування продуктивності застосовуються різні засоби, і існує безліч тестових рейтингів. Одним із сам популярних і об'єктивних є ТРС-аналіз продуктивності систем. Показник ТРС – це відношення кількості запитів оброблюваних за якийсь проміжок часу до вартості всієї системи.

Наявність можливості забезпечення паралельної обробки даних у СУБД грає не менш важливу роль при виборі системи. Можна виділити два підходи: розпаралелювання обробки послідовності запитів на кілька процесорів, або використання декількох комп'ютерів-клієнтів, що працюють із однієї БД, які поєднують у так званий паралельний сервер.

Бажано, щоб обрана система дозволяла робити оптимізацію запитів.

Критерій надійності системи включає можливість:

- відновлення системи після збоїв;
- резервного копіювання;
- відкату змін;
- наявність багаторівневої системи захисту.

Немаловажливо визначити й вимоги до робітничого середовища такі як:

- підтримувані апаратні платформи;
- мінімальні вимоги до встаткування;
- максимальний розмір адресуємої пам'яті;
- операційні системи, під керуванням яких здатна працювати СУБД.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

До змішаних критеріїв можна віднести:

- якість і повноту документації;
- локалізованість, можливість використання національних мов;
- модель формування вартості виробником СУБД;
- стабільність виробника;
- поширеність СУБД.

Чіткий і глибокий порівняльний аналіз на підставі перерахованих вище критеріїв у кожному разі допоможе раціонально вибрати підходящу систему для конкретного проекту, і витрачені зусилля не будуть даремними. Перелік критеріїв допоможе усвідомити масштабність завдання й виконати її адекватну постановку.

### 3.3 Розробка функціональної схеми

Функціональна схема розробленої системи зображена на рисунку 3.2. Програма призначена для автоматизації процесу складання розкладів занять у ЗВО. Вона володіє по-перше, можливістю редагування розкладу, а по-друге його автоматичного добудування. Інтерфейс програми, крім зручності подання інформації на екрані комп'ютера, дозволяє користуватися програмою без спеціальної підготовки.

Функціональна схема складається з наступних блоків:

- Блок складання розкладу на два тижні з обліком "парних-непарних".
- Блок гнучкого налаштування змін.
- Блок складання розкладу для занять у різних корпусах розташованих далеко друг від друга.
- Блок розподілу групи на різні типи підгруп, наприклад по іноземних мовах і по заняттям у комп'ютерних класах.

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

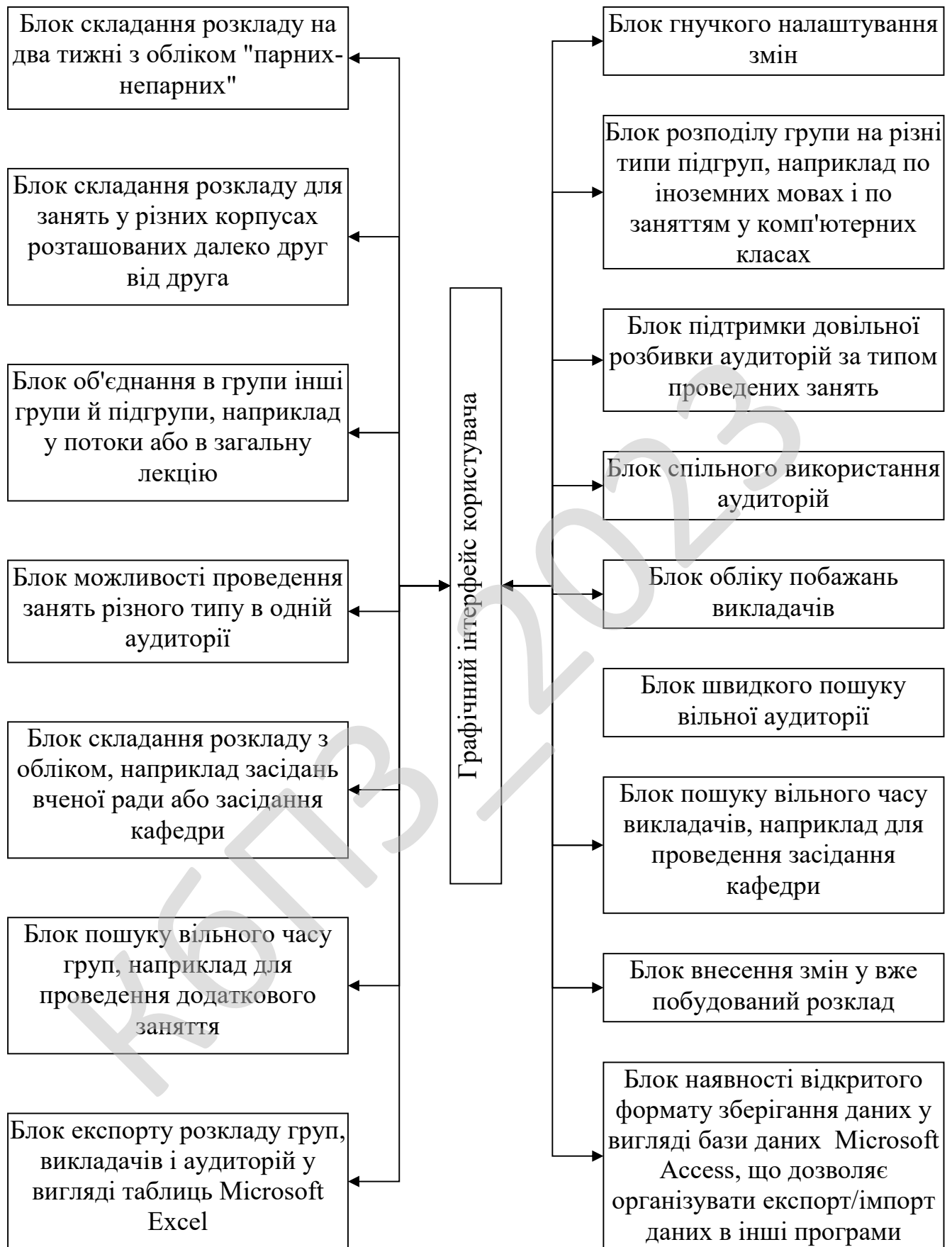


Рисунок 3.2 – Функціональна схема системи

– Блок об'єднання в групи інші групи й підгрупи, наприклад у потоки або в загальну лекцію.

– Блок підтримки довільної розбивки аудиторій за типом проведених занять.

– Блок можливості проведення занять різного типу в одній аудиторії.

– Блок спільного використання аудиторій.

– Блок обліку побажань викладачів.

– Блок складання розкладу з обліком, наприклад засідань вченої ради або засідання кафедри.

– Блок швидкого пошуку вільної аудиторії.

– Блок пошуку вільного часу викладачів, наприклад для проведення засідання кафедри.

– Блок пошуку вільного часу груп, наприклад для проведення додаткового заняття.

– Блок внесення змін у вже побудований розклад.

– Блок експорту розкладу груп, викладачів і аудиторій у вигляді таблиць Microsoft Excel.

– Блок наявності відкритого формату зберігання даних у вигляді бази даних Microsoft Access, що дозволяє організувати експорт/імпорт даних в інші програми.

Ми розглянемо основні типові структури, що включають дані навчальних планів спеціальностей, кафедр і професорсько-викладацького складу, а також підсистему зберігання різної довідкової інформації.

### **Таксономія (Тахоному)**

Таксономія – це убудований класифікатор матеріалів, що додаються. Таксономія дозволяє створювати терміни й підтерміни, які входять у словники. Таксономія застосовується для створення структури сайту, для класифікації матеріалів, що додаються, по розділах/категоріям/типам. Користувачі з

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

відповідними правами можуть додавати словники й терміни на сторінці таксономії. Сторінки зі списками термінів надають можливості по настроюванню кожного терміна і їх упорядкування. Терміни можуть вкладатися друг у друга являючи собою ієрархію з підтермінів. Кожний словник має набір необхідних полів.

У рамках автоматизованої системи використовується 3 типи словників: вид занять, викладач, факультет.

До складу словника «вид занять» входять терміни:

- лабораторне заняття;
- лекція;
- польова практика;
- практичне заняття;
- виробнича практика;
- семінар.

До складу словника «Викладач» входять терміни із професорсько-викладацькому составом. Кожний термін містить наступні поля:

- П.І.Б. викладача.
- Резюме.
- Посада.
- E-mail.
- Телефон.
- Факультет.
- Кафедра.

Словник «Факультет» включає весь перелік спеціальностей і вїм, що підкоряється, навчальним групам.

### **Нода (Node)**

Нода є структурною одиницею матеріалу (контенту). Стандартні типи матеріалів, такі як замітка (story), запис блога (blogs), сторінка (page) є нодами. Для створення навчальних занять ми розробили власний тип нод з назвою «Розклад занять». Основні типи полів і їхнє призначення продемонстровані в таблиці 3.1.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Таблиця 3.1 – Структура ноди «Розклад занять»

Мітка	Машинне ім'я	Тип поля	Віджет
Назва	Title	Елемент модуля Node	
Опис заняття	body	Текст із анонсом	Текстова область
Початок	field_date_rasp	Дата	Pop-up calendar
Закінчення	field_date_rasp_end	Дата	Pop-up calendar
Викладач	field_teacher	Посилання на термін	Вибір зі списку
Місце проведення	field_adress	Список (текст)	Вибір зі списку
Аудиторія	field_place	Текст	Текстове поле
Факультет	field_faculty	Посилання на термін	Вибір зі списку
Кафедра	field_kafedra	Посилання на термін	Вибір зі списку
Вид занять	field_course	Список (текст)	Вибір зі списку

### Подання (views)

Модуль Views дозволяє створювати й управляти списками змісту (документами, користувачами, таблицями з додатковими полями).

У загальному значенні, Views – це інструмент для створення запитів, що дозволяє створювати запити, виконувати їх і виводити отриманий результат різноманітними способами.

Крім створення списків для подання якої-небудь інформації, за допомогою видів можна генерувати звіти, виводити колекцій зображень, створювати сторінки для керування різним змістом і використовувати види для будь-яких інших цілей.

У рамках цілей і завданням магістерської роботи нами був створений окремий views, що використовує плагін FullCalendar jQuery для відображення сітки розкладу занять і виводу інформації з типу матеріалу «Розклад занять» по даті з використанням взаємозалежних фільтрів таксономії «Спеціальність – Група».

Інтерфейс надає користувачеві автоматизованої системи можливість у зручній формі обробляти необхідну інформацію.

При відкритті головної сторінки веб-інтерфейсу користувачеві надається можливість вибору напрямку роботи:

- вибір факультету й групи;
- перегляд сітки розкладу занять по місяцях, тижням і дням;
- перегляд докладної інформації про навчальні заняття;
- перегляд всіх занять по даному предметі в межах навчального періоду;
- перегляд контактної інформації про професорсько-викладацький склад.

Перехід до роботи з обраних напрямків здійснюється натисканням на посилання лівою кнопкою миші.

Кожне навчальне заняття позначається відповідним маркером-кольором, що наочно позначає тип заняття: лекція, лабораторне заняття, семінар, практичне заняття, виробнича практика, польова практика.

При натисканні на відповідне навчальне заняття, можна побачити інформацію про час початку й закінчення заняття по досліджуваному предметі, аудиторію проведення заняття, місце проведення, тип навчального заняття, закріпленого викладача.

При натисканні на пункт «Викладач», система дає можливість переглянути контактну інформацію про викладача й весь навчальний план. Контактна інформація містить у собі Ф.И.Об викладача, посаду, e-mail, контактний телефон, аудиторію, спеціальність і загальне резюме.

У системі існує один суперкористувач (admin) у якого немає обмежень на всі дії в системі, він може редагувати й видаляти будь-які матеріали, управляти конфігурацією проекту й обмежувати дії інших користувачів.

Анонімні користувачі й зареєстровані користувачі є ролями. Таким чином, суперкористувач (admin) може додавати необмежену кількість ролей, у яких можна призначити права доступу до різних областей автоматизованої системи.

Для різних типів ролей передбачена можливість роботи з певними спеціальностями й групами. В адміністративному інтерфейсі було розроблено й кілька різних інструментів для роботи з розкладом занять.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Для швидкого редагування змін у розкладі занять є візуальний редактор, у якому можна швидко змінити час початку заняття, дату й аудиторію. Кожне навчальне заняття має власний унікальний ідентифікатор, що дозволяє використовувати ті самі назви навчальних предметів на різних факультетах, при цьому для кожного словника й терміна настроюються відповідні права на створення й редагування матеріалів.

Таким чином, розроблена система спрощує процес перегляду актуального розкладу занять для студентів і дає можливість співробітникам ЗВО оперативно розраховувати навантаження на кафедри на основі аналізу наявних навчальних планів спеціальностей.

Результати роботи системи надалі можна використовувати для планування аналізу структур навантаження, а також планування структурної доробки й уніфікації наявних навчальних планів.

У результаті проробленої роботи була розроблена автоматизована система розкладу занять навчального закладу з метою її впровадження в навчальний процес університету.

Дана система має зручний користувальницький інтерфейс, що дозволяє легко освоїти роботу в програмі, гнучкість же програмного коду якщо буде потреба дозволить задовольнити зростаючі вимоги до системи.

Переслідувана споконвічно ціль автоматизації процесу розкладу занять досягнута. Розроблену автоматизовану систему можна інтегрувати в єдину інформаційну систему ЗВО, що, безсумнівно, позитивно позначиться на роботі користувачів при аналізі структури навантаження, при плануванні структурної доробки й уніфікації наявних навчальних планів.

Розроблена система дозволить підвищити швидкість обробки інформації, скоротить строки формування звітів і заощадить час роботи користувача користувачів.

Автоматизація процесу адміністрування розкладу занять і гнучкість розробленої автоматизованої системи дають переваги при її використанні в системі утворення, поліпшуючи при цьому діяльність персоналу, а разом з тим і підвищуючи якість надаваного ЗВО утворення.

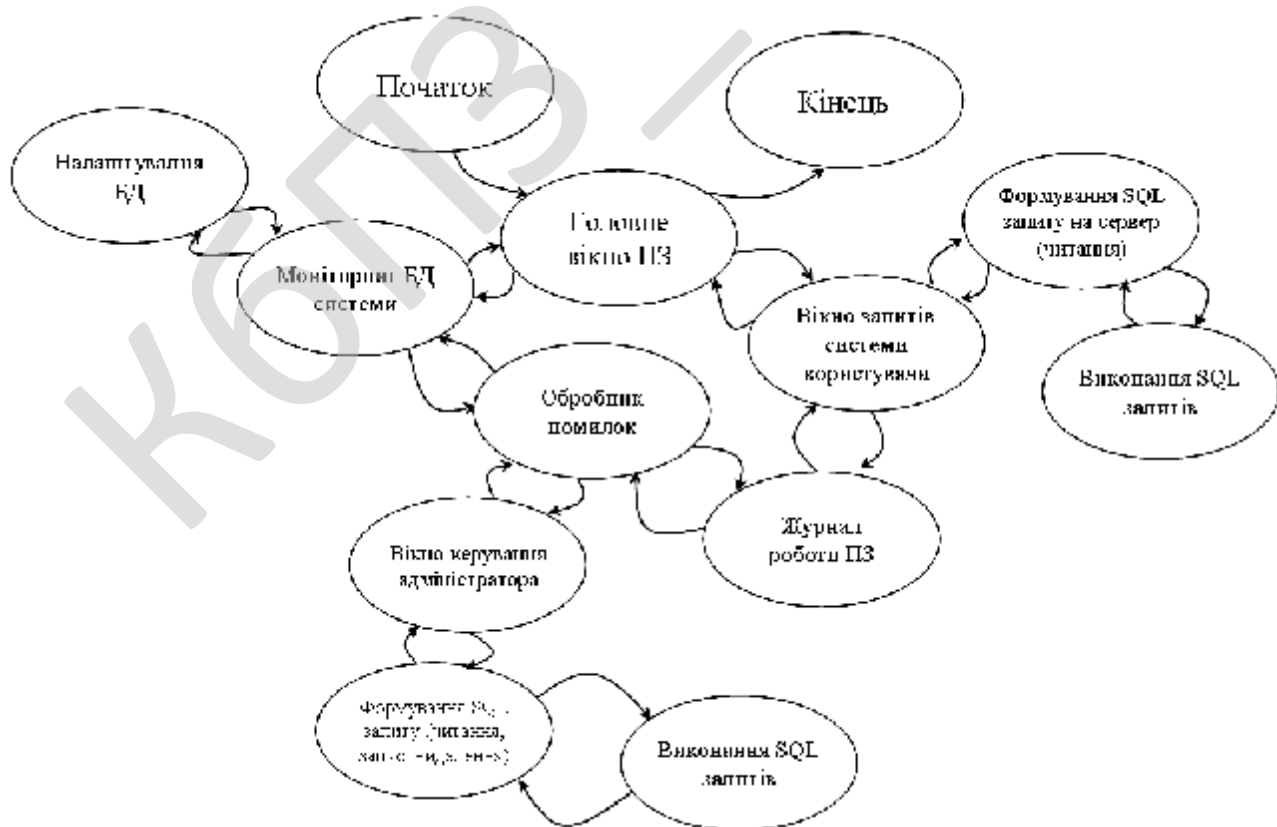
					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Модульна реалізація розробленої системи автоматизованого складання розкладу в структурі загальної автоматизованої системи ведення документообігу забезпечує можливість загального використання баз даних і впровадження загальної політики захисту інформаційного забезпечення системи.

Розроблена система реалізована засобами сучасних технологій з метою автоматизації організаційних процесів і забезпечення можливості одночасного віддаленого доступу користувачів до інформаційних ресурсів у процесі формування службової документації.

### 3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. Після початку роботи розробленого ПЗ ми потрапляємо до головного вікна ПЗ далі можемо потрапити до вікна запитів системи користувача, формування SQL запиту на сервер (читання) та виконання SQL запитів.



### Рисунок 3.3 – Діаграма взаємодії процесів

Крім цього також є перехід до моніторингу БД системи з можливістю налаштування БД та через обробник помилок та ідентифікацію потрапити до вікна керування адміністратора з подальшим формуванням SQL запиту адміністратора (читання, запис, видалення) та виконання SQL запитів.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

КБПЗ-2023

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62



- Ідентифікація пройдена? (запит).
- Читання налаштувань ПЗ.
- Виведення головного вікна ПЗ.
- Очікування дії користувача.
- Є дія користувача? (запит).
- Створення запиту користувача.
- Підпрограма журналу роботи ПЗ.

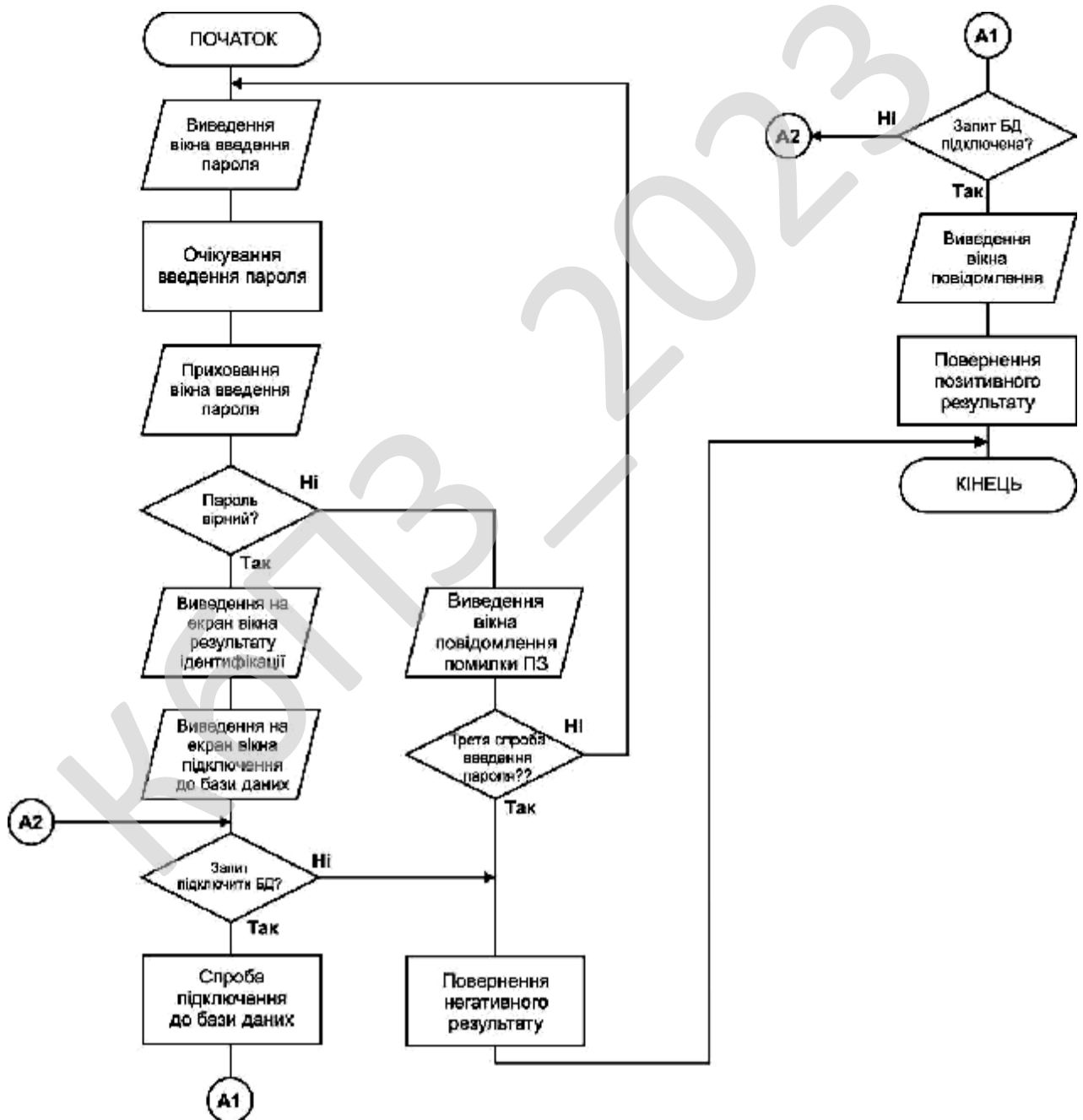


Рисунок 4.2 – Блок-схема підпрограми ідентифікації

- Запит адміністратора? (запит).
- Обробка дій користувача з правами адміністратора.
- Формування SQL запиту (читання, запис, видалення).
- Підпрограма виконання SQL запитів.
- Запит користувача?
- Обробка дій користувача з правами тільки на читання.
- Формування SQL запиту на сервер (читання).
- Підпрограма виконання SQL запиту та виклик журналу роботи.
- Запит WM\_CLOSE?
- Приховання головного вікна ПЗ, знищення динамічних структур.

На рисунку 4.2 зображено роботу підпрограми ідентифікації. Де відбувається наступне:

- Виведення вікна введення пароля.
- Очікування введення пароля.
- Приховання вікна введення пароля.
- Пароль вірний?
- Виведення на екран вікна результату ідентифікації.
- Виведення на екран вікна підключення до бази даних.
- Запит підключити БД?
- Спроба підключення до бази даних.
- Запит БД підключена?
- Виведення вікна повідомлення.
- Повернення позитивного результату.

На рисунку 4.3 зображено роботу підпрограми журналу роботи ПЗ. Де відбувається наступне:

- Пошук файлу журналу роботи ПЗ.
- Запит файл знайдено?
- Перевірка на відповідність.
- Запит файл відповідний?

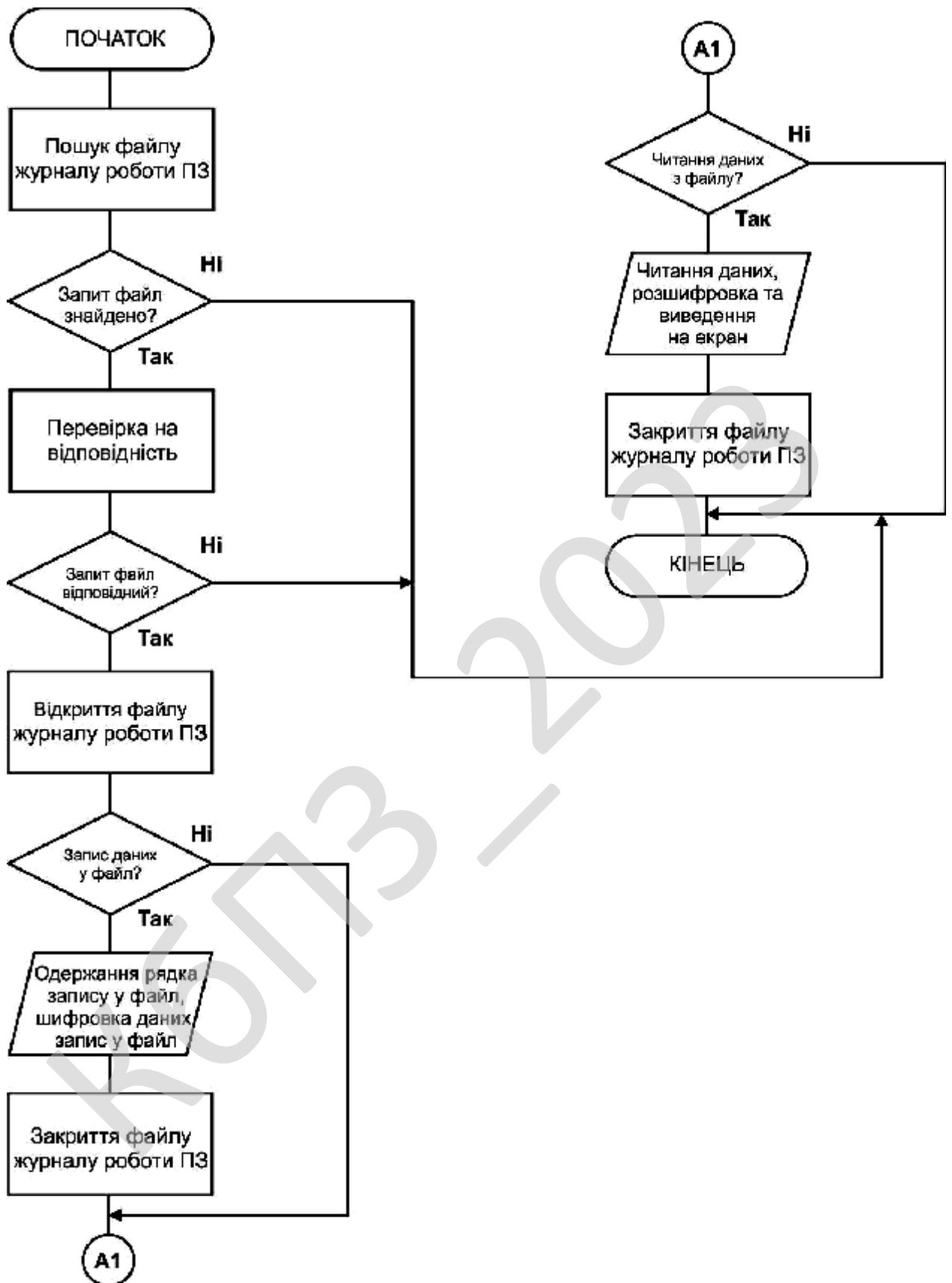


Рисунок 4.3 – Блок-схема підпрограми журнал роботи ПЗ

- Відкриття файлу журналу роботи ПЗ.
- Запис даних у файл? (запит).
- Одержання рядка запису у файл, шифровка даних, запис у файл.
- Закриття файлу журналу роботи ПЗ.
- Читання даних з файлу? (запит).
- Читання даних, розшифровка та виведення на екран.
- Закриття файлу журналу роботи ПЗ.

**Опис алгоритмів функціонування системи.** Особливу роль у створенні програми грає модуль створення звітів, преддрукарської підготовки та друку.

Взятий за основу програмний продукт QuickReport ідеально підходить до наших цілей це набір компонентів для Delphi XE5 що дозволяє побудувати звіт з бази даних швидко й просто.

Дозволяє не тільки віддрукувати звіт, але й переглянути його на екрані, дозволяє користувачеві перевірити результат без псування паперу й експортувати звіт в інші формати, такі як простий ASCII текст, текст розділений комами (CSV) і в HTML.

QuickReport написаний на Delphi XE5 і знає все про модель Delphi XE5 обробки баз даних. Всі компоненти QuickReport розташовані на закладці QReport палітри компонентів Delphi XE5.

Розглянемо використані компоненти при розробці магістерської роботи з короткою вказівкою сфери застосування.

TQuickRep. Це найбільш важливий компонент, це контейнер для всіх інших друкованих компонентів у магістерській роботі використався один глобальний TQuickRep.

Компонент виглядає як аркуш паперу, на якому буде форма попереднього перегляду. Його властивість Page дозволяє встановити розміри паперу, у той же час властивість Dataset визначає джерело даних для звіту. TQuickReport схожий на TDataModule – це спеціальний тип форми, що ніколи не показується користувачеві програми. Можливо, використати TQuickReport на псевдо-формі замість компонента TQuickRep – ніякої різниці в їхніх методах, властивостях і подіях немає.

Компоненти секції (Band) – це компоненти-контейнери, що представляють собою горизонтальні смужки у звіті в магістерській роботі використовуються для оформлення звітності роботи студентів.

TQRSubDetail. Це секція для відбиття даних в master/detail зв'язках. Це дає можливість зробити її майстром для іншої секції detail, і в такий спосіб створити кілька рівнів деталізації для зв'язку з таблицями магістерської роботи.

TQRStringsBand. Даний тип секції представляє механізм створення звітів без використання TDataSet.

TQRBand. Базовий тип секції, що може грати різні ролі залежно від властивості BandType.

TQRChildBand. Використовує об'єкт TQRChildBand коли потрібно розширити існуючу секцію.

Наприклад, при розміщенні декількох компонентів TQRMemo у секції, і додаванні TQRLabel. Оскільки об'єкт TQRMemo може займати більше одного рядка, залежно від змісту, то неефективно розміщати мітку на самій секції. Замість цього об'єкт TQRChildBand, і розміщає мітку на ньому.

TQRGroup. Секція яка друкується при зміні вираження, звичайно при зміні значення поля з бази даних. Дана секція призначена для групування однакових записів. Наприклад, для друку списку адрес.

TQRDBText. Еквівалент компонента TDBText – використовується для відображення вмісту полів бази даних в численних варіаціях на всіх формах розробленої програми. Схоже на стандартні компоненти, але призначені для використання з QuickReport, TQRDBText використовує властивість DataSet для вказівки джерела даних. Звичайні data-aware компоненти використовують властивість DataSource, що вимагає використання окремого компонента TDataSource для зв'язку з набором даних.

TQRExpr. використовується для відображення 'виражень' для попередньої обробки даних перед друком. Наприклад, можливо об'єднати всі частини імені замовника, поля в таблиці customer "Title", "Forename" і "Surname". Просто







```
function Show(AQRPrinter:TQRPrinter)
:TWinControl; override;
function ShowModal(AQRPrinter : TQRPrinter): TWinControl; override; end;
Реалізуємо методи TQRCustomPreviewInterface нашого класу
```

**TStadyWorkPreview, які виглядають так:**

```
//Вивід на екран вікна
Function TQRStadyWorkPreviewInterface.Show(AQRPrinter:
TQRPrinter):TWinControl;

var
frm:TStadyWorkPreview;
begin
frm:=TStadyWorkPreview.Create(Application,AQRPrinter);
frm.Show;
Result := frm;
end;
//Вивід на екран вікна (модально)
function TQRStadyWorkPreviewInterface.ShowModal(
AQRPrinter: TQRPrinter): TWinControl;
var
frm:TStadyWorkPreview;
begin
frm:=TCustPreview.Create(Application, AQRPrinter);
frm.ShowModal;
Result := frm;
end;
//Створення вікна
Constructor TStadyWorkPreview.Create(AOwner: TComponent;
AQRPrinter: TQRPrinter);
begin
inherited Create(AOwner);
fQRPrinter := AQRPrinter;
QRPreview1.QRPrinter := AQRPrinter;
end;
//Видалення вікна
procedure TStadyWorkPreview.CustPreviewClose(Sender: TObject;
var Action: TCloseAction);
begin
fQRPrinter.ClosePreview(Self);
Action := caFree;
end;
```

Для реєстрації альтернативного перегляду необхідно викликати функцію RegisterPreviewClass, з QRPrntr модуля. Виклик виглядає в такий спосіб: RegisterPreviewClass(TQRStadyWorkPreviewInterface).

Розроблені класи. Розглянемо загальну класифікацію алгоритмів та їхню необхідність у застосуванні на практиці. Алгоритми машинної графіки можна розділити на два рівні: нижній і верхній. Група алгоритмів нижнього рівня призначена для реалізації графічних примітивів (ліній, окружностей, заповнень і т.п.). Ці алгоритми або подібні їм, відтворені в графічних бібліотеках мов високого рівня або реалізовані апаратно в графічних процесорах робочих станцій (Silicon Graphics і ін.).

Вивід об'єктів в алгоритмах верхнього рівня забезпечується примітивами, що реалізують алгоритми нижнього рівня, тому не можна ігнорувати проблему вибору й розробки ефективних алгоритмів нижнього рівня.

При складанні блок-схем програмного забезпечення й наробітку алгоритмів взаємодії я зіткнувся з масою проблем, які вимагали наробітку процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи. Далі розглянемо основні алгоритми.

Розглянемо код комбінованої перевірки розроблений у магістерській роботі. При ініціалізації перевіряється присутність певного ключа, якщо його немає – видається відповідне повідомлення й на цьому дії припиняються, якщо ключ присутній, то з'явиться повідомлення про вдале виявлення програми, також перевіряється присутність уже встановленої програми, із пропозицією її видалити.

```
//Ініціалізація
function InitializeSetup(): Boolean;
var
    ResultCode: Integer;
    s, ResultStr:string;
    Result:=True;
begin
    Result:=True;
    If not RegKeyExists(HKLM, 'SOFTWARE\PROGRAM')
    then
```

						<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			73

```

begin
MsgBox('Програма не встановлена або встановлена невірно.' #13#13 'Подальша
робота майстра неможлива. ', mbError, mb_Ok);
Result:=False;
end;
else
begin
MsgBox(' Програма успішно виявлена. ' #13#13 'Установка повністю
автоматична. ', mbInformation, mb_Ok);
begin
If RegQueryStringValue (HKLM, 'SOFTWARE\Microsoft\Windows\
CurrentVersion\Uninstall\Magister_project_v1.0_is1',
'UninstallString', ResultStr)
then
begin
If ResultStr<>' '
then
begin
ResultStr:=RemoveQuotes (ResultStr);
if MsgBox(' Програма установки виявила вже встановлену програму ранньої
версії.' #13#13 ' Ви бажаєте попередньо видалити її? ', mbInformation, MB_YESNO) =
idYes
then
if not Exec (ResultStr, '', '', SW_SHOWNORMAL, ewWaitUntilTerminated,
ResultCode)
then
MsgBox('Помилка видалення. ' #13#13 '' + SysErrorMessage (ResultCode) +
'.' #13#13 'Імовірно, деінсталятор був переміщений, вилучений або
перейменований.', mbError, MB_OK);
end;
end;
end;
end;
end;
end;

```

Далі розглянемо встановлення фоновому зображення під час установки програми.

```

[Setup]
WindowVisible=yes
;Видимість
WindowResizable=no
[Files]

```

						<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			74





```

s:='Виявлений запущений екземпляр програми' #13#13
+'Для продовження установки потрібно закрити його.';
MsgBox(s, mbError, mb_Ok);
Result:= False
end;
end;
end;

```

Щоб інсталятор після завершення роботи автоматично закривав запущену з тимчасової папки програму, була розроблена наступна процедура. Процедура закриває програму й, використовуючи функцію Sleep, дозволяє вчасно вивантажити файли з тимчасової директорії (через клас вікна).

```

procedure DeInitializeSetup();
var
  h:Longint;
begin
  h:=FindWindowByClassName('Temp-Magister-Program-WINDOW');
  If h<>0
  then
    SendMessage(h, 16, 0, 0);
  begin
    Sleep(100);
    //Без функції Sleep тимчасові файли не зможуть вийти через недостачу
    //часу для вивантаження, значення вказується у мілісекундах.
    //Час для затримки визначається залежно від розміру тимчасових файлів.
  end;
end;

```

## 4.2 Захист розробленого програмного забезпечення

Дані які використовуються у даній роботі захищаються алгоритмом ДСТУ 7624:2014 («Калина»). При розробці національного стандарту ДСТУ 7624:2014 (блоковий шифр «Калина» і режими його роботи) було ухвалене рішення забезпечити прозорість проектування й використовувати консервативний підхід із застосуванням добре досліджених конструкцій, що забезпечують запас стійкості для безпечного застосування алгоритму в умовах істотного прогресу криптоаналітичних технік і засобів обробки даних.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Національний стандарт підтримує розмір блоку й довжину ключа шифрування 128, 256 і 512 біт (довжина ключа дорівнює розміру блоку або у два рази перевищує його), забезпечуючи нормальний, високий і надвисокий рівень стійкості (зараз це єдиний у світі стандарт блокового шифрування, що підтримує 512-бітові симетричні ключі). Різні варіанти забезпечують гнучкість вибору параметрів для розроблювачів систем криптографічного захисту, що дозволяє одержати як найвищий рівень швидкодії, так і найбільший запас стійкості перетворення.

Високорівнева конструкція використовує добре досліджену Square-подібну SPN-структуру, застосовувану в алгоритмах AES/Rijndael, Whirlpool, «Стрибог», «Коник» і багатьох інших. Циклове перетворення побудоване на базі таблиць підстановки (S-блоків) і множення на МДР-матрицю над кінцевим полем, забезпечуючи необхідні криптографічні властивості. Застосування саме такої конструкції дозволяє забезпечити доказову стійкість до диференціального, лінійного й ін. видам криптоаналізу, одночасно забезпечуючи ефективну реалізацію на широкому спектрі програмних і програмно-апаратних платформ. При виборі розміру МДР-матриці був прийнятий в увагу розмір кешу L1 сучасних і перспективних процесорів, що дозволило оптимізувати швидкодія програмної реалізації шифру.

Альтернативний варіант, який розглядався при розробці циклової функції, – ARX перетворення (Addition-rotation-xor). Цей підхід реалізований у шифрі SPECK (розроблений Агентством національної безпеки США й переданий у відкритий доступ), у блокових алгоритмах, на основі яких побудовано сімейство геш-функцій SHA-0,1,2 і ін. Перевагою походу є компактність і швидкодія перетворення. Але, у той же час, є й істотний недолік, пов'язаний з відсутністю методів, що дозволяють виконати строге аналітичне обґрунтування криптографічної стійкості таких розв'язків. Навіть із наймогутнішими у світі можливостями для аналізу, США кілька раз були змушено модифікувати свої стандарти гешування через знайдені уразливості: з 1993 по 1995 рр. діяв SHA-0, з

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

1995 по 2001 рр. застосовувався SHA-1, з тих пор використовується SHA-2. Через питання, що піднімаються, до стійкості й цієї версії, у США з 2008 по 2012 рр. був проведений міжнародний конкурс SHA-3. До теперішнього часу розроблений проект стандарту FIPS 202, що описує нову криптографічну геш-функцію.

Таким чином, консервативний і прозорий підхід до проектування нового національного стандарту України обумовив вибір добре перевіреної конструкції на базі S-блоків і лінійного перетворення, для якої можливо забезпечити доказову стійкість до різних видів криптоаналізу.

При порівнянні характеристик S-блоків і інших симетричних перетворень, у т.ч. нових білоруських і російських стандартів, можна відзначити, що саме національний стандарт України забезпечує найбільшу нелінійність булевих функцій, що дає додатковий запас стійкості до лінійного криптоаналізу. Ще більше значення нелінійності для взаємо-однозначної підстановки можна забезпечити, застосовуючи, наприклад, афінно-тквівалентні статечні функції в кінцевому полі, але такі перетворення, використані в AES, Camellia і ін. алгоритмах, ставлять шифр під погрозу реалізації алгебраїчної атаки (цей метод криптоаналізу був успішно застосований проти шифру Keeloq, використовуваного в системах автомобільної безпеки).

У якості схеми розгортання ключів була запропонована нова конструкція з наступними властивостями:

- забезпечення криптографічної стійкості до відомих методів аналізу, відсутність «слабких» ключів, які можуть погіршити властивості перетворення;
- зручність програмної й програмно-апаратної реалізації (для формування циклових ключів застосовуються тільки операції, використовувані при шифруванні);
- висока обчислювальна складність відновлення ключа шифрування по одному або декільком цикловим ключам.

Остання властивість забезпечує додатковий захист до атак на реалізацію, коли зловмисник намагається атакувати інженерні розв'язки (вимірюючи

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

споживаний пристроєм струм, навмисне викликаючи збої в роботі через навмисний перегрів шифратора та ін.). Ця особливість є істотною перевагою для ряду додатків, зокрема, при реалізації шифрування на смарт-картах, USB-токенах та ін., коли ключ прошитий у пристрої й повинен бути захищений від зовнішнього доступу (наприклад, у модулях доступу до платних цифрових ТБ-каналівм і ін.)

Кількість циклів шифрування залежить від довжини ключа: 10 циклів для 128-бітового, 14 циклів для 256-бітового й 18 циклів для 512-бітового ключа шифрування.

У порівнянні з іншими алгоритмами на основі Square-подібної SPN-структури, блоковий шифр «Калина» має наступні істотні конструктивні відмінності:

- початкове й кінцеве забілювання з використанням модульного додавання (264) для підвищення складності криптоаналітичних атак;

- застосування чотирьох різних S-блоків (замість одного) із властивостями для захисту від алгебраїчних атак, і при порівнянні характеристик з іншими блоковими й потоковими шифрами забезпечують найбільшу нелінійність булевих функцій (104), що дає додатковий запас стійкості перетворення;

- збільшений розмір МДР-перетворення, що поліпшує криптографічні властивості й дозволяє оптимізувати швидкодія на сучасних 64-бітових платформах;

- нову односпрямовану схему формування циклових ключів, що забезпечує захист від атак, ефективність програмної й програмно-апаратної реалізації, разом з додатковою стійкістю до методів аналізу спеціального виду.

Оцінка криптографічної стійкості до диференціального, лінійного, алгебраїчного, інтегрального й інших методам аналізу (практичний критерій) показала, що шифр є стійким при 6 циклах для 128-бітового блоку, 7 циклах для 256-бітового й 9 циклах для 512-бітового (кожний додатковий цикл забезпечує експонентний ріст складності криптоаналізу). Таким чином, шифр, що містить 10,

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

14 і 18 циклів для блоку 128, 256 і 512 біт відповідно, забезпечує захист від розглянутих видів аналізу й має істотний запас стійкості.

Крім блокового шифру, ДСТУ 7624:2014 визначає режими роботи, що відповідають як ISO 10116:2006, так і додаткові, призначені для сучасних систем криптографічного захисту IP-трафіка, прозорого шифрування носіїв інформації й ін. У стандарті визначені обсяги повідомлень, після обробки яких потрібна обов'язкова зміна ключа. Крім того, приводяться рекомендації розроблювачам, що обертають увагу на необхідність запобігання атак з використанням особливостей реалізації засобів шифрування. Зокрема, враховані особливості, що дозволяли організацію атак BEAST і CRIME/BREACH у протоколах SSL/TLS, повторне приймання повідомлення, відновлення конфіденційних параметрів на основі залежності часу шифрування від оброблюваних даних (через промахи кешу при табличній реалізації) і ін.

КБГЗ-2023

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>81</b>

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програма має простий та інтуїтивно зрозумілий інтерфейс, який зображений на рисунку 5.1. На рисунку зображено головне вікно програми. З нього видно, що інтерфейс користувача програми складається з таких логічних блоків:

- Меню: Предмети; Факультети; Налаштування; Довідка.
- Запитів системи.



Рисунок 5.1 – Головне вікно програми

– Вікна системи: Вікно розподілу групи на різні типи підгруп; Вікно підтримки довільної розбивки аудиторій; Вікно складання розкладу для занять у різних корпусах; Вікно об'єднання в групи в інші групи й підгрупи; Вікно

складання розкладу з обліком; Вікно пошуку вільного часу груп; Вікно експорту розкладу груп, викладачів і аудиторій у вигляді таблиць Microsoft Excel; Вікно складання розкладу на два тижні з обліком "парних-непарних"; Вікно налаштування змін.

На рисунку 5.2 зображено форму авторського права. Була вибрана ліцензія Freeware.

При розробці ПЗ була створена форма авторського права з ліцензій на використання програмного забезпечення.

Ліцензія на використання програмного забезпечення це вид ліцензії, що визначає умови використання виробу, яким є комп'ютерне програмне забезпечення. Ліцензія може надавати дозвіл робити з ним речі, які були б інакше заборонені законом про авторське право. Наприклад, ліцензія на використання програмного забезпечення може дати дозвіл робити копії програмного забезпечення. Власник авторського права може запропонувати ліцензію на використання ПЗ односторонньо, або як частину ліцензійної угоди на використання ПЗ з іншою стороною. Ліцензія може містити умови. Якщо Ви не дотримуєтесь умов ліцензії, то підпадаєте під дію закону про авторське право.

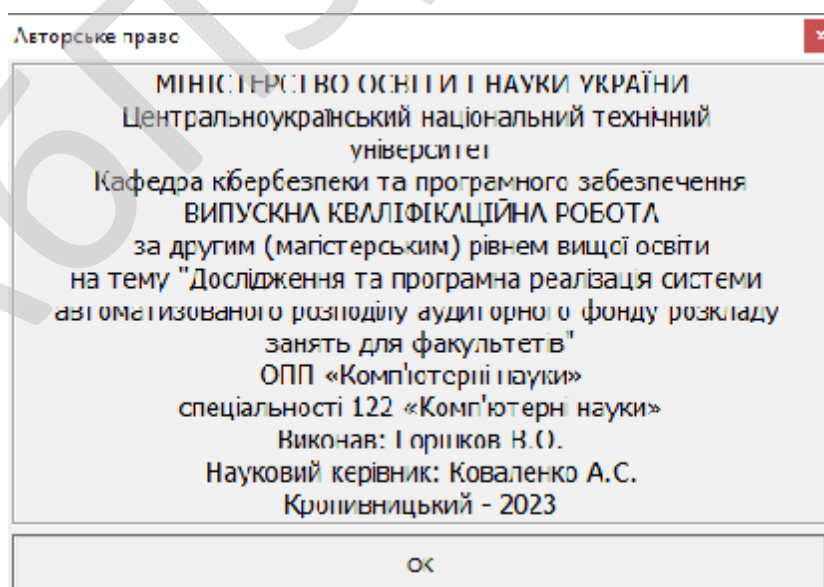


Рисунок 5.2 – Довідка

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

*Метою розробки є дослідження та програмна реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.*

*Об'єктом дослідження є процес автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.*

*Предметом дослідження є методи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.*

*Методи дослідження базуються на методах теорії оптимізації, методах математичної статистики, методах розробки програмного забезпечення.*

**Наукова новизна отриманих результатів.** У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

– Розроблено вітчизняний продукт автоматизованого розподілу аудиторного фонду розкладу занять для факультетів, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

## 7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

### 7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 48 днів (два місяці).

В магістерській роботі проведено дослідження та виконана програмна реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір та системні потреби;
- б) незалежність від встановлених на комп'ютері баз даних;
- в) зручність у користуванні та надійність

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт	N	1
2. Кількість екземплярів програм, шт	Ne	60
3. Запланований термін розробки, днів	Fpq	48 (2 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2
7. Кількість макетів вхідної інформації	–	3

Продовження таблиці 7.1

1	2	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПО для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн	–	60000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20

## 7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B \quad (7.1)$$

де А – коефіцієнт Боема, А=2,45;

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням

$$B = 1,01 + 0,001 \sum W_i \quad (7.2)$$

де  $W_i$  – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,026$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \cdot \Pi V_j, \quad (7.3)$$

де  $\Pi V_j$  – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33+0,2(B-1,01)} S, \quad (7.4)$$

де  $C$  – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);  $S$  – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПО згідно встановленим вимогам. Вибираємо в межах (25...350)%

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 182 = 306 \text{ люд/день}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	306	Ф 7.1-7.4
Впровадження	13	Д13
Всього	347	–

### 7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою

$$Ч = \frac{T_{nz} \cdot N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де  $F_{pq}$  – плановий фонд робочого часу одного спеціаліста, днів,  
 $T_{nz}$  – трудомісткість розробки програмного забезпечення люд-дні,

$$Ч = \frac{347 \cdot 1}{48 \cdot 5} = 8 \text{ ставки}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнан ня	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор-маршрутизатор	155	2	310	5
Кабельні господарства ЛОМ на 1 м.п.	2,5	100	250	4
Кабельне господарство електромережі	48	50	2400	40
Копіювальний апарат	285	2	570	10
Усього за рік:			3 <sub>ч</sub>	227

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{op}^c = \frac{3_{ч} \cdot n_{mic}}{1,2} \quad (7.6)$$

$$\Phi_{op}^c = \frac{227 \cdot 2}{1,2} = 378 \text{ год}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$C_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}} \quad (7.7)$$

$$C_{ел} = 378 / (48 \cdot 8) = 0,98 \text{ ставки}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів–електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2019, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	0,8	0,2
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,2	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,2	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	0,4	
Всього		1,6	

Продовження таблиці 7.4

Посада	Вид роботи	Час	Кількість штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	2	0,5
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	0,5	
	Контроль взаєморозрахунків з постачальниками	0,5	
Всього		4	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	0,5	0,2
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,3	
	Розміщення графіки і контенту на Інтернет сторінках	0,3	
Всього		1,6	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,2
	Верстка друкованих видань	0,2	
	Додрукова підготовка макетів	0,2	
	Розміщення графіки і контенту на Інтернет сторінках	0,2	
Всього		1,6	

Складемо штатний розклад виконавців у таблицю 7.5.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньо-місячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	15000	30000
Продакт-менеджер	0,5	12000	12000
Інженер-програміст	8	14000	224000
Інженер-електронщик	1	9000	18000
Інженер-системотехнік	0,2	9000	3600
Адміністратор мережі	0,2	11000	4400
Системний програміст	0,2	9000	3600
Дизайнер WEB	0,2	9000	3600
Інженер-верстальник	0,2	9000	3600
Бухгалтер-економіст	0,2	10000	4000
Всього за період розробки	$R_{cn}=11,7$	-	$\Phi_{роб}=306800$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де  $\Phi_{роб}$  – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{306800}{11,7 \cdot 48} = 546 \text{ грн.}$$

#### 7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

$$B_{y\delta} = R_{cn}^1 S_y C_{nl}, \quad (7.9)$$

де  $R_{cn}^1$  – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць.

$S_y$  – питома площа на одне робоче місце,  $m^2$ ,

$C_{nl}$  – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 500...1600 у.о./ $m^2$ . Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ $m^2$ . На кожне робоче місце у середньому потрібно 8  $m^2$ . З урахуванням цього:

$$B_{y\delta} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн на одне робоче місце. Тобто

$$I_{нв} = R_{cn}^1 \cdot C_m, \quad (7.10)$$

де  $C_m$  – ціна меблів для одного робочого місця, грн.

$$I_{нв} = 8 \cdot 3500 = 28000 \text{ грн}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7. Дані по оптовій ціні на обладнання та комплектуючі вибирались за комерційною пропозицією фірми Brain за 15.10.23 – джерело <http://brain.com.ua/>

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		11771
Системний блок		7771
Процесор	INTEL Pentium G6405 (BX80701G6405) 1200, 2 ядра, 4 потоки, 4.1 GHz, TDP – 58 Вт, 14nm, BOX	-
Системна плата	ASUS PRIME H510M-K DDR4, LAN – 1 Гбит/с, D-Sub (VGA), HDMI, 1 x M.2 2280, 4 x SATA 6.0 Gb/s, Micro-ATX	-
Жорсткий диск	SSD M.2 2280 240GB Apacer 240 GB, 3D TLC, M.2, PCI Express 3.0 x4	-
Оперативна пам'ять	DDR4 16GB 3200 MHz Fury Beast Black Kingston Fury (ex.HyperX) (KF432C16BB/8) DDR4, 8 ГБ, В наборі – 2	-
Відеоадаптер	Intel UHD Graphics 610	-
Корпус	Vinga CS210B, Miditower, ATX	-
Кардридер внутрішній	Transcend TS-RDF8K USB 3.0	-
інше	Клавіатура, мишка	-
Монітор	Монітор BenQ GL2450HM Black	2600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37	2800
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійного живлення	Powercom BNT-600AP USB	1400

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	8	11771	9416,8	103584,8
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2800	280	3080
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	125216,3

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400

Продовження таблиці 7.8

1	2	3	4
Група 4			
3. Обчислювальна техніка	125216	-	-
Всього по групі	125216	50	62608
Група 5,6			
4. Вимірювальні пристрої	5190	-	-
5. Транспортні засоби	143000	-	-
6. Господарський інвентар	28000	-	-
Всього по групі	176190	20	35238
7. Нематеріальні активи	60000	10	6000
Разом	$K_p = 1769406$		$A_p = 174246$

Примітка: вартість автомобіля Sens (Standard+) взята по даним з автосалону «Кіровоград-Авто», джерело <http://kirovograd-avto.ukravto.ua/catalog/tm-9/model-80/description>, складає 143000 грн.

### 7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де  $N_e$  – Кількість екземплярів програм, шт.

$$Z_o = 546 \cdot 347 / 60 = 3158 \text{ грн}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де  $H_q$  – норматив додаткової зарплати, %

$$Z_d = 3158 \cdot 10 \cdot 0,01 = 316 \text{ грн}$$

Відрахування на соціальні потреби за нормативом  $H_c=22\%$  від суми основної та додаткової зарплати

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де  $H_c$  – відрахування на соціальні потреби, %

$$C_{oc} = 0,01 \cdot 22(3158+316) = 764 \text{ грн}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом  $H_z=15\%$  від основної зарплати

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де  $H_z$  – загальногосподарські витрати, %

$$G_{ocn} = 3158 \cdot 15 \cdot 0,01 = 474 \text{ грн}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де  $Z_{M1}$  – вартість паперу, грн.,  $Z_{M2}$  – вартість запам'ятовуючих пристроїв, грн.,  $Z_{M3}$  – вартість фарби, картриджів, тонеру, грн.,  $N_e$  – кількість екземплярів програм, шт.

Згідно виданих норм приймаємо 0,5 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає  $C_n=160$  грн., визначаємо вартість паперу за період розробки  $N_M=2$  міс:

$$Z_{M1} = C_n \cdot N. \quad (7.16)$$

$$Z_{M1} = 160 \cdot 0,5 = 80 \text{ грн.}$$

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків в кількості 10 примірників:

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

$$Z_{M2} = \sum C_{\partial}, \quad (7.17)$$

де  $C_{\partial}$  – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 49 грн/шт., DVD-R LG 4,7Gb, 16x speed Cake box – 49 грн/шт.

$$Z_{M2} = 49 \cdot 10 = 490 \text{ грн.}$$

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_{з}, \quad (7.18)$$

де:  $C_{з}$  – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (80 + 490 + 1702) / 60 = 38 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ( $H_n = 15\%$ ) від основної зарплати виконавців

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де  $H_n$  – норматив витрат на освоєння нових мов програмування, %

$$O_n = 3158 \cdot 15 \cdot 0,01 = 474 \text{ грн}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ( $N_e = 60$  прим.)

$$A_m = \frac{A_p \cdot N_{\text{міс}}}{N_e \cdot 12}, \quad (7.20)$$

де  $A_p$  – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 174246 \cdot 2 / (60 \cdot 12) = 484 \text{ грн}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції

$$C_n = Z_o + Z_{\partial} + C_{\text{оц}} + \Gamma_{\text{осн}} + Z_M + O_n + A_m. \quad (7.21)$$

$$C_n = 3158 + 316 + 764 + 474 + 38 + 474 + 484 = 5708 \text{ грн.}$$

					<b>БКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

Визначимо плановий прибуток за рівнем рентабельності ( $P_p$ ) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де  $P_c$  – рівень рентабельності, %

$$P_p = 0,01 \cdot 50 \cdot 5708 = 2854 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1	2	3
1. Основна зарплата виконавців	$Z_o$	3158
2. Додаткова зарплата виконавців	$Z_d$	316
3. Відрахування на соціальні потреби	$C_{oc}$	764
4. Загальногосподарські витрати	$G_{ocn}$	474
5. Витрати на матеріали	$Z_M$	38
6. Освоєння нових операційних систем, мов програмування	$O_n$	474
7. Амортизація основних фондів	$A_m$	484
8. Повна собівартість програмного забезпечення	$C_n$	5708
9. Плановий прибуток	$P_p$	2854
10. Ціна підприємства $C_n = C_n + P_p$	$C_n$	8562
11. Податок на додану вартість $ПДВ = 0,01 \cdot H_{об} \cdot C_n$	$ПДВ$	1712,4
12. Відпускна ціна програмної продукції $Ц = Ц_n + ПДВ$	$Ц$	10274,4



$$Z_p = T_p \cdot Z_z \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де  $T_p$  – кількість годин обслуговування системи за рік, год.,

$Z_z$  – заробітна плата обслуговуючого персоналу, грн/год

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 220 годин на рік до 90 годин на рік, тому витрати на технічне обслуговування зменшилися з

$$Z_{p \text{ баз}} = 220 \cdot 100 \cdot 1,1 \cdot 1,22 = 29524 \text{ грн.}$$

до

$$Z_{p \text{ нов}} = 90 \cdot 100 \cdot 1,1 \cdot 1,22 = 12078 \text{ грн.}$$

Витрати на електроенергію визначаються з урахуванням спожитої потужності ( $P_{ел}$ ) в кіловатах, часу експлуатації технічних засобів ( $T_p$ ) в годинах та ціни однієї кіловат-години ( $C_{ел}$ ).

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел} \quad (7.24)$$

$$Z_{ел \text{ баз}} = 0,2 \cdot 7200 \cdot 3,8 = 5472 \text{ грн}$$

$$Z_{ел \text{ нов}} = 0,2 \cdot 3600 \cdot 3,8 = 2736 \text{ грн}$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизац ії %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн., за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	50	–	10274	–	5137
Всього відрахувань	-	–	10274	–	5137

## 7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою

$$E_e = (C_n - C_n) \cdot N_e - \sum_{i=1}^m E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

де:  $K_p$  – балансова вартість основних фондів розробника, грн.;  $E_p$  – розрахунковий коефіцієнт капіталовкладень.

$$E_e = (8562 - 5708) \cdot 60 - (0,05 \cdot 1408000 + 0,5 \cdot 125216 + 0,2 \cdot 176190 + 0,1 \cdot 60000) \cdot 2/12 = 142199 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p^*}{(C_n - C_n) \cdot N_e}, \quad (7.26)$$

де:  $K_p^*$  – балансова вартість основних фондів розробника без врахування вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_e = \frac{361406}{(8562 - 5708) \cdot 60 \cdot 12 / 2} = 0,36 \text{ років}$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\bar{o}} - I_n) - E_n(K_n - K_{\bar{o}}), \quad (7.27)$$

де  $I_{\bar{o}}, I_n$  – величина експлуатаційних витрат за базовим и новим варіантом відповідно,  $K_{\bar{o}}, K_n$  – об'єм капітальних вкладень за варіантами, що порівнюються

$$E_{cn} = (34996 - 19951) - 0,5 \cdot 10274 = 9908 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>103</b>

$$T_{cn} = \frac{K_n - K_0}{I_0 - I_n} \quad (7.28)$$

$$T_{cn} = \frac{10274}{34996 - 19951} = 0,7 \text{ року}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величин а
1. Кількість екземплярів програми	Прим.	60
2. Повна собівартість розробленої програми	Грн.	5708
3. Ціна розробленої програми	Грн.	8562
4. Плановий прибуток від реалізації розробленої програми	Грн.	2854
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1769406
7. Загальний прибуток від реалізації програмної продукції	Грн.	171240
8. Величина економічного ефекту при виготовленні програмної продукції	Грн.	142199
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	0,36
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	10274
11. Величина економічного ефекту у користувача програмної продукції	Грн.	9908
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,7

## 7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

КБПЗ - 2023

					VKPM-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

В кожній ІТ компанії є трудові відносини з працівниками. Згідно закону України “Про охорону праці” кожна компанія впроваджує заходи з охорони праці. Реалізується трудові відносини з вживанням необхідних засобів з охорони праці та розробки відповідних документів:

- Інструкцій з охорони праці по кожній професії і загальні.
- Положення про охорону праці.
- Накази з охорони праці.
- Журнали реєстрації та інструктажу.

Роботодавець створює відділ який працює відповідно до типового положення, яку затверджується центральним органом виконавчої влади і забезпечує виконання вимог державної політики у сфері охорони праці.

За недотриманням вимог, керівники ІТ компаній можуть бути притягнуті до відповідальності, яка виглядає у виді накладання штрафу. Якщо в результаті порушення умов охорони праці є постраждалі працівники то керівні особи ІТ компаній притягуються до кримінальної відповідальності.

### 8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Найвний в даний час в нашій країні комплекс розроблених організаційних заходів та технічних засобів захисту, накопичений передовий досвід роботи ряду обчислювальних центрів показує, що є можливість домогтися значно більших успіхів у справі усунення впливу на працюючих небезпечних і шкідливих виробничих факторів. Проте стан умов праці та його безпеки в ряді ІТ компаній ще не задовольняють сучасним вимогам. Оператори ЕОМ, оператори по

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

підготовці даних, програмісти та інші працівники ІТ стикаються з впливом таких фізично небезпечних і шкідливих виробничих факторів, як підвищений рівень шуму, підвищена температура в приміщенні, відсутність або недостатня освітленість робочої зони, електричний струм, статична електрика і інші [1].

Всі працівники пов'язані з впливом таких психофізичних факторів, як розумова перенапруга, перенапруження зорових і слухових аналізаторів, монотонність праці, емоційні перевантаження. Вплив зазначених несприятливих факторів призводить до зниження працездатності, викликає розвиток втоми. Поява і розвиток втоми пов'язане зі змінами, які виникають під час роботи в центральній нервовій системі, з гальмівними процесами в корі головного мозку. Наприклад сильний шум викликає труднощі з розпізнаванням кольірних сигналів, знижує швидкість сприйняття кольору, гостроту зору, зорову адаптацію, порушує сприйняття візуальної інформації, зменшує на 5 – 12% продуктивність праці. Тривала дія шуму з рівнем звукового тиску 90 ДБ знижує продуктивність праці на 30 – 60% [3].

Медичні обстеження працівників показали, що крім зниження продуктивності праці високі рівні шуму призводять до погіршення слуху. Тривале перебування людини в зоні комбінованого впливу різних несприятливих факторів може призвести до професійного захворювання. Аналіз травматизму серед працівників показує, що в основному нещасні випадки відбуваються від впливу фізично небезпечних виробничих факторів при заправці носія інформації на обертний барабан при зняттю кожусі, під час співробітниками невластивих їм робіт. На другому місці випадки, пов'язані з дією електричного струму [2].

### **8.3 Забезпечення електробезпеки**

Електричні установки, до яких відноситься практично все обладнання в ІТ компаніях, представляють для людини велику потенційну небезпеку, так як в процесі експлуатації, або проведенні профілактичних робіт людина може

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

торкнутися частин, що знаходяться під напругою. Специфічна небезпека електроустановок є струмоведучі провідники, корпуси стійок ЕОМ і іншого устаткування, яка під напругою в результаті пошкодження (пробою) ізоляції, не подає будь-яких сигналів, які попереджають людину про небезпеку. Реакція людини на електричний струм виникає лише при протіканні останнього через тіло людини. Виключно важливе значення для запобігання електротравматизма має правильна організація обслуговування діючих комп'ютерів, проведення ремонтних, монтажних і профілактичних робіт. При цьому під правильною організацією розуміється строге виконання ряду організаційних технічних заходів і засобів, встановлених діючими "Правилами технічної експлуатації електроустановок споживачів і правил техніки безпеки при експлуатації електроустановок споживачів" (ПТЕ і ПТБ споживачів) і "Правила установки електроустановок" (ПУЕ) Залежно від категорії приміщення необхідно вжити певних заходів, що забезпечують достатню електробезпеку при експлуатації та ремонті електроустаткування [4]. В приміщеннях з підвищеною небезпекою електроінструменти, переносні світильники повинні бути виконані з подвійною ізоляцією, або напруга живлення їх не повинна перевищувати. До таких приміщень можуть бути віднесені приміщення машинного залу, приміщення для розміщення сервісної та периферійної апаратури. В особливо небезпечних приміщеннях напруга живлення переносних світильників не повинна перевищувати 12В, а робота з електротранспортуємою напругою не вище 42В дозволяється тільки із застосуванням (діелектричних рукавичок, килимків і т.п.). Роботи без зняття напруги на струповидних частинах і поблизу них, роботи проводяться безпосередньо на цих частинах або при наближенні до них на відстань менше встановленого ПЕУ. До цих робіт можна віднести роботи з налагодження окремих вузлів, блоків. При виконанні такого роду робіт в електроустановках до 1000 В необхідне застосування певних технічних і організаційних заходів, таких як: огорожі, розташовані поблизу робочого місця та інших струмоведучих частин, до яких можливо випадковий дотик; робота в

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		108

діелектричних рукавичках або стоячи на діелектричному килимку; застосування інструменту з ізолюючими рукоятками, за відсутності такого інструменту слід користуватися діелектричними рукавичками. Роботи цього виду повинні виконуватися не менше ніж двома працівниками [4].

Відповідно до ПТЕ і ПТВ споживачам і обслуговуючому персоналу електроустановок пред'являються наступні вимоги:

- особи, які не досягли 18-річного віку, не можуть бути допущені до робіт в електроустановках;
- особи не повинні мати каліцтв і хвороб, що заважають виробничій роботі;
- особи повинні після відповідної теоретичної та практичної підготовки пройти перевірку знань і мати посвідчення на доступ до робіт в електроустановках.

У ВЦ розрядні струми статичної електрики частіше за все виникають при дотику до будь-якого з елементів ЕОМ. Такі розряди небезпеки для людини не представляють, але крім неприємних відчуттів вони можуть привести до виходу з ладу ЕОМ. Для зниження величини виникаючих зарядів статичної електрики в ОЦ покриття технологічних статей повинно бути з одношарового полівінілхлоридного антистатичного лінолеуму. Іншим методом захисту є нейтралізація заряду статичної електрики іонізованим газом. У промисловості широко застосовуються радіоактивні нітралізатори. До загальних заходів захисту від статичної електрики в ОЦ можна віднести загальні та місцеве зволоження повітря [4].

#### **8.4 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста**

Приміщення та їх розміри (площа, об'єм) повинні в першу чергу відповідати кількості працюючих і розміщених в них комплекту технічних

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

засобів. У них передбачаються відповідні параметри температури, освітлення, чистоти повітря, забезпечують ізоляцію від виробничих шумів і т.п. Для забезпечення нормальних умов праці санітарні норми встановлюють на одного працюючого обсяг виробничого приміщення не менше 15 м<sup>3</sup>, площа приміщення огорожена стінами або глухими перегородками не менше 4,5 м<sup>3</sup>.

Для експлуатації ЕОМ слід передбачати такі приміщення:

- машинний зал,
- приміщення для зберігання запасних деталей, інструментів, приладів ;
- приміщення для розміщення припливно-витяжних вентиляторів;
- приміщення для персоналу;
- приміщення для прийому-видачі інформації.

Основні приміщення робочих комп'ютерних станцій розташовуються в безпосередній близькості один від одного. Їх обладнають загальною вентиляцією і штучним освітленням. До приміщення машинного залу і зберігання магнітних носіїв інформації пред'являються особливі вимоги. Площа машинного залу повинна відповідати площі необхідної за заводськими технічними умовам даного типу ЕОМ.

У тих випадках, коли одного природного освітлення не вистачає, встановлюється суміщене освітлення. При цьому додаткове штучне освітлення застосовується не тільки в темний, але і в світлий час доби.

Штучне освітлення по характеру виконуваних завдань ділиться на робоче, аварійне, евакуаційне [3].

Рациональне колірне оформлення приміщення направлено на поліпшення санітарно-гігієнічних умов праці, підвищення його продуктивності і безпеки. Забарвлення приміщень, впливає на нервову систему людини, його настрої і в кінцевому рахунку на продуктивність праці. Основні виробничі приміщення доцільно фарбувати відповідно до кольору технічних засобів. Освітлення приміщення і устаткування повинне бути м'яким, без блиску.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

Зниження шуму, створюваного на робочих місцях внутрішніми джерелами, а також шуму проникаючого ззовні, є дуже важливим завданням. Зниження шуму в джерелі випромінювання можна забезпечити застосуванням пружних прокладок між підставою машини, приладу і опорною поверхнею. Як прокладок використовуються гума, повсть, пробка, різної конструкції амортизатори. Під настільні шумливі апарати можна підкладати м'які килимки з синтетичних матеріалів, а під ніжки столів, на яких вони встановлені, – прокладки з м'якої гуми, повсті, завтовшки 6 – 8 мм. Кріплення прокладок можливо шляхом приклеювання їх до опорних частин.

Застосовують спеціальні звукоізолюючих кожухи які не змінюють технологічний процес. Зниження шуму регулюють завдяки своєчасному змащуванню і заміні механічних вузлів шумливого обладнання.

Правильне планування приміщення є одним з важливих факторів зниження шумів при використанні і устаткуванні ЕОМ.

Зниження рівня шуму, що проникає у виробниче приміщення ззовні, може бути досягнуто збільшенням звукоізоляції огорожувальних конструкцій, ущільненням по периметрі притворів вікон, дверей [3].

Для зниження шуму, створюваного на робочих місцях внутрішніми джерелами, а також шуму, проникаючого з поза слід:

- послабити шум самих джерел (застосування екранів, звукоізолюючих кожухів);
- знизити ефект сумарної дії відбитих звукових хвиль (звукопоглинаюча поверхня конструкцій);
- застосовувати раціональне розташування обладнання;
- використовувати архітектурно-планувальні та технологічні рішення ізоляцій джерел шуму.

## 8.5 Розробка заходів з умов поліпшення охорони праці

Одним з найбільш важливих завдань при розробці нових систем виробництва і технологій, відбувається за допомогою вивчення і вирішення проблем пов'язаних із забезпеченням безпечних умов праці людини. Вивчаючи причини виробничих нещасних випадків, професійних захворювань, пожеж розробляють заходи спрямованні на усунення причин які створюють не безпеку. Одним з основних факторів безпеки праці є комфортні і безпечні умови праці [2].

Аналізуючи останні дослідження і публікації, можна виділити таких вчених як Д.П. Богиня, О.А. Грішнова, Є.П. Желібо, які зробили значний внесок в дослідженнях проблем з удосконалення системи охорони праці, вони проаналізували сучасний стан умов праці і виявили міру впливу їх на результати і ефективність виробництва. В ході досліджень виявили, що на підприємствах більше витрат припадає на пільги і компенсації, які по'язані з небезпечними шкідливими умовами праці, ніж на заходи, що запобігають виробничому травматизму і захворюваності, а також нормалізацію умов праці [1].

Всі працівники проходять інструктаж з охорони праці, правил поведінки при виникненні пожеж, надання першої допомоги потерпілим відповідно до Типового положення про навчання з питань охорони праці. До капітальних витрат відносять, такі як створення основних фондів праце охоронного призначення, вдосконалення техніки і технологій з метою поліпшення умов охорони праці, підвищувати кваліфікацію і відповідне зацікавлення працівників. Велику користь дає преміювання працівників за тривалу роботу без порушень правил охорони праці.

Керівнику треба звернути увагу на посилення контролю з організації служб охорони праці. Правильний підхід до організації, грамотне використання різних способів стимулювання працівників дають почуття надійності, стабільності і зацікавленості керівництва у працівниках [4].

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

## 8.6 Розрахункова частина

Розрахунок занулення.

Розрахунок занулення складається з трьох частин:

1. Розрахунок на відключаючу спроможність.
2. Визначення максимальної напруги на корпусі обладнання відносно землі при замиканні фази на корпус.
3. Розрахунок робочого і повторного заземлювачів.

Початкові дані:

1. Потужність електроприладів, які підлягають зануленню:

$$P = 7 \text{ кВт.}$$

2. Довжина магістрального кабеля:  $L_M = 12 \text{ м.}$

3. Довжина розгалудження (від розподільчого щита до електроприладів) :  
 $l = 25 \text{ м.}$

4. Матеріал провідників кабеля—алюміній.

5. Лінійна напруга  $U = 380 \text{ В.}$

6. Фазна напруга  $U_f = 220 \text{ В.}$

Розрахунок проводиться відповідно до схеми електромережі, яка зображена на рис. 8.1.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113



О. Оришака, А. Е. Солових, С. Е. Катеринич] ; Мін-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – 2-ге вид., перероб. та доп. – Кропивницький : ЦНТУ, 2019. – 27 с. URI: <http://dspace.kntu.kr.ua/jspui/handle/123456789/8769>

3. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення 19.10.23).

4. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. – Кропивницький: ЦНТУ, 2022. – 19 с. [Електронний ресурс]. – Режим доступу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> (дата звернення 19.09.23).

5. НПАОП 40.1-1.21-98. Правила безпечної експлуатації електроустановок споживачів – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0093-98#Text> (дата звернення 19.09.23).

6. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98> (дата звернення 19.09.23).

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		115

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.
- Досліджена система автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.
- На основі отриманих результатів досліджень створена програмна реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		116

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Delphi 10.4.1. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 7624:2014.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 9908 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,7 роки.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		117

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Горшков В.О. Дослідження та програмна реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Fernando Doglio. Skills of a Successful Software Engineer. Manning. 2022. 182 с.
3. M. Holmes He. Creating Apps with React Native. Apress Media. 2022. 445 p.
4. Maurício Aniche. Effective Software Testing. Manning Publications. 2021. 372 p
5. Priscila Heller. Automating Workflows with GitHub Actions. Packt Publishing. 2021. 216 p.
6. JJ Geewax. API Design Patterns. Manning Publications Co. 2021. 481 p.
7. Prateek Prasad. App Design Apprentice. Razeware LLC. 2020. 272 p.
8. Dawn Griffiths, David Griffiths. Head First Android Development. O'Reilly Media, Inc. 2021. 1414 p.
9. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.
10. Aaron Torres. Go Programming Cookbook Second Edition. Packt Publishing Ltd. 2019. 427 p.
11. Мелешко Є.В., Якименко М.С., Поліщук Л.І. Алгоритми та структури даних: Навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. – Кропивницький: Видавець – Лисенко В.Ф., 2019. – 156 с.
12. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.

					ВКРМ-122.23.0032.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		118

13. Knuth D. The Art of Computer Programming: Vol. 3: Sorting and Searching 2nd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 800 p.
14. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.
15. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.
16. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.
17. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». CEUR Workshop Proceedings, Volume 3312, 2022, pp. 47-58.
18. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>.
19. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143
20. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184.
21. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

22. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

23. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

24. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43.

25. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

26. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

27. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

28. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

29. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

30. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P. 707-712.

31. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P.701-706.

32. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

33. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

34. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

35. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		121

абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». Сучасні інформаційні системи, 2023, том 7, № 2, С. 49-56.

36. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А. «Дослідження нормативної документації та стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». VI міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 20-21 квітня 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 35-36.

37. Смірнов, О.А., Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.

38. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.

39. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

40. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральнoукраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.

41. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		122

42. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

43. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

44. O. Smirnov, O. Kovalenko, A. Kovalenko, S. Smirnov, V. Vialkova. The mathematical model of the testing technology for DOM XSS vulnerabilities. Scientific & practical cyber security journal (SPCSJ) Vol 2 Issue 1, 22-28 pp. [Электронный Журнал]. Georgia. Tbilisi: SCSA – 2018.

45. Oleksii Smirnov, Oleksandr Kovalenko, Jamil Al-Azzeh, Anna Kovalenko, Serhii Smirnov. Qualitative risk analysis of software development. Asian Journal of Information Technology. – Volume 17(3). – Medwell Journals. – 2018. – P. 218-230.

46. Смірнов О.А., Коваленко О.В., Коваленко А.С., Смірнов С.А. Розробка методу передтестової компіляції й розподілу доступу. Збірник наукових праць III міжнародної науково-практичної конференції “Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 19-20 квітня 2018р. – Кропивницький: ЦНТУ. – 2018. – С. 214-215

47. Smirnov Oleksii, Kovalenko Oleksandr, Kovalenko Anna, Smirnov Serhii. Method of testing the DOM XSS vulnerability. International Conference «Information technologies, systems and networks ITSН-2017». Chisinau, Republic of Moldova. 17 – 18 October 2017. – Chisinau: Academy of Sciences of Moldova, Military Academy of Armed Forces “Alexandru cel Bun”. 2017. P7.

48. Смірнов О.А., Смірнов С.А., Коваленко О.В., Коваленко А.С. Технологія тестування DOM XSS уразливості. Науково-практичний журнал кібер безпеки (SPCSJ) № 1. [Електронний журнал]. Грузія. Тбілісі: SCSA – 2017.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		123

49. Смірнов О.А., Лисенко І.А. Інформаційна технологія проектування тестових наборів з урахуванням вимог до програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 4 (44). – Полтава: ПолтНТУ. – 2017. – С. 112-115.

50. Смірнов О.А., Смірнов С.А., Рябой Д.К., Рябая О.В. Модель вузла комутації з відносними пріоритетами, резервуванням ресурсів і обліком реальної надійності обслуговуючих приладів .Збірник тез всеукраїнської науково-практичної інтернет-конференції «Автоматика та комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті». м. Кропивницький. 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 198-199.

51. Смірнов О.А., Коваленко О.В. Використання псевдобулевих методів бівалентного програмування для управління ризиками розробки програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 1 (37). – Полтава: ПолтНТУ. – 2016. – С. 98-103.

52. Смірнов О.А., Лисенко І.А. Формалізація процесу проектування тестових наборів. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 3 (48). – Харків: ХУПС. – 2016. – С.96-100.

53. Смірнов О.А., Лисенко І.А. Удосконалення методу перевірки коректності таблиць рішень для подання тестових наборів. Збірник наукових праць "Системи обробки інформації". – Випуск 8 (145). – Х.: ХУПС – 2016. – С. 77-80.

					<b>ВКРМ-122.23.0032.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		124

Додаток А  
(обов'язковий)

**Технічне завдання**

**Зміст**

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					<b>ВКРМ-122.23.0032.00.00.ТЗ</b>		
<i>Вим.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Розробив</i>	<i>Горшков В.О.</i>				<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевірів</i>	<i>Коваленко А.С.</i>			<i>М</i>			
<i>Н. Контр.</i>	<i>Коваленко А.С.</i>				<i>ЦНТУ КН-22М-2</i>		
<i>Затв.</i>	<i>Смірнов О.А.</i>						

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 33-13 від 04.08.2023 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів.

## 4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

## 5 Технічні вимоги

### 5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;

					ВКРМ-122.23.0032.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи автоматизованого розподілу аудиторного фонду розкладу занять для факультетів;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

## 5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

## 5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

## 5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					<b>ВКРМ-122.23.0032.00.00.ТЗ</b>	Арк.
						3
Вим.	Арк.	№ документа	Підпис	Дата		

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

## 5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

## 5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

### 5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

### 5.8.2 Мова програмування

Середовище Delphi 10.4.1.

					ВКРМ-122.23.0032.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

### 5.8.4 Вихідні дані

Робоча програма.

## 6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

## 7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2023 року.

## 8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинно бути розглянуто забезпечення електробезпеки.

					<b>ВКРМ-122.23.0032.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

## 9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 123 аркуші.

## 10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2023 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 19.12.2023 р.

					<b>ВКРМ-122.23.0032.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

**Міністерство освіти і науки України**  
**Центральноукраїнський національний технічний університет**

**ЗАТВЕРДЖУЮ**

Керівник випускної кваліфікаційної роботи за  
другим (магістерським) рівнем вищої освіти

\_\_\_\_\_ Коваленко А.С.

***Дослідження та програмна реалізація  
системи автоматизованого розподілу аудиторного фонду розкладу занять  
для факультетів***

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 58

Літера: РП

Кропивницький – 2023 року

## ОСНОВНИЙ ФАЙЛ ПРОЕКТУ ПЗ DATA.DPR

```

Program Data;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Вихідний код до магістерської роботи
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу
аудиторного фонду розкладу занять для факультетів

Виконав: Горшков Владислав Олегович
студент 6 курсу, групи КН-22М-2

Керівник: Коваленко А.С.
Кропивницький 2023
}

Uses // підключення
  Forms,
  Windows,
  Messages,
  prgData in 'prgData.pas',
  prgTypes in 'prgTypes.pas',
  frmProgressU in 'frmProgressU.pas' {frmProgress},
  frmFormulaU in 'frmFormulaU.pas' {frmFormula},
  frmGroupsU in 'frmGroupsU.pas' {frmGroups},
  frmStudentU in 'frmStudentU.pas' {frmStudent},
  frmReportU in 'frmReportU.pas' {frmReport},
  prgFilter in 'prgFilter.pas',
  frmFilterU in 'frmFilterU.pas' {frmFilter},
  prgFunc_Proc in 'prgFunc_Proc.pas',
  frmSplashU in 'frmSplashU.pas' {frmSplash},
  frmAboutU in 'frmAboutU.pas' {frmAbout},
  frmMainModStatU in 'frmMainModStatU.pas' {frmMainModStat},
  UkrSort in 'UkrSort.pas';

{$R *.res}

Const
  MemFileSize=127;
  MemFileName='Data';
Var // змінні
  MemHnd: HWND;
  cd: TCopyDataStruct;

Begin
  MemHnd:=CreateFileMapping(HWND($FFFFFFFF), Nil, PAGE_READWRITE, 0,
  MemFileSize, MemFileName);
  If (GetLastError<>ERROR_ALREADY_EXISTS) Then
  Begin
    Application.Initialize;
    Application.Title:=' Дослідження та програмна реалізація системи
автоматизованого розподілу аудиторного фонду розкладу занять для факультетів';
    Try
      frmSplash:=TfrmSplash.Create(Application);
      frmSplash.Show;
      frmSplash.Update;

      Application.CreateForm(TfrmMainModStat, frmMainModStat);
      Application.CreateForm(TfrmProgress, frmProgress);
    Repeat
      Application.ProcessMessages;
    Until frmSplash.CloseQuery;
    If ParamCount>0 Then
    Begin
      CurrFile:=ParamStr(1);

```

```

LoadFileData (prgData.CurrFile);
frmMainModStat.UpdateList;
frmMainModStat.cbGroupFilter.ItemIndex:=0;
frmMainModStat.UpdateStudents;
AddRescent (prgData.CurrFile);
frmMainModStat.FillRescent;
frmMainModStat.ShowTXT('Файл відкрито');
Saved:=True;
End
Else
    frmMainModStat.actNewFile.Execute;

    frmSplash.Hide;
Finally
    frmSplash.Free;
End;
Application.Run;
End
Else
Begin
    If ParamCount=0 Then
        Begin
            cd.cbData:=4;
            cd.lpData:=PChar('Act');
            SendMessage(FindWindow('TfrmMainModStat', Дослідження та програмна
реалізація системи автоматизованого розподілу аудиторного фонду розкладу занять
для факультетів '), WM_COPYDATA, 0, LParam(@cd));
            End
        Else
            Begin
                cd.cbData:=Length(ParamStr(1))+1;
                cd.lpData:=PChar(ParamStr(1));
                SendMessage(FindWindow('TfrmMainModStat', ' Дослідження та програмна реалізація
системи автоматизованого розподілу аудиторного фонду розкладу занять для
факультетів '), WM_COPYDATA, 0, LParam(@cd));
                End;
            End;
        CloseHandle (MemHnd);
    End.

```

## ФАЙЛ FRMMAINMODSTATU.PAS

```

Unit frmMainModStatU;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Вихідний код до магістерської роботи
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу
аудиторного фонду розкладу занять для факультетів

Виконав: Горшков Владислав Олегович
студент 6 курсу, групи КН-22М-2

Керівник: Коваленко А.С.
Кропивницький 2023
}
Interface // Інтерфейс

Uses // підключення
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ImgList, ActnList, ExtCtrls, TB2Dock, TB2ToolBar, TB2Item, UkrSort,
TBSkinPlus, ComCtrls, StdCtrls, TFlatComboBoxUnit, ATXPTask, prgData,
TFlatHintUnit, TFlatEditUnit, TFlatSpinEditUnit, Registry, ShellApi;

Type // типи
TfrmMainModStat=Class(TForm)
  ActionList: TActionList;
  actOpenFile: TAction;
  actNewFile: TAction;
  actSaveFile: TAction;
  actExit: TAction;
  actGroupsEdit: TAction;
  dlgOpen: TOpenDialog;
  pnlStudent: TPanel;
  sbMain: TStatusBar;
  actFormula: TAction;
  pnlAllStudents: TPanel;
  lvStudents: TListView;
  actAcceptStu: TAction;
  actPrint: TAction;
  ImageList: TImageList;
  tbSkin: TTBSkin;
  tbDock: TTBDock;
  tbMenu: TTBToolBar;
  tbiFile: TTBSubmenuItem;
  tbiFNewFile: TTBItem;
  tbiFOpenFile: TTBItem;
  tbiFSaveFile: TTBItem;
  tbiFPrint: TTBItem;
  tbiFExit: TTBItem;
  tbiOptions: TTBSubmenuItem;
  tbiOFormula: TTBItem;
  tbiSep1: TTBSeparatorItem;
  tbiSep2: TTBSeparatorItem;
  tbMain: TTBToolBar;
  xptStuData: TXPTask;
  lblFio: TLabel;
  edtFIO: TEdit;
  lblGroup: TLabel;
  edtGroup: TEdit;
  xptM1Data: TXPTask;
  lblM1T: TLabel;
  seM1T: TFlatSpinEditInteger;
  seM1P: TFlatSpinEditInteger;
  seM1V: TFlatSpinEditInteger;
  seM1A: TFlatSpinEditInteger;
  lblM1A: TLabel;
  lblM1V: TLabel;

```

```
lblM1P: TLabel;  
lblM1R: TLabel;  
seM1R: TFlatSpinEditInteger;  
xptM2Data: TXPTask;  
lblM2T: TLabel;  
lblM2A: TLabel;  
lblM2V: TLabel;  
lblM2P: TLabel;  
lblM2R: TLabel;  
seM2T: TFlatSpinEditInteger;  
seM2P: TFlatSpinEditInteger;  
seM2V: TFlatSpinEditInteger;  
seM2A: TFlatSpinEditInteger;  
seM2R: TFlatSpinEditInteger;  
xptM3Data: TXPTask;  
lblM3T: TLabel;  
lblM3A: TLabel;  
lblM3V: TLabel;  
lblM3P: TLabel;  
lblM3R: TLabel;  
seM3T: TFlatSpinEditInteger;  
seM3P: TFlatSpinEditInteger;  
seM3V: TFlatSpinEditInteger;  
seM3A: TFlatSpinEditInteger;  
seM3R: TFlatSpinEditInteger;  
pnlAccept: TPanel;  
btnAccept: TButton;  
dlgSave: TSaveDialog;  
tbiFormula: TTBItem;  
tbiPrint: TTBItem;  
tbiSaveFile: TTBItem;  
tbiNewFile: TTBItem;  
tbStudent: TTBToolbar;  
tbiStudentNew: TTBItem;  
actStudentNew: TAction;  
actStudentEdit: TAction;  
actStudentDel: TAction;  
pnlFilter: TPanel;  
lblGroupFilter: TLabel;  
btnEditGroups: TButton;  
tbiStudentEdit: TTBItem;  
tbiStudentDel: TTBItem;  
cbGroupFilter: TFlatComboBox;  
pnlStop: TPanel;  
Timer: TTimer;  
tbiStudent: TTBSubmenuItem;  
tbiSDel: TTBItem;  
tbiSEdit: TTBItem;  
tbiSNew: TTBItem;  
actSaveFileAs: TAction;  
tbiFSaveAs: TTBItem;  
actAssociation: TAction;  
tbiOAssociate: TTBItem;  
tbiSep3: TTBSeparatorItem;  
tbiOpenFile: TTBSubmenuItem;  
actPassword: TAction;  
tbiSep0: TTBSeparatorItem;  
tbiFPassword: TTBItem;  
tbiHelp: TTBSubmenuItem;  
tbiHAbout: TTBItem;  
tbiHHelp: TTBItem;  
actAbout: TAction;  
actHelp: TAction;  
tbBackground: TTBBackground;  
TBToolbar1: TTBToolbar;  
TBItem1: TTBItem;  
TBItem2: TTBItem;  
TBItem3: TTBItem;  
SuperHint: TFlatHint;
```

```

Procedure RescentFileClick(Sender: TObject);
Procedure FillRescent;
Procedure ShowTXT(S: String);
Procedure SelectStu(idx1, idx2: integer);
Procedure UpDateList;
Procedure UpDateStudents;
Procedure actExitExecute(Sender: TObject);
Procedure actSaveFileExecute(Sender: TObject);
Procedure actOpenFileExecute(Sender: TObject);
Procedure actNewFileExecute(Sender: TObject);
Procedure actFormulaExecute(Sender: TObject);
Procedure actGroupsEditExecute(Sender: TObject);
Procedure ActionListUpdate(Action: TBasicAction; Var Handled: Boolean);
Procedure FormCreate(Sender: TObject);
Procedure lvStudentsClick(Sender: TObject);
Procedure TimerTimer(Sender: TObject);
Procedure actSaveFileAsExecute(Sender: TObject);
Procedure actAssociationExecute(Sender: TObject);
Procedure actPasswordExecute(Sender: TObject);
Procedure FormClose(Sender: TObject; Var Action: TCloseAction);
Procedure cbGroupFilterChange(Sender: TObject);
Procedure actAcceptStuExecute(Sender: TObject);
Procedure actPrintExecute(Sender: TObject);
Procedure actStudentNewExecute(Sender: TObject);
Procedure actStudentEditExecute(Sender: TObject);
Procedure actStudentDelExecute(Sender: TObject);
Procedure seM2TChange(Sender: TObject);
procedure actAboutExecute(Sender: TObject);
procedure actHelpExecute(Sender: TObject);
Private
  Procedure ReceiveMessage(Var Msg: TMessage); Message WM_COPYDATA;
End;

Const RegAN: String='MicronMathModule.Base';
      RegAnEx: String='\shell\open\command';

Var // змінні
    frmMainModStat: TfrmMainModStat;
    idx_1, idx_2: integer;

Implementation // Реалізація

Uses frmFormulaU, frmGroupsU, frmStudentU, frmReportU, frmSplashU, frmAboutU;

{$R *.dfm}

Procedure TfrmMainModStat.RescentFileClick(Sender: TObject);
Var // змінні
    S: String;
Begin
  S:=TTBControlItem(Sender).Caption;
  If FileExists(S) Then
  Begin
    If Saved=False Then
    Begin
      If MessageDlg('Відредагований файл не збережений.'+#13+'Чи бажаєте Ви його зберігти?', mtConfirmation, [mbYes, mbNo], 0)=mrYes Then
      Begin
        actSaveFile.Execute;
      End
    Else
    Begin
      prgData.CurrFile:=S;
      LoadFileData(prgData.CurrFile);
      UpdateList;
      cbGroupFilter.ItemIndex:=0;
      UpdateStudents;
      ShowTXT('Файл відкрито');
      Saved:=True;
    End
  End
End;

```

```

    End;
End
Else
Begin
    prgData.CurrFile:=S;
    LoadFileData (prgData.CurrFile);
    UpdateList;
    cbGroupFilter.ItemIndex:=0;
    UpdateStudents;
    ShowTXT ('Файл відкрито');
    Saved:=True;
End;
End
Else
Begin
    MessageDlg ('Такого файла вже не існує!!!', mtError, [mbOK], 0);
End;
End;

Procedure TfrmMainModStat.FillRescent;
Var // змінні
    CI: TTBCustomItem;
    i: integer;
Begin
    For i:=0 To tbiOpenFile.Count-1 Do
        tbiOpenFile.Items[i].Free;
    If Length(Data.Rescent)>0 Then
        Begin
            For i:=Low(Data.Rescent) To High(Data.Rescent) Do
                Begin
                    If FileExists (Data.Rescent[i]) Then
                        Begin
                            CI:=TTBCustomItem.Create (self);
                            CI.OnClick:=RescentFileClick;
                            CI.Caption:=Data.Rescent[i];
                            tbiOpenFile.Add (CI);
                        End;
                    End;
                End;
            End;
        End;

Procedure TfrmMainModStat.ReceiveMessage;
Var // змінні
    pcd: PCopyDataStruct;
Begin
    pcd:=PCopyDataStruct (Msg.LParam);

    ShowWindow (Handle, SW_SHOW);
    SetForegroundWindow (Handle);

    If pcd.lpData<>PChar ('Act ') Then
        Begin
            If Saved=False Then
                Begin
                    If MessageDlg ('Відредагований файл не збережений.'+#13+'Чи бажаєте Ви його зберігти?', mtConfirmation, [mbYes, mbNo], 0)=mrYes Then
                        Begin
                            actSaveFile.Execute;
                        End
                    Else
                        Begin
                            prgData.CurrFile:=PChar (pcd.lpData);
                            LoadFileData (PChar (pcd.lpData));
                            UpdateList;
                            cbGroupFilter.ItemIndex:=0;
                            UpdateStudents;
                            AddRescent (prgData.CurrFile);
                            FillRescent;
                            ShowTXT ('Файл відкрито');
                        End;
                    End;
                End;
            End;
        End;
    End;

```

```

        Saved:=True;
    End;
End
Else
Begin
    prgData.CurrFile:=PChar(pcd.lpData);
    LoadFileData(PChar(pcd.lpData));
    UpdateList;
    cbGroupFilter.ItemIndex:=0;
    UpdateStudents;
    AddRescent(prgData.CurrFile);
    FillRescent;
    ShowTXT('Файл відкрито');
    Saved:=True;
End;
End;
End;

Function RegValueIs(RKey: HKey; KeyPath: String; ValName: String; Val: String):
Boolean;
Begin
    Result:=false;
    With TRegistry.Create Do
    Try
        RootKey:=RKey;
        If OpenKey(KeyPath, false) Then
            If ReadString(ValName)=Val Then Result:=true;
        Finally
            Free;
        End;
    End;
End;

Procedure RegDelKey(RKey: HKey; KeyPath: String);
Begin
    With TRegistry.Create Do
    Try
        RootKey:=RKey;
        DeleteKey(KeyPath);
    Finally
        Free;
    End;
End;

Procedure RegSetKey(RKey: HKey; KeyPath: String; ValName: String; NewVal:
String);
Begin
    With TRegistry.Create Do
    Try
        RootKey:=RKey;
        OpenKey(KeyPath, True);
        WriteString(ValName, NewVal);
    Finally
        Free;
    End;
End;

Procedure TfrmMainModStat.ShowTXT(S: String);
Begin
    Timer.Enabled:=False;
    sbMain.SimpleText:=S;
    Timer.Enabled:=True;
End;

Procedure TfrmMainModStat.SelectStu(idx1, idx2: integer);
Begin
    idx_1:=idx1;
    idx_2:=idx2;
    studsave:=true;
    edtFIO.Text:=Groups[idx1].Student[idx2].Name;

```

```

edtGroup.Text:=Groups[idx1].Name;
seM1T.Value:=Groups[idx1].Student[idx2].Modules[1].T;
seM1A.Value:=Groups[idx1].Student[idx2].Modules[1].A;
seM1V.Value:=Groups[idx1].Student[idx2].Modules[1].V;
seM1R.Value:=Groups[idx1].Student[idx2].Modules[1].R;
seM2T.Value:=Groups[idx1].Student[idx2].Modules[2].T;
seM2A.Value:=Groups[idx1].Student[idx2].Modules[2].A;
seM2V.Value:=Groups[idx1].Student[idx2].Modules[2].V;
seM2R.Value:=Groups[idx1].Student[idx2].Modules[2].R;
seM3T.Value:=Groups[idx1].Student[idx2].Modules[3].T;
seM3A.Value:=Groups[idx1].Student[idx2].Modules[3].A;
seM3V.Value:=Groups[idx1].Student[idx2].Modules[3].V;
seM3R.Value:=Groups[idx1].Student[idx2].Modules[3].R;
seM1P.Value:=Groups[idx1].Student[idx2].Modules[1].P;
seM2P.Value:=Groups[idx1].Student[idx2].Modules[2].P;
seM3P.Value:=Groups[idx1].Student[idx2].Modules[3].P;
End;

Procedure TfrmMainModStat.UpdateList;
Var // змінні
    i: integer;
Begin
    cbGroupFilter.Items.BeginUpdate;
    cbGroupFilter.Items.Clear;
    cbGroupFilter.Sorted:=True;
    For i:=1 To Length(Groups) Do
    Begin
        cbGroupFilter.Items.Append(Groups[i-1].Name);
    End;
    cbGroupFilter.Sorted:=False;
    cbGroupFilter.Items.Insert(0, 'Bci');
    cbGroupFilter.Items.EndUpdate;
End;

Function CustomSortProc(Item1, Item2: TListItem; ParamSort: integer): integer;
Stdcall;
Begin
    If Item1.Caption<>Item2.Caption Then
        Result:=CompareUkrText(Item1.Caption, Item2.Caption)
    Else
        Result:=CompareUkrText(Item1.SubItems.Strings[0],
Item2.SubItems.Strings[0]);
End;

Procedure TfrmMainModStat.UpdateStudents;
Var // змінні
    i, j, idx: integer;
    Item: Tlistitem;
Begin
    lvStudents.Items.BeginUpdate;
    lvStudents.Items.Clear;
    If cbGroupFilter.Items.Strings[cbGroupFilter.ItemIndex]='Bci' Then
    Begin
        For i:=1 To Length(Groups) Do
        Begin
            For j:=1 To Length(Groups[i-1].Student) Do
            Begin
                Item:=lvStudents.Items.Add;
                Item.Caption:=Groups[i-1].Name;
                Item.SubItems.Append(Groups[i-1].Student[j-1].Name);
            End;
        End;
    End
    Else
    Begin
        For i:=1 To Length(Groups) Do
        Begin
            If Groups[i-1].Name=cbGroupFilter.Items.Strings[cbGroupFilter.ItemIndex]
Then

```

```

    Begin
        idx:=i;
        Break;
    End;
End;
For j:=1 To Length(Groups[idx-1].Student) Do
Begin
    Item:=lvStudents.Items.Add;
    Item.Caption:=Groups[idx-1].Name;
    Item.SubItems.Append(Groups[idx-1].Student[j-1].Name);
End;
End;
lvStudents.CustomSort(@CustomSortProc, 0);
lvStudents.Items.EndUpdate;
End;

Procedure TfrmMainModStat.actExitExecute(Sender: TObject);
Begin
    If Saved=False Then
    Begin
        If MessageDlg('Відредагований файл не збережений.'+#13+'Чи бажаєте Ви його зберегти?', mtConfirmation, [mbYes, mbNo], 0)=mrYes Then
        Begin
            actSaveFile.Execute;
        End;
    End;
    SaveMainData;
    Application.Terminate;
End;

Procedure TfrmMainModStat.actSaveFileExecute(Sender: TObject);
Begin
    If prgData.CurrFile='' Then
    Begin
        If dlgSave.Execute Then
        Begin
            If pos('.mod', dlgSave.FileName)=0 Then
                prgData.CurrFile:=dlgSave.FileName+'.mod'
            Else
                prgData.CurrFile:=dlgSave.FileName;
            SaveFileData(prgData.CurrFile);
            ShowTXT('Файл збережено');
            Saved:=True;
            AddRescent(prgData.CurrFile);
            FillRescent;
        End;
    End
    Else
    Begin
        SaveFileData(prgData.CurrFile);
        ShowTXT('Файл збережено');
        Saved:=True;
        AddRescent(prgData.CurrFile);
        FillRescent;
    End;
End;

Procedure TfrmMainModStat.actOpenFileExecute(Sender: TObject);
Begin
    If (StudSave=False) And (Data.OnDel=True) Then
    Begin
        If MessageDlg('Відредаговані дані не прийняті.'+#13+'Чи бажаєте Ви їх зберегти?', mtConfirmation, [mbYes, mbNo], 0)=mrYes Then
        Begin
            actAcceptStu.Execute;
        End;
    End;
    If Saved=False Then
    Begin

```

```

    If MessageDlg('Відредагований файл не збережений.'+#13+'Чи бажаєте Ви його
зберігти?', mtConfirmation, [mbYes, mbNo], 0)=mrYes Then
    Begin
        actSaveFile.Execute;
    End;
End;
If dlgOpen.Execute Then
Begin
    prgData.CurrFile:=dlgOpen.FileName;
    LoadFileData(prgData.CurrFile);
    UpdateList;
    cbGroupFilter.ItemIndex:=0;
    UpdateStudents;
    AddRescent(prgData.CurrFile);
    FillRescent;
    ShowTXT('Файл відкрито');
    Saved:=True;
End;
End;

Procedure TfrmMainModStat.actNewFileExecute(Sender: TObject);
Begin
    If (StudSave=False) And (Data.OnDel=True) Then
    Begin
        If MessageDlg('Відредаговані дані не прийняті.'+#13+'Чи бажаєте Ви їх
зберігти?', mtConfirmation, [mbYes, mbNo], 0)=mrYes Then
        Begin
            actAcceptStu.Execute;
        End;
    End;
    If Saved=False Then
    Begin
        If MessageDlg('Відредагований файл не збережений.'+#13+'Чи бажаєте Ви його
зберігти?', mtConfirmation, [mbYes, mbNo], 0)=mrYes Then
        Begin
            actSaveFile.Execute;
        End;
    End;
    NewFileData;
    UpdateList;
    cbGroupFilter.ItemIndex:=0;
    UpdateStudents;
    ShowTXT('Новий файл створено');
End;

Procedure TfrmMainModStat.actFormulaExecute(Sender: TObject);
Var // змінні
    frmFormula: TfrmFormula;
Begin
    TempFileData.Formula:=FileData.Formula;
    TempFileData.MaxBal.X:=FileData.MaxBal.X;
    TempFileData.MaxBal.Y:=FileData.MaxBal.Y;
    TempFileData.DownPredel.Y:=FileData.DownPredel.Y;
    TempFileData.DownPredel.X:=FileData.DownPredel.X;
    TempFileData.UpPredel.X:=FileData.UpPredel.X;
    TempFileData.UpPredel.Y:=FileData.UpPredel.Y;
    frmFormula:=TfrmFormula.Create(self);
    If frmFormula.ShowModal=mrOk Then
    Begin
        FileData.Formula:=frmFormula.edtFormula.Text;
        FileData.MaxBal.X:=frmFormula.seMaxX.Value;
        FileData.MaxBal.Y:=frmFormula.seMaxY.Value;
        FileData.DownPredel.X:=frmFormula.seDoX.Value;
        FileData.DownPredel.Y:=frmFormula.seDoY.Value;
        FileData.UpPredel.X:=frmFormula.seUpX.Value;
        FileData.UpPredel.Y:=frmFormula.seUpY.Value;
        prgData.Saved:=False;
    End;
    frmFormula.Free;

```

```

    ShowTXT('Відредаговано опції бази даних');
End;

Procedure TfrmMainModStat.actGroupsEditExecute(Sender: TObject);
Var // змінні
    frmGroups: TfrmGroups;
Begin
    frmGroups:=TfrmGroups.Create(self);
    frmGroups.UpdateList;
    If frmGroups.ShowModal=mrOk Then
    Begin
        prgData.Saved:=False;
    End;
    frmGroups.Free;
    frmMainModStat.UpdateList;
    cbGroupFilter.ItemIndex:=0;
    ShowTXT('Оновлено список');
End;

Procedure TfrmMainModStat.ActionListUpdate(Action: TBasicAction;
    Var Handled: Boolean);
Begin
    pnlStop.Visible:=Not (lvStudents.Selected<>Nil);
    actStudentEdit.Enabled:=(lvStudents.Selected<>Nil);
    actStudentDel.Enabled:=(lvStudents.Selected<>Nil);
    actSaveFile.Enabled:=Not Saved;
    actSaveFileAs.Enabled:=Not Saved;
    tbiOpenFile.DropDownCombo:=(tbiOpenFile.Count>0);
End;

Procedure TfrmMainModStat.FormCreate(Sender: TObject);
Begin
    LoadMainData;
    FillRescent;
    actAssociation.Checked:=(RegValueIs(HKEY_CLASSES_ROOT, RegAn+RegAnEx, '',
Application.ExeName+' %1'))And(RegValueIs(HKEY_CLASSES_ROOT, '.mod', '',
RegAN));
End;

Procedure TfrmMainModStat.lvStudentsClick(Sender: TObject);
Var // змінні
    i, j, idx1, idx2: integer;
Begin
    If (StudSave=False)And(Data.OnDel=True) Then
    Begin
        If MessageDlg('Відредаговані дані не прийняті.'+#13+'Чи бажаєте Ви їх
зберігти?', mtConfirmation, [mbYes, mbNo], 0)=mrYes Then
        Begin
            actAcceptStu.Execute;
        End;
    End;
    If (cbGroupFilter.ItemIndex>-1)And(lvStudents.Selected<>Nil) Then
    Begin
        For i:=Low(Groups)To High(Groups) Do
        Begin
            If Groups[i].Name=lvStudents.Selected.Caption Then
            Begin
                idx1:=i;
                For j:=Low(Groups[idx1].Student)To High(Groups[idx1].Student) Do
                Begin
                    If
Groups[idx1].Student[j].Name=lvStudents.Selected.SubItems.Strings[0] Then
                    Begin
                        idx2:=j;
                        SelectStu(idx1, idx2);
                        StudSave:=True;
                        Exit;
                    End;
                End;
            End;
        End;
    End;

```

```

        End;
    End;
End
Else
Begin
    edtFIO.Text:='';
    edtGroup.Text:='';
    seM1T.Value:=0;
    seM1A.Value:=0;
    seM1V.Value:=0;
    seM1R.Value:=0;
    seM2T.Value:=0;
    seM2A.Value:=0;
    seM2V.Value:=0;
    seM2R.Value:=0;
    seM3T.Value:=0;
    seM3A.Value:=0;
    seM3V.Value:=0;
    seM3R.Value:=0;
End;
End;

Procedure TfrmMainModStat.TimerTimer(Sender: TObject);
Begin
    sbMain.SimpleText:='';
End;

Procedure TfrmMainModStat.actSaveFileAsExecute(Sender: TObject);
Begin
    If dlgSave.Execute Then
    Begin
        If pos('.mod', dlgSave.FileName)=0 Then
            prgData.CurrFile:=dlgSave.FileName+'.mod'
        Else
            prgData.CurrFile:=dlgSave.FileName;
        SaveFileData(prgData.CurrFile);
        ShowTXT('Новий файл збережено');
        Saved:=True;
        AddRescent(prgData.CurrFile);
        FillRescent;
    End;
End;

Procedure TfrmMainModStat.actAssociationExecute(Sender: TObject);
Const S: String=' файли даних (*.mod) ';
Begin
    If actAssociation.Checked Then
        If MessageDlg('Ви дійсно бажаєте деасоціювати'+S+
            ('+Application.ExeName+')!?', mtConfirmation, [mbYes, mbNo], 0)=mrYes Then
            Begin
                RegDelKey(HKEY_CLASSES_ROOT, RegAn);
                RegDelKey(HKEY_CLASSES_ROOT, '.mod');
                ShowTXT('Деасоційовано'+S);
            End
        Else
            Else
                If MessageDlg('Ви дійсно бажаєте асоціювати'+S+'з Статистики успішності
                    ('+Application.ExeName+')!?', mtConfirmation, [mbYes, mbNo], 0)=mrYes Then
                    Begin
                        RegSetKey(HKEY_CLASSES_ROOT, '.mod', '', RegAN);
                        RegSetKey(HKEY_CLASSES_ROOT, RegAN+'\DefaultIcon', '',
                            Application.ExeName+',0');
                        RegSetKey(HKEY_CLASSES_ROOT, RegAN+RegAnEx, '', Application.ExeName+'
                            %1');
                        ShowTXT('Асоційовано'+S);
                    End;
                    actAssociation.Checked:=(RegValueIs(HKEY_CLASSES_ROOT, RegAN+RegAnEx, '',
                        Application.ExeName+' %1'))And(RegValueIs(HKEY_CLASSES_ROOT, '.mod', '',
                            RegAN));

```

```

End;

Procedure TfrmMainModStat.actPasswordExecute(Sender: TObject);
Var // змінні
    S, S1: String;
    Flag: boolean;
Begin
    If FileData.Pass.Password<>' ' Then
    Begin
        Flag:=False;
        S:=InputBox('Встановлення пароля', 'Раніше пароль вже був встановлений.
Введіть його:', '');
        If S<>' ' Then
        Begin
            If S=FileData.Pass.Password Then
            Begin
                Flag:=True;
            End
            Else
            Begin
                MessageDLG('Це невірний пароль!!!', mtError, [mbOK], 0);
            End;
        End;

        If Flag=True Then
        Begin
            FileData.Pass.Password:=InputBox('Встановлення пароля', 'Введіть новий
пароль:', '');
            Saved:=False;
        End;
        Else
        Begin
            FileData.Pass.Password:=InputBox('Встановлення пароля', 'Введіть новий пароль:',
            '');
            Saved:=False;
        End;
    End;

Procedure TfrmMainModStat.FormClose(Sender: TObject;
    Var Action: TCloseAction);
Begin
    actExit.Execute;
End;

Procedure TfrmMainModStat.cbGroupFilterChange(Sender: TObject);
Begin
    UpdateStudents;
End;

Procedure TfrmMainModStat.actAcceptStuExecute(Sender: TObject);
Begin
    Groups[idx_1].Student[idx_2].Name:=edtFIO.Text;
    Groups[idx_1].Name:=edtGroup.Text;
    Groups[idx_1].Student[idx_2].Modules[1].T:=seM1T.Value;
    Groups[idx_1].Student[idx_2].Modules[1].A:=seM1A.Value;
    Groups[idx_1].Student[idx_2].Modules[1].V:=seM1V.Value;
    Groups[idx_1].Student[idx_2].Modules[1].R:=seM1R.Value;
    Groups[idx_1].Student[idx_2].Modules[2].T:=seM2T.Value;
    Groups[idx_1].Student[idx_2].Modules[2].A:=seM2A.Value;
    Groups[idx_1].Student[idx_2].Modules[2].V:=seM2V.Value;
    Groups[idx_1].Student[idx_2].Modules[2].R:=seM2R.Value;
    Groups[idx_1].Student[idx_2].Modules[3].T:=seM3T.Value;
    Groups[idx_1].Student[idx_2].Modules[3].A:=seM3A.Value;
    Groups[idx_1].Student[idx_2].Modules[3].V:=seM3V.Value;
    Groups[idx_1].Student[idx_2].Modules[3].R:=seM3R.Value;
    Groups[idx_1].Student[idx_2].Modules[1].P:=seM1P.Value;
    Groups[idx_1].Student[idx_2].Modules[2].P:=seM2P.Value;
    Groups[idx_1].Student[idx_2].Modules[3].P:=seM3P.Value;

```

```

    StudSave:=True;
    Saved:=False;
End;

Procedure TfrmMainModStat.actPrintExecute(Sender: TObject);
Var // змінні
    frmReport: TfrmReport;
Begin
    frmReport:=TfrmReport.Create(self);
    frmReport.ShowModal;
    frmReport.Free;
End;

Procedure TfrmMainModStat.actStudentNewExecute(Sender: TObject);
Var // змінні
    frmStudent: TfrmStudent;
    Flag: boolean;
    idx, i: integer;
    S: String;
Begin
    If (StudSave=False) And (Data.OnDel=True) Then
    Begin
        If MessageDlg('Відредаговані дані не прийняті.'+#13+'Чи бажаєте Ви їх зберігти?', mtConfirmation, [mbYes, mbNo], 0)=mrYes Then
        Begin
            actAcceptStu.Execute;
        End;
    End;
    frmStudent:=TfrmStudent.Create(self);
    frmStudent.UpdateList;
    Repeat
        Flag:=True;
        If frmStudent.ShowModal=mrOk Then
        Begin
            If frmStudent.cbGroupFilter.ItemIndex=-1 Then
            Begin
                Flag:=False;
            End;
            If frmStudent.edtFIO.Text='' Then
            Begin
                Flag:=False;
                S:='Заповніть поле "Прізвище, ім'я та по батькові "!!!!';
            End;
            If Flag=True Then
            Begin
                For i:=1 To Length(Groups) Do
                Begin
                    If Groups[i-
1].Name=frmStudent.cbGroupFilter.Items.Strings[frmStudent.cbGroupFilter.ItemIndex] Then
                    Begin
                        idx:=i-1;
                        Break;
                    End;
                End;
                SetLength(Groups[idx].Student, Length(Groups[idx].Student)+1);

                Groups[idx].Student[High(Groups[idx].Student)].Name:=frmStudent.edtFIO.Text;
                Groups[idx].Student[High(Groups[idx].Student)].Modules[1].A:=0;
                Groups[idx].Student[High(Groups[idx].Student)].Modules[1].V:=0;
                Groups[idx].Student[High(Groups[idx].Student)].Modules[1].T:=0;
                Groups[idx].Student[High(Groups[idx].Student)].Modules[1].R:=0;
                Groups[idx].Student[High(Groups[idx].Student)].Modules[2].A:=0;
                Groups[idx].Student[High(Groups[idx].Student)].Modules[2].V:=0;
                Groups[idx].Student[High(Groups[idx].Student)].Modules[2].T:=0;
                Groups[idx].Student[High(Groups[idx].Student)].Modules[2].R:=0;
                Groups[idx].Student[High(Groups[idx].Student)].Modules[3].A:=0;
                Groups[idx].Student[High(Groups[idx].Student)].Modules[3].V:=0;
                Groups[idx].Student[High(Groups[idx].Student)].Modules[3].T:=0;

```

```

        Groups[idx].Student[High(Groups[idx].Student)].Modules[3].R:=0;
        Groups[idx].Student[High(Groups[idx].Student)].Modules[1].P:=0;
        Groups[idx].Student[High(Groups[idx].Student)].Modules[2].P:=0;
        Groups[idx].Student[High(Groups[idx].Student)].Modules[3].P:=0;
        prgData.Saved:=False;
        ShowTXT('Додано запис ');
    End
    Else
    Begin
        MessageDlg(S, mtError, [mbOk], 0);
    End;
End;
Until Flag=True;
frmStudent.Free;
S:=cbGroupFilter.Items.Strings[cbGroupFilter.ItemIndex];
UpdateList;
idx:=0;
For i:=2 To cbGroupFilter.Items.Count Do
Begin
    If cbGroupFilter.Items.Strings[i-1]=S Then
    Begin
        idx:=i-1;
        Break;
    End;
End;
cbGroupFilter.ItemIndex:=idx;
UpdateStudents;
End;

Procedure TfrmMainModStat.actStudentEditExecute(Sender: TObject);
Var // змінні
    frmStudent: TfrmStudent;
    Flag: boolean;
    idx, idx2, i: integer;
    S, S2: String;
Begin
    If (StudSave=False) And (Data.OnDel=True) Then
    Begin
        If MessageDlg('Відредаговані дані не прийняті.'+#13+'Чи бажаєте Ви їх зберігти?', mtConfirmation, [mbYes, mbNo], 0)=mrYes Then
        Begin
            actAcceptStu.Execute;
        End;
    End;
    If lvStudents.Selected<>Nil Then
    Begin
        frmStudent:=TfrmStudent.Create(self);
        frmStudent.UpdateList;
        For i:=0 To frmStudent.cbGroupFilter.Items.Count-1 Do
        Begin
            If frmStudent.cbGroupFilter.Items.Strings[i]=lvStudents.Selected.Caption
Then
            Begin
                frmStudent.cbGroupFilter.ItemIndex:=i;
                Break;
            End;
        End;
        frmStudent.edtFIO.Text:=lvStudents.Selected.SubItems.Strings[0];
        S2:=frmStudent.edtFIO.Text;
        Repeat
            Flag:=True;
            If frmStudent.ShowModal=mrOk Then
            Begin
                If frmStudent.cbGroupFilter.ItemIndex=-1 Then
                Begin
                    Flag:=False;
                End;
                If frmStudent.edtFIO.Text='' Then
                Begin

```

```

        Flag:=False;
        S:='Заповніть поле "Прізвище, ім'я та по батькові "!!!';
    End;
    If Flag=True Then
    Begin
        For i:=1 To Length(Groups) Do
        Begin
            If Groups[i-
1].Name=frmStudent.cbGroupFilter.Items.Strings[frmStudent.cbGroupFilter.ItemInde
x] Then
                Begin
                    idx:=i-1;
                    Break;
                End;
            End;
            For i:=Low(Groups[idx].Student)To High(Groups[idx].Student) Do
            Begin
                If Groups[idx].Student[i].Name=S2 Then
                Begin
                    idx2:=i;
                    Break;
                End;
            End;
            Groups[idx].Student[idx2].Name:=frmStudent.edtFIO.Text;
            prgData.Saved:=False;
            ShowTXT('Запис було відредаговано');
        End
        Else
        Begin
            MessageDlg(S, mtError, [mbOk], 0);
        End;
    End;
    Until Flag=True;
    frmStudent.Free;
    S:=cbGroupFilter.Items.Strings[cbGroupFilter.ItemIndex];
    UpdateList;
    idx:=0;
    For i:=2 To cbGroupFilter.Items.Count Do
    Begin
        If cbGroupFilter.Items.Strings[i-1]=S Then
        Begin
            Begin
                idx:=i-1;
                Break;
            End;
        End;
    End;
    cbGroupFilter.ItemIndex:=idx;
    UpdateStudents;
    End;
End;

Procedure TfrmMainModStat.actStudentDelExecute(Sender: TObject);
Var // змінні
    idx, idx2, i: integer;
Begin
    If MessageDlg('Ви впевнені, що хочете знищити запис?', mtConfirmation, [mbYes,
mbNo], 0)=mrYes Then
    Begin
        For i:=Low(Groups)To High(Groups) Do
        Begin
            If Groups[i].Name=lvStudents.Selected.Caption Then
            Begin
                idx:=i;
                Break;
            End;
        End;
    End;
    For i:=Low(Groups[idx].Student)To High(Groups[idx].Student) Do
    Begin
        If Groups[idx].Student[i].Name=lvStudents.Selected.SubItems.Strings[0]
Then

```

```

    Begin
        idx2:=i;
        Break;
    End;
End;

Groups[idx].Student[idx2].Name:=Groups[idx].Student[High(Groups[idx].Student)].Name;

Groups[idx].Student[idx2].Modules[1].A:=Groups[idx].Student[High(Groups[idx].Student)].Modules[1].A;

Groups[idx].Student[idx2].Modules[1].V:=Groups[idx].Student[High(Groups[idx].Student)].Modules[1].V;

Groups[idx].Student[idx2].Modules[1].T:=Groups[idx].Student[High(Groups[idx].Student)].Modules[1].T;

Groups[idx].Student[idx2].Modules[1].R:=Groups[idx].Student[High(Groups[idx].Student)].Modules[1].R;

Groups[idx].Student[idx2].Modules[2].A:=Groups[idx].Student[High(Groups[idx].Student)].Modules[2].A;

Groups[idx].Student[idx2].Modules[2].V:=Groups[idx].Student[High(Groups[idx].Student)].Modules[2].V;

Groups[idx].Student[idx2].Modules[2].T:=Groups[idx].Student[High(Groups[idx].Student)].Modules[2].T;

Groups[idx].Student[idx2].Modules[2].R:=Groups[idx].Student[High(Groups[idx].Student)].Modules[2].R;

Groups[idx].Student[idx2].Modules[3].A:=Groups[idx].Student[High(Groups[idx].Student)].Modules[3].A;

Groups[idx].Student[idx2].Modules[3].V:=Groups[idx].Student[High(Groups[idx].Student)].Modules[3].V;

Groups[idx].Student[idx2].Modules[3].T:=Groups[idx].Student[High(Groups[idx].Student)].Modules[3].T;

Groups[idx].Student[idx2].Modules[3].R:=Groups[idx].Student[High(Groups[idx].Student)].Modules[3].R;

Groups[idx].Student[idx2].Modules[1].P:=Groups[idx].Student[High(Groups[idx].Student)].Modules[1].P;

Groups[idx].Student[idx2].Modules[2].P:=Groups[idx].Student[High(Groups[idx].Student)].Modules[2].P;

Groups[idx].Student[idx2].Modules[3].P:=Groups[idx].Student[High(Groups[idx].Student)].Modules[3].P;
    SetLength(Groups[idx].Student, Length(Groups[idx].Student)-1);
    Saved:=False;
    UpdateStudents;
    ShowTXT('Запис було видалено');
End;
End;

Procedure TfrmMainModStat.seM2TChange(Sender: TObject);
Begin
    StudSave:=False;
End;

procedure TfrmMainModStat.actAboutExecute(Sender: TObject);
Var // змінні
    frmAbout: TfrmAbout;
Begin

```

```
    frmAbout:=TfrmAbout.Create(self);
    frmAbout.ShowModal;
    frmAbout.Free;
End;

procedure TfrmMainModStat.actHelpExecute(Sender: TObject);
begin
    ShellExecute(Application.Handle, 'open', PChar(GetDN('Help\MMModule.hlp')),
    Nil, Nil, SW_MAXIMIZE);
end;

End.
```

K6П3-2023

## ФАЙЛ PRGFILTER.PAS

```

Unit prgFilter;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Вихідний код до магістерської роботи
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу
аудиторного фонду розкладу занять для факультетів

Виконав: Горшков Владислав Олегович
студент 6 курсу, групи КН-22М-2

Керівник: Коваленко А.С.
Кропивницький 2023
}

Interface // Інтерфейс

Uses // підключення
  Windows, SysUtils, prgData, prgTypes, MathParseTypes;

Type // типи
  TPupil=Object
    Name: String;
    Modules: TAllModule;
    SGroup: String;
    Mark: Array[1..3] Of real;
    CurrMark, HoleMark: real;
  End;
  TAllStu=Array Of TPupil;

Var // змінні
  AllStudents: TAllStu;
  Cpos: byte;

Function Filter(Filter: TFilter; M1, M2, M3: boolean; Var AllStu: TAllStu):
integer;
Function Solve(Value: extended): extended;

Implementation // Реалізація

Function DecideExpression(Expression: PChar; BackParse: TBackParse):
TDecideResult; Stdcall; External'MathParse.dll';

Function ProseedChar(Value: String): String;
Var i: Integer;
Begin
  Result:=Value;
  For i:=1 To Length(Value) Do
  Begin
    Case Value[i] Of
      'A'..'Z',
        'А'..'Я': Result[i]:=Chr(Ord(Value[i])+32);
      'Ë': Result[i]:='ë';
      'Í': Result[i]:='í';
      'Ï': Result[i]:='ï';
      'È': Result[i]:='è';
      'İ': Result[i]:='i';
      'I': Result[i]:='i';
      'S': Result[i]:='s';
    End;
  End;
End;

Function StrToExt(Value: String): extended;
Begin
  Result:=-1;

```

```

    Try
      Result:=StrToFloat(Value);
    Except
    End;
End;

Function Solve(Value: extended): extended;
Var // змінні
  x1, y1, x2, y2, k, y, x: extended;
Begin
  Result:=0;
  If (Value>0)And(Value<FileData.DownPredel.X) Then
    Result:=1;
  If (Value>=FileData.DownPredel.X)And(Value<FileData.UpPredel.X) Then
  Begin
    x1:=0;
    y1:=0;
    x2:=FileData.UpPredel.X-FileData.DownPredel.X;
    y2:=FileData.UpPredel.Y-FileData.DownPredel.Y;
    k:=(y2-y1)/(x2-x1);
    x:=Value-FileData.DownPredel.X;
    y:=k*x;
    Result:=y+FileData.DownPredel.Y;
  End;
  If (Value>=FileData.UpPredel.X) Then
    Result:=12;
End;

Function NCount(S: String): word;

// функція пошуку кількості слів в строчці

Var // змінні
  Count: word;
  I: Word;
  Len: Word;
Begin
  Count:=0;
  Len:=length(S);
  I:=1;
  While I<=Len Do
  Begin
    While (I<=Len)And(S[I]=' ') Do Inc(I);
    If I<=Len Then Inc(Count);
    While (I<=Len)And Not(S[I]=' ') Do Inc(I)
  End;
  Result:=Count;
End;

Function NPosition(N: Byte; S: String): word;

// визначення позиції в строчці

Var // змінні
  Count: word;
  I: Word;
  Len: Word;
Begin
  Count:=0;
  Len:=length(S);
  I:=1;
  Result:=0;

  While (I<=Len)And(Count<>N) Do
  Begin
    While (I<=Len)And(S[I]=' ') Do Inc(I);
    If I<=Len Then Inc(Count);
    If Count<>N Then
      While (I<=Len)And Not(S[I]=' ') Do Inc(I)

```

```

    Else
        Result:=I
    End
End;

Function ExtractN(N: word; S: String): String;

// отримання слова N з строчки S

Var // змінні
    I: word;
    SLen: Word;
    sExtract: String;
Begin
    sExtract:='';
    SLen:=length(S);
    I:=NPosition(N, S);
    If I<>0 Then
        While (I<=SLen)And Not(S[I]=' ') Do
            Begin
                sExtract:=sExtract+S[I];
                Inc(I)
            End;
        result:=sExtract;
    End;
End;

Procedure SortI(idx1, idx2: integer; Var AllStu: TAllStu);
Var // змінні
    p: integer;
    S: String;
    r: extended;
Begin
    S:=AllStu[idx1].Name;
    AllStu[idx1].Name:=AllStu[idx2].Name;
    AllStu[idx2].Name:=S;

    S:=AllStu[idx1].SGroup;
    AllStu[idx1].SGroup:=AllStu[idx2].SGroup;
    AllStu[idx2].SGroup:=S;

    r:=AllStu[idx1].HoleMark;
    AllStu[idx1].HoleMark:=AllStu[idx2].HoleMark;
    AllStu[idx2].HoleMark:=r;

    r:=AllStu[idx1].Mark[1];
    AllStu[idx1].Mark[1]:=AllStu[idx2].Mark[1];
    AllStu[idx2].Mark[1]:=r;

    r:=AllStu[idx1].Mark[2];
    AllStu[idx1].Mark[2]:=AllStu[idx2].Mark[2];
    AllStu[idx2].Mark[2]:=r;

    r:=AllStu[idx1].Mark[3];
    AllStu[idx1].Mark[3]:=AllStu[idx2].Mark[3];
    AllStu[idx2].Mark[3]:=r;

    p:=AllStu[idx1].Modules[1].A;
    AllStu[idx1].Modules[1].A:=AllStu[idx2].Modules[1].A;
    AllStu[idx2].Modules[1].A:=p;

    p:=AllStu[idx1].Modules[1].V;
    AllStu[idx1].Modules[1].V:=AllStu[idx2].Modules[1].V;
    AllStu[idx2].Modules[1].V:=p;

    p:=AllStu[idx1].Modules[1].T;
    AllStu[idx1].Modules[1].T:=AllStu[idx2].Modules[1].T;
    AllStu[idx2].Modules[1].T:=p;

    p:=AllStu[idx1].Modules[1].R;

```

```

AllStu[idx1].Modules[1].R:=AllStu[idx2].Modules[1].R;
AllStu[idx2].Modules[1].R:=p;

p:=AllStu[idx1].Modules[2].A;
AllStu[idx1].Modules[2].A:=AllStu[idx2].Modules[2].A;
AllStu[idx2].Modules[2].A:=p;
p:=AllStu[idx1].Modules[2].V;
AllStu[idx1].Modules[2].V:=AllStu[idx2].Modules[2].V;
AllStu[idx2].Modules[2].V:=p;
p:=AllStu[idx1].Modules[2].T;
AllStu[idx1].Modules[2].T:=AllStu[idx2].Modules[2].T;
AllStu[idx2].Modules[2].T:=p;
p:=AllStu[idx1].Modules[2].R;
AllStu[idx1].Modules[2].R:=AllStu[idx2].Modules[2].R;
AllStu[idx2].Modules[2].R:=p;
p:=AllStu[idx1].Modules[3].A;
AllStu[idx1].Modules[3].A:=AllStu[idx2].Modules[3].A;
AllStu[idx2].Modules[3].A:=p;
p:=AllStu[idx1].Modules[3].V;
AllStu[idx1].Modules[3].V:=AllStu[idx2].Modules[3].V;
AllStu[idx2].Modules[3].V:=p;
p:=AllStu[idx1].Modules[3].T;
AllStu[idx1].Modules[3].T:=AllStu[idx2].Modules[3].T;
AllStu[idx2].Modules[3].T:=p;
p:=AllStu[idx1].Modules[3].R;
AllStu[idx1].Modules[3].R:=AllStu[idx2].Modules[3].R;
AllStu[idx2].Modules[3].R:=p;
p:=AllStu[idx1].Modules[1].P;
AllStu[idx1].Modules[1].P:=AllStu[idx2].Modules[1].P;
AllStu[idx2].Modules[1].P:=p;
p:=AllStu[idx1].Modules[2].P;
AllStu[idx1].Modules[2].P:=AllStu[idx2].Modules[2].P;
AllStu[idx2].Modules[2].P:=p;
p:=AllStu[idx1].Modules[3].P;
AllStu[idx1].Modules[3].P:=AllStu[idx2].Modules[3].P;
AllStu[idx2].Modules[3].P:=p;
End;

Function Filter(Filter: TFilter; M1, M2, M3: boolean; Var AllStu: TAllStu):
integer;
Var // змінні
    max, i, j, k, d, i1, i2: integer;
    Sum: real;
    R: TDecideResult;
    Str: PChar;
    S, SB, SV: String;
    SVe, SBe: extended;
    Flag, CanAdd, sf: boolean;
    SCData: Array[1..3] Of Array[1..6] Of extended;
    SCDataAll: extended;
Begin

    // Ініціалізація

    max:=0;
    SetLength(AllStu, 0);

    // Фільтрація
    For i:=Low(Groups) To High(Groups) Do
    Begin
        For j:=Low(Groups[i].Student) To High(Groups[i].Student) Do
        Begin
            CanAdd:=False;
            SCData[1, 6]:=Groups[i].Student[j].Modules[1].A;
            SCData[1, 2]:=Groups[i].Student[j].Modules[1].T;
            SCData[1, 5]:=Groups[i].Student[j].Modules[1].R;
            SCData[1, 4]:=Groups[i].Student[j].Modules[1].V;
            SCData[1, 3]:=Groups[i].Student[j].Modules[1].P;
            SCData[2, 6]:=Groups[i].Student[j].Modules[2].A;

```

```

SCData[2, 2]:=Groups[i].Student[j].Modules[2].T;
SCData[2, 5]:=Groups[i].Student[j].Modules[2].R;
SCData[2, 4]:=Groups[i].Student[j].Modules[2].V;
SCData[2, 3]:=Groups[i].Student[j].Modules[2].P;
SCData[3, 6]:=Groups[i].Student[j].Modules[3].A;
SCData[3, 2]:=Groups[i].Student[j].Modules[3].T;
SCData[3, 5]:=Groups[i].Student[j].Modules[3].R;
SCData[3, 4]:=Groups[i].Student[j].Modules[3].V;
SCData[3, 3]:=Groups[i].Student[j].Modules[3].P;
For k:=1 To 3 Do
Begin
  S:=prgData.FileData.Formula;
  Insert(FloatToStr(SCData[k, 4]), S, pos('V', S));
  Insert(FloatToStr(SCData[k, 6]), S, pos('A', S));
  Insert(FloatToStr(SCData[k, 5]), S, pos('R', S));
  Insert(FloatToStr(Solve(SCData[k, 2]+SCData[k, 3])), S, pos('T', S));
  Delete(S, pos('V', S), 1);
  Delete(S, pos('A', S), 1);
  Delete(S, pos('T', S), 1);
  Delete(S, pos('R', S), 1);
  While pos(',', S)>0 Do
  Begin
    Insert('.', S, pos(',', S));
    Delete(S, pos(',', S), 1);
  End;
  Str:=PChar(S);
  R:=DecideExpression(Str, Nil);
  SCData[k, 1]:=R.Result;
End;
Sum:=0;
d:=0;
If M1 Then
Begin
  Sum:=Sum+SCData[1, 1];
  Inc(d);
End;
If M2 Then
Begin
  Sum:=Sum+SCData[2, 1];
  Inc(d);
End;
If M3 Then
Begin
  Sum:=Sum+SCData[3, 1];
  Inc(d);
End;
SCDataAll:=Sum/d;

If Length(Filter)>0 Then
Begin
  For d:=Low(Filter)To High(Filter) Do
  Begin

    // Перевірка строки

    If Filter[i].Ident='name' Then
    Begin
      If Filter[i].Value[1]='^' Then
      Begin
        SB:=Copy(Filter[i].Value, 4, Length(Filter[i].Value)-3);
        If Filter[i].Value[2]='G' Then
        Begin
          SV:=Groups[i].Name;
        End
        Else
        Begin
          SV:=Groups[i].Student[j].Name;
        End;
      End
    End
  End
End

```

```

Else
Begin
  SB:=ProseedChar(Copy(Filter[i].Value, 4, Length(Filter[i].Value)-
3));

  If Filter[i].Value[2]='G' Then
  Begin
    SV:=ProseedChar(Groups[i].Name);
  End
  Else
  Begin
    SV:=ProseedChar(Groups[i].Student[j].Name);
  End;
End;
If Pos(SB, SV)<>0 Then
  CanAdd:=True;
End;

// Перевірка числа
If Filter[i].Ident='module' Then
Begin
  Case Filter[i].Value[1] Of
    'M': i1:=1;
    't': i1:=2;
    'p': i1:=3;
    'v': i1:=4;
    'r': i1:=5;
    'a': i1:=6;
  End;
  Case Filter[i].Value[2] Of
    '1': i2:=1;
    '2': i2:=2;
    '3': i2:=3;
  End;

  SB:=Copy(Filter[i].Value, 4, Length(Filter[i].Value)-3);
  SBe:=StrToExt(ExtractN(1, SB));
  SVe:=StrToExt(ExtractN(2, SB));

  If Filter[i].Value[3]='^' Then
  Begin
    If (SBe<=SCData[i1, i2])And(SCData[i1, i2]<=SVe) Then
      CanAdd:=True;
    End
  Else
  Begin
    If (SBe<SCData[i1, i2])And(SCData[i1, i2]<SVe) Then
      CanAdd:=True;
    End;
  End;
End;

// Перевірка числа 2
If Filter[i].Ident='module' Then
Begin
  sf:=False;
  If Filter[i].Value[1]='A' Then sf:=True;
  If sf=True Then
  Begin
    SB:=Copy(Filter[i].Value, 4, Length(Filter[i].Value)-3);
    SBe:=StrToExt(ExtractN(1, SB));
    SVe:=StrToExt(ExtractN(2, SB));

    If Filter[i].Value[3]='^' Then
    Begin
      If (SBe<=SCDataAll)And(SCDataAll<=SVe) Then
        CanAdd:=True;
      End
    Else
    Begin

```

```

        If (SBe<SCDataAll) And (SCDataAll<Sve) Then
            CanAdd:=True;
        End;
    End;
End;

    // Кінець перевірок
End;
End
Else
Begin
    CanAdd:=True;
End;

If CanAdd Then
Begin
    SetLength (AllStu, Length (AllStu)+1);
    Inc (max);
    AllStu[High (AllStu)].Name:=Groups[i].Student[j].Name;
    AllStu[High (AllStu)].SGroup:=Groups[i].Name;
    AllStu[High (AllStu)].Modules[1].T:=Round (SCData[1, 2]);
    AllStu[High (AllStu)].Modules[1].P:=Round (SCData[1, 3]);
    AllStu[High (AllStu)].Modules[1].V:=Round (SCData[1, 4]);
    AllStu[High (AllStu)].Modules[1].R:=Round (SCData[1, 5]);
    AllStu[High (AllStu)].Modules[1].A:=Round (SCData[1, 6]);
    AllStu[High (AllStu)].Modules[2].T:=Round (SCData[2, 2]);
    AllStu[High (AllStu)].Modules[2].P:=Round (SCData[2, 3]);
    AllStu[High (AllStu)].Modules[2].V:=Round (SCData[2, 4]);
    AllStu[High (AllStu)].Modules[2].R:=Round (SCData[2, 5]);
    AllStu[High (AllStu)].Modules[2].A:=Round (SCData[2, 6]);
    AllStu[High (AllStu)].Modules[3].T:=Round (SCData[3, 2]);
    AllStu[High (AllStu)].Modules[3].P:=Round (SCData[3, 3]);
    AllStu[High (AllStu)].Modules[3].V:=Round (SCData[3, 4]);
    AllStu[High (AllStu)].Modules[3].R:=Round (SCData[3, 5]);
    AllStu[High (AllStu)].Modules[3].A:=Round (SCData[3, 6]);
    For k:=1 To 3 Do
        Begin
            AllStu[High (AllStu)].Mark[k]:=SCData[k, 1];
            If Cpos=k Then
                AllStu[High (AllStu)].CurrMark:=AllStu[High (AllStu)].Mark[k];
            End;
            AllStu[High (AllStu)].HoleMark:=SCDataAll;
        End;
    End;
End;

// Сортування
For i:=1 To max*max*max Do
Begin
    Flag:=False;
    For k:=Low (AllStu) To High (AllStu)-1 Do
        Begin
            If AllStu[k].HoleMark<AllStu[k+1].HoleMark Then
                Begin
                    SortI (k, k+1, AllStu);
                    Flag:=True;
                End;
            End;
        End;
    If Flag=False Then Break;
End;

// Результат
Result:=Max;
End;

End.

```

## ФАЙЛ PRGDATA.PAS

```

Unit prgData;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Вихідний код до магістерської роботи
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу
аудиторного фонду розкладу занять для факультетів

Виконав: Горшков Владислав Олегович
студент 6 курсу, групи КН-22М-2

Керівник: Коваленко А.С.
Кропивницький 2023
}

Interface // Інтерфейс

Uses // підключення
  prgTypes, SysUtils, abIniFiles, frmProgressU, Forms, prgFunc_Proc, Dialogs,
  Controls;

Var // змінні
  Data: TData;
  FileData, tempFileData: TFileData;
  Groups: TAllGroups;
  CurrFile: String;
  prCode: integer=425958;
  Saved: boolean=True;
  StudSave: boolean=True;
  RepFilter: TFilter;

Function GetDN(SName: String): String;
Procedure NewFileData;
Procedure LoadFileData(DataFile: String);
Procedure SaveFileData(DataFile: String);
Procedure AddRepFilter(I, V: String);
Procedure DelRepFilter(I: integer);

Procedure LoadMainData;
Procedure SaveMainData;
Procedure AddRescent(FileName: String);

Implementation // Реалізація

Function GetDN(SName: String): String;
Var // змінні
  sDir: String;
Begin
  sDir:=ExtractFilePath(Application.ExeName);
  If (sDir[Length(sDir)]<>'\'')
  Then sDir:=sDir+'\'';
  Result:=sDir+sName;
End;

Procedure AddRescent(FileName: String);
Var // змінні
  i: integer;
  Flag: boolean;
Begin
  If (FileName<>'')And(FileExists(FileName)) Then
  Begin
    Flag:=False;
    For i:=Low(Data.Rescent)To High(Data.Rescent) Do
    Begin
      If Data.Rescent[i]=FileName Then
      Begin

```

```

        Flag:=True;
        Break;
    End;
End;
If Flag=False Then
Begin
    Setlength(Data.Rescent, length(Data.Rescent)+1);
    For i:=High(Data.Rescent)-1 Downto Low(Data.Rescent) Do
    Begin
        Data.Rescent[i+1]:=Data.Rescent[i];
    End;
    Data.Rescent[0]:=FileName;
    SaveMainData;
End;
End;
End;

Procedure LoadMainData;
Var // змяшні
    Ini: TabIniFile;
    Coll: integer;
    i: integer;
    DataFile: String;
Begin
    DataFile:=GetDN('Data.mcd');
    If FileExists(DataFile) Then
    Begin
        CryptFile(DataFile, 567344);
    End;
    Ini:=TabIniFile.Create(DataFile);

    Data.OnDel:=Ini.ReadBool('Data', 'OnDel', True);
    Data.RescentMax:=Ini.ReadInteger('Data', 'RescentMax', 7);
    Coll:=Ini.ReadInteger('Rescent', 'Count', 0);
    Setlength(Data.Rescent, coll);
    For i:=1 To coll Do
    Begin
        Data.Rescent[i-1]:=Ini.ReadString('Rescent', IntToStr(i), '');
    End;
    Ini.Free;
    If FileExists(DataFile) Then
    Begin
        CryptFile(DataFile, 567344);
    End;
End;

Procedure SaveMainData;
Var // змяшні
    Ini: TabIniFile;
    i: integer;
    DataFile: String;
Begin
    DataFile:=GetDN('Data.mcd');
    If FileExists(DataFile) Then
        Deletefile(DataFile);

    Ini:=TabIniFile.Create(DataFile);
    Ini.WriteBool('Data', 'OnDel', Data.OnDel);
    Ini.WriteInteger('Data', 'RescentMax', Data.RescentMax);
    Ini.WriteInteger('Rescent', 'Count', Length(Data.Rescent));
    For i:=1 To Length(Data.Rescent) Do
    Begin
        Ini.WriteString('Rescent', IntToStr(i), Data.Rescent[i-1]);
    End;
    Ini.Free;
    CryptFile(DataFile, 567344);
End;

Procedure NewFileData;

```

```

Begin
  FileData.MaxBal.Y:=12;
  FileData.MaxBal.X:=72;
  FileData.UpPredel.Y:=12;
  FileData.UpPredel.X:=58;
  FileData.DownPredel.Y:=4;
  FileData.DownPredel.X:=24;
  FileData.Formula:='(75*T+2*V+3*A+20*R)/100';
  FileData.Pass.Password:='';
  FileData.Pass.OnOpen:=False;
  FileData.Pass.OnEdit:=False;
  SetLength(Groups, 0);
  CurrFile:='';
  Saved:=True;
End;

Procedure AddRepFilter(I, V: String);
Begin
  SetLength(RepFilter, Length(RepFilter)+1);
  RepFilter[High(RepFilter)].Ident:=I;
  RepFilter[High(RepFilter)].Value:=V;
End;

Procedure DelRepFilter(I: integer);
Begin
  RepFilter[i].Ident:=RepFilter[High(RepFilter)].Ident;
  RepFilter[i].Value:=RepFilter[High(RepFilter)].Value;
  SetLength(RepFilter, Length(RepFilter)-1);
End;

Procedure LoadFileData(DataFile: String);
Var // змінні
  Ini: TabIniFile;
  Col1, Col2: integer;
  i, j: integer;
  Flag, Flag2: boolean;
  S: String;
Begin
  CryptFile(DataFile, prCode);
  Ini:=TabIniFile.Create(DataFile);

  FileData.Pass.Password:=Ini.ReadString('Password', 'Pass', '');
  FileData.Pass.OnOpen:=Ini.ReadBool('Password', 'OnOpen', False);
  FileData.Pass.OnEdit:=Ini.ReadBool('Password', 'OnEdit', False);
  Flag:=True;
  If (FileData.Pass.Password<>'') {And(FileData.Pass.OnOpen=True)} Then
  Begin
    Flag:=False;
    Repeat
      Flag2:=False;
      S:=InputBox('Пароль', 'Файл захищен паролем. Введіть його:', '');
      If S='' Then Flag2:=True
      Else
      Begin
        If S=FileData.Pass.Password Then
        Begin
          Flag2:=True;
          Flag:=True;
        End
        Else
          If MessageDLG('Це невірний пароль. Спробуйте ще?', mtError, [mbYes,
mbNo], 0)=mrNo Then Flag2:=True;
        End;
      Until Flag2=True;
    End;
  End;

  If Flag=True Then
  Begin
    Col1:=Ini.ReadInteger('Data', 'Groups', 0);

```

```

frmProgress.pbMain.Max:=0;
For i:=1 To Col1 Do
Begin

frmProgress.pbMain.Max:=frmProgress.pbMain.Max+Ini.ReadInteger('Group'+IntToStr(
i), 'StudentsCol', 0);
End;
frmProgress.pbMain.Position:=0;
frmProgress.lblMain.Caption:='Відкриття даних';
frmProgress.Visible:=True;

FileData.MaxBal.Y:=Ini.ReadFloat('Data', 'MaxBal_Y', 12);
FileData.MaxBal.X:=Ini.ReadFloat('Data', 'MaxBal_X', 72);
FileData.UpPredel.Y:=Ini.ReadFloat('Data', 'UpP_Y', 12);
FileData.UpPredel.X:=Ini.ReadFloat('Data', 'UpP_X', 58);
FileData.DownPredel.Y:=Ini.ReadFloat('Data', 'DoP_Y', 4);
FileData.DownPredel.X:=Ini.ReadFloat('Data', 'DoP_X', 24);
FileData.Formula:=Ini.ReadString('Data', 'Formula',
'(75*T+2*V+3*A+20*R)/100');

For i:=1 To Col1 Do
Begin
SetLength(Groups, i);
Col2:=Ini.ReadInteger('Group'+IntToStr(i), 'StudentsCol', 0);
Groups[i-1].Name:=Ini.ReadString('Group'+IntToStr(i), 'Name', '');
For j:=1 To Col2 Do
Begin
SetLength(Groups[i-1].Student, j);
Groups[i-1].Student[j-1].Name:=Ini.ReadString('Stud'+IntToStr(i)+IntToStr(j),
'Name', '');
Groups[i-1].Student[j-
1].Modules[1].A:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'A1', 0);
Groups[i-1].Student[j-
1].Modules[1].V:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'V1', 0);
Groups[i-1].Student[j-
1].Modules[1].T:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'T1', 0);
Groups[i-1].Student[j-
1].Modules[1].R:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'R1', 0);
Groups[i-1].Student[j-
1].Modules[1].P:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'P1', 0);
Groups[i-1].Student[j-
1].Modules[2].A:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'A2', 0);
Groups[i-1].Student[j-
1].Modules[2].V:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'V2', 0);
Groups[i-1].Student[j-
1].Modules[2].T:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'T2', 0);
Groups[i-1].Student[j-
1].Modules[2].R:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'R2', 0);
Groups[i-1].Student[j-
1].Modules[2].P:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'P2', 0);
Groups[i-1].Student[j-
1].Modules[3].A:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'A3', 0);
Groups[i-1].Student[j-
1].Modules[3].V:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'V3', 0);
Groups[i-1].Student[j-
1].Modules[3].T:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'T3', 0);
Groups[i-1].Student[j-
1].Modules[3].R:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'R3', 0);
Groups[i-1].Student[j-
1].Modules[3].P:=Ini.ReadInteger('Stud'+IntToStr(i)+IntToStr(j), 'P3', 0);
frmProgress.pbMain.Position:=frmProgress.pbMain.Position+1;
End;
End;
Ini.Free;
CryptFile(DataFile, prCode);
End
Else
Begin

```

```

    Ini.Free;
    CryptFile(DataFile, prCode);
    NewFileData;
End;
frmProgress.Visible:=False;
End;

Procedure SaveFileData(DataFile: String);
Var // зміни
    Ini: TabIniFile;
    i, j: integer;
Begin
    If FileExists(DataFile) Then
        Deletefile(DataFile);
    frmProgress.pbMain.Max:=0;
    For i:=1 To Length(Groups) Do
        Begin
            frmProgress.pbMain.Max:=frmProgress.pbMain.Max+Length(Groups[i-1].Student);
        End;
    frmProgress.pbMain.Position:=0;
    frmProgress.lblMain.Caption:='Збереження даних';
    frmProgress.Visible:=True;

    Ini:=TabIniFile.Create(DataFile);
    Ini.WriteString('Password', 'Pass', FileData.Pass.Password);
    Ini.WriteBool('Password', 'OnOpen', FileData.Pass.OnOpen);
    Ini.WriteBool('Password', 'OnEdit', FileData.Pass.OnEdit);

    Ini.WriteInteger('Data', 'Groups', Length(Groups));
    Ini.WriteFloat('Data', 'MaxBal_X', FileData.MaxBal.X);
    Ini.WriteFloat('Data', 'MaxBal_Y', FileData.MaxBal.Y);
    Ini.WriteFloat('Data', 'UpP_Y', FileData.UpPredel.Y);
    Ini.WriteFloat('Data', 'UpP_X', FileData.UpPredel.X);
    Ini.WriteFloat('Data', 'DoP_Y', FileData.DownPredel.Y);
    Ini.WriteFloat('Data', 'DoP_X', FileData.DownPredel.X);
    Ini.WriteString('Data', 'Formula', FileData.Formula);
    For i:=1 To Length(Groups) Do
        Begin
            Ini.WriteInteger('Group'+IntToStr(i), 'StudentsCol', Length(Groups[i-1].Student));
            Ini.WriteString('Group'+IntToStr(i), 'Name', Groups[i-1].Name);
            For j:=1 To Length(Groups[i-1].Student) Do
                Begin
                    Ini.WriteString('Stud'+IntToStr(i)+IntToStr(j), 'Name', Groups[i-1].Student[j-1].Name);
                    Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'A1', Groups[i-1].Student[j-1].Modules[1].A);
                    Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'V1', Groups[i-1].Student[j-1].Modules[1].V);
                    Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'T1', Groups[i-1].Student[j-1].Modules[1].T);
                    Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'R1', Groups[i-1].Student[j-1].Modules[1].R);
                    Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'A2', Groups[i-1].Student[j-1].Modules[2].A);
                    Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'V2', Groups[i-1].Student[j-1].Modules[2].V);
                    Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'T2', Groups[i-1].Student[j-1].Modules[2].T);
                    Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'R2', Groups[i-1].Student[j-1].Modules[2].R);
                    Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'A3', Groups[i-1].Student[j-1].Modules[3].A);
                    Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'V3', Groups[i-1].Student[j-1].Modules[3].V);
                    Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'T3', Groups[i-1].Student[j-1].Modules[3].T);
                    Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'R3', Groups[i-1].Student[j-1].Modules[3].R);
                End;
            End;
        End;
    End;
End;

```

```
Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'P1', Groups[i-1].Student[j-1].Modules[1].P);
Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'P2', Groups[i-1].Student[j-1].Modules[2].P);
Ini.WriteInteger('Stud'+IntToStr(i)+IntToStr(j), 'P3', Groups[i-1].Student[j-1].Modules[3].P);
frmProgress.pbMain.Position:=frmProgress.pbMain.Position+1;
  End;
  End;
  Ini.Free;
  CryptFile(DataFile, prCode);
  frmProgress.Visible:=False;
End;

End.
```

K6П3 - 2023

## ФАЙЛ FRMREPORTU.PAS

```

Unit frmReportU;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Вихідний код до магістерської роботи
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу
аудиторного фонду розкладу занять для факультетів

Виконав: Горшков Владислав Олегович
студент 6 курсу, групи КН-22М-2

Керівник: Коваленко А.С.
Кропивницький 2023
}

Interface // Інтерфейс

Uses // підключення
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, FR_DSet, FR_Class, prgData, prgTypes,
  FR_E_TXT, FR_E_RTF, glWizardHeader, ExtCtrls, RxDGIF, ActnList, FR_E_HTM,
  FR_E_CSV, prgFilter, frmFilterU;

Type // типи
  TfrmReport=Class(TForm)
    CurrReport: TfrReport;
    SomeDataset: TfrUserDataset;
    Header: TglWizardHeader;
    Panel: TPanel;
    btnNext: TButton;
    Pages: TNotebook;
    PicPanel: TPanel;
    Image: TImage;
    btnOK: TButton;
    btnPrev: TButton;
    ActionList: TActionList;
    lblTitle: TLabel;
    edtTitle: TEdit;
    lblReportType: TLabel;
    cbReportType: TComboBox;
    GroupBox1: TGroupBox;
    PagesStat: TNotebook;
    lblM: TLabel;
    cbM1: TCheckBox;
    cbM2: TCheckBox;
    cbM3: TCheckBox;
    lblGenerate: TLabel;
    rgFinish: TRadioGroup;
    btnPreview: TButton;
    frText: TfrTextExport;
    frRTF: TfrRTFExport;
    frCSV: TfrCSVExport;
    frHTM: TfrHTMExport;
    dlgSave: TSaveDialog;
    actDelFilter: TAction;
    actAddFilter: TAction;
    lblFilter: TLabel;
    lbFilter: TListBox;
    btnDelFilter: TButton;
    btnAddFilter: TButton;
    rgFull: TRadioGroup;
    Procedure UpdateFilterList;
    Function RoundMy(Value: extended): extended;
    Procedure CurrReportGetValue(Const ParName: String;
      Var ParValue: Variant);
    Procedure ActionListUpdate(Action: TBasicAction; Var Handled: Boolean);

```

```

Procedure btnPrevClick(Sender: TObject);
Procedure GenerateR;
Procedure CheckPage0;
Procedure CheckPage1;
Procedure CheckPage3;
Procedure btnNextClick(Sender: TObject);
Procedure btnPreviewClick(Sender: TObject);
Procedure FormCreate(Sender: TObject);
Procedure actDelFilterExecute(Sender: TObject);
Procedure actAddFilterExecute(Sender: TObject);
Procedure cbReportTypeChange(Sender: TObject);
Private
  { Private declarations }
Public
  { Public declarations }
End;

Var // змінні
  frmReport: TfrmReport;
  currp: integer=0;
  CurrT: integer=1;
  RepFile: String;

Implementation // Реалізація

{$R *.dfm}

Procedure TfrmReport.UpdateFilterList;
Var // змінні
  i: integer;
  S: String;
Begin
  lbFilter.Items.BeginUpdate;
  lbFilter.Items.Clear;
  If Length(RepFilter)>0 Then
  Begin
    For i:=Low(RepFilter)To High(RepFilter) Do
    Begin
      If RepFilter[i].Ident='name' Then
      Begin
        S:='Стр.: ';
        If RepFilter[i].Value[1]='^' Then
          S:=S+'Пер.; ';
        If RepFilter[i].Value[2]='G' Then
          S:=S+'Група; '
        Else
          S:=S+'ІІІБ; ';
        S:=S+Copy(RepFilter[i].Value, 4, Length(RepFilter[i].Value)-3);
      End;
      If RepFilter[i].Ident='module' Then
      Begin
        S:='Чис.: ';
        Case RepFilter[i].Value[1] Of
          'A': S:=S+'Зар.; ';
          'M': S:=S+'Мод.; ';
          't': S:=S+'Теор.; ';
          'p': S:=S+'Практ.; ';
          'r': S:=S+'РГР; ';
          'a': S:=S+'Ауд.; ';
          'v': S:=S+'Відв.; ';
        End;
        Case RepFilter[i].Value[2] Of
          '1': S:=S+'M1; ';
          '2': S:=S+'M2; ';
          '3': S:=S+'M3; ';
        End;
        S:=S+Copy(RepFilter[i].Value, 4, Length(RepFilter[i].Value)-3);
      End;
      lbFilter.Items.Append(S);
    End;
  End;

```

```

    End;
End
Else
    lbFilter.Items.Append('Всі записи');
lbFilter.Items.EndUpdate;
End;

Function TfrmReport.RoundMy(Value: extended): extended;
Begin
    Result:=Round(Value*10)/10;
End;

Procedure TfrmReport.CurrReportGetValue(Const ParName: String;
    Var ParValue: Variant);
Begin
    If ParName='Name' Then
        ParValue:=AllStudents[SomeDataset.RecNo].Name;
    If ParName='Group' Then
        ParValue:=AllStudents[SomeDataset.RecNo].SGroup;
    If ParName='No' Then
        ParValue:=IntToStr(SomeDataset.RecNo+1)+'.';
    If ParName='Teor' Then
        ParValue:=AllStudents[SomeDataset.RecNo].Modules[CurrT].T;
    If ParName='A' Then
        ParValue:=AllStudents[SomeDataset.RecNo].Modules[CurrT].A;
    If ParName='V' Then
        ParValue:=AllStudents[SomeDataset.RecNo].Modules[CurrT].V;
    If ParName='R' Then
        ParValue:=AllStudents[SomeDataset.RecNo].Modules[CurrT].R;
    If ParName='Pract' Then
        ParValue:=AllStudents[SomeDataset.RecNo].Modules[CurrT].P;
    If ParName='M1' Then
        ParValue:=RoundMy(AllStudents[SomeDataset.RecNo].Mark[1]);
    If ParName='M2' Then
        ParValue:=RoundMy(AllStudents[SomeDataset.RecNo].Mark[2]);
    If ParName='M3' Then
        ParValue:=RoundMy(AllStudents[SomeDataset.RecNo].Mark[3]);
    If ParName='AC' Then
        ParValue:=RoundMy(AllStudents[SomeDataset.RecNo].Mark[CurrT]);
    If ParName='Rang' Then
        ParValue:=FloatToStr(RoundMy(AllStudents[SomeDataset.RecNo].HoleMark));
    If ParName='T' Then

ParValue:=FloatToStr(RoundMy(Solve(AllStudents[SomeDataset.RecNo].Modules[CurrT]
.P+AllStudents[SomeDataset.RecNo].Modules[CurrT].T)));
    If ParName='All' Then
        ParValue:=RoundMy(AllStudents[SomeDataset.RecNo].CurrMark);
    If ParName='Title' Then
        ParValue:=edtTitle.Text;
End;

Procedure TfrmReport.ActionListUpdate(Action: TBasicAction;
    Var Handled: Boolean);
Begin
    btnNext.Enabled:=(Pages.PageIndex<3)And(Pages.PageIndex<>2);
    btnOK.Enabled:=(Pages.PageIndex<>2);
    actDelFilter.Enabled:=(lbFilter.ItemIndex>-
1)And(lbFilter.Items.Strings[lbFilter.ItemIndex]<>'Всі записи');
End;

Procedure TfrmReport.btnPrevClick(Sender: TObject);
Begin
    Case currp Of
        1: Begin Header.PageNo:=0; Pages.PageIndex:=0; currp:=0;
btnPrev.Enabled:=False; End;
        3: Begin Header.PageNo:=1; Pages.PageIndex:=1; currp:=1;
btnNext.Caption:='Наступна >>'; End;
    End;
End;

```

```

Procedure TfrmReport.GenerateR;
Begin
  SomeDataset.RangeEndCount:=Filter(RepFilter, cbM1.Checked, cbM2.Checked,
cbM3.Checked, AllStudents);
End;

Procedure TfrmReport.CheckPage0;
Var // змінні
  S: String;
  MySet: Set Of 1..250;
Begin
  S:='';
  If cbReportType.ItemIndex=-1 Then S:='Виберіть тип статистики!!!';
  If edtTitle.Text='' Then S:='Треба заповнити поле "Заголовок"!!!';
  If S<>'' Then
  Begin
    MessageDlg(S, mtError, [mbOk], 0);
  End
  Else
  Begin
    currp:=1;
    Header.PageNo:=1;
    Pages.PageIndex:=1;
    MySet:= [0, 1];
    Case cbReportType.ItemIndex Of
      0: RepFile:=prgData.GetDN('Reports\Simple.rpt');
      1: RepFile:=prgData.GetDN('Reports\Complex.rpt');
    End;
    If cbReportType.ItemIndex In MySet Then
      rgFull.Visible:=True
    Else
      rgFull.Visible:=False;
    CurrReport.LoadFromFile(RepFile);
  End;
End;

Procedure TfrmReport.CheckPage1;
Var // змінні
  S: String;
  Flag: boolean;
Begin
  S:='Виберіть хоча б один пункт!!!';
  Flag:=True;
  If (cbM1.Checked=False)And(cbM2.Checked=False)And(cbM3.Checked=False) Then
  Begin
    Flag:=False;
  End;
  If Flag=False Then
  Begin
    MessageDlg(S, mtError, [mbOk], 0);
  End
  Else
  Begin
    currp:=2;
    Header.PageNo:=2;
    Pages.PageIndex:=2;
    Cpos:=rgFull.ItemIndex+1;
    CurrT:=cpos;
    GenerateR;
    btnNext.Caption:='Закінчити >>';
    Currp:=3;
    Header.PageNo:=3;
    Pages.PageIndex:=3;
  End;
End;

Procedure TfrmReport.CheckPage3;
Var // змінні

```

```

FN: String;
Begin
  Case rgFinish.ItemIndex Of
    0: Begin
      CurrReport.PrepareReport;
      CurrReport.PrintPreparedReport('1', 1, false, frAll);
    End;
    1: Begin
      If dlgSave.Execute Then
        Begin
          FN:=dlgSave.FileName;
          If pos('.rtf', FN)=0 Then FN:=FN+'.rtf';
          CurrReport.PrepareReport;
          CurrReport.ExportTo(frRTF, FN);
        End;
      End;
    2: Begin
      If dlgSave.Execute Then
        Begin
          FN:=dlgSave.FileName;
          If pos('.csv', FN)=0 Then FN:=FN+'.csv';
          CurrReport.PrepareReport;
          CurrReport.ExportTo(frCSV, FN);
        End;
      End;
    3: Begin
      If dlgSave.Execute Then
        Begin
          FN:=dlgSave.FileName;
          If pos('.htm', FN)=0 Then FN:=FN+'.htm';
          CurrReport.PrepareReport;
          CurrReport.ExportTo(frHTM, FN);
        End;
      End;
    4: Begin
      If dlgSave.Execute Then
        Begin
          FN:=dlgSave.FileName;
          If pos('.txt', FN)=0 Then FN:=FN+'.txt';
          CurrReport.PrepareReport;
          CurrReport.ExportTo(frText, FN);
        End;
      End;
    End;
  ModalResult:=mrOK;
End;

Procedure TfrmReport.btnNextClick(Sender: TObject);
Begin
  Case currp Of
    0: CheckPage0;
    1: CheckPage1;
    3: CheckPage3;
  End;
  btnPrev.Enabled:=True;
End;

Procedure TfrmReport.btnPreviewClick(Sender: TObject);
Begin
  CurrReport.ShowReport;
End;

Procedure TfrmReport.FormCreate(Sender: TObject);
Begin
  currp:=0;
  Header.PageNo:=0;
  Pages.PageIndex:=0;
  PagesStat.PageIndex:=0;
  UpdateFilterList;

```

```
End;

Procedure TfrmReport.actDelFilterExecute(Sender: TObject);
Begin
  DelRepFilter(lbFilter.ItemIndex);
  UpdateFilterList;
End;

Procedure TfrmReport.actAddFilterExecute(Sender: TObject);
Var // змінні
  frmFilter: TfrmFilter;
Begin
  frmFilter:=TfrmFilter.Create(self);
  If frmFilter.ShowModal=mrOk Then UpdateFilterList;
  frmFilter.Free;
End;

Procedure TfrmReport.cbReportTypeChange(Sender: TObject);
Begin
  PagesStat.PageIndex:=cbReportType.ItemIndex+1;
  Case PagesStat.PageIndex Of
    0: PagesStat.PageIndex:=2;
    1: PagesStat.PageIndex:=2;
  End;
End;

End.
```

## ФАЙЛ FRMGROUPSU.PAS

```

Unit frmGroupsU;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Вихідний код до магістерської роботи
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу
аудиторного фонду розкладу занять для факультетів

Виконав: Горшков Владислав Олегович
студент 6 курсу, групи КН-22М-2

Керівник: Коваленко А.С.
Кропивницький 2023
}

Interface // Інтерфейс

Uses // підключення
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, ActnList, prgData;

Type // типи
  TfrmGroups=Class(TForm)
    Panel: TPanel;
    lbGroups: TListBox;
    lblGroups: TLabel;
    btnAdd: TButton;
    btnEdit: TButton;
    btnDel: TButton;
    ShapeBar: TShape;
    btnAccept: TButton;
    ActionList: TActionList;
    actAdd: TAction;
    actEdit: TAction;
    actDel: TAction;
    Procedure UpDateList;
    Procedure actAddExecute(Sender: TObject);
    Procedure actEditExecute(Sender: TObject);
    Procedure actDelExecute(Sender: TObject);
    Procedure ActionListUpdate(Action: TBasicAction; Var Handled: Boolean);
  Private
    { Private declarations }
  Public
    { Public declarations }
  End;
Var // змінні
  frmGroups: TfrmGroups;

Implementation // Реалізація

{$R *.dfm}

Procedure TfrmGroups.UpDateList;
Var // змінні
  i: integer;
Begin
  lbGroups.Items.BeginUpdate;
  lbGroups.Items.Clear;
  For i:=1 To Length(Groups) Do
  Begin
    lbGroups.Items.Append(Groups[i-1].Name);
  End;
  lbGroups.Items.EndUpdate;
End;

Procedure TfrmGroups.actAddExecute(Sender: TObject);

```

```

Var // змінні
  S: String;
  i: integer;
  Flag: boolean;
Begin
  S:=InputBox('Додавання запису', 'Введіть назву нової групи:', '');
  If S<>' ' Then
  Begin
    Flag:=False;
    For i:=1 To Length(Groups) Do
    Begin
      If Groups[i-1].Name=S Then
      Begin
        Flag:=True;
        Break;
      End;
    End;
    If Flag=False Then
    Begin
      SetLength(Groups, Length(Groups)+1);
      Groups[High(Groups)].Name:=S;
    End
    Else
      MessageDlg('Така група вже існує!!!', mtError, [mbOK], 0);
  End;
  UpDateList;
End;

Procedure TfrmGroups.actEditExecute(Sender: TObject);
Var // змінні
  S: String;
  i, idx: integer;
  Flag: boolean;
Begin
  For i:=1 To Length(Groups) Do
  Begin
    If Groups[i-1].Name=lbGroups.Items.Strings[lbGroups.ItemIndex] Then
    Begin
      idx:=i;
      Break;
    End;
  End;
  S:=InputBox('Редагування запису', 'Введіть нову назву групи:',
lbGroups.Items.Strings[lbGroups.ItemIndex]);
  If S<>' ' Then
  Begin
    Flag:=False;
    For i:=1 To Length(Groups) Do
    Begin
      If (i<>idx) And (Groups[i-1].Name=S) Then
      Begin
        Flag:=True;
        Break;
      End;
    End;
    If Flag=False Then
    Begin
      Groups[idx-1].Name:=S;
    End
    Else
      MessageDlg('Така група вже існує!!!', mtError, [mbOK], 0);
  End;
  UpDateList;
End;

Procedure TfrmGroups.actDelExecute(Sender: TObject);
Var // змінні
  S: String;
  i, idx: integer;

```

```

Begin
  For i:=1 To Length(Groups) Do
    Begin
      If Groups[i-1].Name=lbGroups.Items.Strings[lbGroups.ItemIndex] Then
        Begin
          idx:=i-1;
          Break;
        End;
      End;
      If Length(Groups[idx].Student)>0 Then
        S:='Ця група має записи!!!'+#13+'Всеодно знищити?'
      Else
        S:='Ви рішуче бажаєте видалити цю групу?';
      If MessageDlg(S, mtConfirmation, [mbYes, mbNo], 0)=mrYes Then
        Begin
          Groups[idx].Name:=Groups[High(Groups)].Name;
          SetLength(Groups[idx].Student, Length(Groups[High(Groups)].Student));
          For i:=1 To Length(Groups[High(Groups)].Student) Do
            Begin
              Groups[idx].Student[i].Name:=Groups[High(Groups)].Student[i].Name;
              Groups[idx].Student[i].Modules[1].A:=Groups[High(Groups)].Student[i].Modules[1].
              A;
              Groups[idx].Student[i].Modules[1].V:=Groups[High(Groups)].Student[i].Modules[1].
              V;
              Groups[idx].Student[i].Modules[1].R:=Groups[High(Groups)].Student[i].Modules[1].
              R;
              Groups[idx].Student[i].Modules[1].T:=Groups[High(Groups)].Student[i].Modules[1].
              T;
              Groups[idx].Student[i].Modules[2].A:=Groups[High(Groups)].Student[i].Modules[2].
              A;
              Groups[idx].Student[i].Modules[2].V:=Groups[High(Groups)].Student[i].Modules[2].
              V;
              Groups[idx].Student[i].Modules[2].R:=Groups[High(Groups)].Student[i].Modules[2].
              R;
              Groups[idx].Student[i].Modules[2].T:=Groups[High(Groups)].Student[i].Modules[2].
              T;
              Groups[idx].Student[i].Modules[3].A:=Groups[High(Groups)].Student[i].Modules[3].
              A;
              Groups[idx].Student[i].Modules[3].V:=Groups[High(Groups)].Student[i].Modules[3].
              V;
              Groups[idx].Student[i].Modules[3].R:=Groups[High(Groups)].Student[i].Modules[3].
              R;
              Groups[idx].Student[i].Modules[3].T:=Groups[High(Groups)].Student[i].Modules[3].
              T;
              Groups[idx].Student[i].Modules[3].p:=Groups[High(Groups)].Student[i].Modules[3].
              P;
              Groups[idx].Student[i].Modules[2].p:=Groups[High(Groups)].Student[i].Modules[2].
              P;
              Groups[idx].Student[i].Modules[1].p:=Groups[High(Groups)].Student[i].Modules[1].
              P;
            End;
          SetLength(Groups, Length(Groups)-1);
          UpdateList;
        End;
      End;
    End;

  Procedure TfrmGroups.ActionListUpdate(Action: TBasicAction;
    Var Handled: Boolean);
  Begin
    actAdd.Enabled:=(lbGroups.Items.Count<200);
    actEdit.Enabled:=(lbGroups.ItemIndex>-1);
    actDel.Enabled:=(lbGroups.ItemIndex>-1);
  End;

End.

```

## ФАЙЛ FRMFORMULA.U.PAS

```

Unit frmFormulaU;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Вихідний код до магістерської роботи
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу
аудиторного фонду розкладу занять для факультетів

Виконав: Горшков Владислав Олегович
студент 6 курсу, групи КН-22М-2

Керівник: Коваленко А.С.
Кропивницький 2023
}

Interface // Інтерфейс

Uses // підключення
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  prgData,
  Dialogs, ExtCtrls, StdCtrls, ComCtrls, glPage, glTab, Math, Grids,
  TFlatEditUnit, TFlatSpinEditUnit, TFlatTitlebarUnit;

Type // типи
  TfrmFormula=Class(TForm)
    Panel: TPanel;
    btnAccept: TButton;
    btnCancel: TButton;
    pcOptions: TglPageControl;
    tsFormula: TTabSheet;
    tsPredeli: TTabSheet;
    lblCaption: TLabel;
    lblDesc: TLabel;
    edtFormula: TEdit;
    pbLim: TPaintBox;
    lblMaxX: TLabel;
    seMaxX: TFlatSpinEditInteger;
    lblMaxY: TLabel;
    seMaxY: TFlatSpinEditInteger;
    lblDoX: TLabel;
    seDoX: TFlatSpinEditInteger;
    lblUpX: TLabel;
    seUpX: TFlatSpinEditInteger;
    lblDoY: TLabel;
    seDoY: TFlatSpinEditInteger;
    lblUpY: TLabel;
    seUpY: TFlatSpinEditInteger;
    ftMain: TFlatTitlebar;
    Procedure RepaintPB;
    Procedure FormKeyPress(Sender: TObject; Var Key: Char);
    Procedure pbLimPaint(Sender: TObject);
    Procedure FormCreate(Sender: TObject);
    Procedure seMaxXChange(Sender: TObject);
    Procedure seMaxYChange(Sender: TObject);
    Procedure seDoXChange(Sender: TObject);
    Procedure seUpXChange(Sender: TObject);
    Procedure seDoYChange(Sender: TObject);
    Procedure seUpYChange(Sender: TObject);
  Private
    { Private declarations }
  Public
    { Public declarations }
  End;

Var // змінні
  frmFormula: TfrmFormula;

```

```

Function F(x: Double): Double;

Implementation // Реализация

{$R *.dfm}

Function F(x: Double): Double;
Begin
  Try
    Result:=0;
    If (x<prgData.tempFileData.DownPredel.X) And (x>=0) Then
      Result:=0;
    If
      (x>=prgData.tempFileData.DownPredel.X) And (x<prgData.tempFileData.UpPredel.X)
    Then
      Begin
        Result:=(prgData.tempFileData.MaxBal.Y/prgData.tempFileData.MaxBal.X)*x;
      End;
      If x>=prgData.tempFileData.UpPredel.X Then
        Begin
          Result:=prgData.tempFileData.MaxBal.Y;
        End;
      Except
        Result:=0;
      End;
    End;
  End;

Procedure TfrmFormula.FormKeyPress(Sender: TObject; Var Key: Char);
Begin
  If Key=#27 Then
    Begin
      Key:=#0;
      btnCancel.Click;
    End;
  If Key=#13 Then
    Begin
      Key:=#0;
      btnAccept.Click;
    End;
  End;

Procedure TfrmFormula.pbLimPaint(Sender: TObject);
Begin
  RepaintPB;
End;

Procedure TfrmFormula.FormCreate(Sender: TObject);
Begin
  edtFormula.Text:=tempFileData.Formula;
  seMaxX.Value:=Round(tempFileData.MaxBal.X);
  seMaxY.Value:=Round(tempFileData.MaxBal.Y);
  seDoX.Value:=Round(tempFileData.DownPredel.X);
  seDoY.Value:=Round(tempFileData.DownPredel.Y);
  seUpX.Value:=Round(tempFileData.UpPredel.X);
  seUpY.Value:=Round(tempFileData.UpPredel.Y);
  RepaintPB;
End;

Procedure TfrmFormula.seMaxXChange(Sender: TObject);
Begin
  Try
    tempFileData.MaxBal.X:=seMaxX.Value;
    RepaintPB;
  Except
  End;
End;

Procedure TfrmFormula.seMaxYChange(Sender: TObject);

```

```
Begin
  Try
    tempFileData.MaxBal.Y:=seMaxY.Value;
    RepaintPB;
  Except
  End;
End;

Procedure TfrmFormula.seDoXChange(Sender: TObject);
Begin
  Try
    tempFileData.DownPredel.X:=seDoX.Value;
    RepaintPB;
  Except
  End;
End;

Procedure TfrmFormula.seUpXChange(Sender: TObject);
Begin
  Try
    tempFileData.UpPredel.X:=seUpX.Value;
    RepaintPB;
  Except
  End;
End;

Procedure TfrmFormula.seDoYChange(Sender: TObject);
Begin
  Try
    tempFileData.DownPredel.Y:=seDoY.Value;
    RepaintPB;
  Except
  End;
End;

Procedure TfrmFormula.seUpYChange(Sender: TObject);
Begin
  Try
    tempFileData.UpPredel.Y:=seUpY.Value;
    RepaintPB;
  Except
  End;
End;

End.
```

## ФАЙЛ FRMFILTERU.PAS

```

Unit frmFilterU;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Вихідний код до магістерської роботи
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу
аудиторного фонду розкладу занять для факультетів

Виконав: Горшков Владислав Олегович
студент 6 курсу, групи КН-22М-2

Керівник: Коваленко А.С.
Кропивницький 2023
}

Interface // Інтерфейс

Uses // підключення
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, ActnList, prgData, TFlatEditUnit,
  TFlatSpinEditUnit;

Type // типи
  TfrmFilter=Class(TForm)
    Panel: TPanel;
    btnNext: TButton;
    btnCancel: TButton;
    Pages: TNotebook;
    rgFilterType: TRadioGroup;
    btnPrev: TButton;
    lblText: TLabel;
    rgFilterField: TRadioGroup;
    cbRegister: TCheckBox;
    edtText: TEdit;
    rgModulFilter: TRadioGroup;
    rgWhereFilter: TRadioGroup;
    gbRange: TGroupBox;
    cbInclude: TCheckBox;
    Label1: TLabel;
    Label2: TLabel;
    seFrom: TFlatSpinEditFloat;
    seTo: TFlatSpinEditFloat;
    Procedure FormKeyPress(Sender: TObject; Var Key: Char);
    Procedure btnPrevClick(Sender: TObject);
    Procedure btnNextClick(Sender: TObject);
    Procedure rgModulFilterClick(Sender: TObject);
  Private
    { Private declarations }
  Public
    { Public declarations }
  End;

Var // змінні
  frmFilter: TfrmFilter;

Implementation // Реалізація

{$R *.dfm}

Procedure TfrmFilter.FormKeyPress(Sender: TObject; Var Key: Char);
Begin
  If Key=#27 Then
  Begin
    Key:=#0;
    btnCancel.Click;
  End;
End;

```

```

End;

Procedure TfrmFilter.btnPrevClick(Sender: TObject);
Begin
  btnPrev.Enabled:=(Pages.PageIndex>0);
  Pages.PageIndex:=0;
  btnNext.Caption:='Наступна >>';
End;

Procedure TfrmFilter.btnNextClick(Sender: TObject);
Var // зміни
  Idt, Val: String;
  c1, c2, c3: char;
Begin
  rgWhereFilter.Enabled:=(rgModulFilter.ItemIndex>0);
  Case Pages.PageIndex Of
    0: Pages.PageIndex:=rgFilterType.ItemIndex+1;
    1: Begin
      Idt:='name';
      c1:=' ';
      If cbRegister.Checked Then c1:='^';
      c2:='N';
      If rgFilterField.ItemIndex=0 Then c2:='G';
      c3:=' ';
      Val:=c1+c2+c3+edtText.Text;
      AddRepFilter(Idt, Val);
      ModalResult:=mrOK;
    End;
    2: Begin
      Idt:='module';
      If rgModulFilter.ItemIndex=0 Then
      Begin
        c1:='A';
        c2:=' ';
        c3:=' ';
        If cbInclude.Checked Then c3:='^';
      End
      Else
      Begin
        c2:=IntToStr(rgModulFilter.ItemIndex)[1];
        Case rgWhereFilter.ItemIndex Of
          0: c1:='M';
          1: c1:='t';
          2: c1:='p';
          3: c1:='r';
          4: c1:='a';
          5: c1:='v';
        End;
        c3:=' ';
        If cbInclude.Checked Then c3:='^';
      End;
      Val:=c1+c2+c3+FloatToStr(seFrom.Value)+' '+FloatToStr(seTo.Value);
      AddRepFilter(Idt, Val);
      ModalResult:=mrOK;
    End;
  End;
  btnPrev.Enabled:=(Pages.PageIndex>0);
  btnNext.Caption:='Закінчити >>';
End;

Procedure TfrmFilter.rgModulFilterClick(Sender: TObject);
Begin
  rgWhereFilter.Enabled:=(rgModulFilter.ItemIndex>0);
End;

End.

```

## ФАЙЛ FRMSTUDENTU.PAS

```

Unit frmStudentU;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Вихідний код до магістерської роботи
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу
аудиторного фонду розкладу занять для факультетів

Виконав: Горшков Владислав Олегович
студент 6 курсу, групи КН-22М-2

Керівник: Коваленко А.С.
Кропивницький 2023
}

Interface // Інтерфейс

Uses // підключення
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, prgData;

Type // типи
  TfrmStudent=Class(TForm)
    Panel: TPanel;
    lblFio: TLabel;
    edtFIO: TEdit;
    lblGroup: TLabel;
    cbGroupFilter: TComboBox;
    btnEditGroups: TButton;
    btnAccept: TButton;
    btnCancel: TButton;
    Procedure UpDateList;
    Procedure FormCreate(Sender: TObject);
    Procedure btnEditGroupsClick(Sender: TObject);
    Procedure FormKeyPress(Sender: TObject; Var Key: Char);
  Private
    { Private declarations }
  Public
    { Public declarations }
  End;

Var // змінні
  frmStudent: TfrmStudent;

Implementation // Реалізація

Uses frmMainModStatU, frmGroupsU;

{$R *.dfm}

Procedure TfrmStudent.UpDateList;
Var // змінні
  i: integer;
Begin
  cbGroupFilter.Items.BeginUpdate;
  cbGroupFilter.Items.Clear;
  For i:=1 To Length(Groups) Do
  Begin
    cbGroupFilter.Items.Append(Groups[i-1].Name);
  End;
  cbGroupFilter.Items.EndUpdate;
End;

Procedure TfrmStudent.FormCreate(Sender: TObject);
Begin
  UpDateList;

```

```

    If cbGroupFilter.Items.Count>0 Then
        cbGroupFilter.ItemIndex:=0;
    End;

    Procedure TfrmStudent.btnEditGroupsClick(Sender: TObject);
    Var // змінні
        frmGroups: TfrmGroups;
        idx, k: integer;
        S: String;
    Begin
        frmGroups:=TfrmGroups.Create(self);
        frmGroups.UpDateList;
        If frmGroups.ShowModal=mrOk Then
            Begin
                prgData.Saved:=False;
            End;
        frmGroups.Free;
        If cbGroupFilter.ItemIndex>-1 Then
            Begin
                S:=cbGroupFilter.Items.Strings[cbGroupFilter.ItemIndex];
                UpdateList;
                k:=-1;
                For idx:=1 To cbGroupFilter.Items.Count Do
                    Begin
                        If cbGroupFilter.Items.Strings[idx-1]=S Then
                            Begin
                                k:=idx-1;
                                Break;
                            End;
                        End;
                    End;
                cbGroupFilter.ItemIndex:=k;
            End
        Else
            UpdateList;
        End;

    Procedure TfrmStudent.FormKeyPress(Sender: TObject; Var Key: Char);
    Begin
        If Key=#27 Then
            Begin
                Key:=#0;
                btnCancel.Click;
            End;
        If Key=#13 Then
            Begin
                Key:=#0;
                btnAccept.Click;
            End;
        End;
    End.

```

## ФАЙЛ UKRSORT.PAS

```

Unit UkrSort;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Вихідний код до магістерської роботи
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу
аудиторного фонду розкладу занять для факультетів

Виконав: Горшков Владислав Олегович
студент 6 курсу, групи КН-22М-2

Керівник: Коваленко А.С.
Кропивницький 2023
}

Interface // Інтерфейс

Function CompareUkrText(S1, S2: String): integer;

Var // змінні
  Chars: Array[1..136] Of char=
    ('1', '2', '3', '4', '5', '6', '7', '8', '9', '0', 'A', 'B', 'C', 'D', 'E',
    'F', 'G',
    'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V',
    'W', 'X',
    'Y', 'Z', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm',
    'n', 'o',
    'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'B', 'Г',
    'Г', 'Д',
    'E', 'E', 'E', 'Ж', 'З', 'И', 'I', 'I', 'Й', 'K', 'Л', 'M', 'H', 'O', 'П',
    'P', 'C',
    'T', 'Y', 'Ф', 'X', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я', 'a',
    'б', 'в',
    'г', 'г', 'д', 'e', 'e', 'e', 'ж', 'з', 'и', 'i', 'i', 'й', 'к', 'л', 'м',
    'н', 'о',
    'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э',
    'ю', 'я');
Implementation // Реалізація

Function CompareUkrText(S1, S2: String): integer;

  Function GetNum(C: char): integer;
  Var // змінні
    i: integer;
  Begin
    Result:=0;
    For i:=1 To 136 Do
      If Chars[i]=C Then
        Begin
          Result:=i;
          exit;
        End;
    End;

  Function CompCh(C1, C2: integer): integer;
  Begin
    If C1=C2 Then Result:=0;
    If C1>C2 Then Result:=1;
    If C1<C2 Then Result:=-1;
  End;
  Var // змінні
    i, xS1, xS2, CurrR: integer;
  Begin
    Result:=0;
    CurrR:=0;
    If S1<>S2 Then

```

```
Begin
  For i:=1 To Length(S1) Do
    Begin
      If Length(S2)>=i Then
        Begin
          xS1:=GetNum(S1[i]);
          xS2:=GetNum(S2[i]);
          If (xS1<>0)And(xS2<>0)And(xS1<>xS2) Then
            CurrR:=CompCh(xS1, xS2)
          Else
            Begin
              If (xS1=0)Or(xS2=0) Then
                Begin
                  If xS2=0 Then CurrR:=1;
                  If xS1=0 Then CurrR:=-1;
                End;
              End;
              If CurrR<>0 Then
                Begin
                  Result:=CurrR;
                  Exit;
                End;
              End
            End
          Else
            Begin
              Result:=CurrR;
              Exit;
            End;
          End;
        End;
      End;
    End;
  End.

```

## ФАЙЛ PRGFUNC\_PROC.PAS

```

Unit prgFunc_Proc;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Вихідний код до магістерської роботи
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу
аудиторного фонду розкладу занять для факультетів

Виконав: Горшков Владислав Олегович
студент 6 курсу, групи КН-22М-2

Керівник: Коваленко А.С.
Кропивницький 2023
}

Interface // Інтерфейс

Uses // підключення
  Windows, SysUtils, Forms;

Function CryptFile(FileName: String; numer: integer): String;

Implementation // Реалізація

Function GetDN(SName: String): String;
Var // змінні
  sDir: String;
Begin
  sDir:=ExtractFilePath(Application.ExeName);
  If (sDir[Length(sDir)]<>'\'')
  Then sDir:=sDir+'\'';
  Result:=sDir+sName;
End;

Function CryptFile(FileName: String; numer: integer): String;
Var // змінні
  buf: Array[1..1024] Of char;
  fp1, fp2: File;
  k: word;
  n, m, q: integer;
  tab: byte;
Begin
  CopyFile(PChar(FileName), PChar(GetDN('coder.tmp')), True);
  FileSetAttr(GetDN('coder.tmp'), 0);
  AssignFile(fp1, GetDN('coder.tmp'));
  Reset(fp1, 1);
  If IOResult<>0 Then
  Begin
    Result:='Ошибка ввода-вывода.';
    Exit;
  End;
  randseed:=numer;
  AssignFile(fp2, FileName);
  Case UpCase(FileName[1]) Of
    'A': q:=1;
    'B': q:=2;
    'C': q:=3;
    'D': q:=4;
    'E': q:=5;
    'F': q:=6;
    'G': q:=7;
    'H': q:=8;
    'I': q:=9;
    'J': q:=10;
    'K': q:=11;
    'L': q:=12;

```

```
'M': q:=13;
'N': q:=14;
'O': q:=15;
'P': q:=16;
'Q': q:=17;
'R': q:=18;
'S': q:=19;
'T': q:=20;
'U': q:=21;
'V': q:=22;
'W': q:=23;
'X': q:=24;
'Y': q:=25;
'Z': q:=26;
End;
If DiskFree(q)<filesize(fp1) Then
Begin
  Result:='Мало места на выходящем диске.';
  Exit;
End;
Rewrite(fp2, 1);
Repeat
  blockread(fp1, buf, sizeof(buf), n);
  For k:=1 To sizeof(buf) Do
  Begin
    tab:=random(256);
    buf[k]:=char(ord(buf[k])Xor tab);
  End;
  blockwrite(fp2, buf, n, m);
Until (n=0)Or(n<>m);
CloseFile(fp1);
CloseFile(fp2);
DeleteFile(GetDN('coder.tmp'));
Result:='Ok';
End;
End.
```

## ФАЙЛ FRMSPLASHU.PAS

```

Unit frmSplashU;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Вихідний код до магістерської роботи
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу
аудиторного фонду розкладу занять для факультетів

Виконав: Горшков Владислав Олегович
студент 6 курсу, групи КН-22М-2

Керівник: Коваленко А.С.
Кропивницький 2023
}

Interface // Інтерфейс

Uses // підключення
  Windows, Forms, ExtCtrls, StdCtrls, TFlatAnimationUnit,
  TFlatProgressBarUnit, jpeg, Classes, Controls;

Type // типи
  TfrmSplash=Class(TForm)
    PB: TFlatProgressBar;
    Animation: TFlatAnimation;
    lblProc: TLabel;
    lblTitle: TLabel;
    Label1: TLabel;
    T1: TTimer;
    Label2: TLabel;
    Procedure FormCreate(Sender: TObject);
    Procedure T1Timer(Sender: TObject);
    Procedure FormCloseQuery(Sender: TObject; Var CanClose: Boolean);
  End;

Var // змінні
  frmSplash: TfrmSplash;

Implementation // Реалізація

{$R *.dfm}

Uses frmMainModStatU;

Procedure TfrmSplash.FormCreate(Sender: TObject);
Begin
  T1.Enabled:=True;
End;

Procedure TfrmSplash.T1Timer(Sender: TObject);
Begin
  Case PB.Position Of
    0: lblProc.Caption:='Ініціалізація...';
    20: lblProc.Caption:='Авторизація...';
    40: Begin lblProc.Caption:='Створення форм...'; End;
    60: lblProc.Caption:='Налагодження опцій...';
    80: lblProc.Caption:='Завантаження бібліотек...';
    90: lblProc.Caption:='Завантаження програми...';
    100: Begin
      T1.Enabled:=False;
      frmMainModStat.WindowState:=wsMaximized;
    End;
  End;
  If PB.Position<100 Then
    PB.Position:=PB.Position+1;
End;

```

```
Procedure TfrmSplash.FormCloseQuery(Sender: TObject;  
  Var CanClose: Boolean);  
Begin  
  CanClose:=Not T1.Enabled;  
End;  
  
End.
```

К6П3 - 2023

## ФАЙЛ PRGTYPES.PAS

```

Unit prgTypes;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Вихідний код до магістерської роботи
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу
аудиторного фонду розкладу занять для факультетів

Виконав: Горшков Владислав Олегович
студент 6 курсу, групи КН-22М-2

Керівник: Коваленко А.С.
Кропивницький 2023
}

Interface // Інтерфейс

Uses // підключення
  Controls;

Type // типи
  {=====}
  TRescent=Array Of String;

  TData=Object
    OnDel: boolean;
    Rescent: TRescent;
    RescentMax: integer;
  End;
  {=====}
  TCoord=Object
    X, Y: extended;
  End;

  TPassword=Object
    Password: String;
    OnOpen, OnEdit: boolean;
  End;

  TFileData=Object
    MaxBal: TCoord;
    UpPredel, DownPredel: TCoord;
    Formula: String;
    Pass: TPassword;
  End;

  TModule=Object
    A, V, R, T, P: integer;
  End;
  TAllModule=Array[1..3] Of TModule;

  TPupil=Object
    Name: String;
    Modules: TAllModule;
  End;
  TAllPupils=Array Of TPupil;

  TGroup=Object
    Name: String;
    Student: TAllPupils;
  End;

  TAllGroups=Array Of TGroup;

  TFilterItem=Object

```

```
    Ident: String;  
    Value: String;  
End;  
  
TFilter=Array Of TFilterItem;
```

```
Implementation // Реалізація
```

```
End.
```

КБПЗ - 2023

## ФАЙЛ FRMABOUTU.PAS

```
Unit frmAboutU;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Вихідний код до магістерської роботи
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу
аудиторного фонду розкладу занять для факультетів

Виконав: Горшков Владислав Олегович
студент 6 курсу, групи КН-22М-2

Керівник: Коваленко А.С.
Кропивницький 2023
}

Interface // Інтерфейс
Uses // підключення
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, jpeg, ExtCtrls, MTCoolButton;

Type // типи
  TfrmAbout=Class(TForm)
    btnOK: TMTCoolButton;
    lblTitle: TLabel;
    Label1: TLabel;
    Label3: TLabel;
    Label2: TLabel;
    Label4: TLabel;
    Image1: TImage;
    Procedure btnOKClick(Sender: TObject);
  Private
    { Private declarations }
  Public
    { Public declarations }
  End;

Var // змінні
  frmAbout: TfrmAbout;

Implementation // Реалізація

{$R *.dfm}

Procedure TfrmAbout.btnOKClick(Sender: TObject);
Begin
  ModalResult:=mrOK;
End;
End.
```

## ФАЙЛ FRMPROGRESSU.PAS

```
Unit frmProgressU;  
{  
Міністерство освіти і науки України  
Центральноукраїнський національний технічний університет  
Факультет механіко-технологічний  
Кафедра кібербезпеки та програмного забезпечення  
Вихідний код до магістерської роботи  
на тему: Дослідження та програмна реалізація системи автоматизованого розподілу  
аудиторного фонду розкладу занять для факультетів  
  
Виконав: Горшков Владислав Олегович  
студент 6 курсу, групи КН-22М-2  
  
Керівник: Коваленко А.С.  
Кропивницький 2023  
}  
  
Interface // Інтерфейс  
Uses // підключення  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, ExtCtrls, ComCtrls, StdCtrls;  
Type // типи  
  TfrmProgress=Class(TForm)  
    Panel: TPanel;  
    pbMain: TProgressBar;  
    lblMain: TLabel;  
  Private  
    { Private declarations }  
  Public  
    { Public declarations }  
  End;  
Var // змінні  
  frmProgress: TfrmProgress;  
Implementation // Реалізація  
{ $R *.dfm }  
End.
```