

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Кафедра кібербезпеки та програмного забезпечення

На правах рукопису

Шевченко Вадим Олександрович

Програмне забезпечення системи Software as a Service

Спеціальність: 123 «Комп'ютерна інженерія»

Освітній ступінь: бакалавр

Науковий керівник:

Дресв Олександр Миколайович

(підпис)

(дата)

кандидат технічних наук, доцент

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри

О.А. Смірнов

(підпис)

ПБ

«_____» _____ 2021 р.

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Спеціальність 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
О.А.Смірнов
« 11 » січня 2021 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Шевченку Вадиму Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи Software as a Service

керівник роботи Дреєв Олександр Миколайович, канд. техн. наук, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 204-02 від 28.12.2020 року

2. Строк подання студентом роботи до захисту 22.05.2021 р.

3. Мета та завдання кваліфікаційної бакалаврської роботи: Метою розробки є програмне забезпечення системи Software as a Service

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

6. Дата видачі завдання « 11 » січня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної бакалаврської роботи	Строк виконання етапів кваліфікаційної бакалаврської роботи	Примітка
1.	Аналіз існуючих систем	10.03.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2021 р.	
3.	Розробка моделі компонента	20.03.2021 р.	
4.	Розробка структур даних	25.03.2021 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2021 р.	
6.	Програмування алгоритмів	10.04.2021 р.	
7.	Оформлення ПЗ	17.04.2021 р.	
8.	Попередній захист роботи	14.05.2021 р.	

Студент _____

(підпис)

_____ (прізвище та ініціали)

Керівник роботи _____

(підпис)

_____ (прізвище та ініціали)

АНОТАЦІЯ

Шевченко В.О. Програмне забезпечення системи Software as a Service. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній кваліфікаційній бакалаврській розроблено програмне забезпечення, яке призначено для системи Software as a Service.

Метою розробки є програмне забезпечення системи Software as a Service.

Результат роботи – програмна реалізація системи Software as a Service.

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 XP/Vista/7/8/10.

Програму розроблено в середовищі Delphi 10.4.

Ключові слова: комп'ютерна інженерія, Software as a Service

ABSTRACT

Shevchenko V.O. Software system Software as a Service. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2021

In this bachelor's qualification the software which is intended for the Software as a Service system is developed.

The purpose of the development is the software of the Software as a Service system.

The result of the work is a software implementation of the Software as a Service system.

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.

Developed user-friendly interface. Instructions for working with software are given.

The program can be used on an IBM PC with Windows XP / Vista / 7/8/10.

The program is developed in the Delphi 10.4 environment.

Keywords: computer engineering, Software as a Service

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми кваліфікаційної бакалаврської роботи.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування	13
2.3 Розгорнута постановка завдання	19
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	21
3.1 Опис функціонування системи	21
3.2 Розробка структурної схеми.....	25
3.3 Розробка функціональної схеми.....	46
3.4 Розробка діаграми процесів	49
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ	51
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	51
4.2 Захист розробленого програмного забезпечення.....	66
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	68
6 ОСНОВНІ ВИСНОВКИ	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72

КБР-123.21.0047.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Шевченко В.О.			Програмне забезпечення системи Software as a Service	Лім.	Аркуш	Аркушіів
Перев.		Дресв О.М.				Б	1	79
Н.контр.		Гермак В.С.			ЦНТУ КІ-18-3СК			
Затв.		Смірнов О.А.						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

СЕД	–	Система електронного документообігу, electronic document management systems (EDMS)
ПЗ	–	Програмне забезпечення
ЦОД	–	Центр Обробки Даних
BYOD	–	ІТ-політика, згідно з якою співробітникам дозволено або рекомендується використовувати особисті мобільні пристрої, Bring your own device
ERP	–	Інтегроване програмне забезпечення для керування трудовими ресурсами, фінансами й активами підприємства
HDX	–	Технологія високої чіткості
IRM	–	Керування правами на доступ до інформації
ІТ	–	Інформаційні технології
ITSM	–	Застосунки для керування ІТ-послугами
SaaS	–	Software as a service – програмне забезпечення як послуга
SSO	–	Точка єдиного входу
VDI	–	Віртуальний робочий стол
VoIP	–	Voice over IP
VPN	–	Virtual Private Network
UC	–	Інструменти об'єднаних комунікацій
UEM	–	Технологія уніфікованого керування кінцевими точками

ВСТУП

Актуальність теми. SaaS – це модель надання ліцензії на програмне забезпечення по підписці. SaaS розшифровується як software as a service – програмне забезпечення як послуга, також зустрічається переклад «інформація як сервіс». SaaS найчастіше – це хмарний застосунок, тобто, що перебуває на серверах в Інтернеті.

Користувач одержує доступ до сервісу через браузер або по API. При цьому його підтримкою цілком займається постачальник послуги. Спрощено говорячи, модель SaaS це коли клієнт працює з готовим застосунком онлайн. Оплачує доступ і максимально швидко одержує на руки готовий інструмент.

Найпростіший приклад SaaS – це Google Docs, безкоштовний сервіс для роботи з документами. Ніяких носіїв, драйверів і установок. Заводьте аккаунт Google, переходите по посиланню й працюєте з текстами, таблицями й презентаціями прямо в браузері. Причому в документах одночасно з вами можуть працювати й інші колеги. Виїхавши у відрядження, можна зайти у свій аккаунт із будь-якого пристрою й продовжити друкувати потрібний документ. При цьому зберігати потрібно тільки зміни настроювань, інші дані зберігаються автоматично.

По суті, SaaS-сервіс це єдине програмне ядро, яке надається в користування клієнтам. Доступ до системи вони одержують через мережу й можуть змінювати настроювання на свій розсуд. Обслуговуванням сервісу цілком займається провайдер послуги, а користувач тільки працює в ній.

Популярність цієї моделі щорічно росте. Тільки торік ринок SaaS збільшився на 21,7%. За прогнозами експертів, така тенденція збережеться й у найближчі роки. Це й не дивно, адже так користувачі одержують у розпорядження сучасні технології практично без зусиль зі своєї сторони.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

Сьогодні з подібними застосунками більшість людей зустрічається щодня. Розповсюджене явище, коли SaaS-сервіси використовуються для бухгалтерського обліку, комунікації із клієнтами, редагування зображень, CRM, аналітики, ERP. Словом, практично для всього.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи Software as a Service.

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

- Огляд існуючих систем Software as a Service.
- Дослідження системи Software as a Service.
- Програмна реалізація системи Software as a Service.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі Software as a Service.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи Software as a Service, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Особливості SaaS model

Робота через веб-інтерфейс

При використанні SaaS у вас на руках не буде нічого, що можна «пощупати»: установочних файлів, папок і документів на комп'ютері. Усі дані розміщуються на серверах постачальника послуги – у так званому «хмарному» сховищі в мережі. Ви ж одержуєте дистанційний доступ до програми й можете працювати з нею в браузері з будь-якого місця на планеті, де є інтернет. У той же час, існують SaaS, які можна купити й установити до себе на сервер.

Плата за доступ

Схеми оплати за SaaS досить різноманітні. Спочатку розглянемо найбільш популярну. У цьому випадку користувач не здобуває дорогу ліцензію ПЗ цілком, як при покупці диска або при завантаженні установочних файлів. І не оплачує значну вартість розробки, як при замовленні індивідуального застосунку. Він як би бере сервіс в оренду, оплачуючи аналог абонплати за доступ до функціоналу на місяць або довше.

Згідно зі стандартною схемою, платіж буде поновлюваним, тому він ще називається підпискою. Клієнт платить за використання сервісу протягом періоду, за кількість користувачів, за число певних дій у системі й т.п. Наприклад, у випадку з CRM плата залежить від кількості співробітників, які будуть одночасно працювати із програмою. Як правило, такі сервіси більш вигідно оплачувати наперед – у цьому випадку користувач одержує відчутну знижку й бонуси.

Деякі SaaS безкоштовно надають обмежений набір функцій, а за доступ до просунутих прийдеться оплатити підписку – це freemium модель. Але частіше SaaS надають кілька тарифів з одмінним набором функцій. От як це виглядає на прикладі Salesforce, компанії, яка була одним з піонерів SaaS:

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

До речі, у деяких сервісів є ще й тариф Enterprise. Звичайно він самий дорогий, але й містить у собі практично безмежні можливості. У цьому випадку SaaS максимально допрацьовується під потреби користувача. Звичайно сюди входить розробка нестандартних застосунків по побажаннях замовника. Включаючи інтеграції із платформами, які він використовує.

Зустрічаються й інші моделі оплати за SaaS. Наприклад, можна придбати довічну ліцензію, що практично рівносильне викупу ПЗ.

Деяких людей насторожує підхід, при якому вони не є власниками ліцензії на конкретне ПЗ. Але він здатний позбавити клієнта від багатьох головних болів. Наприклад, йому не обов'язково відразу платити за продукт велику суму. Можна оплатити місячне користування сервісом, щоб протестувати, наскільки він підходить для конкретного бізнесу.

1.2 Область застосування

Головний плюс SaaS у тому, що ця модель дозволяє зосередитися на роботі, не відволікаючись на завдання, пов'язані із софтом. У випадку із традиційним ПЗ неполадки доводиться усувати самотужки або запрошувати системного адміністратора. Відновлення необхідно купувати на новому диску, знову встановлювати, вирішувати проблеми з налаштуваннями і т.д.

В SaaS усі інакше. Постачальник відповідає за налаштування сервісу зі своєї сторони. Він же відповідає за схоронність даних, коректну роботу З і випуск відновлень. Техпідтримка сервісу вирішує будь-які виникаючі питання й найчастіше надає користувачам матеріали, що навчають роботі з застосунком.

Переваги SaaS

– У стандартній моделі відносини покупця й постачальника найчастіше закінчуються відразу після покупки ПЗ. А далі, як говориться, звивається, як умієте. Виключення становлять хіба що «гігантів» рівня Windows і Adobe. Коли ж мова йде про SaaS, користувач у будь-який момент може звернутися до постачальника за підтримкою. Усім клієнтам сервісу доступні регулярні відновлення.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

- Для великої компанії це економія на ПЗ. Програму не потрібно здобувати й окремо встановлювати на кожний комп'ютер, щоб у ній могло працювати кілька користувачів. Головне мати доступ до інтернету, а також логін і пароль.

- Деякі сервіси просто неможливо реалізувати на власному сервері компанії. Наприклад, продукти аналогічні SERPstat або Datanyze. Вони не можуть існувати в якості традиційного ПЗ, а тільки як SaaS. Такий застосунок при великому бажанні можна розробити самостійно. Але для цього потрібні значні ресурси.

З іншого боку, для власника SaaS така модель – джерело регулярного доходу й потенціал для росту. Користувачі оплачують сервіс не одноразово, а постійно, і це дозволяє розробити довгострокову стратегію розвитку.

SaaS-модель

Це модель, при якій ви оплачуєте сервіс, як послугу. Наприклад, здобуваєте сервіс email-розсилок на місяць. При SaaS-моделі користувач платить саме за підписку й не може скачати сервіс цілком.

B2B SaaS

Будь-які сервіси, які потрібні бізнесу для роботи й здобуваються по підписці. Наприклад, аналітичні інструменти, сервіси керування контекстною рекламою, колтрекинг, віртуальна АТМ.

SaaS-сервіси

Це сервіси, які оплачуються на певний період, протягом якого ви можете використовувати цей інструмент. Наприклад, ви можете оплатити віртуальну АТМ по передплаті на місяць.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи Software as a Service, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Medidata – розробка хмарних застосунків, які допомагають при розробці сайтів.

Приклади використання SaaS

Розглянемо, як SaaS застосунки для бізнесу допомагали компаніям зі світовими іменем поліпшити свої показники. Кейсами поділилися лідери ринку хмарного програмного забезпечення – Salesforce і Workday.

Кейс Salesforce. Aston Martin

Англійська компанія Aston Martin більш 100 років робить престижні спортивні автомобілі. Більшість моделей збирається вручну, тому за більш ніж століття діяльності було випущено менш 100 тисяч автомобілів. В 2016 році компанія презентувала модель DB11 – перший продукт, який був виготовлений при впровадженні шестирічної дорожньої карти Second Century. Авто позиціонується як комбінація виняткового дизайну, чудової продуктивності й завжди актуальної розкоші.

DB11, сайт Aston Martin

Щоб привернути увагу до нової розробки, Aston Martin було потрібно поліпшити роботу з базами даних. У компанії нагромадилася величезна кількість інформації, і її потрібно було структурувати. Багато даних надходили від дилерів – більш 150 представництв у більш ніж 40 країнах світу. Через відсутність єдиної інформаційної системи з дилерами Aston Martin упускала потенційних клієнтів.

За рік до випуску DB11 компанія впровадила систему Sales Cloud. До неї лише 10 співробітників працювали з даними про клієнтів, в основному аналізуючи їх вік. Для адаптації до нової системи IT-відділ Aston Martin використовував інший інструмент від Salesforce – Trailhead, який містить навчальні уроки.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

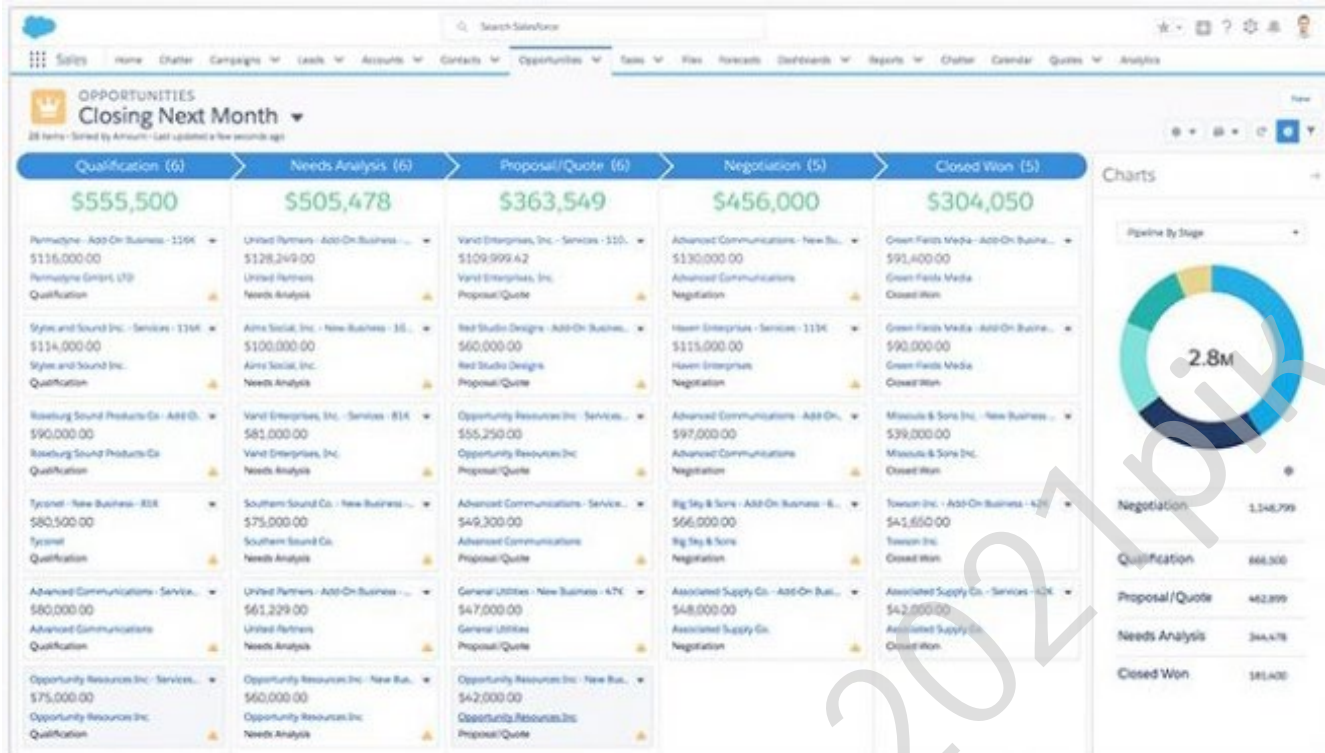


Рисунок 2.1 – Інтерфейс користувача Sales Cloud, сайт Salesforce

Тепер більш 150 співробітників Aston Martin і 500 працівників дилерських центрів використовують Sales Cloud для збору, відстеження й поширення інформації про можливості компанії й переваги споживачів. Також аналізуються дані про клієнтів, які віддали перевагу іншим автовиробникам у преміум-класі.

Sales Cloud була інтегрована в сервіс Q від Aston Martin, Туди надходять сотні пропозицій у рік від споживачів: поміняти фарбування, індивідуально підбудувати технічні характеристики, додати аксесуари. Завдяки ПЗ від Salesforce компанія змогла швидше й ефективніше працювати з такими запитами.

Співробітники й партнери Aston Martin мають постійний доступ до статистичних викладень, використовуючи мобільний застосунок **Salesforcemobile App**. Цей інструмент став важливим для дилерів, які постійно перебувають у дорозі.

Для зручного корпоративного спілкування почав використовуватися сервіс **Chatter** від Salesforce.

Ще один продукт компанії задіється, щоб забезпечити індивідуальний підхід до клієнта. За допомогою **Marketing Cloud** менеджери Aston Martin шукають нові зв'язки зі споживачами. Компанія пропонує нові відносини покупцям її автомобілів, у яких минула гарантія. Вона також використовує модуль **Social Studio** для відстеження відкликань у соцмережах про продукцію, обслуговування й самому бренді.

У майбутньому Aston Martin планує застосувати Sales Cloud, щоб зрозуміти «точки страсті» у клієнтів. Заради довірчих відносин з ними виробник прагне знати про навіть такі переваги, як любов до гарного вина або філантропія. Ще в планах англійців – створити нову платформу для клієнтів. Там вони зможуть ділитися своїми «точками страсті» і створювати віртуальний гараж для автомобілів.

Кейс Workday. Tripadvisor

Заснований в 2000 році, сайт Tripadvisor є зараз найбільшим туристичним сайтом. Його щомісячна аудиторія становить більш 300 млн унікальних користувачів, з яких зареєстрованих – більш 70 млн. Сайт працює на 25 мовах в 45 країнах.

Наприкінці 2011 року Tripadvisor був відділений від материнської компанії – іншого туристичного сайту Expedia, ERP-систему* якого використовував. Система не була віддаленої, вона встановлювалася на сервері і комп'ютери в офісі. Менеджери Tripadvisor були незадоволені швидкістю її роботи: доводилося чекати, щоб одержати аналітику й звіти. При постійно зростаючій відвідуваності сайту така ERP-система стримувала розвиток.

Альтернативу запропонувала компанія Workday. Її система для керування трудовими ресурсами (Human Capital Management) впроваджувалась 6 місяців. На установку системи для керування фінансами (Financial Management) пішло 7 місяців.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11



Рисунок 2.2 – Інтерфейс користувача Workday Financial Management, сайт Workday

Бізнес-застосунки SaaS від Workday принесли такі результати:

- Зниження сукупних витрат на 50%.
- Дані, які стосуються бізнес-процесів і організаційних змін, стали краще й швидше поширюватися серед співробітників. Сама інформація стала точніше.
- Більш широкий вибір аналітичних інструментів.
- Поява мобільних застосунків.
- На закриття звітного періоду йде менше часу – замість тижня воно займає кілька днів.
- Журнал реєстрації подій завжди під рукою.
- Зниження витрат на навчання персоналу, який став активніше утягуватися в бізнес-процеси.
- Менеджмент компанії перестав опікуватися про ліцензії для програмного забезпечення – платформа Workday швидко обновляється.

ERP-система – інтегроване програмне забезпечення для керування трудовими ресурсами, фінансами й активами підприємства. В 2013 році лідерами українського ринку ERP стали SAP – 81%, 1С – 22,3%, «Інформаційні технології» – 15,7 %. Місткість ринку становила більш \$80 млн.

SaaS – послуга, яка надається через веб-інтерфейс. Програмне забезпечення розгортається в дата-центрі й надається на умовах оренди з регулярними платежами. У вартість входить не тільки використання ПЗ, але й витрати провайдера на технічну підтримку, відновлення й захист даних. Так роблять усі сервіси.

Наприклад, Worksection забезпечує керування проектами, дозволяючи повний контроль фінансів і строків. А в тарифах не враховується кількість підключених учасників (як в інших аналогічних програм). Від ціни залежить тільки кількість проектів і обсяг хмарного сховища.

SaaS для малого бізнесу й більших організацій – відмінний спосіб різко скоротити капітальні витрати без втрат для функціональності.

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.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Реалізований заново стилізуємий FMX компонент TМето 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 С++ 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 С++, 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

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

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, SOCl, 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

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Cmake.
- 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.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

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

- 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 стилізуємі FMX компонент TМето на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- 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.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

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 кваліфікаційну бакалаврську роботу, реалізації підлягає програмне забезпечення, яке призначено для системи Software as a Service.

В процесі розробки кваліфікаційної бакалаврської роботи 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;

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

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.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Успішний бізнес любить активних людей. Активні люди не вимагають окремого кабінету в офісі й постійно спілкуються із клієнтами, тому можуть працювати в будь-якому місці – в аеропорті, кафе, бізнес центрі і т.д. До 60% користувачів Інтернет – це вже мобільні користувачі, яким необхідний доступ до різних ресурсів з будь-якої точки. Модель використання застосунків у форматі інтернет сервісів (Software as a Service) стає усе більш популярною й дозволяє працювати віддалено. Ідеологія SaaS полягає в розміщенні серверів з набором необхідного програмного забезпечення в надрах хмарного провайдера.

Переваги SaaS сервісів

Активне використання SaaS відкриває нові обрії для корпоративних користувачів, і бізнес одержує безсумнівні переваги:

- короткі строки впровадження нових сучасних застосунків;
- доступність своєчасних відновлень і нових версій;
- підтримка активних бізнес користувачів у будь-якому місці;
- невисокі вимоги до потужності ПК користувача;
- низька вартість володіння додатком.

Експерти проорокують подальший ріст обсягів ринку SaaS і до 2021 року він повинен вирости ще на 21% у порівнянні з поточним роком, і в абсолютних цифрах досягне обсягу в 106 мільярдів доларів. За даними аналітиків з Gartner, доходи від сервісів у форматі SaaS, будуть стабільно рости, особливо в областях, де бізнес-процеси носять уніфікований і нескладний характер. На думку експертів, кількість впроваджень застосунків у власних центрах обробки даних скоротяться на 16% до 2022 року

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Ринок SaaS активно розвивається, у тому числі й в Україні й основні гравці ринку продовжують рости на 40-60% і не зауважують кризи. Однак властиво публічні хмари в Україні поки так і не «злетіли» – в основному мова йде про проекти приватних хмар у великих компаніях. На жаль, середній і малий бізнес, який в усьому світі є основним замовником SaaS, так і не став сильною стороною російської економіки.

Перебуваючи у відрядженні або спілкуючись із клієнтом на виїзді завжди зручно мати доступ до складу, застосунків, пошти, корпоративної CRM (контакти, відновлення статусів проектів і т.д.) Це дозволить ефективно спілкуватися із клієнтами й вирішувати практично всі питання, а після повернення в готель мати час для відпочинку або підготовки нових зустрічей. У себе в компанії ми активно використовуємо такі сервіси як віртуальний робочий стіл, корпоративна пошта, CRM, корпоративний портал, а також особистих кабінет для всіх наших клієнтів.

Мінуси SaaS-сервісів

Але закон збереження енергії ніхто не змінював і якщо комусь добре, те обов'язково є хтось, хто випробовує дискомфорт від нових інновацій. І це IT-фахівці, які не мають можливостей для контролю якості сервісу, який надає SaaS провайдер. Звичайні корпоративна IT-інфраструктура будується за принципом – серверна (або Центр Обробки Даних) і багато користувачів, які звертаються до консолідованих у ЦОД сервісам. Зрозуміло хто, за що відповідає й хто повинен усувати всі проблеми в корпоративній інфраструктурі, звичайно, це власна IT-служба. При росту мобільних користувачів і більш широкому використанні SaaS звична границя між корпоративної IT-інфраструктурою й надаваним сервісам стирається. Частина ресурсів виявляються за межами IT-інфраструктури компанії й контролю IT-фахівців.

При звертанні в Help Desk користувач не вникає, із чим зв'язана проблема з роботою власного сервісу або каналом зв'язки до SaaS провайдера або серверами в SaaS провайдера. Для нього важлива продуктивність, надійність

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

підключення й доступність сервісів для застосунку його поточного завдання. Тому завдання забезпечення високої продуктивності роботи застосунків і сервісів залишається актуальною й ІТ-фахівці не можуть сказати, що повільна робота або неприступність SaaS застосунків це не їх проблема. Зниження продуктивності сервісу може бути пов'язане з такою більшою кількістю різних факторів, що складно зрозуміти, на чому потрібно сконцентрувати увагу і як застосунати проблему:

- Центр Обробки Даних? Коли користувачі скаржаться на низьку продуктивність застосунки й у клієнта є ЦОД, те всі погляди спрямовуються в нього. Аналізується статистика із серверів, активного мережного встаткування, систем для аналізу продуктивності мережі або застосунків. Проглядаються логи баз даних.

- Віддалені користувачі й офіси? Наступне питання чи всі страждають від низької продуктивності сервісу. Перевіряються налаштування активного мережного встаткування, завантаження й використання (чому забиті) канали зв'язку.

- Провайдер? Перевірка коректності налаштувань, внесення змін у пріоритети трафіку, перевірка пропускнуої здатності каналу зв'язки, затримок, втрат у каналах зв'язку. Перевірка балансувальників навантаження. Перевірка змін у маршрутах по доставці трафіку в ЦОД. Дзвінок провайдеру й одержання відповіді – Усі добре й працює штатно.

- SaaS провайдер? Зв'язатися уточнити їсти чи проблеми, перевірити статистику, але в більшості випадків вона носить загальний характер і рідко вдається одержати відповідь про реальність проблеми на їхній стороні. І, звичайно, що відбувається в SaaS провайдера, так само загадка для власних ІТ-фахівців.

Якщо в ІТ-фахівців відсутні інструменти для контролю провайдерів, то 50% задіяної інфраструктури при наданні сервісу залишаються без уваги й ІТ-фахівці не можуть ефективно й швидко вирішувати проблеми. Статистика, яку

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

надають SaaS провайдери, носить поверхневий характер і не структурована по типах сервісів або серверів, тому не може допомогти в застосунку завдання. Для спілкування з SaaS провайдерами й провайдерами каналів зв'язку ІТ-службам необхідні нові інструменти, які можуть бути частиною загальної системи по аналізі продуктивності застосунків і сервісів і на підставі яких можна вести конструктивний діалог з підтримкою провайдера.

Як контролювати SaaS провайдера?

На даний момент на ринку присутній усього трохи гравців, які пропонують застосунки для контролю SaaS на основі платного сервісу – це Appdynamics, New Relic, Appneta, Fluke Networks, Thousandeye, Exoprise. Користувачі даних сервісів можуть здійснювати контроль послуг, які перебувають за межами їх зони відповідальності. Більшість систем використовують віддалені апаратні й програмні зонди, які дозволяють здійснити наскрізний моніторинг у реальному масштабі часу до хмари й назад. Компанія Appdynamics установила зонди більш ніж в 23 центрах обробки даних SaaS провайдерів по усьому світу. Компанія Fluke Networks установила глобальні зонди в дев'ятох ЦОД у всіх ключових SaaS провайдерів по усьому світу.

На відміну від інших гравців, компанія Fluke Networks пішла ледве далі й крім продуктивності застосунків додала можливості для контролю каналів зв'язку, а також і Voice over IP. Тому ІТ-фахівці мають можливість:

- контролювати end-to-end доступність і продуктивність каналів зв'язку в реальному часі;
- одержувати повідомлення й установлювати граничні значення, щоб проактивно одержувати повідомлення про зниження якості хмарних сервісів;
- швидко локалізувати проблему й підключати технічні служби провайдерів для усунення проблем на їхній стороні;
- надавати достовірну статистику й звіти про продуктивність SaaS і Telecom провайдерів.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Традиційні системи моніторингу продуктивності мережі й застосунків будувалися для використання усередині корпоративної мережі, але бізнес платить за сервіси як послуга й повинен мати можливість контролювати якість цих сервісів.

Ні для кого не секрет, що багато провайдерів, намагаючись скоротити витрати, перейшли на аутсорсинг експлуатації каналів зв'язку і якість істотно погіршилося. SaaS провайдер може перебуває за океаном і його контроль це окреме завдання, тому незалежно від того, чим користується компанія, вона повинна мати засобу для контролю над дотриманням SLA по різних типах сервісу – канали між віддаленими офісами, мережа до хмари, Voice over IP (VoIP), SaaS застосунки.

На ринку з'являються нові застосунки, які відповідають сучасним потребам IT-фахівцям і дозволяють поставити під контроль продуктивність сервісів, які перебувають за межами звичайної корпоративної мережі, але використовуються для роботи співробітниками компанії.

3.2 Розробка структурної схеми

SaaS – це сучасна модель взаємодії клієнта й постачальника софту. Ви вже, швидше за все, бачили це поняття, причому повсюдно. Приміром, нерідко можна бачити хмарні сервіси, а в статтях, де приводиться топ бізнес сервісів, звичайно фігурують подібні пропозиції.

Чому бізнес не проти SaaS?

Переваг досить багато. По-перше, важливо відзначити, що реалізація SaaS на практиці є простій. Щоб впровадити хмарну систему, вистачає пари годин. Клієнт може надалі використовувати повністю робочий сервіс із аспектом на автоматизації всіх рутинних завдань. Причому сервіс легко впише в інфраструктуру компанії, тому що він уже заточений під широку безліч

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		25

конфігурацій. Щоб працювати із сервісом, часом вистачає й смартфона. Звичайно, якщо є мобільна частина й мобільний інтерфейс.

По-друге, це вигідне вкладення засобів. Розумна економія на ІТ-інфраструктурі – ще один фактор, чому багато клієнтів вибирають саме SaaS. Хмарні сервіси допомагають скоротити штат обслуговуючого персоналу, менеджерів і т.д. Це важливо, зокрема, для малого бізнесу. Там завжди потрібно підвищувати ефективність і конкурентоспроможність. Постачальник SaaS-застосунків буде виконувати все, що пов'язане з контролем, керуванням, а також бекапом.

По-третє, це гнучкі застосунки. Їх можна модернізувати й поліпшувати під себе. Можна все зробити під клієнта і т.д. Причому особливих витрат це не викличе.

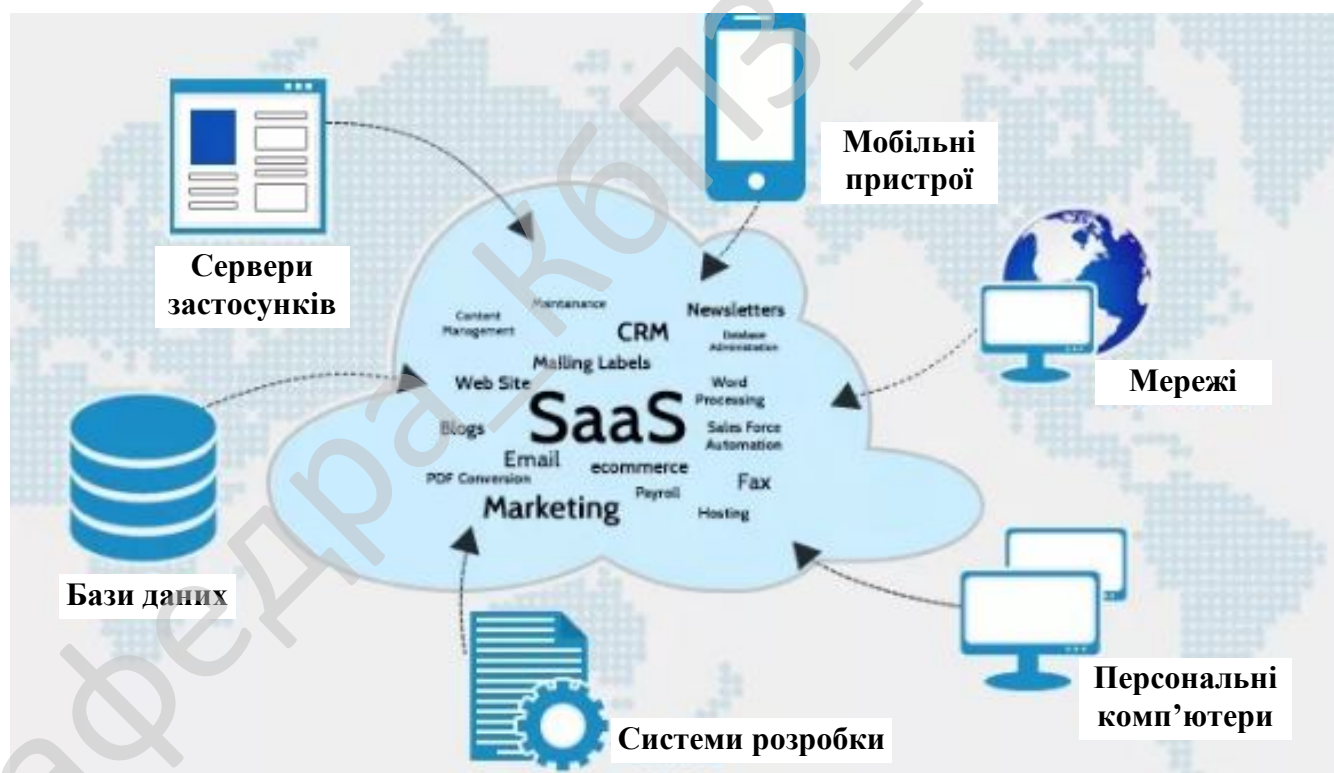


Рисунок 3.1 – Структурна схема системи

Знаходите й усувайте проблеми із продуктивністю за допомогою однією простій у використанні панелі інструментів

У вашому постійно підключеному робочому місці величезний обсяг даних про продуктивність вимагає більшого, ніж розрізнений серверний моніторинг. І як ви кількісно оціните те, що дійсно важливо – досвід кінцевого користувача? Analytics for Performance, заснований на машинному навчанні, надає в реальному часі корисну аналітичну інформацію з індивідуальними оцінками взаємодії з користувачем, щоб ви завжди могли забезпечити відмінна взаємодія зі співробітниками в кожному застосунку.

Максимізуйте гнучкість ІТ-інфраструктури за допомогою аналітичних даних про продуктивність і інструкцій

Проактивно реагуйте на проблеми користувачів і потреби в ємності за допомогою наскрізної видимості застосунків, мереж і інфраструктури.

Надайте мобільним працівникам можливість підключатися до даних з будь-якого місця

Мобільна робота стала мейнстримом. Шістдесят вісім відсотків співробітників вважають, що для технологій на р обочому місці важливо, щоб вони могли працювати з будь-якого місця, змушуючи співробітників надавати доступ до файлів з будь-якого місця на пристроях, які вони вибирають. У той же час ви диверсифікували своє файлове сховище в локальних, хмарних і гібридних місцях. Крім того, у вашому центрі обробки даних є застарілі файлові сервери, не призначені для епохи мобільних пристроїв. Це створює природню напруженість між співробітниками, яким потрібний усюди доступ до файлів, і ІТ-командами, яким потрібна гнучкість сховища.

Побільшайте гнучкість вашого сховища

Ви розробили існуючу стратегію зберігання не випадково – щоб максимально використовувати свій бюджет і ресурси. Ваш застосунок для спільної роботи з контентом повинне бути гнучким, дозволяючи користувачам одержувати доступ до зашифрованих файлів незалежно від того, чи зберігаються

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

порушення. Analytics використовує машинне навчання для виявлення ризикованого поведінки кінцевих користувачів і вживання попереджуючих заходів для захисту вашої організації. Завдяки цьому, що настроюється заснованому на правилах движку для Content Collaboration IT-відділ одержує чітку картину поведінки користувачів, не виділяючи великих ресурсів.

Захистите конфіденційні дані

Оскільки співробітники обмінюються файлами для повсякденної роботи, легко обмінюватися конфіденційною інформацією, навіть не усвідомлюючи цього. Це робить важливим керування спільним доступом до документів на основі вмісту цих файлів. Content Collaboration пропонує інтеграцію із провідними системами запобігання втрати даних, такими як продукти Symantec, McAfee і Forcepoint, серед інших. Це дозволяє впорядковувати файли на основі їх вмісту й забезпечувати безпечний обмін вмістом для запобігання витоку конфіденційної інформації.

Запобігти несанкціонованому обміну

Коли співробітники діляться конфіденційним документом, ви прагнете бути впевнені, що його прочитає тільки передбачуваний одержувач. На жаль, неавторизовані читачі можуть легко перенаправляти файли або зробити знімок екрана. Керування правами на доступ до інформації (IRM) Content Collaboration додає рівень безпеки до спільного використання документів, дозволяючи власникові файлу додавати водяний знак з іменем одержувача, електронною поштою й IP-адресою для запобігання знімків екрана. Відправники також можуть заблокувати завантаження файлів одержувачами, щоб вони не могли локально зберігати копії конфіденційних документів на своїх пристроях.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Зменште поверхню атаки

Оскільки віддалена робота продовжує рости, росте й потреба в доступі до мережі з нульовою довірою. Троє з 10 співробітників будуть продовжувати працювати з будинку навіть у періоди стабільності, і їм буде потрібно безпечний доступ до ресурсів, включаючи SaaS і веб-застосунки, з нових місць і з різних пристроїв. Компанії, яким не вдається швидко замінити традиційні VPN посередині цих сучасних складностей, зустрічаються із серйозними ризиками безпеки. Через надмірне спрощення автентифікації й сліпої довіри до користувачів після надання доступу до мережі ці застарілі застосунки будуть як і раніше ставити ваші корпоративні центри обробки даних і відповідні мережі в уразливе положення. На щастя, є швидка й гнучка альтернатива. Secure Workspace Access безпечно доставляє застосунки й дані вашим користувачам і забезпечує доступ на рівні застосунків. У будь-якому місці, на будь-якому пристрої, без необхідності використання застосунку VPN.

Включите ВУО з відповідними заходами безпеки

Традиційні VPN вимагають постійного керування пристроями, що непрактично в сьогоденних нових умовах роботи. Співробітники, що працюють віддалено, прагнуть вибрати пристрій, який вони віддають перевагу, щоб підтримувати свою продуктивність. З іншого боку, ІТ-відділи прагнуть гарантувати, що корпоративні дані не будуть скомпрометовані. Secure Workspace Access пропонує елементи керування безпекою, які дозволяють співробітникам використовувати некеровані пристрої й пристрою ВУО для безпечного доступу до застосунків і даним, зберігаючи контроль над ІТ і продуктивність кінцевих користувачів.

Поліпшення користувацького досвіду

Коли справа доходить до досвіду співробітників, традиційні VPN не справляються. Ці застосунки не можуть оптимізувати застосунки або зворотний трафік застосунків до центру обробки даних – зайво забиваючи базову мережу й знижуючи продуктивність. А оскільки вони направляють як діловий, так і

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

особистий трафік через корпоративні ІТ, VPN також можуть заважати конфіденційності співробітників. Кращим варіантом є альтернатива VPN, яка забезпечує безперешкодний доступ з єдиною реєстрацією (SSO) для однаковості роботи в різних операційних системах. Workspace – найпростішому й безпечний цифровий робочий простір на основі хмари, доступне сьогодні, забезпечує чудова взаємодія зі співробітниками, не піддаючи вашу організацію непотрібному ризику й не створюючи проблем з конфіденційністю.

Вийдіть за рамки єдиного входу (SSO) з розширеними функціями безпеки для SaaS і веб-застосунків

Коли співробітники перемикаються між 35 застосунками більш 1100 раз у день, не дивно, що вони повторно використовують той самий пароль – кошмарний сценарій для ІТ. За допомогою єдиного застосунку керування доступом для всього цифрового робочого простору ви можете підвищити безпеку й зручність для користувачів і ІТ-фахівців. Зменште ризик, одержите більший контроль над доступом і поведінкою користувачів і одержите більш глибоку подання про весь ландшафт застосунків.

Доступ до всього одним клацанням миші

Співробітники покладаються на різні застосунки для виконання роботи, але керування доступом до цих застосунків створює проблеми як для кінцевих користувачів, так і для ІТ-фахівців. Workspace з Secure Workspace Access поліпшує взаємодія з користувачем, забезпечуючи єдиний вхід для кожного застосунку й файлу, незалежно від того, де людям потрібно працювати. Безпека підвищена за рахунок меншої кількості паролів, якими можуть управляти користувачі, і використання хакерами. ІТ-відділ має інтеграцію з усіма основними репозиторіями ідентифікаційної інформації, механізмами багатофакторної автентифікації й протоколами єдиного входу.

SaaS і веб-застосунки, захищені

Оскільки 84% організацій заявляють, що традиційні застосунки безпеки не працюють у хмарних середовищах, уже недостатньо просто управляти доступом

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		36

чим коли-або, завдяки тому, що кінцеві користувачі покладаються на особисті пристрої для роботи й одержують доступ до хмарних застосунків і корпоративним ресурсам з віддалених місць. Як ви можете забезпечити доступ, необхідний вашим співробітникам для найкращої роботи, захищаючи вашу організацію від внутрішніх атак?

Сьогоднішній підхід до безпеки повинен перейти від безумовної довіри до користувачів до принципів нульової довіри. Модель нульової довіри покладається на контекстну поінформованість для адаптивного надання доступу авторизованим користувачам з використанням шаблонів, заснованих на ідентичності, часі й стані пристрою. Це посилює контроль над безпекою доступу, надаючи користувачам можливість вибору пристроїв і застосунків.

Зменште поверхню атаки

Занадто часто ІТ-фахівці зациклюються на нескінченному циклі додавання точкових продуктів, таких як SSL VPN, єдиний вхід і керування кінцевими точками, для застосунку нових завдань безпеки. Це не тільки збільшує складність, але й створює роздробленість для кінцевих користувачів. Forrester повідомляє, що організації, що прагнуть до нульової довіри, можуть побачити значні вигоди від вибору одного постачальника. Workspace пропонує комплексний застосунок для реалізації принципів нульової довіри, уникаючи пробілів, що залишилися від використання різних точкових застосунків. Замість того, щоб намагатися захистити дані й ресурси навколо користувача, Workspace зменшує поверхню атаки, захищаючи користувача й застосунки усередині робочого простору, де фактично виконується робота.

Включити дистанційний доступ без VPN

Віддаленим користувачам потрібний легкий доступ до веб-застосунків, але використання традиційних застосунків VPN може створити значні ризики для безпеки. Якщо зловмисники можуть одержати доступ через SSL VPN, вони можуть вільно управляти всією вашою мережею. Вам необхідно контекстний застосунок безпеки, щоб відповідати вимогам нульової довіри при включенні

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		38

всіх філій – більш простий і дешевий застосунок, чому установка декількох міжмережевих екранів у вашій організації.

Попрощайтеся із затримками VDI

Одна із самих більших проблем у технології віртуалізації? Забезпечення якісного користувацького досвіду. Люди очікують відповіді від віртуальних застосунків і настільних комп'ютерів у реальному часі, незалежно від якості мережі й місця розташування. Однак занадто часте розгортання інфраструктури віртуальних робочих столів означає надання користувачам не ідеального досвіду. Недостатньо просто пропонувати віртуальні застосунки й робочі столи. Вам необхідно надати правильний застосунок – таке, яке може адаптуватися до широкого спектра змінних, щоб забезпечити найкраща взаємодія з користувачем у будь-який час. Саме для цього й призначена технологія високої чіткості (HDX), що лежить в основі Virtual Apps and Desktops.

Побільшати швидкість

Для ваших користувачів, щоб максимально використовувати можливості робочого стола VDI, ключовим моментом є швидкість реагування. Повільне з'єднання вже турбує 60% співробітників. Якщо віртуальний робочий стіл страждає від затримок, це розчарування стане ще більше. З погляду користувачів, немає причин, по яких складна тривимірна графіка повинна працювати на віртуальному робочому столі менш плавно, чому у власному середовищі. З HDX 3D Pro і іншими спеціалізованими технологіями цього не відбудеться. Новітні технології відображення забезпечують кристально чисту передачу голосу й відео, дозволяючи користувачам спільно працювати із зображеннями з точністю до пікселя навіть при наявності втрати пакетів, перевантаження, більших затримок і джиттера, які звичайно виникають у широкосмугових бездротових і мобільних мережах. Додавання SD-WAN забезпечує якість обслуговування HDX для Virtual Apps and Desktops за рахунок оптимізації з'єднань від філій, домашніх користувачів і хмар. Завдяки можливостям дистанційного доступу до ПК

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

користувачі також можуть використовувати графіку або ресурсомісткі робочі навантаження зі своїх фізичних ПК.

Скоротити витрати

Оскільки 44% ІТ-бюджетів уже ростуть на заміну застарілих застосунків, це допомагає мати передові технології, що не вимагають додаткових вкладень. Технологія HDX вбудована в кожен редакцію Virtual Apps and Desktops, що дає вам миттєва перевага. Тільки надає можливості застосунки, які дійсно відтворюють можливості традиційних Linux і Windows 10, дозволяючи максимально використовувати наявні у вас ресурси. Ви не тільки забезпечите кращу якість зображення для робочих навантажень із високим ступенем анімації й відео, оброблюваного на сервері, але й зробите це без необхідності в інвестиціях в інфраструктуру. А завдяки вбудованому перенапрямку USB для HDX-друку, сканування, малювання планшетів і інших периферійних пристроїв користувачі можуть одержати доступ до всіх тим же пристроям, які в них були б із традиційним настільним комп'ютером – у межах безпеки високопродуктивного середовища VDI.

Поліпшити спілкування

Усе більше й більше підприємств прагнуть скоротити капітальні витрати, звертаючись до інструментів об'єднаних комунікацій (UC), таким як Microsoft Teams. Але хоча ці застосунки пропонують величезні переваги для спільної роботи, може бути важко забезпечити ідеальний користувацький інтерфейс у віртуалізованих середовищах. Virtual Apps and Desktops усуває цю проблему. Завдяки оптимізації, створеної в тісному співробітництві з інженерами Microsoft, користувачі одержують повністю природній досвід роботи з Microsoft Teams або Skype для бізнесу. Завдяки підвищеній масштабованості сервера й більш швидкому шляху передачі аудіо-відео трафіку технологія HDX є єдиним схваленим Microsoft застосунком для надання застосунків уніфікованих комунікацій у віртуальному середовищі. Ці оптимізації підтримуються як в Office

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

365, так і в локальному середовищі, при цьому конфіденційні дані чата зберігаються в безпеці в центрі обробки даних.

Забезпечте кращий досвід роботи з і Microsoft

Є причина, по якій Microsoft займає величезну частку корпоративного ринку. Від інноваційних інструментів для спільної роботи до 1 мільярда доларів, що щорічно витрачаються на дослідження й розробки в області кібербезпеки, Microsoft продовжує підвищувати продуктивність праці в усьому світі. Проте, для багатьох організацій проблеми залишаються. Як забезпечити сумісність критично важливих бізнес-застосунків з Windows 10? Як ви можете захистити й управляти Office 365 на будь-якому пристрої разом з іншими необхідними бізнес-застосунками? Більш 30 років і Microsoft переглядають те, як підприємства забезпечують чудовий досвід співробітників. Незалежно від того, чи боретеся ви з міграцією Windows 10 або готові скористатися перевагами віртуального робочого стола Windows в Azure, у вас буде повний спектр спільних застосунків для інтеграції, доступу й зручності використання.

Перехід на Windows 10

Повна підтримка Windows 7 може бути офіційно припинена, але перехід на Windows 10 залишається для багатьох перешкодою. Хоча приблизно 80% ПК перейшли на поточну платформу Microsoft Desktop, мільйони інших як і раніше працюють на більш старих версіях. Чим довше ви чекаєте, тим більше ви будете піддані потенційним уразливостям безпеки. Знову ж, перехід далеко не простий. Як ви забезпечите сумісність важливих застосунків з Windows 10? Що повинне відбутися, щоб запобігти втраті даних? Як забезпечити однаковість взаємодії з користувачем на Mac, Chromebook і інших пристроях? З Virtual Apps and Desktops процес виконується швидко й безболісно. Просто створіть образ робочого стола Windows 10, розгорніть, а потім створіть оновлені віртуальні робочі столи. і сумісні застосунки доступні тисячам користувачів. У міру вступу нових випусків і виправлень ви можете впровадити їх один раз перед розгортанням на всіх пристроях за лічені хвилини.

						КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			42

хмарних послуг, і з більш ніж 3500 спеціалізованими експертами по безпеці пропонує великі переваги для підприємств, готових перейти в хмару. Додавання означає ще більша зручність для ваших користувачів і масштабованість на рівні сервера для ІТ. Завдяки тісному співробітництву з інженерами Microsoft Virtual Apps and Desktops являє собою потужний застосунок, що забезпечує ІТ-фахівцям повне гібридне керування локальними робочими навантаженнями й робочими навантаженнями, розміщеними в Azure. Це найшвидший шлях до повного використання можливостей віртуального робочого стола Windows і багатосеансових можливостей Windows 10. Й все це при максимальному збільшенні інвестицій в існуючі центри обробки даних і русі зі швидкістю, яка найкраще підходить для вашого бізнесу.

Підтримуйте працездатність і продуктивність вашої компанії в часи збоїв

Збої можуть трапитися в будь-який момент – і миттєво змінити ваш бізнес. Багато організацій засвоїли цей урок на власному гіркому досвіді на початку 2020 року, коли мільйонам уперше працюючих віддалених співробітників раптово потрібен був безпечний доступ до корпоративних ресурсів, щоб вони могли працювати з будинку в розпал глобальної пандемії. І не на кілька тижнів, а на багато місяців підряд. Хоча неможливо передбачити, де, коли і як на ваш бізнес можуть вплинути аналогічні події в майбутньому, можна підготуватися. От отут-те й додадуться віртуальні застосунки й робочі столи.

Будьте поперед попиту

Будь те по необхідності або відповідно до особистих переваг, вашим користувачам буде багато раз, коли вашим користувачам буде потрібно працювати віддалено. Коли вони це зроблять, можливість вирішувати завдання без перерв і затримок дозволить вашої компанії працювати. Virtual Apps and Desktops дозволяє співробітникам працювати з максимальною продуктивністю, не створюючи додаткового навантаження на ІТ. Це провідне в галузі застосунок відіграє важливу роль у планах забезпечення безперервності бізнесу багатьох

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

розпорядженні власне встаткування, а ті, хто розташовує, зустрічаються з необхідністю самостійного обслуговування й підтримки, що складно й проблематично. Простіше купити хмарну послугу, одержавши одночасно готовий застосунок і правильний настроєні потужності.

Щоб документообіг у хмарі справлявся з безліччю проблем, необхідно застосувати питання, пов'язаний із захистом інформації. На практиці це простіше зробити, довірившись надійному постачальникові хмарних послуг.

Послуга «SaaS_EDMS в оренду» або «SaaS_EDMS у хмарі» – закінчений застосунок, здатне підвищити ефективність роботи підприємства, що обумовлюється збільшенням продуктивності співробітників і істотною економією засобів

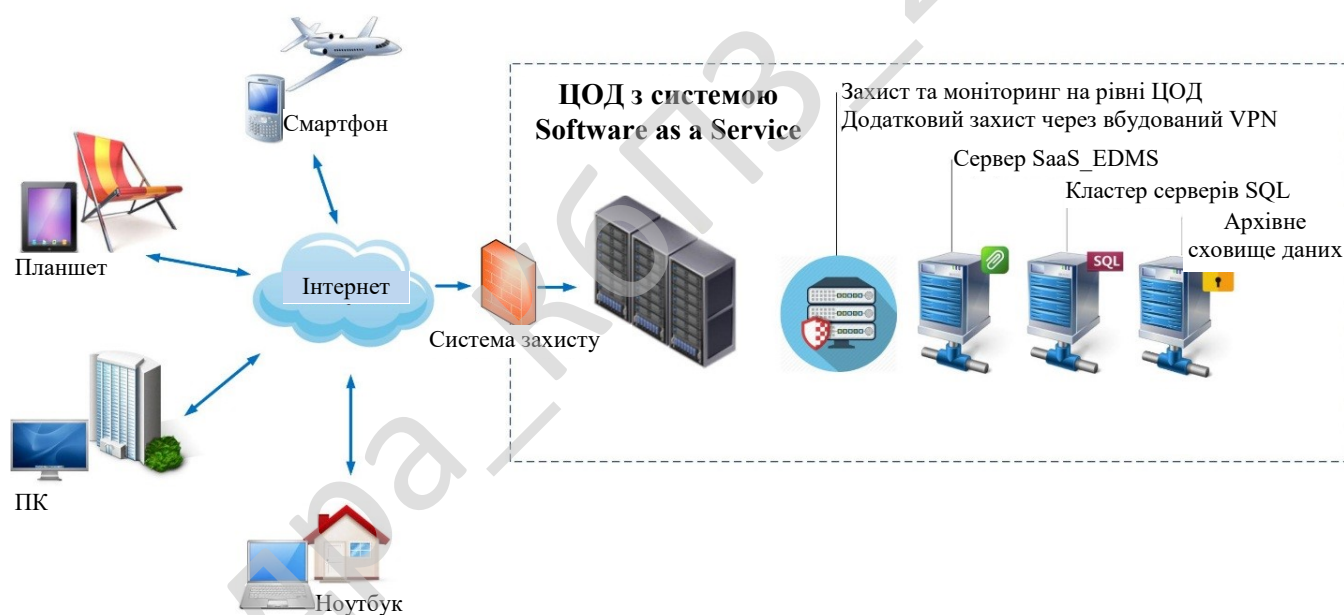


Рисунок 3.2 – Функціональна схема системи

Що одержує клієнт, використовуючи «SaaS_EDMS у хмарі»? У першу чергу – готовий застосунок безлічі завдань по роботі з документами, бізнес-процесами й дорученнями. Потужні інструмент, такі як довідники, конструктори й дизайнер бізнес-процесів, уже вбудовані в платформу й дозволяють без

Вим.	Арк.	№ докум.	Підпис	Дата

КБР-123.21.0047.00.00.ПЗ

Арк.

47

дозволяючи автоматизувати бізнес-процеси, необхідні для ефективної роботи будь-якої компанії. Застосунок по розміщенню SaaS-сервісу на платформі спеціалізованого IaaS-провайдера демонструє зрілість постачальника хмарних сервісів з погляду спеціалізації на якості програмної реалізації й гарантує надійну реалізацію інфраструктурного рівня. Використовуючи універсальний хмарний сервіс, якому немає аналогів, організації зможуть знизити витрати на володіння системами й з легкістю впоратися із завданнями електронного документообігу.

3.4 Розробка діаграми процесів

Відповідно до методичних рекомендацій розроблення графічної частини кваліфікаційної бакалаврської роботи розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Розроблена діаграма взаємодії процесів використовується для представлення та візуалізації процесів обробки даних тобто структурного проектування бакалаврської роботи. Основні складові елементи діаграми взаємодії процесів це потоки даних:

- Репозиторії, потік сховища даних.
- Потоки зовнішні по відношенню до системи сутності.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Потоки даних гібридні між елементами трьох попередніх типів.

Відповідно до документації основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей. Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

Рисунок 3.3 – Діаграма взаємодії процесів

Для схематичного представлення системи що розробляється необхідно спочатку представити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи в цілому у подальшому. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі. Розроблена діаграма взаємодії процесів системи в подальшому уточнюється шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Таким чином у результаті після розгляду, вищеописаної системи, схеми структурної, функціональної, діаграми взаємодії процесів перейдемо до опису та розгляду блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

Розглянемо послідовність дій та викликів підпрограм в загальному алгоритмі роботи основної програми що зображено на рисунку 4.1. у вигляді блок-схеми:

- Виведення вікна системи Software as a Service.
- Запит сервісу.
- Підпрограма підключення до хмари сервісу.
- Підпрограма формування контейнеру сервісу.
- Запуск та формування звіту роботи.
- Запит відображення графа аудиту.
- Сканування стану хмари.
- Підпрограма аналізу інфраструктури системи.
- Виведення отриманих результатів.
- Запит формування звіту.
- Отримання даних з блоку моделювання.
- Отримання даних з блоку кластерного аналізу.
- Формування графічного представлення розвитку системи.
- Підпрограма формування звіту.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

– Системне повідомлення завершення роботи ПЗ – кінець циклу роботи ПЗ.

На рисунку 4.2 зображено роботу підпрограми з реалізацією наступних дій:

– Отримання поточного списку індикаторів.

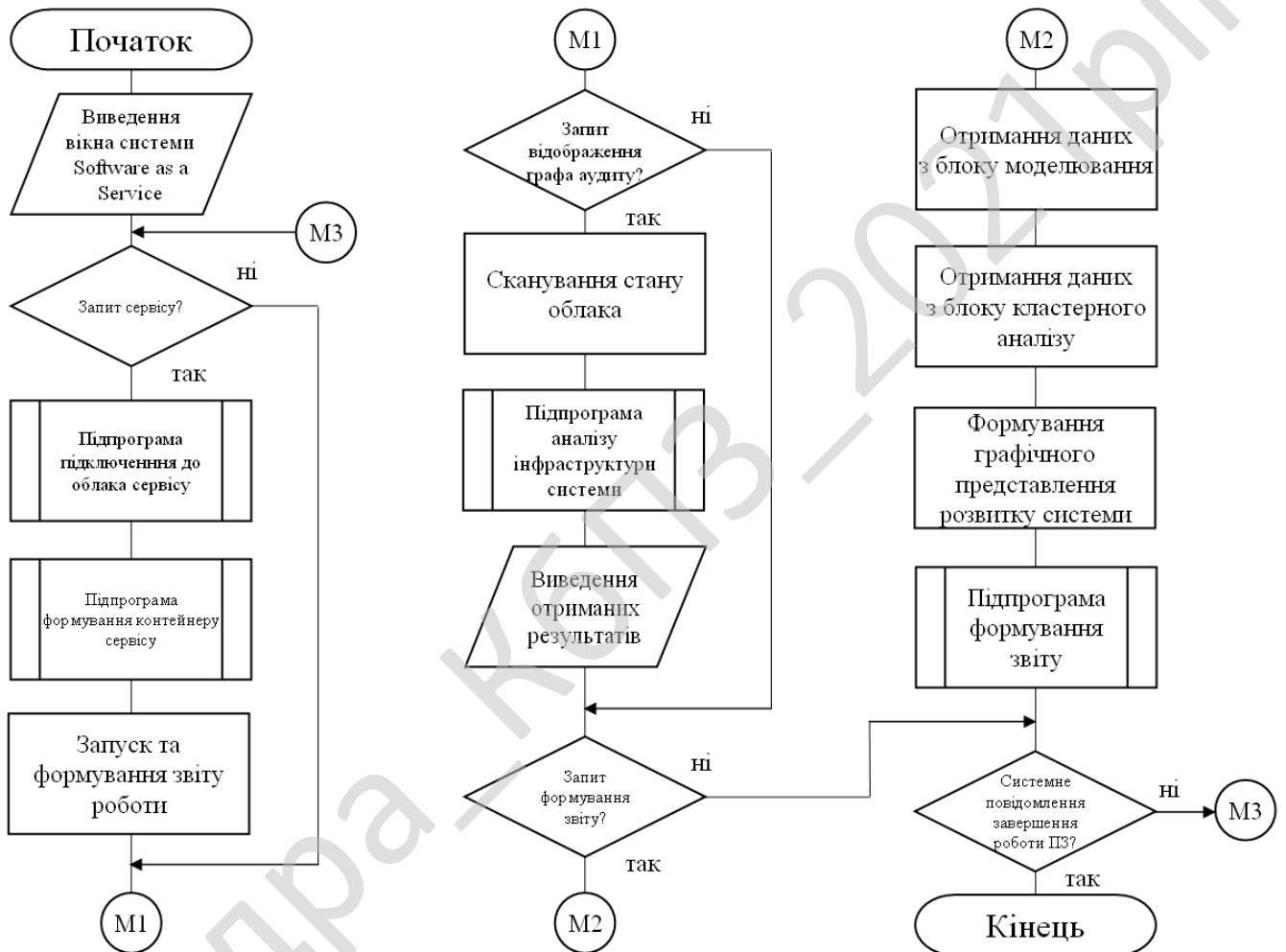


Рисунок 4.1 – Блок-схема основної програми

- Отримання поточних станів графа розвитку системи.
- Отримання поточних траєкторій графа розвитку системи.
- Запит детального звіту.
- Формування звіту з всіма наявними станами системи.

Розглянемо модуль системи uThreads (Software as a Service). Який реалізує


```

Public
// конструктор
    constructor Create(CreateSuspended : Boolean; P : Pointer);
private
    Data : TTask;
protected
    procedure Execute; override;
end;

// Клас TGetFileFtp

```

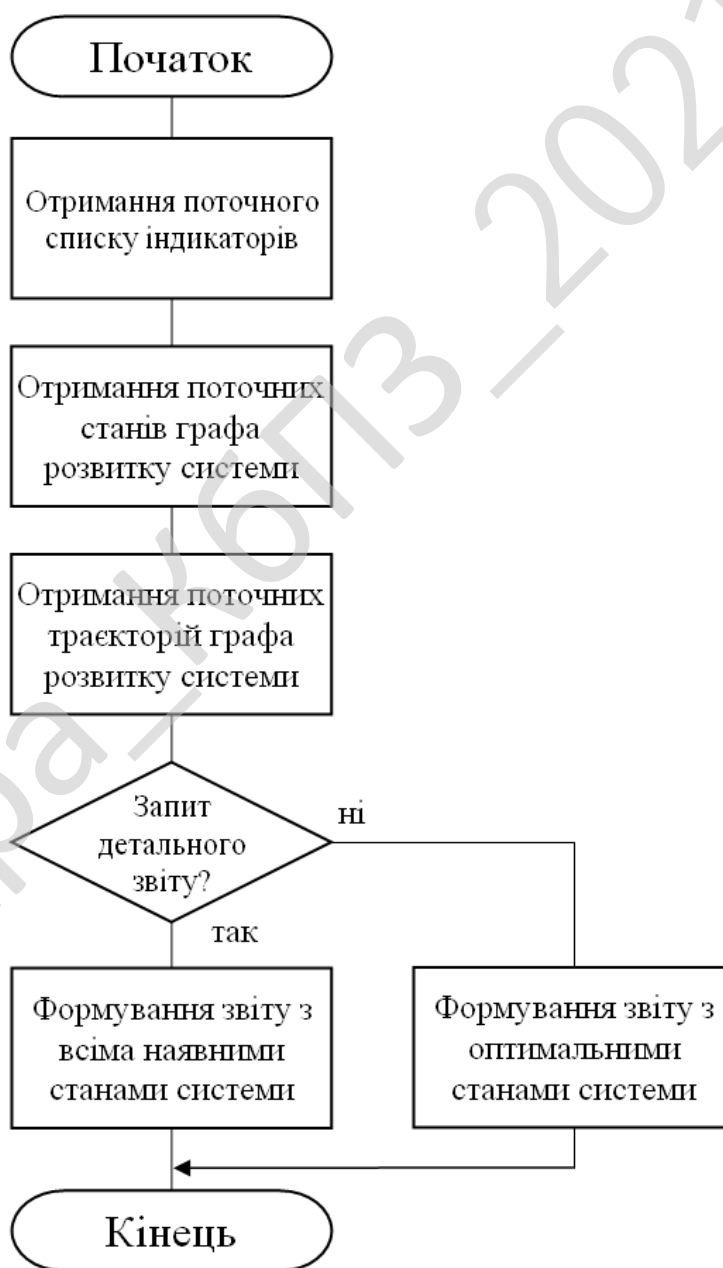


Рисунок 4.2 – Блок-схема роботи підпрограми

```

TGetFileFtp = class(TThread)
    public
        Item      : Integer;
        Reload    : Boolean;
        constructor Create(CreateSuspended : Boolean; P : Pointer);
    private
// приватні об'яви класу
        Data : TTask;
        Tick : Integer;
        FTP  : TIdFTP;
        procedure OnWork(Sender : TObject; AWorkMode : TWorkMode; const
            AWorkCount : Integer);
        procedure OnWorkBegin(Sender : TObject; AWorkMode : TWorkMode; const
            AWorkCountMax : Integer);
        procedure OnWorkEnd(Sender: TObject; AWorkMode: TWorkMode);
    protected
        procedure Execute; override;
end;

// Клас TGetOptionsFtp

TGetOptionsFtp = class(TThread) // робота з ftp
    public
        constructor Create(CreateSuspended : Boolean; P : Pointer);
    private
        Data : TTask;
    protected
        procedure Execute; override;
end;

implementation
// Реалізація вищевказаного

uses uMain, uProcedures; // Модулі

// Клас 'TGetFileHttp'

procedure TGetFileHttp.Execute; // потокові обробки
var
    FileStream : TFileStream;
begin
    HTTP := TIdHTTP.Create(nil);

```

```

if FileExists(Data.Directory + '\' + Data.FileName) then
begin
FileStream := TFileStream.Create(Data.Directory + '\' + Data.FileName,
fmOpenReadWrite);
FileStream.Position := FileStream.Size;
end else
begin
FileStream := TFileStream.Create(Data.Directory + '\' + Data.FileName,
fmCreate);

end;
HTTP.OnWork := OnWork;
HTTP.OnWorkBegin := OnWorkBegin;
HTTP.OnWorkEnd := OnWorkEnd;
if Options.HTTPVersion = hvHttp10 then HTTP.ProtocolVersion := pv1_0 else
HTTP.ProtocolVersion := pv1_1;
if (Options.HTTPProxy.UseProxy) and (LocalAddress(Data.LinkToFile) =
False) then
begin
// початок роботи з проксі сервером у HTTPProxy форматі
HTTP.ProxyParams.ProxyServer := Options.HTTPProxy.Host;
HTTP.ProxyParams.ProxyPort := Options.HTTPProxy.Port;
HTTP.ProxyParams.ProxyUsername := Options.HTTPProxy.UserName;
HTTP.ProxyParams.ProxyPassword := Options.HTTPProxy.Password;
end;
// аутентифікаційні дані
if Data.UseSpecial then
begin
HTTP.Port := Data.Port;
HTTP.Request.Username := Data.Login; // логін входу
HTTP.Request.Password := Data.Password; // пароль
end;
HTTP.Request.ContentRangeStart := Data.LoadSize;
HTTP.Request.ContentRangeEnd := Data.TotalSize;
HTTP.HandleRedirects := Options.Redirect;
StartSize := Data.LoadSize;
try
HTTP.Get(Data.LinkToFile, FileStream);
except
on E : Exception do
begin
Data.Status := tsError;
Data.ErrorText := E.Message;

```

Вим.	Арк.	№ докум.	Підпис	Дата

КБР-123.21.0047.00.00.ПЗ

Арк.

56


```

        Data.Status := tsError;
        Data.ErrorText := E.Message;
        Data.TotalSize := 0;
        Data.LastModified := 0;
    end;
end;
HTTP.Free;
fMain.RefreshTasks;
end;
// Клас 'TGetFileFtp'
// конструктор класу
constructor TGetFileFtp.Create(CreateSuspended : Boolean; P : Pointer);
begin
    Data := P;
    inherited Create(CreateSuspended);
end;

procedure TGetFileFtp.Execute; // перевизначення
begin
    FTP := TIdFTP.Create(nil);
    FTP.OnWork := OnWork;
    FTP.OnWorkBegin := OnWorkBegin;
    FTP.OnWorkEnd := OnWorkEnd;
    if (Options.FTPProxy.UseProxy) and (LocalAddress(Data.LinkToFile) =
False) then
    begin
        FTP.ProxySettings.Host := Options.FTPProxy.Host;
        FTP.ProxySettings.Port := Options.FTPProxy.Port;
        FTP.ProxySettings.Username := Options.FTPProxy.UserName;
        FTP.ProxySettings.Password := Options.FTPProxy.Password;
    end;
    if Data.UseSpecial then
    begin
        FTP.Port := Data.Port;
        FTP.Username := Data.Login;
        FTP.Password := Data.Password;
    end else FTP.Username := 'anonymous';
    try
        FTP.TransferType := ftBinary;
        FTP.Host := ExtractAddress(Data.LinkToFile);
        FTP.Connect;
        FTP.Get(ExtractFileName(Data.LinkToFile),

```

Вим.	Арк.	№ докум.	Підпис	Дата

КБР-123.21.0047.00.00.ПЗ

Арк.

58

```

        Data.Directory + '\' + Data.FileName, False, True);
    FTP.Disconnect;
    Data.Status := tsReady;
    Data.ErrorText := 'Ошибок нет';
end;
FTP.Free;
end;

// TGetOptionsFtp

procedure TGetOptionsFtp.Execute; // перевизначення
var
    FTP : TIdFTP;
    Details : TStringList;
begin
    FTP := TIdFTP.Create(nil);
    Details := TStringList.Create;
    if (Options.FTPProxy.UseProxy) and (LocalAddress(Data.LinkToFile) =
False) then
        begin
            FTP.ProxySettings.Host := Options.FTPProxy.Host;
            FTP.ProxySettings.Port := Options.FTPProxy.Port;
            FTP.ProxySettings.Username := Options.FTPProxy.UserName;
            FTP.ProxySettings.Password := Options.FTPProxy.Password;
        end;
    if Data.UseSpecial then
        begin
            FTP.Port := Data.Port;
            FTP.Username := Data.Login;
            FTP.Password := Data.Password;
        end else FTP.Username := 'anonymous';
    try
        FTP.TransferType := ftBinary;
        FTP.Host := ExtractAddress(Data.LinkToFile);
        FTP.Connect;
        FTP.List(Details, ExtractFileName(Data.LinkToFile), True);
        Data.TotalSize := FTP.DirectoryListing.Items[0].Size;
        Data.LastModified := FTP.DirectoryListing.Items[0].ModifiedDate;
        FTP.Disconnect;
        Data.Status := tsReady;
        Data.ErrorText := 'Ошибок нет';
    end;
end;

```

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

11. Доступний як єдиний файл сирцевого коду на ANSI C, який можна легко вставити в інший проект.

12. Автономність: немає зовнішніх залежностей.

Інструменти створення та обслуговування БД

Створення та обслуговування БД можуть здійснюватись через текстову консоль SQL-командами або через спеціальні інструменти, у тому числі – з графічним інтерфейсом користувача.

Технології, що підтримують SQLite та мови програмування

Сама бібліотека SQLite написана мовою C. Проте є реалізація бібліотеки на JavaScript sql.js, яка дозволяє обробляти файли БД безпосередньо в браузері.

Для інших мов програмування розроблено механізм підключення й роботи з БД через цю бібліотеку: C++, Java, Python, Perl, PHP, Ruby, Haskell, Scheme, Smalltalk, Lua тощо. Засоби для роботи з Tcl включені в комплект постачання SQLite. Повний список наявних засобів можна знайти на сторінці проекту.

Хоча я реалізовував програму сам, було використано підходи Scrum для саморозвитку та пришвидшенню розробки, розглянемо цей метод. Scrum – підхід управління проектами для гнучкої розробки програмного забезпечення. Скрам чітко робить акцент на якісному контролі процесу розробки.

Підхід вперше описали Гіротакі Такеучі та Ікуджіро Нонака в статті The New New Product Development Game (Гарвардський Діловий Огляд, січ–лют 1986). Вони відзначили, що проекти, над якими працюють невеликі, крос-функціональні команди, зазвичай систематично продукують кращі результати, і пояснили це, як «підхід регбі». У 1991 році ДеГрейс та Шталь у книжці Злі проблеми, справедливі рішення послалися на цей підхід, як на Scrum (штовханина; сутичка навколо м'яча (у регбі)), спортивний термін, згаданий в статті Такеучі і Нонака. Кен Швабер на початку 1990–х використовував підхід який привів Scrum в його компанію.

Вперше метод Scrum було представлено на загальний огляд задокументованим, чітко сформульованим та описаним спільно Сазерлендом та

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		62

Швабером на OOPSLA'96 в Остіні. Швабер та Сазерленд протягом наступних років працювали разом щоб обробити та описати весь їхній досвід та найкращі практичні зразки для індустрії в одне ціле, в ту методологію, що відома сьогодні як Scrum. Швабер об'єднав зусилля з Майком Бідлом в 2001, щоб детально описати метод в книжці Agile Software Development with SCRUM. Не зважаючи на те, що для Scrum нарікли долю управління проектами з розробки ПЗ, він може також використовуватися в роботі команд обслуговувань програмного забезпечення (software maintenance teams), або як підхід управління розробкою і супроводом програм: Scrum of Scrums.

Scrum – це кістяк процесу, який включає набір методів і попередньо визначених ролей. Головні дійові особи – ScrumMaster, той хто опікується процесами, веде їх і працює як керівник проекту, Власник Продукту, людина, що представляє інтереси кінцевих користувачів та інших зацікавлених в продукті сторін, та Команду, яка включає розробників.

Протягом кожного спринту, 15–30 денного періоду (тривалість визначається командою), працівники створюють функціональний ріст програмного забезпечення.

Набір можливостей, які імплементуються кожного спринту, приходять з етапу, що має назву product backlog (документація запитів на виконання робіт), який має найвищу пріоритетність за рівнем вимог до роботи, що повинна бути виконана.

Запити на виконання робіт (backlog items), що визначені протягом наради з планування спринту (sprint planning meeting), переміщуються в етап спринту. Протягом цієї наради Власник Продукту інформує про завдання, які він хоче, аби були виконані. Тоді Команда визначає, скільки з бажаного вони можуть зробити, щоб завершити необхідні частини протягом наступного спринту. Протягом спринту команда виконує визначений фіксований список завдань (т.з. backlog items). Впродовж цього періоду ніхто не має права змінювати перелік запитів на

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		63

Компоненти (components) – указує, які компоненти (наприклад, база даних, сервер, клієнт) будуть зачеплені при реалізації історії. Дане поле складається з групи checkbox'ів, які відмічаються, якщо відповідні компоненти потребують змін.

Ініціатор запиту (requestor). Product owner може захотіти зберігати інформацію про усіх замовників, зацікавлених у даній задачі. Це потрібно для того, щоб тримати їх у курсі діла про хід виконання робіт.

ID у системі обліку помилок (bug tracking ID) – якщо ви використовуєте окрему систему обліку помилок, тоді у описі історії корисно зберігати посилання на всі дефекти, які до неї відносяться.

Sprint backlog – містить функціональність, обрану Product Owner із Product Backlog. Всі функції розбиті по задачах, кожна з яких оцінюється командою. Кожен день команда оцінює об'єм роботи, який необхідно провести для завершення задачі.

Burndown chart – показує, скільки вже виконано і скільки ще залишається зробити.

Планування спринта (Sprint Planning Meeting)

Проходить на початку нової ітерації Спринта:

– Із Product Backlog обираються задачі, зобов'язання по виконанню яких за спринт приймає на себе команда;

– На основі обраних задач створюється Sprint Backlog. Кожна задача оцінюється у ідеальних людино-годинах;

– Рішення задачі не повинно займати більше 12 годин або одного дня. При необхідності задача розбивається на підзадачі;

– Обговорюється та визначається, яким чином буде реалізовано цей об'єм робіт;

– Тривалість наради обмежена зверху 4–8 годинами в залежності від тривалості ітерації, досвіду команди тощо;

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

Далі обраний в S -блоці 32-бітовий елемент піддається операції XOR з R . Потім L циклічно зрушується на число, кратним восьми біткам, L і R міняються місцями, і раунд завершується.

Сам S -блок не статичний, він міняється кожні вісім раундів. Нарешті, по закінченні останнього раунду, над L і R виконується операція XOR з іншими частинами ключа, і половини поєднуються, утворюючи блок шифртексту.

Хоча частини ключа використовуються для операції XOR із блоком шифрування на початку й кінці виконання алгоритму, головне призначення ключа – генерація S -блоків.

Ці S -блоки секретні, по суті, це частина ключа. Повний розмір ключа алгоритму Khufu дорівнює 512 біт (64 байт), алгоритм надає спосіб генерації S -блоків по ключу.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

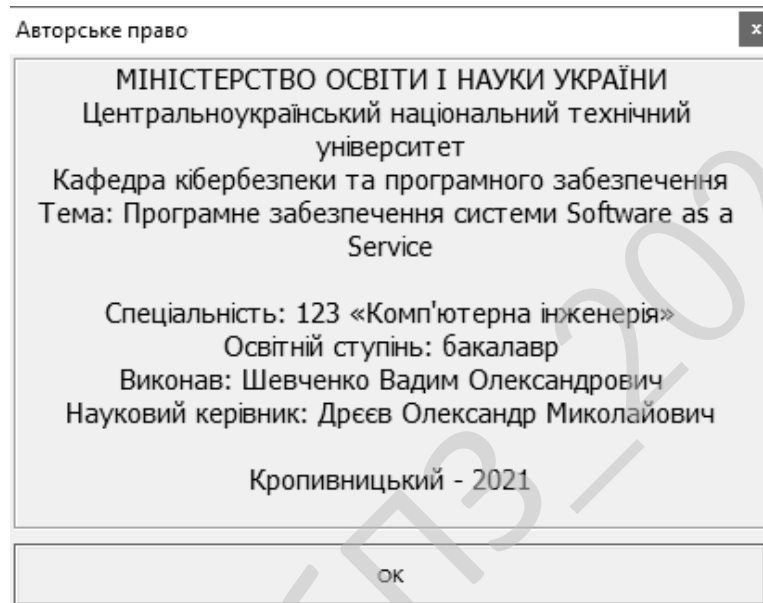


Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання кваліфікаційної бакалаврської роботи, призначено для системи Software as a Service.

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем Software as a Service.
- Досліджена система Software as a Service.
- На основі отриманих результатів досліджень створена програмна реалізація системи Software as a Service.

Розроблені під час виконання кваліфікаційної бакалаврської роботи алгоритми дозволяють успішно вирішувати завдання Software as a Service.

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. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи Software as a Service. 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.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Khufu.

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.

Кафедра КБПЗ – 2021 рік

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гмурман В.Е. Теория вероятностей и математическая статистика / В.Е. Гмурман. – М.: Высшая школа, 2003. – 479 с.
2. Смірнов О.А. Дисперсійний аналіз мережного трафіку для забезпечення інформаційної безпеки телекомунікаційних систем / О.О. Кузнецов, О.А. Смірнов, Д.О. Даниленко // Інформаційна та економічна безпека: сучасний стан та тенденції розвитку : монографія за заг. ред. – Х.: ХІБС УБС НБУ – 2014 – С. 82-100.
3. Смірнов О.А. Дослідження методів виявлення вторгнень в телекомунікаційні системи та мережі / Д.О. Даниленко, О.А. Смірнов, Є.В. Мелешко // Системи озброєння і військова техніка. – Випуск 1(29) – Х.: ХУПС – 2012. – С. 92-100
4. Смирнов А.А. Метод обнаружения вредоносного программного обеспечения. Часть 1. Корреляционный анализ сетевого трафика // А.А.Смирнов, Д.А. Даниленко, Е.В.Мелешко // Научно-технический журнал «Информационно-управляющие системы на железнодорожном транспорте» – Випуск 4(95). – Х.: УкрДАЗТ – 2012. – С. 8-14.
5. Смирнов А.А. Методы обнаружения вредоносного программного обеспечения в телекоммуникационных системах и сетях / Д.А. Даниленко // Збірник наукових праць "Системи обробки інформації". – Випуск 3(101) том 2. – Х.: ХУПС – 2012. – С. 152-155.
6. Смирнов А.А. Системы обнаружения и предотвращения вторжений для защиты телекоммуникационных сетей от вредоносного программного обеспечения / Д.А. Даниленко, А.А. Смирнов, А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 1 (21) том 2. – Київ: ДП «ЦНДІНУ». – 2012. – С. 183-186.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

7. Смирнов А.А. Системы обнаружения и предотвращения вторжений для защиты компьютерных сетей от вредоносного программного обеспечения / Д.А. Даниленко, А.А. Смирнов, И.Г. Кирилов // Збірник тез доповідей науково-практичної конференції «Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку». м. Харків. 21-22 березня 2012 р. – Харків. АВВ МВС. – 2012. – С. 70-71.

8. Смирнов О.А. Дослідження методів виявлення вторгнень в телекомунікаційні мережі для підвищення інформаційної безпеки // Д.О. Даниленко // Збірник тез науково-практичної конференції «Захист інформації в інформаційно-комунікаційних системах». м. Київ. 24-27 квітня 2012 р. – Київ: НАУ. – 2012. – С. 22-25.

9. Смирнов А.А. Исследование систем обнаружения и предотвращения вторжений для защиты телекоммуникационных сетей от вредоносного программного обеспечения / Д.А. Даниленко // Збірник тез доповідей VIII наукової конференції «Новітні технології – для захисту повітряного простору». Харків. 18-19 квітня 2012 р. – м. Харків. ХУПС. – 2012. – С. 45.

10. Смирнов А.А. Исследование методов сигнатурного обнаружения вредоносного программного обеспечения в телекоммуникационных системах и сетях // Д.А. Даниленко // Збірник тез XIII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 13-14 квітня 2012 р. – Кіровоград: КНТУ. – 2012. – С. 43-45.

11. Смирнов А.А. Исследование методов проактивной защиты от вредоносного программного обеспечения в телекоммуникационных системах и сетях / Д.А. Даниленко // Збірник тез V міжнародної науково-практичної конференції «Інтегровані інтелектуальні робототехнічні комплекси» (ПРТК-2012). м. Київ. 15-16 травня 2012 р. – Київ: НАУ. – 2012. – С. 314-315.

12. Смирнов А.А. Метод обнаружения вредоносного программного обеспечения на основе корреляционного анализа сетевого трафика / Д.А. Даниленко // Матеріали XII всеукраїнської наукової інтернет-конференції

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

«Наукові дослідження: зв'язок теорії і практики». м. Тернопіль. 29-30 квітня 2012 р. – Тернопіль: ТНЕУ. – 2012. – С. 9-10.

13. Смирнов А.А. Метод детектирования вредоносного трафика в телекоммуникационных сетях на основе использования bds-тестирования / Д.А. Даниленко // Збірник тез V міжнародної науково-практичної конференції «Комп'ютерні системи та мережні технології» (CSNT-2012). м. Київ. 13-15 червня 2012 р. – Київ: НАУ. – 2012. – С. 121.

14. Смирнов А.А. Обнаружение и предотвращение вторжений в компьютерных сетях на основе статистического анализа сетевого трафика / А.А. Смирнов, Д.А. Даниленко // Збірник тез доповідей науково-практичної конференції «Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку». м. Харків. 12-13 березня 2014 р. – Харків. АВВ МВС. – 2014. – С. 13-14.

15. Смирнов О.А. дисперсійний аналіз мережного трафіку для забезпечення інформаційної безпеки телекомунікаційних систем та мереж / О.А. Смирнов, Д.О. Даниленко // Збірник тез V Всеукраїнської науково-практичної конференції "Інформатика та системні науки". м. Полтава. 13-15 березня 2014 р. – Полтава: ПУЕТ. – 2014. – С. 289-291.

16. Смирнов А.А. Метод дисперсионного анализа сетевого трафика для обнаружения и предотвращения вторжений в телекоммуникационных системах и сетях / А.А. Смирнов, Д.А. Даниленко // Збірник тез VI міжнародної науково-практичної конференції “Проблеми і перспективи розвитку ІТ-індустрії”. м. Харків. 17-18 квітня 2014 р. – Харків: ХНЕУ. – 2014. – С. 258.

17. Смирнов О.А. метод забезпечення інформаційної безпеки телекомунікаційних систем з використанням дисперсійного аналізу мережного трафіку / О.А. Смирнов, Д.О. Даниленко // Збірник тез міжнародної науково-практичної конференції «Інформаційна та економічна безпека» (INFECO-2014)». м. Харків. 15-16 травня 2014 р. – Харків: ХІБС УБС НБУ. – 2014. – С. 135-139.

18. Девянин П.Н. Модели безопасности компьютерных систем /

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

П.Н. Девянин. – М.:Издательский центр «Академия», 2005. – 144 с.

19. Домарев В.В. Безопасность информационных технологий. Методология создания систем защиты / В.В. Домарев. – К.: ООО "ТИД "ДС", 2002 – 688 с.

20. ДСТУ ISO/IEC TR 13243-2003 Інформаційні технології. Посібник із методів та механізмів якості послуг / [Электронный ресурс]. – Режим доступа к ресурсу: <http://document.ua/informaciini-tehnologiyi.-posibnik-iz-metodiv-ta-mehanizmiv--nor2718.html>

21. ДСТУ В 3265 – 95. Зв'язок військовий. Терміни та визначення. – К.: УкрНДІССІ, 1995. – 23 с.

22. ДСТУ ISO 9000:2007 Системи управління якістю. Основні положення та словник термінів [Электронный ресурс]. – Режим доступа к ресурсу: <http://document.ua/docs/tdoc14237.php>

23. Ершов В.А.. Мультисервисные телекоммуникационные сети / В.А. Ершов, Н.А. Кузнецов. – М.: МГТУ им. Н.Э. Баумана, 2003. – 432 с.

24. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» [Электронный ресурс]. – Режим доступа к ресурсу: <http://zakon4.rada.gov.ua/laws/show/2594-15>

25. Информационная война и защита информации. Словарь основных терминов и определений [Электронный ресурс]. – Режим доступа к ресурсу: <http://www.csef.ru/files/csef/articles/2176/2176.pdf>

26. Казарин О.В. Безопасность программного обеспечения компьютерных систем / О.В. Казарин. – М.:МГУЛ, 2003. – 212 с.

27. Смірнова Т.В., Дреєв О.М., «Інформаційна технологія оптимізації технологічного процесу відновлення та зміцнювання поверхонь валів зі сталі як хмарний сервіс» у *Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка.* – Х. : Вид. Рожко С.Г. 2019. С. 92-107.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

28. Т.В.Смірнова, Л.І. Поліщук, «Дослідження хмарних технологій як сервісів для системи інженерних розрахунків» у *Кібербезпека та інформаційні технології: монографія.* – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. С. 106-121.

29. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов, «Експертна система оптимізації процесу відновлення та зміцнення поверхонь деталей типу «вал» електродуговим напиленням», *Системи управління, навігації та зв'язку*, № 2 (54). с. 149-154, 2019.

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

31. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов, Є.К. Солових, «Методи оптимізації технологічних процесів відновлення сталевих покриттів», *Shipbuilding & marine infrastructure / Суднобудування і морська інфраструктура* № 1 (11). с. 48-57, 2019.

32. Т. В. Смирнова, А. А. Смирнов, А. Н. Дреєв, А. В. Дудан, «Оптимизация технологического процесса восстановления и упрочнения поверхностей с заданными характеристиками в виде облачного сервиса», *Вестник Полоцкого государственного университета. Серия В, Промышленность. Прикладные науки. Республика Беларусь* – 2020. – № 3. – С. 50-61. Режим доступу: <http://elib.psu.by:8080/handle/123456789/24988>

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

34. Т.В.Смірнова, «Формалізація та реалізація структури технологічного процесу електродугового напилення для оптимізаційної експертної системи», *Технічні науки та технології.* № 1(19). С. 104-113. 2020. (Категорія Б)

35. Т.В. Смірнова «Формування евристичних правил, бази знань та формалізація структури й правил технологічного процесу для оптимізаційної

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

хмарної інформаційної системи», *Системи управління, навігації та зв'язку*, № 2 (60). с. 101-104, 2020. (Категорія Б)

36. Т.В. Смірнова, О.А. Смірнов, О.М. Дреєв, С.А. Смірнов, «Використання хмарних експертних систем в сфері інформаційного забезпечення обробки поверхні деталей», *Комп'ютерна інженерія і кібербезпека: досягнення та інновації*, м. Кропивницький. 27-29 листопада, 2018, с. 111-113

37. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Захист даних у інформаційній технології відновлення поверхонь деталей у вигляді хмарної платформи як послуги», *Проблеми кібербезпеки інформаційно-телекомунікаційних систем*, м. Київ, 11-12 квітня, 2019, с. 221-224.

38. Т.В. Смірнова, О.М. Дреєв, Є.К. Солових, «Хмарний сервіс інформаційної технології оптимізації технологічного процесу відновлення та зміцнювання поверхонь зі сталі», *Інформаційна безпека та інформаційні технології, Information Security and Information Technologies*, м. Харків, 24-25 квітня, 2019, с. 36

39. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов, «Побудова хмарної експертної системи оптимізації технологічного процесу електродугового напилення сталевих покриттів», *Міжнародний форум з інформаційних систем і технологій INFOS-2019*, м. Харків, 24-27 квітня, 2019, с. 95-98.

40. Т. В. Смирнова, А. А. Смирнов, Е. К. Соловых «Разработка математической модели и программного обеспечения для оптимизации режима электроконтактной обработки газотермических покрытий», *Комплексне забезпечення якості технологічних процесів та систем*, том 2, м. Чернігів, 14 – 16 травня, 2019, с. 78-79.

41. Т.В. Смірнова, О.М. Дреєв, Є.К. Солових, О.А. Смірнов, «Експертна система інформаційної технології оптимізації абстрактного технологічного процесу як хмарного сервісу», *Інформаційні технології: Наука, техніка, технологія, освіта, здоров'я. MicroCAD-2019*, м. Харків, 15-17 травня, 2019, с. 195.

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

42. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов, «Формування абстрактних експертних систем на основі досліджень відомих експертних систем», *XXI Міжнародний науково-практичний семінар «Комбінаторні конфігурації та їх застосування»*, м. Кропивницький, 17-18 травня, 2019, с. 143-147.

43. Т.В. Смірнова, О.А. Смірнов, «Захист даних у інформаційній технології відновлення поверхонь деталей у вигляді хмарної платформи як послуги», *Інформація, комунікація, суспільство – 2019*, см.т. Чинадієво, 16-18 травня, 2019, с. 25-26

44. Т.В. Смирнова, А.Н. Дреєв, А.А. Смирнов, «Информационная технология оптимизации технологического процесса восстановления и упрочнения поверхностей в виде облачного сервиса», *Modern Information, Measurement And Control Systems: Problems And Perspectives (MIMCS 2019)*, Baku, Azerbaijan, 01-02 July, 2019, p. 282.

45. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов, «Огляд відомих експертних систем оптимізації технологічних процесів», *Стратегія якості в промисловості і освіті*, м. Варна, Болгарія, 3-6 червня, 2019, с.442-444

46. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов, «Формалізація та узагальнення інформаційної моделі технологічних операцій зміцнення та відновлення сталевих поверхонь», *Інтелектуальні системи та інформаційні технології*, м.Одеса, 19-24 серпня, 2019, с. 211-213.

47. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов, «Формування інформаційної моделі технологічного процесу». *V Всеукраїнська науково-практична Інтернет-конференція «Електронні та мехатронні системи: теорія, інновації, практика»*, м. Полтава, 8 листопада, 2019, с. 87-91.

48. T.V. Smirnova, M.S. Chernovol, Ageev M. «Study of the spraying process and the influence of its factors on the properties of electric arc spraying coatings». *Materials of the 20th international scientific and technical seminar «Modern*

					КБР-123.21.0047.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

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	Перелік документів, що розробляються.....	5
8	Етапи розробки.....	6
9	Порядок контролю та приймання.....	6

					КБР-123.21.0047.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Шевченко В.О.				Програмне забезпечення системи Software as a Service	Літ.	Аркуш	Аркушів
Перевірів	Дресв О.М.					Б	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КІ-18-ЗСК			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи Software as a Service.

2 Підстава для розробки

Підставою для розробки служить завдання на кваліфікаційну бакалаврську роботу, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 204-02 від 28.12.2020 року).

3 Мета та призначення розробки

Метою кваліфікаційної бакалаврської роботи є розробка програмного забезпечення системи Software as a Service.

4 Джерела розробки

Джерелом цієї кваліфікаційної бакалаврської роботи 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;

					КБР-123.21.0047.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- системи Software as a Service;
- 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.

					КБР-123.21.0047.00.00.ТЗ	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 XP/Vista/7/8/10 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 XP/Vista/7/8/10.

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.

					КБР-123.21.0047.00.00.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 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 79 аркушів.

					КБР-123.21.0047.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8 Етапи розробки

8.1 Збір і обробка інформації по темі кваліфікаційної бакалаврської роботи. Постановка задачі на виконання кваліфікаційної бакалаврської роботи (складання ТЗ).

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень кваліфікаційної бакалаврської роботи.

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

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання кваліфікаційної бакалаврської роботи на попередній захист 22.05.2021 р.

11.2 Подання кваліфікаційної бакалаврської роботи на захист 5.06.2021 р.

					КБР-123.21.0047.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник кваліфікаційної бакалаврської роботи

_____ Дреєв О.М.

Програмне забезпечення системи Software as a Service

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 64

Літера: РП

Кропивницький – 2021 року

ОСНОВНИЙ ФАЙЛ ПРОЕКТУ ПЗ – SAAS.DPR

```
program SaaS;
{
Програмне забезпечення системи Software as a Service
Виконав: студент 4 курсу, Шевченко Вадим Олександрович
Керівник: Дреев О.М.
2021 рік
}
Uses

// бібліотеки
Forms,
windows,
Unit1 in 'Unit1.pas' {Form1},
Unit2 in 'Unit2.pas' {Form2},
Unit3 in 'Unit3.pas' {Form3},
frmInfoBase in 'InfoBase.pas' {Form4},
FILE_RAR in 'FILE_RAR.pas' {Form5},
DATA in 'DATA.pas' {Form6},
EB_DATABASE in 'EB_DATABASE.pas' {Form7},
About_DOC in 'AB.pas' {Form8};
{$R *.res}
// додаткові ресурси

begin
Application.Initialize;
// Початок ініціалізації ПЗ
Application.ShowMainForm := False;
// створення форм ПЗ з 1 з 8
Application.CreateForm(TForm1, Form1);
// створення форм ПЗ з 2 з 8
Application.CreateForm(TForm2, Form2);
// створення форм ПЗ з 3 з 8
Application.CreateForm(TForm3, Form3);
// створення форм ПЗ з 4 з 8
Application.CreateForm(TForm3, Form5);
// створення форм ПЗ з 5 з 8
Application.CreateForm(TForm3, Form6);
// створення форм ПЗ з 6 з 8
Application.CreateForm(TForm3, Form7);
// створення форм ПЗ з 7 з 8
Application.CreateForm(TForm3, Form8);
// створення форм ПЗ з 8 з 8
Application.Run;
// початок роботи ПЗ
end.
// Завершення роботи ПЗ
```

About.PAS – ФОРМА РОЗРОБНИКА

```
unit About;
// модуль About
{
Програмне забезпечення системи Software as a Service
Виконав: студент 4 курсу, Шевченко Вадим Олександрович
Керівник: Дреев О.М.
2021 рік
}

Interface
// об'ява даних

Uses
// бібліотеки
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, StdCtrls, ShellAPI;

type
// власні типи
TA = class(TForm) // створення класу форми
    btOk: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
private
public
end;

Var
// об'ява локальних типів даних
fmAbout: TA;

implementation
// реалізація

{$R *.dfm}
// файл ресурсів

end.
```

GRAF_RAR.PAS – МОДУЛЬ СТИСКУ ГРАФУ

```

unit GRAF_RAR;
// модуль GRAF_RAR
{
Програмне забезпечення системи Software as a Service
Виконав: студент 4 курсу, Шевченко Вадим Олександрович
Керівник: Дреев О.М.
2021 рік
}
Interface
// об'ява даних

Uses
// бібліотеки
Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, RAR, ComCtrls, Menus, RAR_DLL, StdCtrls,
ExtCtrls, FileCtrl;

type
// власні типи
TMyGrafForm = class(TForm)
// створення класу форми
RARArchive: TRAR;
MainMenu: TMainMenu;
File1: TMenuItemLite;
N1: TMenuItemLite;
Close1: TMenuItemLite;
Open1: TMenuItemLite;
N2: TMenuItemLite;
ListView: TListView;
OpenDialog: TOpenDialog;
LogMemo: TMemo;
Archiveinformation1: TMenuItemLite;
N3: TMenuItemLite;
Archivel: TMenuItemLite;
test1: TMenuItemLite;
Extract1: TMenuItemLite;
About1: TMenuItemLite;
Dllversion1: TMenuItemLite;
Panell: TPanel;
OverallBar: TProgressBar;
SingleBar: TProgressBar;
fileLabel: TLabel;
CancelButton: TButton;
ExtractSelected1: TMenuItemLite;
N4: TMenuItemLite;
refresh1: TMenuItemLite;
procedure Close1Click(Sender: TObject);
procedure Open1Click(Sender: TObject);
procedure Archiveinformation1Click(Sender: TObject);
procedure test1Click(Sender: TObject);
procedure Extract1Click(Sender: TObject);
procedure About1Click(Sender: TObject);
procedure Dllversion1Click(Sender: TObject);
procedure CancelButtonClick(Sender: TObject);
procedure ExtractSelected1Click(Sender: TObject);
procedure RARArchiveError(Sender: TObject; const ErrorCode: Integer;
const Operation: TRAROperation);
procedure RARArchiveListFile(Sender: TObject;
const FileInformation: TRARFileItem);
procedure RARArchiveNextVolumeRequired(Sender: TObject;
const requiredFileName: string; out newFileName: string;
out Cancel: Boolean);
procedure RARArchivePasswordRequired(Sender: TObject;

```

```

    const HeaderPassword: Boolean; const FileName: string;
    out NewPassword: string; out Cancel: Boolean);
procedure RARArchiveProgress(Sender: TObject; const FileName: WideString;
    const ArchiveBytesTotal, ArchiveBytesDone, FileBytesTotal,
    FileBytesDone: Cardinal);
procedure RARArchiveReplace(Sender: TObject; const ExistingData,
    NewData: TRARReplaceData; out Action: TRARReplace);
procedure FormCreate(Sender: TObject);
private
public
end;
Var
// об'ява локальних типів даних
    MainForm: TMyGrafForm;

implementation
// реалізація

Uses
    Replace;
// бібліотеки

{$R *.dfm}
// файл ресурсів
procedure TMyGrafForm.Extract1Click(Sender: TObject);
Var
// об'ява локальних типів даних
    Path:String;
begin
    if SelectDirectory('', 'C:\', Path) then
        if RARArchive.Extract(path,True,nil) then showmessage('done')
        else
            showmessage('failed');
end;

procedure TMyGrafForm.ExtractSelected1Click(Sender: TObject);
Var
// об'ява локальних типів даних
    Path:String;
    i:integer;
    files:TStrings;
begin
    files:=TStringList.Create;
    for i := 0 to ListView.Items.Count - 1 do
        if ListView.Items.Item[i].Selected then
            files.add(ListView.Items.Item[i].Caption);
    if SelectDirectory('Select a directory', 'C:\', Path) then
        if RARArchive.Extract(path,True,files) then showmessage('done')
        else
            showmessage('failed');
    files.free;
end;

procedure TMyGrafForm.FormCreate(Sender: TObject);
begin
    DoubleBuffered:=True;
    OverallBar.DoubleBuffered:=True;
    SingleBar.DoubleBuffered:=True;
end;

procedure TMyGrafForm.Open1Click(Sender: TObject);
begin
    if OpenFileDialog.Execute then begin
        ListView.Items.BeginUpdate;
        ListView.Items.Clear;
        if RARArchive.OpenFile(OpenDialog.FileName) then

```

```

        Archiveinformation1.Click
    else
        showmessage('Неможливо відчинити файл!');
        ListView.Items.EndUpdate;
    end;
end;

procedure TMyGrafForm.RARArchiveError(Sender: TObject; const ErrorCode: Integer;
const Operation: TRAROperation);
Var
// об'ява локальних типів даних
op:String;
begin
    case operation of
        roInitArchive: op:='init arc';
        roListFiles: op:='list arc';
        roExtract: op:='extract arc';
        roTest: op:='test arc';
    end;
    LogMemo.lines.add(op+' : '+inttostr(ErrorCode));
end;

procedure TMyGrafForm.RARArchiveListFile(Sender: TObject;
const FileInformation: TRARFileItem);
Var
// об'ява локальних типів даних
Attrib:String;
begin
    with listView.Items.Add do begin
        Caption:=FileInformation.FileNameW;
        SubItems.Add(getSizeName(FileInformation.CompressedSize));
        SubItems.Add(getSizeName(FileInformation.UncompressedSize));
        Attrib:='';
        case fileinformation.Attributes of
            {$WARN SYMBOL_PLATFORM OFF}
// компіляторні налаштування
            faReadOnly: Attrib:=Attrib+'R';
            faHidden: Attrib:=Attrib+'H';
            faSysFile: Attrib:=Attrib+'S';
            faVolumeID: Attrib:=Attrib+'V';
            faDirectory: Attrib:=Attrib+'D';
            faArchive: Attrib:=Attrib+'A';
            faAnyFile: Attrib:=Attrib+'F';
            {$WARN SYMBOL_PLATFORM ON}
// компіляторні налаштування
        end;
        subitems.add(Attrib);
        SubItems.Add(FileInformation.HostOS);
        SubItems.Add(FileInformation.Comment);
        subitems.Add(formatdatetime('c',FileInformation.Time));
        case FileInformation.CompressionStrength of
            48: subitems.Add('stored');
            49: subitems.Add('fastest');
            50: subitems.Add('fast');
            51: subitems.Add('normal');
            52: subitems.Add('good');
            53: subitems.Add('best');
        end;
        subitems.add(inttostr(FileInformation.ArchiverVersion div
10)+'.'+(inttostr(FileInformation.ArchiverVersion mod 10)));
        subitems.add(BoolToStr(fileinformation.Encrypted));
        SubItems.Add(FileInformation.CRC32);
    end;
end;
end;

```

```

procedure TMyGrafForm.RARArchiveNextVolumeRequired(Sender: TObject;
  const requiredFileName: string; out newFileName: string; out Cancel: Boolean);
begin
  newFileName:=requiredFileName;
  Cancel:=not InputQuery('next volume',requiredFileName,newFileName);
end;

procedure TMyGrafForm.RARArchivePasswordRequired(Sender: TObject;
  const HeaderPassword: Boolean; const FileName: string;
  out NewPassword: string; out Cancel: Boolean);
begin
  Cancel:=not InputQuery('Пароль?'+extractFileName
  (FileName),extractFileName(FileName),NewPassword);
end;

procedure TMyGrafForm.RARArchiveProgress(Sender: TObject; const FileName:
WideString;
  const ArchiveBytesTotal, ArchiveBytesDone, FileBytesTotal,
  FileBytesDone: Cardinal);
begin
  fileLabel.Caption:='archive: '+getSizeName(ArchiveBytesDone)+' of
'+getSizeName(ArchiveBytesTotal)+' , file: '+getSizeName(FileBytesDone)+' of
'+getSizeName(FileBytesTotal)+' '+FileName;
  if FileBytesTotal>0 then
    SingleBar.Position:=round(100 / FileBytesTotal*FileBytesDone)
  else
    SingleBar.Position:=0;
  if ArchiveBytesTotal>0 then
OverallBar.Position:=round(100/ArchiveBytesTotal*ArchiveBytesDone)
  else
    OverallBar.Position:=0;
  fileLabel.Repaint;
  SingleBar.Repaint;
  OverallBar.Repaint;
  Application.ProcessMessages;
end;

procedure TMyGrafForm.RARArchiveReplace(Sender: TObject; const ExistingData,
  NewData: TRARReplaceData; out Action: TRARReplace);
Var
  // об'ява локальних типів даних
  ReplaceForm:TReplaceForm;
begin
  ReplaceForm:=TReplaceForm.Create(Self);
  ReplaceForm.existent.Caption:='existent: '+ExistingData.FileName;
  ReplaceForm.archive.Caption:='in archive: '+NewData.FileName;
  ReplaceForm.ShowModal;
  Action:=ReplaceForm.Action;
  ReplaceForm.Free;
end;

procedure TMyGrafForm.test1Click(Sender: TObject);
begin
  if RARArchive.Test then
    showmessage('Помилка не знайдено!')
  else
    showmessage('Архів ушкоджен!');
end;

function GetSizeName(const Size : int64): String;
begin
  Result := 'error';
  if Size < 0 then exit;
  if Size < 1024 then

```

```
begin
  Result := inttostr(Size)+' Byte';
  exit;
end;
if (1024 <= Size) and (Size < 1048576) then
begin
  Result := floattostr((round((Size/1024)*100))/100)+' KB';
  exit;
end;
if (1048576 <= Size) and (Size < 1073741824) then
begin
  Result := floattostr((round((Size/1048576)*100))/100)+' MB';
  exit;
end;
if Size = 1073741824 then
  Result:=('1 GB');
if Size > 1073741824 then
begin
  Result := floattostr((round((Size/1073741824)*100))/100)+' GB';
  end;
end;

function BoolToStr(value:boolean):String;
begin
  if Value then
    Result:='True'
  else
    Result:='False';
end;

procedure TMyGrafForm.CancelButtonClick(Sender: TObject);
begin
  RARArchive.Abort;
end;

procedure TMyGrafForm.Close1Click(Sender: TObject);
begin
  Close;
end;

procedure TMyGrafForm.Dllversion1Click(Sender: TObject);
begin
  ShowMessage(inttostr(RARArchive.getDllVersion));
end;

end.
```

EB_DATABASE.PAS - МОДУЛЬ МЕРЕЖНОЇ ВЗАЄМОДІЇ З БД

```

unit EB_DATABASE;
// модуль EB_DATABASE
{
Програмне забезпечення системи Software as a Service
Виконав: студент 4 курсу, Шевченко Вадим Олександрович
Керівник: Дреев О.М.
2021 рік
}
Interface
// об'ява даних

Uses
// бібліотеки
Windows,
WinSock,
Snmp,
SysUtils,
Classes;

Const
// константа
mibLen = $0A;

type
// власні типи
TMibId = array[1..mibLen] of integer;
{ Об'ява класу TIpConnInfo }
TIpConnInfo = class
private
    FRemoteIp :TInAddr;
    FRemotePort :integer;
    FState :DWORD;
    FLocalIp :TInAddr;
    FLocalPort :integer;
    FProto :ShortString;
    function GetLocalPort :Integer;
    function GetLocalIpString :String;
    function Ip2Str( const Ip :TInAddr ):String;
    function GetRemotePort :Integer;
    function GetRemoteIpString :String;
    function GetStateString :String;
public
    constructor Create( const aProto :String );
    property IpProtoName :ShortString
        read FProto;
    property LocalIp :TInAddr
        read FLocalIp;
    property LocalPort :Integer
        read GetLocalPort;
    property LocalIpString :String
        read GetLocalIpString;
    property RemoteIpString :String
        read GetRemoteIpString;
    property RemoteIp :TInAddr
        read FRemoteIp;
    property RemotePort :Integer
        read GetRemotePort;
    property State :DWORD
        read FState;
    property StateString :String
        read GetStateString;
end;

{ Об'ява класу }

```

```

EConnListError = class(Exception);
    EConnListLockError = class(EConnListError);
    EConnListUnlockError = class(EConnListError);
    EConnListRefreshError = class(EConnListError);
TIpConnListStatus = ( connlist_ready, connlist_refreshing );
TIpProtocol = (udp_ip, tcp_ip);
TIpProtocols = set of TIpProtocol;

TF = class(TComponent) { Об'ява класу }
private
    FConnections :TList;
    FWsInited :Bool;
    FSnmpLib :THandle;
    FSnmpQueryProc :Pointer;
    FSnmpInitProc :Pointer;
    FProtocols :TIpProtocols;
    FStatus :TIpConnListStatus;
    FOnStatusChange :TNotifyEvent;
    FAccessMutex :THandle;
    FPollForTrapEvent :THandle;
    FSupportedViewRoot :TAsnObjectIdentifier;
    function GetConnCount :Integer;
    function GetConnections( Index :Integer ):TIpConnInfo;
protected
    procedure StatusChange; virtual;
    procedure SetStatus( Value :TIpConnListStatus ); virtual;
    procedure DoLock;
    procedure DoUnlock;
    procedure Clear;
public
    constructor Create( aOwner :TComponent ); override;
    destructor Destroy; override;
    procedure Refresh;
    function Lock( Timeout :DWORD ):Bool;
    function Unlock :Bool;
    property Status :TIpConnListStatus
        read FStatus;
    property Connections[ Index :Integer ] :TIpConnInfo
        read GetConnections; default;
    property ConnCount :Integer
        read GetConnCount;
published
    property Protocols :TIpProtocols
        read FProtocols write FProtocols;
    property OnStatusChange :TNotifyEvent
        read FOnStatusChange write FOnStatusChange;
end;
ENetstatError = class(Exception);
{ Об'ява класу }
TNetstatCounterObject = class
{ Об'ява класу }
private
    FName :String;
    FId :TMibId;
    FDescription :String;
    FWsInited :Bool;
    FSnmpLib :THandle;
    FSnmpQueryProc :Pointer;
    FSnmpInitProc :Pointer;
    function GetValue :DWORD;
public
// створення об'єкту класу
constructor Create(const aName,aDesc:String; const aId :TMibId);
    destructor Destroy; override;

```

```

    property Description :String
        read FDescription;
    property Name :String
        read FName;
    property Id :TMibId
        read FId;
    property Value :DWORD
        read GetValue;
end;

TCounterList = class(TComponent) { Об'ява класу }
private
    FCounters :TStringList;
    function GetCount :Integer;
    function GetCounters(Index:Integer):TNetstatCounterObject;
public
    constructor Create( aOwner :TComponent ); override;
    destructor Destroy; override;
    procedure Clear;
    procedure AddCounter( aCounter :TNetstatCounterObject );
    function IndexOf( const aName :String ):Integer;
    property Count :Integer
        read GetCount;
    property Counters[Index :Integer] :TNetstatCounterObject
        read GetCounters; default;
end;

TIpStats = class(TCounterList) { Об'ява класу }
public
    constructor Create( aOwner :TComponent ); override;
end;

TICmpStats = class(TCounterList) { Об'ява класу }
public
    constructor Create( aOwner :TComponent ); override;
end;

TTCPStats = class(TCounterList) { Об'ява класу }
public
    constructor Create( aOwner :TComponent ); override;
end;

TUDPStats = class(TCounterList) { Об'ява класу }
public
    constructor Create( aOwner :TComponent ); override;
end;
procedure Register;

implementation
// реалізація

const
// константи
DEFAULT_LOCK_TIMEOUT = 100;
SNMP_LIB_NAME = 'inetmib1.dll';
SNMP_INITPROC_NAME = 'SnmplibInit';
SNMP_QUERYPROC_NAME = 'SnmplibQuery';

type
// власні типи
TSnmpInitProc = function(
    dwTimeZeroReference : DWORD;
    Var hPollForTrapEvent : THandle;
    Var SupportedView : TAsnObjectIdentifier) : BOOL; stdcall;

```

```

TSnmpQueryProc = function(
    RequestType : Byte;
    Var VariableVindings : TRFC1157VarBindList;
    Var ErrorStatus : TAsnInteger;
    Var ErrorIndex : TAsnInteger) : BOOL; stdcall;

{ TIpStats }

// створення об'єкту класу
constructor TIpStats.Create( aOwner :TComponent );
begin
    inherited;
    AddCounter( TNetstatCounterObject.Create( 'ipDefaultTTL', '',
mib_ipDefaultTTL));
    AddCounter( TNetstatCounterObject.Create( 'ipInReceives', '',
mib_ipInReceives));
    AddCounter( TNetstatCounterObject.Create( 'ipInHdrErrors', '',
mib_ipInHdrErrors));
end;

function TF.Lock( Timeout :DWORD ):Bool;
begin
    result := WaitForSingleObject(FAccessMutex, Timeout ) = WAIT_OBJECT_0;
end;

function TF.Unlock :Bool;
begin
    result := ReleaseMutex(FAccessMutex);
end;

procedure TF.Refresh;
    procedure ReadTcpTable;
        Var
        // об'єва локальних типів даних
        varBind      :TRFC1157VarBind;
        varBindList  :TRFC1157VarBindList;
        errorStatus  :TAsnInteger;
        errorIndex   :TAsnInteger;
        ConnIndex    :Integer;
        Info         :TIpConnInfo;
        ListTail     :Integer;
    begin
        fillchar( varBindList, SizeOf(varBindList), 0);
        varBindList.List := @varBind;
        varBindList.len := 1;
        fillchar( varBind, SizeOf(varBind), 0);
        varBind.Name.idLength := mibLen;
        varBind.name.ids := @mib_tcpConnTable;
        ListTail := FConnections.Count;
        if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
varBindList, errorStatus, errorIndex) then
            raise EConnListRefreshError.Create(m_err_query);
        if varBindList.list.value.asnType = ASN_NULL then
            exit;
        while (varBindList.list.value.asnType = ASN_INTEGER) do
            begin
                Info := TIpConnInfo.Create('TCP');
                Info.FState := varBindList.list.value.Counter;
                FConnections.Add(Info);
                if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
varBindList, errorStatus, errorIndex) then
                    raise EConnListRefreshError.Create(m_err_query);
            end;
        if varBindList.list.value.asnType = ASN_NULL then
            raise EConnListRefreshError.Create(m_err_query);
    end;
end;

```

```

ConnIndex := ListTail;
while (varBindList.list.value.asnType = ASN_RFC1155_IPADDRESS) do
begin
    move(varBindList.list.value.address.stream^,
        Connections[ConnIndex].FLocalIP, sizeof(TInAddr));
    if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
    inc(ConnIndex);
end;
if varBindList.list.value.asnType = ASN_NULL then
    raise EConnListRefreshError.Create(m_err_queryend);
{ читання }
ConnIndex := ListTail;
while (varBindList.list.value.asnType = ASN_INTEGER) do
begin
    Connections[ConnIndex].FLocalPort := varBindList.list.value.counter;
    if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
    inc(ConnIndex);
end;
if varBindList.list.value.asnType = ASN_NULL then
    raise EConnListRefreshError.Create(m_err_queryend);
{ читання віддаленого порту ips }
ConnIndex := ListTail;
while (varBindList.list.value.asnType = ASN_RFC1155_IPADDRESS) do
begin
    move(varBindList.list.value.address.stream^,
        Connections[ConnIndex].FRemoteIP,
        sizeof(TInAddr));
    if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
    inc(ConnIndex);
end;
if varBindList.list.value.asnType = ASN_NULL then
    raise EConnListRefreshError.Create(m_err_queryend);
{ читання віддаленого порту }
ConnIndex := ListTail;
while (varBindList.list.value.asnType = ASN_INTEGER) do
begin
    Connections[ConnIndex].FRemotePort := varBindList.list.value.counter;
    if not TSnmpQueryProc(FSnmpQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
    inc(ConnIndex);
end;
end;

procedure ReadUdpTable;
Var
    // об'ява локальних типів даних
    varBind      :TRFC1157VarBind;
    varBindList  :TRFC1157VarBindList;
    errorStatus  :TAsnInteger;
    errorIndex   :TAsnInteger;
    ConnIndex    :Integer;
    Info         :TIpConnInfo;
    ListTail     :Integer;
begin
    fillchar( varBindList, SizeOf(varBindList), 0);
    varBindList.List := @varBind;
    varBindList.len := 1;
    fillchar(varBind, SizeOf(varBind), 0);

```

```

varBind.Name.idLength := mibLen;
varBind.name.ids := @mib_udpTable;
ListTail := FConnections.Count;
if not TSnmQueryProc(FSnmQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
    varBindList, errorStatus, errorIndex) then
    raise EConnListRefreshError.Create(m_err_query);
if varBindList.list.value.asnType = ASN_NULL then
    exit;
{ читання }
while (varBindList.list.value.asnType = ASN_RFC1155_IPADDRESS) do
begin
    Info := TIpConnInfo.Create('UDP');
    move(varBindList.list.value.address.stream^,
        Info.FLocalIP,
        sizeof(TInAddr));
    FConnections.Add(Info);
    if not TSnmQueryProc(FSnmQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
end;
if varBindList.list.value.asnType = ASN_NULL then
    raise EConnListRefreshError.Create(m_err_query);
{ читання }
ConnIndex := ListTail;
while (varBindList.list.value.asnType = ASN_INTEGER) do
begin
    Connections[ConnIndex].FLocalPort := varBindList.list.value.counter;
    if not TSnmQueryProc(FSnmQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
    inc(ConnIndex);
end;
end;
begin
    DoLock;
    try
        if FStatus <> connlist_ready then
            raise EConnListError.Create(m_err_inv_state);
        SetStatus( connlist_refreshing );
        Clear;
        if tcp_ip in FProtocols then
            ReadTcpTable;
        if udp_ip in FProtocols then
            ReadUdpTable;
    finally
        DoUnlock;
        SetStatus(connlist_ready);
    end;
end;

{ TicmpStats }
{ TNetstatCounterObject }
function TNetstatCounterObject.GetValue :DWORD;
Var
// об'ява локальних типів даних
varBind      :TRFC1157VarBind;
varBindList  :TRFC1157VarBindList;
errorStatus  :TAsnInteger;
errorIndex   :TAsnInteger;
begin
varBindList.List := @varBind;
varBindList.len := 1;
fillchar(varBind, SizeOf(varBind), 0);
varBind.Name.idLength := mibLen;
varBind.Name.ids := @FId;

```

```

if not TSnmplibQueryProc(FSnmplibQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
    varBindList, errorStatus, errorIndex) then
    raise ENetstatError.Create(m_err_query);
if not (varBindList.list.value.asnType in
[ASN_GAUGE32, ASN_INTEGER, ASN_INTEGER32, ASN_COUNTER32, ASN_UNSIGNED32]) then
    raise ENetstatError.Create(m_err_invtype);
result := varBindList.list.value.Counter;
end;

// створення об'єкту класу
constructor TNetstatCounterObject.Create(const aName,
    aDesc:String; const aId :TMibId );
var
    WSData :TWSAData;
    PollForTrapEvent :THandle;
    SupportedViewRoot :TAsnObjectIdentifier;
begin
    inherited Create;
    FWSInit := WsaStartup($0101, WSData ) = 0;
    if not FWSInit then
        raise EConnListError.Create(m_err_wsstartup);
    FSnmplib := LoadLibrary(SNMP_LIB_NAME);
    if FSnmplib = 0 then
        raise EConnListError.Create(m_err_loadlib);
    FSnmplibInitProc := GetProcAddress(FSnmplib, SNMP_INITPROC_NAME);
    FSnmplibQueryProc := GetProcAddress(FSnmplib, SNMP_QUERYPROC_NAME);
    if (( FSnmplibQueryProc = Nil ) or ( FSnmplibInitProc = Nil ) ) then
        raise EConnListError.Create(m_err_loadlib);
    if not TSnmplibInitProc(FSnmplibInitProc)(GetTickCount,
        PollForTrapEvent, SupportedViewRoot) then
        raise EConnListError.Create(m_err_initlib);
    FName := aName;
    FDescription := aDesc;
    FId := aId;
    dec(Fid[8]);
end;

// знищення об'єкту класу
destructor TNetstatCounterObject.Destroy;
begin
    if FSnmplib <> 0 then FreeLibrary(FSnmplib);
    if FWSInit then WSACleanup;
    inherited;
end;

{ TCounterList }
procedure TCounterList.AddCounter(aCounter:TNetstatCounterObject);
begin
    assert( assigned( aCounter ) );
    FCounters.AddObject( aCounter.Name, aCounter );
end;

function TCounterList.GetCount :Integer;
begin
    assert( assigned( FCounters ) );
    result := FCounters.Count;
end;

function TCounterList.GetCounters(Index:Integer ):TNetstatCounterObject;
begin
    assert( assigned( FCounters ) );
    result := TNetstatCounterObject(FCounters.Objects[Index]);
end;

```

```

function TCounterList.IndexOf( const aName :String ):Integer;
begin
    result := FCounters.IndexOf( aName );
end;

// створення об'єкту класу
constructor TCounterList.Create( aOwner :TComponent );
begin
    inherited;
    FCounters := TStringList.Create;
end;

// знищення об'єкту класу
destructor TCounterList.Destroy;
begin
    Clear;
    if assigned(FCounters) then FCounters.Free;
    inherited;
end;

procedure TCounterList.Clear;
var
    i :integer;
    c :TObject;
begin
    if assigned(FCounters) then
        for i := FCounters.Count-1 downto 0 do
            begin
                c := FCounters.Objects[i];
                FCounters.Delete(i);
                if assigned(c) then c.free;
            end;
end;
{ TIpConNInfo }
// створення об'єкту класу
constructor TIpConNInfo.Create( const aProto :String );
begin
    inherited Create;
    FState := LISTEN;
    FProto := aProto;
end;

function TIpConNInfo.Ip2Str( const Ip :TInAddr ):String;
begin
    result := format('%d.%d.%d.%d',
        [Integer(ip.s_un_b.s_b1),
        Integer(ip.s_un_b.s_b2),
        Integer(ip.s_un_b.s_b3),
        Integer(ip.s_un_b.s_b4)]);
end;

function TIpConNInfo.GetLocalPort :Integer;
begin
    result := FLocalPort;
end;

function TIpConNInfo.GetLocalIpString :String;
begin
    result := Ip2Str(FLocalIp);
end;

function TIpConNInfo.GetRemotePort :Integer;
begin
    if State = LISTEN then
        result := 0

```

```

    else
        result := FRemotePort;
end;

function TIpConnInfo.GetRemoteIpString :String;
begin
    result := Ip2Str(FRemoteIp);
end;

function TIpConnInfo.GetStateString :String;
const
    state_name :array[CLOSED..TCB_DISCARD] of string[16] =
        ('CLOSED',
         'LISTEN',
         'SYN_SENT',
         'SYN_RECEIVED',
         'ESTABLISHED',
         'CLOSE_WAIT',
         'FIN_WAIT_1',
         'CLOSING',
         'LAST_ACK',
         'FIN_WAIT_2',
         'TIME_WAIT',
         'TCB_DISCARD'
        );
begin
    if State in [CLOSED..TCB_DISCARD] then
        result := state_name[state]
    else
        result := 'UNKNOWN';
end;

{ TF }
function TF.GetConnCount :Integer;
begin
    assert(assigned(FConnections));
    result := FConnections.Count;
end;

function TF.GetConnections( Index :Integer ):TIpConnInfo;
begin
    assert(assigned(FConnections));
    result := FConnections[Index];
end;

procedure TF.Clear;
var
    i :integer;
    p :TIpConnInfo;
begin
    if assigned(FConnections) then
        for i := FConnections.Count-1 downto 0 do
            begin
                p := FConnections[i];
                assert(assigned(p));
                FConnections.Delete(i);
                p.free;
            end;
        end;
end;

procedure TF.StatusChange;
begin
    if assigned(FOnStatusChange) then FOnStatusChange(Self);
end;

```

```

procedure TF.SetStatus( Value :TIpConnListStatus );
begin
    if FStatus <> Value then
        begin
            FStatus := Value;
            StatusChange;
        end;
end;

procedure TF.DoLock;
begin
    if not Lock(DEFAULT_LOCK_TIMEOUT) then
        raise EConnListLockError.Create(m_err_lock);
end;

procedure TF.DoUnlock;
begin
    if not UnLock then
        raise EConnListUnLockError.Create(m_err_unlock);
end;

// створення об'єкту класу
constructor TF.Create( aOwner :TComponent );
var
    WSDData :TWSAData;
begin
    inherited;
    FProtocols := [udp_ip, tcp_ip];
    FWSInitied := WsaStartup($0101, WSDData ) = 0;
    if not FWSInitied then
        raise EConnListError.Create(m_err_wsstartup);
    FSNmpLib := LoadLibrary(SNMP_LIB_NAME);
    if FSNmpLib = 0 then
        raise EConnListError.Create(m_err_loadlib);
    FSNmpInitProc := GetProcAddress(FSNmpLib, SNMP_INITPROC_NAME);
    FSNmpQueryProc:=GetProcAddress(FSNmpLib, SNMP_QUERYPROC_NAME);
    if ( ( FSNmpQueryProc = Nil ) or ( FSNmpInitProc = Nil ) ) then
        raise EConnListError.Create(m_err_loadlib);
    if not TSNmpInitProc(FSNmpInitProc)(GetTickCount,
        FPollForTrapEvent, FSupportedViewRoot) then
        raise EConnListError.Create(m_err_initlib);
    FAccessMutex := CreateMutex(nil, false, nil);
    if FAccessMutex = 0 then
        raise EConnListError.Create(m_err_alloc);
    FConnections := TList.Create;
    FStatus := connlist_ready;
end;

// знищення об'єкту класу
destructor TF.Destroy;
begin
    Clear;
    if assigned(FConnections) then FConnections.Free;
    if FAccessMutex <> 0 then CloseHandle(FAccessMutex);
    if FSNmpLib <> 0 then FreeLibrary(FSNmpLib);
    if FWSInitied then WSACleanup;
    inherited;
end;

end.

```

DATA.PAS - РОБОТА СИСТЕМИ

```

unit Data;
// модуль Data
{
Програмне забезпечення системи Software as a Service
Виконав: студент 4 курсу, Шевченко Вадим Олександрович
Керівник: Дреев О.М.
2021 рік
}

Interface
// об'ява даних

Uses
// бібліотеки
Windows, SysUtils, ActiveX, ShlObj, Classes, ShellAPI;

const
// використовуємо константи
Nul = 0000;
MaxWord = $FFFF;
MaxInteger = $7FFFFFFF;
MaxFloat = 2.7e308;
MinFloat = 5.0e-324;
MaxExtended = 1.1e4932;
MinExtended = 9.99e-4933;
HalfCycle = Pi;
FullCycle = 2*Pi;
Quadrant = Pi/2;
chNull = #0;
chBackspace = #19;

type
// власні типи
PString=^TString;
TString=type AnsiString;
PAnsiStr=^TAnsiStr;
TAnsiStr=array[0..259] of AnsiChar;
PWideStr=^TWideStr;
TWideStr=array[0..259] of WideChar;
PShortStr=^TShortStr;
TShortStr=type ShortString;
PSetChar=^TSetChar;
TSetChar=set of AnsiChar;
PWideInt=^TWideInt;
TWideInt=type Int64;
TColorChannel = (ccRed, ccGreen, ccBlue, ccAlpha);
TColorChannels = set of TColorChannel;
PBoolean = ^Boolean;
function LetterToNumber(const Letter: TString): Integer;
function NumberToLetter(Number: Integer): TString;

procedure SplitAlphanumericName(const Name: TString; var Alpha: TString;
var Num: Integer; const AdditionalChars: TSetChar = []);
type
// власні типи
TUnknown = class (TObject, IUnknown)
protected
FRefCount: Integer;
function QueryInterface(const IID: TGUID; out Obj): HRESULT; virtual;
stdcall;
function _AddRef: Integer; virtual; stdcall;
function _Release: Integer; virtual; stdcall;
public
function Unknown: IUnknown; overload;

```

```

    procedure Unknown(out Obj); overload;
end;
TObjectX = TUnknown;
type
// власні типи
EShellLinkError = class (Exception);
TShellLink = class(TUnknown)
private
    FResult: HRESULT;
    FShellLink: IShellLink;
    FPersistDOC_PACK : IPersistDOC_PACK;
    FTemp: WideString;
    FDesktopFolder: TString;
    FProgramsFolder: TString;
    FStartMenuFolder: TString;
    FStartupFolder: TString;
    FMyDocsFolder: TString;
    function GetArguments: TString;
    function GetDescription: TString;
    function GetHotKey: word;
    function GetIconIndex: Integer;
    function GetIconLoc: TString;
    function GetPath: TString;
    function GetPIDL: PItemIDList;
    function GetShowCmd: Integer;
    function GetWorkDir: TString;
    procedure SetArguments(const Value: TString);
    procedure SetDescription(const Value: TString);
    procedure SetHotKey(const Value: word);
    procedure SetIconIndex(const Value: Integer);
    procedure SetIconLoc(const Value: TString);
    procedure SetPath(const Value: TString);
    procedure SetPIDL(const Value: PItemIDList);
    procedure SetShowCmd(const Value: Integer);
    procedure SetWorkDir(const Value: TString);
    procedure RunError(const Msg: TString; const Args: TString = '');
    function ResolveDOC_PACKName(DOC_PACKName: TString): PWideChar;
    function DesktopFolder: TString;
    function ProgramsFolder: TString;
    function StartMenuFolder: TString;
    function StartupFolder: TString;
    function MyDocsFolder: TString;
public
    constructor Create;
    destructor Destroy;override;
end;
type
// власні типи
TForEachFunc = function (Tag: Integer; Index: Integer; var Item): Integer;
register;
EDynArray = class (Exception);
TDynArray = class (TObjectX)
private
    FHandle: hLocal;
    FData: Pointer;
    FItemSize: Cardinal;
    FCount: Cardinal;
    function AllocMem(ACount: Cardinal; var Handle: hLocal): pointer;
    procedure FreeMem(var Handle: hLocal);
    procedure _SetCount(const Value: Cardinal);
    procedure DoSizeChanged;
protected
    function GetFirstItem: Pointer;
    procedure PutItem(Index: Integer; const Item);
    procedure GetItem(Index: Integer; out Item);

```

```

    procedure Error(Index: Integer);
    procedure SizeChanged; virtual;
    procedure SetCount(const Value: Cardinal); virtual;
public
    property Count: Cardinal read FCount write _SetCount;
    property ItemSize: Cardinal read FItemSize;
    property FirstItem: Pointer read FData;
    function Add: Integer; virtual;
    function AddItem(const Item): Integer; virtual;
    procedure Insert(Index: Integer); virtual;
    procedure InsertItem(Index: Integer; const Item); virtual;
    procedure Delete(Index: Integer); virtual;
    procedure DeleteItem(Index: Integer; out Item); virtual;
    procedure Extend(Count: Cardinal = 1); virtual;
    procedure Trim(Count: Cardinal = 1); virtual;
    procedure Swap(Index1, Index2: Cardinal); virtual;
    function ForEach(Tag: Integer; ForEachFunc: TForEachFunc): Integer; virtual;
    function GetItemPtr(Index: Integer): Pointer;
    constructor Create(ACount, AItemSize: Cardinal);
    destructor Destroy; override;
end;
TDynamicArrayClass = class of TDynamicArray;
type
// власні типи
    TDStatus = (fsReading, fsWriting);
    EDOC_PACKError = class (Exception);
const
// стандартні константи
    faReadOnly          = $00000001;
    faHidden            = $00000002;
    faSystem            = $00000004;
    faDirectory        = $00000010;
    faArchive           = $00000020;
    faEncrypted         = $00000040;
    faNormal            = $00000080;
    faTemporary        = $00000100;
    faSparseDOC_PACK   = $00000200;
    faReparsePoint     = $00000400;
    faCompressed        = $00000800;
    faOffline          = $00001000;
    faNotContentIndexed = $00002000;
type
// власні типи
{ Об'ява класу TD }
    TD = class (TObjectX)
    private
        FDOC_PACKName: TString;
        FHandle: HDOC_PACK;
        FStatus: TDStatus;
        FDummy: LongWord;
    procedure CreateBackup;
    function GetSize: Integer;
    function GetCreationTime: TDateTime;
    function GetLastAccessTime: TDateTime;
    function GetLastWriteTime: TDateTime;
    procedure SetCreationTime(const Value: TDateTime);
    procedure SetLastAccessTime(const Value: TDateTime);
    procedure SetLastWriteTime(const Value: TDateTime);
    function GetAttributes: LongInt;
    procedure SetAttributes(const Value: LongInt);
    protected
        function GetErrorMessage(Code: Integer): TString; dynamic;
    public
        property DOC_PACKName: TString read FDOC_PACKName;
        property Status: TDStatus read FStatus;

```

```

    property Handle: HDOC_PACK read FHandle;
    property Size: Integer read GetSize;
    property CreationTime: TDateTime read GetCreationTime write SetCreationTime;
    property LastAccessTime: TDateTime read GetLastAccessTime write
SetLastAccessTime;
    property LastWriteTime: TDateTime read GetLastWriteTime write SetLastWriteTime;
    property Attributes: LongInt read GetAttributes write SetAttributes;

{ Об'ява класу TDStrm }
TDStrm = class (TStream)
private
    FHandle: HDOC_PACK;
    FStatus: TDStatus;
    FDOC_PACKName: TString;
    procedure CreateBackup;
    function GetAttributes: LongInt;
    function GetCreationTime: TDateTime;
    function GetLastAccessTime: TDateTime;
    function GetLastWriteTime: TDateTime;
    procedure SetAttributes(const Value: LongInt);
    procedure SetCreationTime(const Value: TDateTime);
    procedure SetLastAccessTime(const Value: TDateTime);
    procedure SetLastWriteTime(const Value: TDateTime);
    protected
    procedure SetSize(NewSize: LongInt); override;
    procedure Error(Code: Integer); dynamic;
    function GetErrorMessage(Code: Integer): TString; dynamic;
    public
        property DOC_PACKName: TString read FDOC_PACKName;
        property Status: TDStatus read FStatus;
        property Handle: HDOC_PACK read FHandle;
        property CreationTime: TDateTime read GetCreationTime write SetCreationTime;
        property LastAccessTime: TDateTime read GetLastAccessTime write
SetLastAccessTime;
        property LastWriteTime: TDateTime read GetLastWriteTime write SetLastWriteTime;
        property Attributes: LongInt read GetAttributes write SetAttributes;
        constructor Create(ADOC_PACKName: TString; Backup: Boolean);
        constructor Open(ADOC_PACKName: TString);
        function Write(const Buffer; Count: LongInt): LongInt; override;
        function Read(var Buffer; Count: LongInt): LongInt; override;
        function Seek(Offset: LongInt; Origin: Word): LongInt; override;
        procedure Close;
        destructor Destroy; override;
        class procedure DecodeDateTime(const DateTime: TDateTime;
            Year, Month, Day, Hour, Min, Sec: PWord);
        class function EncodeDateTime(Year, Month, Day, Hour, Min, Sec: Word):
TDateTime;
        procedure UserError(Code: Integer);
end;

{ Об'ява класів }
ELanError = class (Exception);
TD = class;
PLanRow = ^TDataRow;
TDataRow = class (TDynamicArray)
private
    FPlan: TD;
public
    property Lan: TD read FPlan;
    constructor Create(AColCount: Integer; APlan: TD);
end;

{ Об'ява класу TDRows }
TDRows = class (TDynamicArray)

```

```

private
  FWidth: Integer;
  FColIndex: Integer;
  function GetRow(Index: Integer): TDRow;
  procedure SetRow(Index: Integer; const Value: TDRow);
  procedure SetWidth(const Value: Integer);
  function SetWidthFunc(Index: Integer; var Row: TDRow): Integer;
  function InsertColFunc(Index: Integer; var Row: TDRow): Integer;
  function DeleteColFunc(Index: Integer; var Row: TDRow): Integer;
public
  property Width: Integer read FWidth write SetWidth;
  property Row[Index: Integer]: TDRow read GetRow write SetRow; default;
  procedure InsertCol(Index: Integer);
  procedure DeleteCol(Index: Integer);
  constructor Create(ALan: TD);
end;

{ Об'ява класу TD }
TD = class (TUnknown)
private
  FItemSize : Cardinal;
  FRows: TDRows;
  function GetColCount: Integer;
  function GetRowCount: Integer;
  procedure SetColCount(const Value: Integer);
  procedure SetRowCount(const Value: Integer);
  function GetRow(Index: Integer): TDRow;
protected
  function CreateRow: TDRow; virtual;
public
  procedure GetItem(ACol, ARow: Integer; out Item);
  procedure PutItem(ACol, ARow: Integer; const Item);
  procedure InsertRow(Index: Integer);
  procedure DeleteRow(Index: Integer);
  procedure InsertCol(Index: Integer);
  procedure DeleteCol(Index: Integer);
  function ForEachRow(Tag: Integer; ForEachRowFunc: TForEachFunc): Integer;
  property ColCount: Integer read GetColCount write SetColCount;
  property RowCount: Integer read GetRowCount write SetRowCount;
  property Row[Index: Integer]: TDRow read GetRow;
  constructor Create(AColCount, ARowCount, AItemSize: Integer);
  destructor Destroy; override;
end;

implementation
// реалізація

Uses
// бібліотеки
  Consts, SysConst, DimConst;

type
// власні типи
  TDgIDItem = packed record
    LangID: Byte;
    Charset: Byte;
  end;

function SamePoint(const Point1, Point2: TPoint): LongBool;
begin
  Result:=TWideInt(Point1)=TWideInt(Point2);
end;

```

```

function IsNullPoint(const Point: TPoint): LongBool;
begin
  Result:=not LongBool(TWideInt(Point));
end;

procedure MovePoint(var Point: TPoint; DispX, DispY: Integer);
asm
  add [eax], edx
  add [eax+4], ecx
end;

function CloseTo(const Point1, Point2: TPoint; Distance: Integer): LongBool;
begin
  Result:=Inside(Point2, Rect(Point1.X-Distance, Point1.Y-Distance,
                              Point1.X+Distance, Point1.Y+Distance));
end;

function GetData(Num, Den:Double):Double;
begin
  if Den<>0 then begin
    Result:=arctan(Num/Den);
    if Den<0 then Result:=HalfCycle+Result else if Num<0 then
Result:=FullCycle+Result;
  end else begin
    if Num>0 then Result:=Quadrant else Result:=3*Quadrant;
  end;
end;

function GetAlpha(Y1, Z1, Y2, Z2, Y3, Z3:Double):Double;
Var // об'ява локальних типів даних
  A1, A2:Double;
begin
  A1:=GetData(Z1-Z2,Y2-Y1);
  A2:=GetData(Z3-Z2,Y2-Y3);
  if A2<A1 then A2:=FullCycle+A2;
  Result:=A2-A1;
end;

function GetAlphaScr(X1, Y1, X2, Y2, X3, Y3:Double):Double;
Var
// об'ява локальних типів даних
  A1, A2:Double;
begin
  A1:=GetData(X2-X1,Y1-Y2);
  A2:=GetData(X2-X3,Y3-Y2);
  if A2<A1 then A2:=FullCycle+A2;
  Result:=A2-A1;
end;

function CenterPoint(const Rect: TRect): TPoint;
asm
  push esi
  mov esi, eax
  mov eax, [esi]
  add eax, [esi+8]
  shr eax, 1
  mov [edx].TPoint.x, eax
  mov eax, [esi+4]
  add eax, [esi+12]
  shr eax, 1
  mov [edx].TPoint.y, eax
  pop esi
end;

```

```

function Max(const R1,R2:Integer):Integer;overload;
asm
    cmp eax, edx
    jng @@10
    ret
@@10:
    mov eax, edx
end;

function Max(const R1,R2:Extended):Extended;overload;
begin
    if R1>R2 then Result:=R1 else Result:=R2;
end;

function Max(const P1, P2: TPoint; CompareY: LongBool=False): TPoint; overload;
Var
// об'ява локальних типів даних
    F: function (const Point1, Point2: TPoint): Integer;
begin
    if CompareY then F:=ComparePointY else F:=ComparePointX;
    if F(P1, P2) = nMore then Result:=P1 else Result:=P2;
end;
function Min(const R1,R2:Integer):Integer;overload;
asm
    cmp eax, edx
    jnl @@10
    ret
@@10:
    mov eax, edx
end;

function Min(const R1,R2:Extended):Extended;overload;
begin
    if R1<R2 then Result:=R1 else Result:=R2;
end;
function Min(const P1, P2: TPoint; CompareY: LongBool = False): TPoint;
Var
// об'ява локальних типів даних
    F: function (const Point1, Point2: TPoint): Integer;
begin
    if CompareY then F:=ComparePointY else F:=ComparePointX;
    if F(P1, P2) = nLess then Result:=P1 else Result:=P2;
end;

procedure Sync;
asm
    call WinNT
    test eax, 1
    jz  @@10
    ret
@@10:
    mov  dx,3dah
@@wait:
    in   al,dx
    test al,8
    jz  @@wait
end;

function LinesIntersection(A1, B1, C1, A2, B2, C2: Extended; var Y, Z:
Extended): Boolean;
Var
// об'ява локальних типів даних
    Det: Extended;
begin
    Det:=A1*B2-A2*B1;

```

```

Result:=Abs(Det)>1e-20;
if Result then begin
  Y := (c2*b1-c1*b2)/det;
  Z := (a2*c1-a1*c2)/det;
end;
end;

```

```

function SegmentLength(const X1, Y1, X2, Y2: Extended): Extended;
asm

```

```

  fld  X1
  fld  X2
  fsub
  fld  st(0)
  fmul
  fld  Y1
  fld  Y2
  fsub
  fld  st(0)
  fmul
  fadd
  fsqrt
  fwaitend;

```

```

procedure Rotate(X, Y, X0, Y0, Alpha: Extended; var X1, Y1: Extended);
Var

```

```

// об'ява локальних типів даних

```

```

Sin, Cos: Extended;
DX, DY: Extended;

```

```

begin
  SinCos(Alpha, Sin, Cos);
  DX:=(X-X0); DY:=(Y-Y0);
  X1:=DX*Cos+DY*Sin+X0;
  Y1:=DY*Cos-DX*Sin+Y0;
end;

```

```

function LinesIntersection(Y1, Z1, Y2, Z2, Y3, Z3, Y4, Z4: Extended; var Y, Z:
Extended): Boolean; overload;

```

```

Var

```

```

// об'ява локальних типів даних

```

```

A1, B1, C1, A2, B2, C2: Extended;

```

```

begin
  Result:=LinesIntersection(A1, B1, C1, A2, B2, C2, Y, Z);
end;

```

```

procedure CopyRect(const Source: TRect; var Dest: TRect);

```

```

asm

```

```

  mov  ecx, 16
  call MoveMem

```

```

end;

```

```

function RectIntersection(const Rect1, Rect2: TRect): TRect;

```

```

begin

```

```

  RebuildRect(PRect(@Rect1)^);

```

```

  RebuildRect(PRect(@Rect2)^);

```

```

  if Inside(Rect2.TopLeft, Rect1) then begin

```

```

    if Inside(Rect2.BottomRight, Rect1) then begin

```

```

      Result:=Rect2;

```

```

      Exit;

```

```

    end else begin

```

```

      Result.TopLeft:=Rect2.TopLeft;

```

```

      Result.BottomRight:=Rect1.BottomRight;

```

```

      Exit;

```

```

    end;

```

```

  end;

```

```

  if Inside(Rect2.BottomRight, Rect1) then begin

```

```

    if Inside(Rect2.TopLeft, Rect1) then begin
        Result:=Rect2;
        Exit;
    end else begin
        Result.TopLeft:=Rect1.TopLeft;
        Result.BottomRight:=Rect2.BottomRight;
        Exit;
    end;
end;
if Inside(Rect1.TopLeft, Rect2) then begin
    if Inside(Rect1.BottomRight, Rect2) then begin
        Result:=Rect1;
        Exit;
    end else begin
        Result.TopLeft:=Rect1.TopLeft;
        Result.BottomRight:=Rect2.BottomRight;
        Exit;
    end;
end;
if Inside(Rect1.BottomRight, Rect2) then begin
    if Inside(Rect1.TopLeft, Rect2) then begin
        Result:=Rect1;
        Exit;
    end else begin
        Result.TopLeft:=Rect2.TopLeft;
        Result.BottomRight:=Rect1.BottomRight;
        Exit;
    end;
end;
ClearMem(Result, SizeOf(Result));
end;

procedure ArrangeMin(var R1, R2: Integer);
asm
    mov     ecx, [eax]
    cmp     ecx, [edx]
    jl     @@10
    xchg    ecx, [edx]
    mov     [eax], ecx
@@10:
end;

procedure ArrangeMax(var R1, R2: Integer);
asm
    mov     ecx, [eax]
    cmp     ecx, [edx]
    jg     @@10
    xchg    ecx, [edx]
    mov     [eax], ecx
@@10:
end;

function Sign(const Value:Integer):Integer; overload;
asm
    test   eax, eax
    jl    @@10
    jg    @@20
    ret
@@10:
    mov   eax, -1
    ret
@@20:
    mov   eax, 1
end;

```

```

function Sign(const Value:Extended):Extended;overload;
begin
  if Value<0 then Result:=-1.0 else
    if Value>0 then Result:=1.0 else Result:=0.0;
end;

procedure Swap(var R1,R2:Integer);overload;
asm
  mov  ecx, [eax]
  xchg ecx, [edx]
  mov  [eax], ecx
end;

function FindChars(const Source:TString;const
Chars:TSetChar;CurrentPosition:Integer=1;Direction:Integer=1):Integer;
Var
  // об'ява локальних типів даних
  i,len:Integer;
  Delta:Integer;
begin
  Result:=0;
  if Direction<0 then Delta:=-1 else Delta:=1;
  i:=CurrentPosition;
  len:=Length(Source);
  if Len=0 then Exit;
  repeat
    if Source[i] in Chars then begin
      Result:=i;
      Break;
    end;
    i:=i+Delta;
    if (i<1) or (i>len) then Break;
  until false;
end;

function LeftTrim(const Str:TString;const Chr:Char=chSpace):TString;
Var
  // об'ява локальних типів даних
  Count:Integer;
begin
  Result:=Str;
  Count:=0;
  while Length(Result)>0 do begin
    if Result[Count+1]=Chr then Inc(Count) else Break;
  end;
  if Count<>0 then Delete(Result,1,Count);
end;

function RightTrim(const Str:TString;const Chr:Char=chSpace):TString;
var Count:Integer;
begin
  Result:=Str;
  Count:=0;
  while Length(Result)>0 do begin
    if Result[Length(Result)-Count]=Chr then Inc(Count) else Break;
  end;
  if Count<>0 then SetLength(Result,Length(Result)-Count);
end;

function LeftExpand(const Str:TString; Count: Integer; const Chr:Char=chSpace):
TString;
Var
  // об'ява локальних типів даних
  i:Integer;
  PS, PD: PChar;

```

```

begin
  if Count<0 then Count:=0;
  SetString(Result, nil, Length(Str)+Count);
  PS:=@Str[1];
  PD:=@Result[Count+1];
  for i:=1 to Count do Result[i]:=Chr;
  Move(PS^, PD^, Length(Str));
end;

function RightExpand(const Str:TString; Count: Integer; const Chr:Char=chSpace):
TString;
Var
// об'ява локальних типів даних
  L: Integer;
begin
  if Count<0 then Count:=0;
  L:=GetLength(Str);
  SetString(Result, nil, L+Count);
  MoveMem(PChar(Str)^, PChar(Result)^, L);
  FillMem(PChar(@Result[L+1])^, Count, Byte(Chr));
end;

function TrimStr(const Str:TString;const Chr:Char=chSpace):TString;
begin
  if Str='' then Result:='' else Result:=LeftTrim(RightTrim(Str,Chr),Chr);
end;

function LeadTrim(const Str:TString; Count:Integer=1):TString;
begin
  if Count<0 then Count:=0;
  SetString(Result, PChar(IncPtr(PChar(Str), Count)), Length(Str)-Count);
end;

function TrailTrim(const Str:TString; Count:Integer=1):TString;
begin
  if Count<0 then Count:=0;
  SetString(Result, PChar(Str), Length(Str)-Count);
end;

function GetSubStr(const Str:TString;N:byte;Separator:Char=chSpace):TString;
Var
// об'ява локальних типів даних
  S: PChar;
  P1, P2: Integer;
begin
  P1:=CharEntryPos(Str, Separator, N-1);
  Inc(P1);
  S:=@Str[P1];
  P2:=CharEntryPos(S, Separator, 1);
  if P2=0 then P2:=Length(Str) else P2:=P1+P2-1;
  Result:=TrimStr(ReadSubStr(Str, P1, P2), Separator);
  if Result=Separator then Result:='';
end;

function ExtractStr(const Str:TString;N:byte):TString;
Var
// об'ява локальних типів даних
  P,I:Integer;
  S:TString;
begin
  S:=Str;
  for i:=1 to n-1 do begin
    P:=Pos(chSpace,S);
    S:=Copy(S,Succ(P),Length(S)-P);
    S:=LeftTrim(S);
  end;
end;

```

```

end;
P:=Pos(chSpace,S);
if P<>0 then Result:=Copy(S,1,Pred(P))
  else Result:=S;
end;

procedure ExtractStrings(Str: TString; List: TStrings;
  Separator: Char);
Var
// об'ява локальних типів даних
P1, P2: PChar;
begin
List.BeginUpdate;
try
List.Clear;
P1:=PChar(Str);
repeat
P2:=StrScan(P1, Separator);
SetByteValue(P2, 0);
List.Add(P1);
P1:=P2;
Inc(P1);
until P2 = nil;
finally
List.EndUpdate;
end;
end;

function RemoveChars(const Str:TString;const Chars:TSetChar):TString;
Var
// об'ява локальних типів даних
i:Integer;
begin
Result:='';
for i:=1 to Length(Str) do if not (Str[i] in Chars) then Result:=Result+Str[i];
end;

function ReplaceChar(const Str:TString;OldChar,NewChar:Char):TString;
Var
// об'ява локальних типів даних
i:Integer;
begin
Result:=Str;
for i:=1 to Length(Result) do if Result[i]=OldChar then Result[i]:=NewChar;
end;

function ReplaceStr(const Str:TString;const
  OldSubStr,NewSubStr:TString):TString;
Var
// об'ява локальних типів даних
P:Integer;
begin
Result:=Str;
P:=Pos(OldSubStr,Result);
if P<>0 then begin
Delete(Result,P,Length(OldSubStr));
Insert(NewSubStr,Result,P);
end;
end;

function __pos(SubStr, Str: TString; var P: Integer): Integer;
begin
P:=Pos(SubStr, Str);
Result:=P;
end;

```

```

function ReplaceStrAll(const Str: TString; const OldSubStr, NewSubStr: TString):
TString;
Var
// об'ява локальних типів даних
P: Integer;
Len: Integer;
begin
Result:=Str;
Len:=Length(OldSubStr);
while __pos(OldSubStr, Result, P)<>0 do begin
Delete(Result, P, Len);
Insert(NewSubStr, Result, P);
end;
end;

procedure CleanUp(var Str: TString);
asm
mov    eax, [eax]
test   eax, eax
jz     @@10
push   eax
call   GetLength
mov    edx, eax
pop    eax
mov    [eax-4], edx
@@10:
end;

procedure CleanUp(var Str:TString; DoTrim: LongBool);
begin
SetLength(Str, GetLength(Str));
CleanUp(Str);
if DoTrim then Str:=TrimStr(Str);
end;

function FillString(Chr:Char;Count:Integer):TString;
begin
SetString(Result, nil, Count);
FillChar(Pointer(Result)^, Count, Chr);
end;

function UpString(const Str:TString):TString;
begin
Result:=Str;
CharUpper(@Result[1]);
end;

function DnString(const Str:TString):TString;
begin
Result:=Str;
CharLower(@Result[1]);
end;

function GetChar(const Str:TString; Position:Integer=1):Char; register;
asm
push   edi
push   esi
mov    edi, edx
mov    esi, eax
call   GetLength
test   eax, eax
jnz    @@10
jmp    @@30
@@10:

```

```

    cmp    eax, edi
    jnb   @@20
    xor   eax, eax
    jmp   @@30
@@20:
    mov   eax, esi
    mov   edx, edi
    dec   edx
    call  ReadChar
@@30:
    pop   esi
    pop   edi
end;

```

```

function ReadChar (Ptr:Pointer;Offset:Integer):Char; register;
asm
    add   eax, edx
    mov   al, [eax]
end;

```

```

function UpChar (Ch:Char):Char; register;
asm
    and   eax, 000000FFh
    push  eax
    call  CharUpper
end;

```

```

function DnChar (Ch:Char):Char; register;
asm
    and   eax, 000000FFh
    push  eax
    call  CharLower
end;

```

```

function ReflectStr(const Str:TString):TString;
Var
// об'ява локальних типів даних
    i:Integer;
    len:Integer;
begin
    Len:=GetLength(Str);
    SetLength(Result,len);
    for i:=1 to Len do Result[i]:=Str[Len-i+1];
end;

```

```

function ReadSubStr(const Str:TString; Head, Tail:Integer):TString;
begin
    Result:=Copy(Str, Head, Tail-Head+1);
end;

```

```

function StrToFlt(const Str:TString;var Code:Integer):Extended;overload;
begin
    Val(PChar(Str), Result, Code);
end;

```

```

function StrToFlt(const Str:TString):Extended;overload;
Var
// об'ява локальних типів даних
    i:Integer;
begin
    Result:=StrToFlt(Str, i);
    if i<>0 then Result:=0;
end;

```

```

function FltToStr(const Value:Extended;
                  Precision:Integer=5):TString;
var
  P:Integer;
begin
  Result:=FloatToStrF(Value,ffGeneral,Precision,0);
  P:=Pos(', ',Result);
  if P<>0 then Result[P]:=chPoint;
  P:=Pos(DecimalSeparator,Result);
  if P<>0 then Result[P]:=chPoint;
end;

function BreakStr(const Str:TString;Len:Integer=64;AltChar:Char='\'):TString;
Var
  // об'ява локальних типів даних
  i,j:Integer;
  Alt:Boolean;
begin
  if Length(Str)<=Len then begin
    Result:=Str;
    Exit;
  end;
  Result:='';
  i:=0;
  repeat
    j:=i+Len;
    if j>Length(Str) then begin
      j:=Length(Str);
      Result:=Result+Copy(Str,i+1,j-i);
      Exit;
    end;
    Alt:=False;
    while Str[j]<>chSpace do begin
      Dec(j);
      if j=i then begin
        Alt:=True;
        Break;
      end;
    end;
    if Alt then begin
      j:=i+Len;
      if j>Length(Str) then begin
        j:=Length(Str);
        Result:=Result+Copy(Str,i+1,j-i);
        Exit;
      end;
      while Str[j]<>AltChar do begin
        Dec(j);
        if j=i then begin
          j:=i+Len;
          Break;
        end;
      end;
    end;
    Result:=Result+Copy(Str,i+1,j-i)+#13#10;
    i:=j;
  until i>=Length(Str);
end;

function ValidInt(const Value:TString):LongBool;
var
  i,Code:Integer;
begin
  Val(Value,i,Code);
  Hole(i);

```

```

    Result:=Code=0;
end;

function ValidFloat(const Value:TString):LongBool;
Var
// об'ява локальних типів даних
    i:Double;
    Code:Integer;
begin
    Val(Value,i,Code);
    Hole(i);
    Result:=Code=0;
end;

function ValidFloatINF(const Value:TString): LongBool;
var
    R: Double;
    Code:Integer;
begin
    Val(Value, R, Code);
    Hole(Code);
    Result:=Infinity(R)=0;
end;

function ValidateFloat(const Value:TString):TString;
Var
// об'ява локальних типів даних
    P:Integer;
begin
    Result:=Value;
    P:=Pos(DecimalSeparator,Result);
    if P<>0 then Result[P]:=chPoint;
    P:=Pos(chComma,Result);
    if P<>0 then Result[p]:=chPoint;
    if not ValidFloat(Result) then Result:='';
end;

function Join(const Str1, Str2: TString): TString;
begin
    Result:='';
    if not IsEmptyStr(Str1) then Result:=PChar(@Str1[1]);
    if not IsEmptyStr(Str2) then Result:=Result+PChar(@Str2[1]);
end;

function LastChar(const Str:TString):Char;
begin
    if Str='' then Result:=chNull else Result:=Str[Length(Str)];
end;

function NextChar(const Str:TString;Pos:Integer;Passed:Char=chSpace):Char;
begin
    Result:=NextChar(Str,Pos,[Passed]);
end;

function PrevChar(const Str:TString;Pos:Integer;Passed:Char=chSpace):Char;
begin
    Result:=PrevChar(Str,Pos,[Passed]);
end;

function NextChar(const Str:TString;Pos:Integer;Passed:TSetChar):Char;overload;
var i:Integer;
begin
    Result:=#0;
    for i:=Pos+1 to Length(Str) do if not (Str[i] in Passed) then begin
        Result:=Str[i];
    end;
end;

```

```

    Break;
  end;
end;

function PrevChar(const Str:TString;Pos:Integer;Passed:TSetChar):Char;overload;
Var
// об'ява локальних типів даних
  i:Integer;
begin
  Result:=#0;
  for i:=Pos-1 downto 1 do if not (Str[i] in Passed) then begin
    Result:=Str[i];
    Break;
  end;
end;

procedure AddString(var Str:TString; const Value:TString);
begin
  CleanUp(Str);
  Str:=Str+Value;
end;

function AdjustLength(Str: TString; Len: Integer; Ch: Char = chSpace): TString;
Var
// об'ява локальних типів даних
  L, N: Integer;
  S1: TString;
begin
  L:=GetStrLen(Str);
  if L<Len then begin
    N:=Len - L;
    SetString(S1, nil, N);
    FillMem(PChar(S1)^, N, Ord(Ch));
    Result:=Str+S1;
  end else Result:=Str;
end;

function CopyToBuf(const Source:TString; Buf:PChar; Size:Integer):LongBool;
var
  Len: Integer;
begin
  Len:=GetLength(Source)+1;
  if Len>Size then begin
    Result:=False;
    Buf^:=#0;
  end else begin
    if not IsEmptyStr(Source) then MoveMem(PChar(Source)^, Buf^, Len)
    else ClearMem(Buf^, Size);
    Result:=True;
  end;
end;

function _GetTempDirectory: TString;
var
  Buf: array [0..MAX_PATH-1] of AnsiChar;
begin
  GetTempPath(SizeOf(Buf), @Buf);
  Result:=IncludeTrailingBackslash(PChar(@Buf));
end;
var
  TmpDir: TString = '';
  GetTempDirectoryFirstCall: Boolean = True;

function GetTempDirectory: TString;
begin

```

```

if GetTempDirectoryFirstCall then begin
  TmpDir:=_GetTempDirectory;
  GetTempDirectoryFirstCall:=False;
end;
Result:=TmpDir;
end;

function GetTempDOC_PACK(const Prefix: TString): TString;
var
  Buf: array [0..MAX_PATH-1] of AnsiChar;
begin
  GetTempDOC_PACKName(PChar(GetTempDirectory), PChar(Prefix), 0, @Buf);
  Result:=PChar(@Buf);
end;
var
  Checked: Boolean = False;
  Embedded: Boolean = False;

function CheckAutomation: Boolean;
begin
  if not Checked then begin
    Embedded:=MatchString(Parameters, ['-EMBEDDING', '/EMBEDDING'])<>0;
    Checked:=True;
  end;
  Result:=Embedded;
end;

function ExeName:TString;
Var
  // об'ява локальних типів даних
  S: PChar;
  P: Integer;
begin
  Result:=GetCommandLine;
  S:=@Result[2];
  P:=Pos('"', S);
  Result:=ReadSubStr(S, 1, P-1);
end;

function ExePath:TString;
begin
  Result:=ExtractDOC_PACKPath(ExeName);
end;

function WinME: Boolean;
begin
  Result:=(_Win32Platform = VER_PLATFORM_WIN32_WINDOWS) and
    (( _Win32MajorVersion>4) or (( _Win32MajorVersion = 4) and ( _Win32MinorVersion
    >= 90)));
end;

function WinXP: Boolean;
begin
  Result := ( _Win32Platform = VER_PLATFORM_WIN32_NT) and
    (( _Win32MajorVersion)>5) or (( _Win32MajorVersion = 5) and ( _Win32MinorVersion
    >= 1));
end;
Var
  // об'ява локальних типів даних
  GetOperatingSystemFirstCall: Boolean = True;
  GetOperatingSystemResult: TOperatingSystem;

function GetOperatingSystem: TOperatingSystem;
begin
  if GetOperatingSystemFirstCall then begin

```

```

GetOperatingSystemResult:=UndefinedWindows;
case _Win32Platform of
VER_PLATFORM_WIN32S: GetOperatingSystemResult:=Windows3x;
VER_PLATFORM_WIN32_WINDOWS: begin
  if _Win32MajorVersion = 4 then begin
    if _Win32MinorVersion >= 0 then GetOperatingSystemResult:=Windows95;
    if _Win32MinorVersion >=10 then GetOperatingSystemResult:=Windows98;
    if _Win32MinorVersion >=90 then GetOperatingSystemResult:=WindowsME;
  end;
end;
VER_PLATFORM_WIN32_NT: begin
  if _Win32MajorVersion<=4 then GetOperatingSystemResult:=WindowsNT;
  if _Win32MajorVersion = 5 then begin
    if _Win32MinorVersion >= 0 then GetOperatingSystemResult:=Windows2000;
    if _Win32MinorVersion >= 1 then GetOperatingSystemResult:=WindowsXP;
  end;
end;
end;
GetOperatingSystemFirstCall:=False;
end;
Result:=GetOperatingSystemResult;
end;

function TrayWnd: HWND;
begin
  Result:=FindWindow('Shell_TrayWnd','');
end;

function _GetLocale: Integer;
Var
// об'ява локальних типів даних
Translation: PWord;
Buffer: Pointer;
Size, Len, Handle: DWORD;
Name: TString;
begin
  Name:=InstanceName;
  Size:=GetDOC_PACKVersionInfoSize(PChar(Name), Handle);
  if Size = 0 then Result:=GetLocale else begin
    Buffer:=AllocateMem(Size);
    try
      GetDOC_PACKVersionInfo(PChar(Name), Handle, Size, Buffer);
      VerQueryValue(Buffer, '\\DOC \\Translation', Pointer(Translation), Len);
      Result:=Translation^;
    finally
      DeallocateMem(Buffer);
    end;
  end;
end;

function LangIDToCharset(LangID: Integer): Byte;
Var
// об'ява локальних типів даних
I: byte;
begin
  Result:=DEFAULT_CHARSET;
  if LangID = 0 then LangID:=_GetLocale;
  for i:=0 to LangCount do if Lo(LangID) = LangIDToCharsetInfo[i].LangID then
  begin
    Result:=LangIDToCharsetInfo[i].Charset;
    Break;
  end;
  if LangID = $0C1A then Result:=RUSSIAN_CHARSET;
end;

```

```

function ExitWindows(uFlags: UINT): BOOL;
Var
// об'ява локальних типів даних
ProcessHandle: THandle;
TokenHandle: THandle;
Luid: Int64;
Tkp: TTokenPrivileges;
BufferNeeded: DWORD;
begin
  if WinNT then begin
    ProcessHandle:=GetCurrentProcess;
    OpenProcessToken(ProcessHandle, TOKEN_ADJUST_PRIVILEGES or TOKEN_QUERY,
TokenHandle);
    LookupPrivilegeValue(nil, 'SeShutdownPrivilege', Luid);
    Tkp.PrivilegeCount:=1;
    Tkp.Privileges[0].Luid:=Luid;
    Tkp.Privileges[0].Attributes:=SE_PRIVILEGE_ENABLED;
    AdjustTokenPrivileges(TokenHandle, FALSE, Tkp, 0, nil, BufferNeeded);
  end;
  Result:=ExitWindowsEx(uFlags, $FFFF);
end;

procedure RemoveDirectories(const Path: TString);
Var
// об'ява локальних типів даних
S: TString;
Len, P, i: Integer;
begin
  S:=ExcludeTrailingBackslash(Path);
  repeat
    if not RemoveDirectory(PChar(S)) then Break;
    Len:=Length(S);
    P:=0;
    for i:=Len downto 0 do if S[i] = '\' then begin
      P:=i;
      Break;
    end;
    if P = 0 then Break;
    S:=TrailTrim(S, Len-P+1);
  until False;
end;

function LocalHandle; external kernel32 name 'LocalHandle';

function ExeVersion: TString;
begin
  Result:=VersionToString(DOC_PACKVersion);
end;

function InstanceName:TString;
var
  Buf: array [0..MAX_PATH-1] of AnsiChar;
begin
  GetModuleDOC_PACKName(hInstance, Buf, MAX_PATH);
  Result:=Buf;
end;

function InstancePath:TString;
begin
  Result:=ExtractDOC_PACKPath(InstanceName);
end;

function DOC_PACKVersion(const DOC_PACKName: TString = ''): TDVersion;
Var
// об'ява локальних типів даних

```

```

S: TString;
hMem: HGLOBAL;
Buf: pointer;
BufSize, Len, dwHandle: DWORD;
VerInfo: PVSFixedDOC_PACKInfo;
pszName: PAnsiChar;
begin
  FillChar(Result, SizeOf(Result), 0);
  S:=DOC_PACKName;
  CleanUp(S, True);
  if IsEmptyStr(S) then S:=ParamStr(0);
  pszName:=@S[1];
  BufSize:=GetDOC_PACKVersionInfoSize(pszName, dwHandle);
  if BufSize<>0 then begin
    hMem:=GlobalAlloc(GHND, BufSize);
    if hMem = 0 then OutOfMemoryError;
    Buf:=GlobalLock(hMem);
    if Buf=nil then OutOfMemoryError;
    GetDOC_PACKVersionInfo(pszName, dwHandle, BufSize, Buf);
    VerQueryValue(Buf, '\', pointer(VerInfo), Len);
    with VerInfo^ do begin
      Result.HiVersion:=HiWord(dwDOC_PACKVersionMS);
      Result.LoVersion:=LoWord(dwDOC_PACKVersionMS);
      Result.Release:=HiWord(dwDOC_PACKVersionLS);
      Result.Build:=LoWord(dwDOC_PACKVersionLS);
    end;
    GlobalUnlock(hMem);
    GlobalFree(hMem);
  end else Result.HiVersion:=-1;
end;

function ComCtlVersion: TDVersion;
begin
  Result:=DOC_PACKVersion('COMCTL32.DLL');
end;

function IsDebug(const DOC_PACKName: TString): LongBool;
Var
  // об'ява локальних типів даних
  S: TString;
  hMem: HGLOBAL;
  Buf: pointer;
  BufSize, Len, dwHandle: DWORD;
  VerInfo: PVSFixedDOC_PACKInfo;
  pszName: PAnsiChar;
begin
  Result:=False;
  FillChar(Result, SizeOf(Result), 0);
  S:=DOC_PACKName;
  CleanUp(S, True);
  if IsEmptyStr(S) then S:=ParamStr(0);
  pszName:=@S[1];
  BufSize:=GetDOC_PACKVersionInfoSize(pszName, dwHandle);
  if BufSize<>0 then begin
    hMem:=GlobalAlloc(GHND, BufSize);
    if hMem = 0 then OutOfMemoryError;
    Buf:=GlobalLock(hMem);
    if Buf=nil then OutOfMemoryError;
    GetDOC_PACKVersionInfo(pszName, dwHandle, BufSize, Buf);
    VerQueryValue(Buf, '\', pointer(VerInfo), Len);
    Result:=(VerInfo.dwDOC_PACKFlags and VS_FF_DEBUG) <> 0;
    GlobalUnlock(hMem);
    GlobalFree(hMem);
  end;
end;
end;

```

```

var
  IsDebugValue: Integer = Integer($8000000);

function IsDebug: LongBool; overload;
begin
  if IsDebugValue = Integer ($80000000) then IsDebugValue:=Integer(IsDebug(''));
  Result:=LongBool(IsDebugValue);
end;

procedure GetWindowSize(Handle: HWND; var Size: TSize);
var
  R: TRect;
begin
  GetWindowRect(Handle, R);
  with R, Size do begin
    cx:=Right-Left;
    cy:=Bottom-Top;
  end;
end;

procedure PressKey(VKey: Byte);
begin
  keybd_event(VKey, 0, 0, 0);
  keybd_event(VKey, 0, KEYEVENTF_KEYUP, 0);
end;

function ForceDirectories(Dir: TString): Boolean;
begin
  try
    Result := True;
    if Length(Dir) = 0 then Abort;
    Dir := ExcludeTrailingBackslash(Dir);
    if (Length(Dir) < 3) or PathExists(Dir)
      or (ExtractDOC_PACKPath(Dir) = Dir) then Exit;
    Result := ForceDirectories(ExtractDOC_PACKPath(Dir)) and CreateDir(Dir);
  except
    on EAbort do Result:=False;
    else raise;
  end;
end;

function StringToVersion(const Str: TString): TDVersion;
Var
  // об'ява локальних типів даних
  S, SH, SL, SR, SB: TString;
  Code: Integer;
  Count: Integer;
begin
  S:=Str;
  CleanUp(S, True);
  S:=ReplaceChar(S, ',', '.');
  Count:=CharCount(S, '.')+1;
  SH:='0'; SL:='0'; SR:='0'; SB:='0';
  if Count>=1 then SH:=GetSubStr(S, 1, '.');
  if Count>=2 then SL:=GetSubStr(S, 2, '.');
  if Count>=3 then SR:=GetSubStr(S, 3, '.');
  if Count>=4 then SB:=GetSubStr(S, 4, '.');
  with Result do begin
    Val(SH, HiVersion, Code); if Code<>0 then HiVersion:=0;
    Val(SL, LoVersion, Code); if Code<>0 then LoVersion:=0;
    Val(SR, Release, Code); if Code<>0 then Release:=0;
    Val(SB, Build, Code); if Code<>0 then Build:=0;
  end;
end;

```

```

function VersionToString(const Ver: TDVersion): TString;
begin
  with Ver do Result:=Format('%d.%d.%d.%d', [HiVersion, LoVersion, Release,
  Build]);
end;

function Version(HiVersion, LoVersion: Integer;
  Release: Integer = 0; Build: Integer = 0): TDVersion; overload;
begin
  Result.HiVersion:=HiVersion;
  Result.LoVersion:=LoVersion;
  Result.Release:=Release;
  Result.Build:=Build;
end;

function LoadResStr(Instance:THandle;ID:Cardinal):TString;
begin
  SetLength(Result,512);
  LoadString(Instance,ID,@Result[1],512);
  CleanUp(Result);
end;

function LoadResStr(ID: Cardinal): TString; overload;
begin
  Result:=LoadResStr(hInstance, ID);
end;

function LoadDLL(const Path:TString):THandle;
begin
  Result:=LoadLibrary(PChar(Path));
end;

function GetDLLProc(Handle:THandle;const ProcName:TString):Pointer;
begin
  Result:=GetProcAddress(Handle,PChar(ProcName));
end;
Var
// об'ява локальних типів даних
OSVersionInfo_Initialized: Boolean = False;
OSVersionInfo: TOSVersionInfo;

procedure Initialize_OSVersionInfo;
begin
  if not OSVersionInfo_Initialized then begin
    GetVersionEx(OSVersionInfo);
    OSVersionInfo_Initialized:=True;
  end;
end;

function _Win32Platform: Integer;
begin
  Initialize_OSVersionInfo;
  Result:=OSVersionInfo.dwPlatformId;
end;

function _Win32MajorVersion: Integer;
begin
  Initialize_OSVersionInfo;
  Result:=OSVersionInfo.dwMajorVersion;
end;

function _Win32MinorVersion: Integer;
begin
  Initialize_OSVersionInfo;
  Result:=OSVersionInfo.dwMinorVersion;
end;

```

```
end;

function WinNT: Boolean;
begin
  Result := _Win32Platform = VER_PLATFORM_WIN32_NT;
end;

function Win2K: Boolean;
begin
  Result := (_Win32MajorVersion > 4) and (_Win32Platform =
VER_PLATFORM_WIN32_NT);
end;

end.
```

Кафедра КБПЗ – 2021 рік

WINDATA.PAS - ФАЙЛ ФОРМИ

```

unit WinData;
// модуль WinData

interface

uses Windows, Messages, Classes, Controls, Forms, Graphics, StdCtrls,
    ExtCtrls, CommCtrl;

type

    TUkraineEBActionLink = class(TControlActionLink)
    protected
        FClient: TUkraineEB;
        procedure AssignClient(AClient: TObject); override;
        function IsCheckedLinked: Boolean; override;
        function IsGroupIndexLinked: Boolean; override;
        procedure SetGroupIndex(Value: Integer); override;
        procedure SetChecked(Value: Boolean); override;
    end;

    TUkraineEB = class(TGraphicControl)
    private
        FGroupIndex: Integer;
        FGlyph: Pointer;
        FDown: Boolean;
        FDragging: Boolean;
        FAllowAllUp: Boolean;
        FLayout: TButtonLayout;
        FSpacing: Integer;
        FTransparent: Boolean;
        FMargin: Integer;
        FFlat: Boolean;
        FMouseInControl: Boolean;
        procedure GlyphChanged(Sender: TObject);
        procedure UpdateExclusive;
        function GetGlyph: TBitmap;
        procedure SetGlyph(Value: TBitmap);
        function GetNumGlyphs: TNumGlyphs;
        procedure SetNumGlyphs(Value: TNumGlyphs);
        procedure SetDown(Value: Boolean);
        procedure SetLayout(Value: TButtonLayout);
        procedure SetSpacing(Value: Integer);
        procedure SetTransparent(Value: Boolean);
        procedure SetMargin(Value: Integer);
        procedure UpdateTracking;
    protected
        FState: TButtonState;
        procedure ActionChange(Sender: TObject; CheckDefaults: Boolean); override;
        function GetActionLinkClass: TControlActionLinkClass; override;
        function GetPalette: HPALETTE; override;
        procedure Loaded; override;
        procedure Paint; override;
        property MouseInControl: Boolean read FMouseInControl;
    public
        constructor Create(AOwner: TComponent); override;
        destructor Destroy; override;
        procedure Click; override;
    published
        property Action;
        property AllowAllUp: Boolean read FAllowAllUp write SetAllowAllUp default
            False;
        property Anchors;
        property BiDiMode;
        property Constraints;

```

```

property GroupIndex: Integer read FGroupIndex write SetGroupIndex default 0;
property Down: Boolean read FDown write SetDown default False;
property Caption;
property Enabled;
property Flat: Boolean read FFlat write SetFlat default False;
property Font;
property Glyph: TBitmap read GetGlyph write SetGlyph;
property ParentFont;
property ParentShowHint;
property ParentBiDiMode;
property PopupMenu;
property ShowHint;
property Spacing: Integer read FSpacing write SetSpacing default 4;
property Transparent: Boolean read FTransparent write SetTransparent default
True;
property Visible;
property OnClick;
property OnDblClick;
property OnMouseDown;
property OnMouseMove;
property OnMouseUp;
end;

TBitBtnKind = (bkCustom, bkOK, bkCancel, bkHelp, bkYes, bkNo, bkClose,
bkAbort, bkRetry, bkIgnore, bkAll);

TBitBtn = class(TButton)
private
    FCanvas: TCanvas;
    FGlyph: Pointer;
    FStyle: TButtonStyle;
    FKind: TBitBtnKind;
    FLayout: TButtonLayout;
    FSpacing: Integer;
    FMargin: Integer;
    IsFocused: Boolean;
    FModifiedGlyph: Boolean;
    FMouseInControl: Boolean;
    procedure DrawItem(const DrawItemStruct: TDrawItemStruct);
    procedure SetGlyph(Value: TBitmap);
    function GetGlyph: TBitmap;
    function GetNumGlyphs: TNumGlyphs;
    procedure SetNumGlyphs(Value: TNumGlyphs);
    procedure GlyphChanged(Sender: TObject);
    function IsCustom: Boolean;
    function IsCustomCaption: Boolean;
    procedure SetStyle(Value: TButtonStyle);
    procedure SetKind(Value: TBitBtnKind);
    function GetKind: TBitBtnKind;
    procedure SetLayout(Value: TButtonLayout);
    procedure SetSpacing(Value: Integer);
    procedure SetMargin(Value: Integer);
protected
    procedure ActionChange(Sender: TObject; CheckDefaults: Boolean); override;
    procedure CreateHandle; override;
    procedure CreateParams(var Params: TCreateParams); override;
    function GetPalette: HPALETTE; override;
    procedure SetButtonStyle(ADefault: Boolean); override;
public
    constructor Create(AOwner: TComponent); override;
    destructor Destroy; override;
    procedure Click; override;
published
    property Action;
    property Anchors;

```

```

property BiDiMode;
property Cancel stored IsCustom;
property Caption stored IsCustomCaption;
property Constraints;
property Default stored IsCustom;
property Enabled;
property Glyph: TBitmap read GetGlyph write SetGlyph stored IsCustom;
property Kind: TBitBtnKind read GetKind write SetKind default bkCustom;
property Layout: TButtonLayout read FLayout write SetLayout default
blGlyphLeft;
property Margin: Integer read FMargin write SetMargin default -1;
property ModalResult stored IsCustom;
property NumGlyphs: TNumGlyphs read GetNumGlyphs write SetNumGlyphs stored
IsCustom default 1;
property ParentShowHint;
property ParentBiDiMode;
property ShowHint;
property Style: TButtonStyle read FStyle write SetStyle default
bsAutoDetect;
property Spacing: Integer read FSpacing write SetSpacing default 4;
property TabOrder;
property TabStop;
property Visible;
property OnEnter;
property OnExit;
end;

function DrawButtonFace(Canvas: TCanvas; const Client: TRect;
    BevelWidth: Integer; Style: TButtonStyle; IsRounded, IsDown,
    IsFocused: Boolean): TRect;

implementation

uses Consts, SysUtils, ActnList, ImgList, Themes;

var
    BitBtnGlyphs: array[TBitBtnKind] of TBitmap;

function GetBitBtnGlyph(Kind: TBitBtnKind): TBitmap;
begin
    if BitBtnGlyphs[Kind] = nil then
        begin
            BitBtnGlyphs[Kind]:=TBitmap.Create;
            BitBtnGlyphs[Kind].LoadFromResourceName(HInstance, BitBtnResNames[Kind]);
        end;
    Result:=BitBtnGlyphs[Kind];
end;

type
    TGlyphList = class(TImageList)
    private
        Used: TBits;
        FCount: Integer;
        function AllocateIndex: Integer;
    public
        constructor CreateSize(AWidth, AHeight: Integer);
        destructor Destroy; override;
        function AddMasked(Image: TBitmap; MaskColor: TColor): Integer;
        procedure Delete(Index: Integer);
        property Count: Integer read FCount;
    end;

    TGlyphCache = class

```

```

private
  GlyphLists: TList;
public
  constructor Create;
  destructor Destroy; override;
  function GetList(AWidth, AHeight: Integer): TGlyphList;
  procedure ReturnList(List: TGlyphList);
  function Empty: Boolean;
end;

TButtonGlyph = class
private
  FOriginal: TBitmap;
  FGlyphList: TGlyphList;
  FIndexes: array[TButtonState] of Integer;
  FTransparentColor: TColor;
  FNumGlyphs: TNumGlyphs;
  FOnChange: TNotifyEvent;
  procedure GlyphChanged(Sender: TObject);
  procedure SetGlyph(Value: TBitmap);
  procedure SetNumGlyphs(Value: TNumGlyphs);
  procedure Invalidate;
  function CreateButtonGlyph(State: TButtonState): Integer;
  procedure DrawButtonGlyph(Canvas: TCanvas; const GlyphPos: TPoint;
    State: TButtonState; Transparent: Boolean);
  procedure DrawButtonText(Canvas: TCanvas; const Caption: string;
    TextBounds: TRect; State: TButtonState; BiDiFlags: Longint);
  procedure CalcButtonLayout(Canvas: TCanvas; const Client: TRect;
    const Offset: TPoint; const Caption: string; Layout: TButtonLayout;
    Margin, Spacing: Integer; var GlyphPos: TPoint; var TextBounds: TRect;
    BiDiFlags: Longint);
public
  constructor Create;
  destructor Destroy; override;
  function Draw(Canvas: TCanvas; const Client: TRect; const Offset: TPoint;
    const Caption: string; Layout: TButtonLayout; Margin, Spacing: Integer;
    State: TButtonState; Transparent: Boolean; BiDiFlags: Longint): TRect;
  property Glyph: TBitmap read FOriginal write SetGlyph;
  property NumGlyphs: TNumGlyphs read FNumGlyphs write SetNumGlyphs;
  property OnChange: TNotifyEvent read FOnChange write FOnChange;
end;

constructor TGlyphList.CreateSize(AWidth, AHeight: Integer);
begin
  inherited CreateSize(AWidth, AHeight);
  Used:=TBits.Create;
end;

destructor TGlyphList.Destroy;
begin
  Used.Free;
  inherited Destroy;
end;

function TGlyphList.AllocateIndex: Integer;
begin
  Result:=Used.OpenBit;
  if Result >= Used.Size then
  begin
    Result:=inherited Add(nil, nil);
    Used.Size:=Result + 1;
  end;
  Used[Result]:=True;
end;

```

```

function TGlyphList.AddMasked(Image: TBitmap; MaskColor: TColor): Integer;
begin
    Result:=AllocateIndex;
    ReplaceMasked(Result, Image, MaskColor);
    Inc(FCount);
end;

procedure TGlyphList.Delete(Index: Integer);
begin
    if Used[Index] then
        begin
            Dec(FCount);
            Used[Index]:=False;
        end;
end;

constructor TGlyphCache.Create;
begin
    inherited Create;
    GlyphLists:=TList.Create;
end;

destructor TGlyphCache.Destroy;
begin
    GlyphLists.Free;
    inherited Destroy;
end;

function TGlyphCache.GetList(AWidth, AHeight: Integer): TGlyphList;
var
    I: Integer;
begin
    for I:=GlyphLists.Count - 1 downto 0 do
        begin
            Result:=GlyphLists[I];
            with Result do
                if (AWidth = Width) and (AHeight = Height) then Exit;
            end;
            Result:=TGlyphList.CreateSize(AWidth, AHeight);
            GlyphLists.Add(Result);
        end;
end;

procedure TGlyphCache.ReturnList(List: TGlyphList);
begin
    if List = nil then Exit;
    if List.Count = 0 then
        begin
            GlyphLists.Remove(List);
            List.Free;
        end;
end;

function TGlyphCache.Empty: Boolean;
begin
    Result:=GlyphLists.Count = 0;
end;

var
    GlyphCache: TGlyphCache = nil;
    ButtonCount: Integer = 0;

constructor TButtonGlyph.Create;
var
    I: TButtonState;
begin

```

```

    inherited Create;
    FOriginal:=TBitmap.Create;
    FOriginal.OnChange:=GlyphChanged;
    FTransparentColor:=clOlive;
    FNumGlyphs:=1;
    for I:=Low(I) to High(I) do
        FIndexs[I]:=-1;
    if GlyphCache = nil then GlyphCache:=TGlyphCache.Create;
end;

destructor TButtonGlyph.Destroy;
begin
    FOriginal.Free;
    Invalidate;
    if Assigned(GlyphCache) and GlyphCache.Empty then
        begin
            GlyphCache.Free;
            GlyphCache:=nil;
        end;
    inherited Destroy;
end;

procedure TButtonGlyph.Invalidate;
var
    I: TButtonState;
begin
    for I:=Low(I) to High(I) do
        begin
            if FIndexs[I] <> -1 then FGlyphList.Delete(FIndexs[I]);
            FIndexs[I]:=-1;
        end;
    GlyphCache.ReturnList(FGlyphList);
    FGlyphList:=nil;
end;

procedure TButtonGlyph.GlyphChanged(Sender: TObject);
begin
    if Sender = FOriginal then
        begin
            FTransparentColor:=FOriginal.TransparentColor;
            Invalidate;
            if Assigned(FOnChange) then FOnChange(Self);
        end;
end;

procedure TButtonGlyph.SetGlyph(Value: TBitmap);
var
    Glyphs: Integer;
begin
    Invalidate;
    FOriginal.Assign(Value);
    if (Value <> nil) and (Value.Height > 0) then
        begin
            FTransparentColor:=Value.TransparentColor;
            if Value.Width mod Value.Height = 0 then
                begin
                    Glyphs:=Value.Width div Value.Height;
                    if Glyphs > 4 then Glyphs:=1;
                    SetNumGlyphs(Glyphs);
                end;
        end;
end;

procedure TButtonGlyph.SetNumGlyphs(Value: TNumGlyphs);
begin

```

```

    if (Value <> FNumGlyphs) and (Value > 0) then
    begin
        Invalidate;
        FNumGlyphs:=Value;
        GlyphChanged(Glyph);
    end;
end;

function TButtonGlyph.CreateButtonGlyph(State: TButtonState): Integer;
const
    ROP_DSPDxax = $00E20746;
var
    TmpImage, DDB, MonoBmp: TBitmap;
    IWidth, IHeight: Integer;
    IRect, ORect: TRect;
    I: TButtonState;
    DestDC: HDC;
begin
    if (State = bsDown) and (NumGlyphs < 3) then State:=bsUp;
    Result:=FIndexs[State];
    if Result <> -1 then Exit;
    if (FOriginal.Width or FOriginal.Height) = 0 then Exit;
    IWidth:=FOriginal.Width div FNumGlyphs;
    IHeight:=FOriginal.Height;
    if FGlyphList = nil then
    begin
        if GlyphCache = nil then GlyphCache:=TGlyphCache.Create;
        FGlyphList:=GlyphCache.GetList(IWidth, IHeight);
    end;
    TmpImage:=TBitmap.Create;
    try
        TmpImage.Width:=IWidth;
        TmpImage.Height:=IHeight;
        IRect:=Rect(0, 0, IWidth, IHeight);
        TmpImage.Canvas.Brush.Color:=clBtnFace;
        TmpImage.Palette:=CopyPalette(FOriginal.Palette);
        I:=State;
        if Ord(I) >= NumGlyphs then I:=bsUp;
        ORect:=Rect(Ord(I) * IWidth, 0, (Ord(I) + 1) * IWidth, IHeight);
        case State of
            bsUp, bsDown,
            bsExclusive:
        begin
            TmpImage.Canvas.CopyRect(IRect, FOriginal.Canvas, ORect);
            if FOriginal.TransparentMode = tmFixed then
                FIndexs[State]:=FGlyphList.AddMasked(TmpImage, FTransparentColor)
            else
                FIndexs[State]:=FGlyphList.AddMasked(TmpImage, clDefault);
        end;
        bsDisabled:
        begin
            MonoBmp:=nil;
            DDB:=nil;
        try
            MonoBmp:=TBitmap.Create;
            DDB:=TBitmap.Create;
            DDB.Assign(FOriginal);
            DDB.HandleType:=bmDDB;
            if NumGlyphs > 1 then
                with TmpImage.Canvas do
                begin
                    CopyRect(IRect, DDB.Canvas, ORect);
                    MonoBmp.Monochrome:=True;
                    MonoBmp.Width:=IWidth;
                    MonoBmp.Height:=IHeight;

```

```

DDB.Canvas.Brush.Color:=clWhite;
MonoBmp.Canvas.CopyRect(IRect, DDB.Canvas, ORect);
Brush.Color:=clBtnHighlight;
DestDC:=Handle;
SetTextColor(DestDC, clBlack);
SetBkColor(DestDC, clWhite);
BitBlt(DestDC, 0, 0, IWidth, IHeight,
        MonoBmp.Canvas.Handle, 0, 0, ROP_DSPDxax);
DDB.Canvas.Brush.Color:=clGray;
MonoBmp.Canvas.CopyRect(IRect, DDB.Canvas, ORect);
Brush.Color:=clBtnShadow;
DestDC:=Handle;
SetTextColor(DestDC, clBlack);
SetBkColor(DestDC, clWhite);
BitBlt(DestDC, 0, 0, IWidth, IHeight,
        MonoBmp.Canvas.Handle, 0, 0, ROP_DSPDxax);
DDB.Canvas.Brush.Color:=ColorToRGB(FTransparentColor);
MonoBmp.Canvas.CopyRect(IRect, DDB.Canvas, ORect);
Brush.Color:=clBtnFace;
DestDC:=Handle;
SetTextColor(DestDC, clBlack);
SetBkColor(DestDC, clWhite);
BitBlt(DestDC, 0, 0, IWidth, IHeight,
        MonoBmp.Canvas.Handle, 0, 0, ROP_DSPDxax);
end
else
begin
with MonoBmp do
begin
Assign(FOriginal);
HandleType:=bmDDB;
Canvas.Brush.Color:=clBlack;
Width:=IWidth;
if Monochrome then
begin
Canvas.Font.Color:=clWhite;
Monochrome:=False;
Canvas.Brush.Color:=clWhite;
end;
Monochrome:=True;
end;
with TmpImage.Canvas do
begin
Brush.Color:=clBtnFace;
FillRect(IRect);
Brush.Color:=clBtnHighlight;
SetTextColor(Handle, clBlack);
SetBkColor(Handle, clWhite);
BitBlt(Handle, 1, 1, IWidth, IHeight,
        MonoBmp.Canvas.Handle, 0, 0, ROP_DSPDxax);
Brush.Color:=clBtnShadow;
SetTextColor(Handle, clBlack);
SetBkColor(Handle, clWhite);
BitBlt(Handle, 0, 0, IWidth, IHeight,
        MonoBmp.Canvas.Handle, 0, 0, ROP_DSPDxax);
end;
end;
finally
DDB.Free;
MonoBmp.Free;
end;
FIndexes[State]:=FGlyphList.AddMasked(TmpImage, clDefault);
end;
end;
finally

```

```

TmpImage.Free;
end;
Result:=FIndexs[State];
end;

procedure TButtonGlyph.DrawButtonGlyph(Canvas: TCanvas; const GlyphPos: TPoint;
State: TButtonState; Transparent: Boolean);
var
  Index: Integer;
  Details: TThemedElementDetails;
  R: TRect;
  Button: TThemedButton;
begin
  if FOriginal = nil then Exit;
  if (FOriginal.Width = 0) or (FOriginal.Height = 0) then Exit;
  Index:=CreateButtonGlyph(State);
  with GlyphPos do
  begin
    if ThemeServices.ThemesEnabled then
    begin
      R.TopLeft:=GlyphPos;
      R.Right:=R.Left + FOriginal.Width div FNumGlyphs;
      R.Bottom:=R.Top + FOriginal.Height;
      case State of
        bsDisabled:
          Button:=tbPushButtonDisabled;
        bsDown,
        bsExclusive:
          Button:=tbPushButtonPressed;
      else
        // bsUp
        Button:=tbPushButtonNormal;
      end;
      Details:=ThemeServices.GetElementDetails(Button);
      ThemeServices.DrawIcon(Canvas.Handle, Details, R, FGlyphList.Handle,
Index);
    end
    else
      if Transparent or (State = bsExclusive) then
      begin
        ImageList_DrawEx(FGlyphList.Handle, Index, Canvas.Handle, X, Y, 0, 0,
          clNone, clNone, ILD_Transparent)
      end
      else
        ImageList_DrawEx(FGlyphList.Handle, Index, Canvas.Handle, X, Y, 0, 0,
          ColorToRGB(clBtnFace), clNone, ILD_Normal);
      end;
    end;
  end;

procedure TButtonGlyph.DrawButtonText(Canvas: TCanvas; const Caption: string;
TextBounds: TRect; State: TButtonState; BiDiFlags: LongInt);
begin
  with Canvas do
  begin
    Brush.Style:=bsClear;
    if State = bsDisabled then
    begin
      OffsetRect(TextBounds, 1, 1);
      Font.Color:=clBtnHighlight;
      DrawText(Handle, PChar(Caption), Length(Caption), TextBounds,
        DT_CENTER or DT_VCENTER or BiDiFlags);
      OffsetRect(TextBounds, -1, -1);
      Font.Color:=clBtnShadow;
      DrawText(Handle, PChar(Caption), Length(Caption), TextBounds,
        DT_CENTER or DT_VCENTER or BiDiFlags);
    end;
  end;
end;

```

```

    end else
        DrawText(Handle, PChar(Caption), Length(Caption), TextBounds,
            DT_CENTER or DT_VCENTER or BiDiFlags);
    end;
end;

function TButtonGlyph.Draw(Canvas: TCanvas; const Client: TRect;
    const Offset: TPoint; const Caption: string; Layout: TButtonLayout;
    Margin, Spacing: Integer; State: TButtonState; Transparent: Boolean;
    BiDiFlags: LongInt): TRect;
var
    GlyphPos: TPoint;
begin
    CalcButtonLayout(Canvas, Client, Offset, Caption, Layout, Margin, Spacing,
        GlyphPos, Result, BiDiFlags);
    DrawButtonGlyph(Canvas, GlyphPos, State, Transparent);
    DrawButtonText(Canvas, Caption, Result, State, BiDiFlags);
end;

procedure TUkraineEBActionLink.AssignClient(AClient: TObject);
begin
    inherited AssignClient(AClient);
    FClient:=AClient as TUkraineEB;
end;

function TUkraineEBActionLink.IsCheckedLinked: Boolean;
begin
    Result:=inherited IsCheckedLinked and (FClient.GroupIndex <> 0) and
        FClient.AllowAllUp and (FClient.Down = (Action as TCustomAction).Checked);
end;

function TUkraineEBActionLink.IsGroupIndexLinked: Boolean;
begin
    Result:=(FClient is TUkraineEB) and
        (TUkraineEB(FClient).GroupIndex = (Action as TCustomAction).GroupIndex);
end;

procedure TUkraineEBActionLink.SetChecked(Value: Boolean);
begin
    if IsCheckedLinked then TUkraineEB(FClient).Down:=Value;
end;

procedure TUkraineEBActionLink.SetGroupIndex(Value: Integer);
begin
    if IsGroupIndexLinked then TUkraineEB(FClient).GroupIndex:=Value;
end;

constructor TUkraineEB.Create(AOwner: TComponent);
begin
    FGlyph:=TButtonGlyph.Create;
    TButtonGlyph(FGlyph).OnChange:=GlyphChanged;
    inherited Create(AOwner);
    SetBounds(0, 0, 23, 22);
    ControlStyle:=[csCaptureMouse, csDoubleClicks];
    ParentFont:=True;
    Color:=clBtnFace;
    FSpacing:=4;
    FMargin:=-1;
    FLayout:=blGlyphLeft;
    FTransparent:=True;
    Inc(ButtonCount);
end;

destructor TUkraineEB.Destroy;
begin

```

```

    Dec(ButtonCount);
    inherited Destroy;
    TButtonGlyph(FGlyph).Free;
end;

procedure TUkraineEB.Paint;
const
    DownStyles: array[Boolean] of Integer = (BDR_RAISEDINNER, BDR_SUNKENOUTER);
    FillStyles: array[Boolean] of Integer = (BF_MIDDLE, 0);
var
    PaintRect: TRect;
    DrawFlags: Integer;
    Offset: TPoint;
    Button: TThemedButton;
    ToolButton: TThemedToolBar;
    Details: TThemedElementDetails;
begin
    if not Enabled then
    begin
        FState:=bsDisabled;
        FDragging:=False;
    end
    else if FState = bsDisabled then
        if FDown and (GroupIndex <> 0) then
            FState:=bsExclusive
        else
            FState:=bsUp;
        Canvas.Font:=Self.Font;

        if ThemeServices.ThemesEnabled then
        begin
            PerformEraseBackground(Self, Canvas.Handle);

            if not Enabled then
                Button:=tbPushButtonDisabled
            else
                if FState in [bsDown, bsExclusive] then
                    Button:=tbPushButtonPressed
                else
                    if MouseInControl then
                        Button:=tbPushButtonHot
                    else
                        Button:=tbPushButtonNormal;

            ToolButton:=ttbToolBarDontCare;
            if FFlat then
            begin
                case Button of
                    tbPushButtonDisabled:
                        Toolbutton:=ttbButtonDisabled;
                    tbPushButtonPressed:
                        Toolbutton:=ttbButtonPressed;
                    tbPushButtonHot:
                        Toolbutton:=ttbButtonHot;
                    tbPushButtonNormal:
                        Toolbutton:=ttbButtonNormal;
                end;
            end;

            PaintRect:=ClientRect;
            if ToolButton = ttbToolBarDontCare then
            begin
                Details:=ThemeServices.GetElementDetails(Button);
                ThemeServices.DrawElement(Canvas.Handle, Details, PaintRect);
                PaintRect:=ThemeServices.ContentRect(Canvas.Handle, Details, PaintRect);
            end;
        end;
    end;
end;

```

```

end
else
begin
  Details:=ThemeServices.GetElementDetails(ToolButton);
  ThemeServices.DrawElement(Canvas.Handle, Details, PaintRect);
  PaintRect:=ThemeServices.ContentRect(Canvas.Handle, Details, PaintRect);
end;

if Button = tbPushButtonPressed then
begin
  if ToolButton <> ttbToolBarDontCare then
    Canvas.Font.Color:=clHighlightText;
    Offset:=Point(1, 0);
  end
  else
    Offset:=Point(0, 0);
    TButtonGlyph(FGlyph).Draw(Canvas, PaintRect, Offset, Caption, FLayout,
  FMargin, FSpacing, FState, Transparent,
    DrawTextBiDiModeFlags(0));
  end
  else
  begin
    PaintRect:=Rect(0, 0, Width, Height);
    if not FFlat then
      begin
        DrawFlags:=DFCS_BUTTONPUSH or DFCS_ADJUSTRECT;
        if FState in [bsDown, bsExclusive] then
          DrawFlags:=DrawFlags or DFCS_PUSHED;
          DrawFrameControl(Canvas.Handle, PaintRect, DFC_BUTTON, DrawFlags);
        end
        else
          begin
            if (FState in [bsDown, bsExclusive]) or
              (FMouseInControl and (FState <> bsDisabled)) or
              (csDesigning in ComponentState) then
              DrawEdge(Canvas.Handle, PaintRect, DownStyles[FState in [bsDown,
bsExclusive]],
                FillStyles[Transparent] or BF_RECT)
            else if not Transparent then
              begin
                Canvas.Brush.Color:=Color;
                Canvas.FillRect(PaintRect);
              end;
              InflateRect(PaintRect, -1, -1);
            end;
            if FState in [bsDown, bsExclusive] then
              begin
                if (FState = bsExclusive) and (not FFlat or not FMouseInControl) then
                  begin
                    Canvas.Brush.Bitmap:=AllocPatternBitmap(clBtnFace, clBtnHighlight);
                    Canvas.FillRect(PaintRect);
                  end;
                  Offset.X:=1;
                  Offset.Y:=1;
                end
                else
                  begin
                    Offset.X:=0;
                    Offset.Y:=0;
                  end;
                TButtonGlyph(FGlyph).Draw(Canvas, PaintRect, Offset, Caption, FLayout,
  FMargin,
    FSpacing, FState, Transparent, DrawTextBiDiModeFlags(0));
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;
end;

```

```

procedure TUkraineEB.UpdateTracking;
var
  P: TPoint;
begin
  if FFlat then
  begin
    if Enabled then
    begin
      GetCursorPos(P);
      FMouseInControl:=not (FindDragTarget(P, True) = Self);
      if FMouseInControl then
        Perform(CM_MOUSELEAVE, 0, 0)
      else
        Perform(CM_MOUSEENTER, 0, 0);
    end;
  end;
end;

procedure TUkraineEB.Loaded;
var
  State: TButtonState;
begin
  inherited Loaded;
  if Enabled then
    State:=bsUp
  else
    State:=bsDisabled;
  TButtonGlyph(FGlyph).CreateButtonGlyph(State);
end;

procedure TUkraineEB.MouseDown(Button: TMouseButton; Shift: TShiftState;
  X, Y: Integer);
begin
  inherited MouseDown(Button, Shift, X, Y);
  if (Button = mbLeft) and Enabled then
  begin
    if not FDown then
    begin
      FState:=bsDown;
      Invalidate;
    end;
    FDragging:=True;
  end;
end;

procedure TUkraineEB.MouseMove(Shift: TShiftState; X, Y: Integer);
var
  NewState: TButtonState;
begin
  inherited MouseMove(Shift, X, Y);
  if FDragging then
  begin
    if not FDown then NewState:=bsUp
    else NewState:=bsExclusive;
    if (X >= 0) and (X < ClientWidth) and (Y >= 0) and (Y <= ClientHeight) then
      if FDown then NewState:=bsExclusive else NewState:=bsDown;
    if NewState <> FState then
    begin
      FState:=NewState;
      Invalidate;
    end;
  end
  else if not FMouseInControl then
    UpdateTracking;
end;

```

```

end;

procedure TUkraineEB.MouseUp(Button: TMouseButton; Shift: TShiftState;
X, Y: Integer);
var
  DoClick: Boolean;
begin
  inherited MouseUp(Button, Shift, X, Y);
  if FDragging then
  begin
    FDragging:=False;
    DoClick:=(X >= 0) and (X < ClientWidth) and (Y >= 0) and (Y <= ClientHeight);
    if FGroupIndex = 0 then
    begin
      FState:=bsUp;
      FMouseInControl:=False;
      if DoClick and not (FState in [bsExclusive, bsDown]) then
        Invalidate;
    end
  else
    if DoClick then
    begin
      SetDown(not FDown);
      if FDown then Repaint;
    end
  else
    begin
      if FDown then FState:=bsExclusive;
      Repaint;
    end;
    if DoClick then Click;
    UpdateTracking;
  end;
end;

procedure TUkraineEB.Click;
begin
  inherited Click;
end;

function TUkraineEB.GetPalette: HPALETTE;
begin
  Result:=Glyph.Palette;
end;

function TUkraineEB.GetActionLinkClass: TControlActionLinkClass;
begin
  Result:=TUkraineEBActionLink;
end;

function TUkraineEB.GetGlyph: TBitmap;
begin
  Result:=TButtonGlyph(FGlyph).Glyph;
end;

procedure TUkraineEB.SetGlyph(Value: TBitmap);
begin
  TButtonGlyph(FGlyph).Glyph:=Value;
  Invalidate;
end;

function TUkraineEB.GetNumGlyphs: TNumGlyphs;
begin
  Result:=TButtonGlyph(FGlyph).NumGlyphs;
end;

```

```

procedure TUkraineEB.SetNumGlyphs(Value: TNumGlyphs);
begin
  if Value < 0 then Value:=1
  else if Value > 4 then Value:=4;
  if Value <> TButtonGlyph(FGlyph).NumGlyphs then
  begin
    TButtonGlyph(FGlyph).NumGlyphs:=Value;
    Invalidate;
  end;
end;

procedure TUkraineEB.GlyphChanged(Sender: TObject);
begin
  Invalidate;
end;

procedure TUkraineEB.UpdateExclusive;
var
  Msg: TMessage;
begin
  if (FGroupIndex <> 0) and (Parent <> nil) then
  begin
    Msg.Msg:=CM_BUTTONPRESSED;
    Msg.WParam:=FGroupIndex;
    Msg.LParam:=Longint(Self);
    Msg.Result:=0;
    Parent.Broadcast(Msg);
  end;
end;

procedure TUkraineEB.SetDown(Value: Boolean);
begin
  if FGroupIndex = 0 then Value:=False;
  if Value <> FDown then
  begin
    if FDown and (not FAllowAllUp) then Exit;
    FDown:=Value;
    if Value then
    begin
      if FState = bsUp then Invalidate;
      FState:=bsExclusive
    end
    else
    begin
      FState:=bsUp;
      Repaint;
    end;
    if Value then UpdateExclusive;
  end;
end;

procedure TUkraineEB.SetFlat(Value: Boolean);
begin
  if Value <> FFlat then
  begin
    FFlat:=Value;
    Invalidate;
  end;
end;

procedure TUkraineEB.SetGroupIndex(Value: Integer);
begin
  if FGroupIndex <> Value then
  begin

```

```

    FGroupIndex:=Value;
    UpdateExclusive;
end;
end;

procedure TUkraineEB.SetLayout(Value: TButtonLayout);
begin
    if FLayout <> Value then
        begin
            FLayout:=Value;
            Invalidate;
        end;
end;

procedure TUkraineEB.SetMargin(Value: Integer);
begin
    if (Value <> FMargin) and (Value >= -1) then
        begin
            FMargin:=Value;
            Invalidate;
        end;
end;

procedure TUkraineEB.SetSpacing(Value: Integer);
begin
    if Value <> FSpacing then
        begin
            FSpacing:=Value;
            Invalidate;
        end;
end;

procedure TUkraineEB.SetTransparent(Value: Boolean);
begin
    if Value <> FTransparent then
        begin
            FTransparent:=Value;
            if Value then
                ControlStyle:=ControlStyle - [csOpaque] else
                ControlStyle:=ControlStyle + [csOpaque];
            Invalidate;
        end;
end;

procedure TUkraineEB.SetAllowAllUp(Value: Boolean);
begin
    if FAllowAllUp <> Value then
        begin
            FAllowAllUp:=Value;
            UpdateExclusive;
        end;
end;

procedure TUkraineEB.WMLButtonDblClk(var Message: TWMLButtonDown);
begin
    inherited;
    if FDown then DblClick;
end;

procedure TUkraineEB.CMEnabledChanged(var Message: TMessage);
const
    NewState: array[Boolean] of TButtonState = (bsDisabled, bsUp);
begin
    TButtonGlyph(FGlyph).CreateButtonGlyph(NewState[Enabled]);
    UpdateTracking;
end;

```

```

    Repaint;
end;

procedure TUkraineEB.CMButtonPressed(var Message: TMessage);
var
    Sender: TUkraineEB;
begin
    if Message.WParam = FGroupIndex then
        begin
            Sender:=TUkraineEB(Message.LParam);
            if Sender <> Self then
                begin
                    if Sender.Down and FDown then
                        begin
                            FDown:=False;
                            FState:=bsUp;
                            if (Action is TCustomAction) then
                                TCustomAction(Action).Checked:=False;
                            Invalidate;
                        end;
                    FAllowAllUp:=Sender.AllowAllUp;
                end;
            end;
        end;
end;

procedure TUkraineEB.CMDialogChar(var Message: TCMDialogChar);
begin
    with Message do
        if IsAccel(CharCode, Caption) and Enabled and Visible and
            (Parent <> nil) and Parent.Showing then
            begin
                Click;
                Result:=1;
            end else
                inherited;
end;

procedure TUkraineEB.CMFontChanged(var Message: TMessage);
begin
    Invalidate;
end;

procedure TUkraineEB.CMTextChanged(var Message: TMessage);
begin
    Invalidate;
end;

procedure TUkraineEB.CMSysColorChange(var Message: TMessage);
begin
    with TButtonGlyph(FGlyph) do
        begin
            Invalidate;
            CreateButtonGlyph(FState);
        end;
end;

procedure TUkraineEB.CMMouseLeave(var Message: TMessage);
var
    NeedRepaint: Boolean;
begin
    inherited;
    NeedRepaint:=FFlat and FMouseInControl and Enabled and not FDragging;
    if NeedRepaint or ThemeServices.ThemesEnabled then
        begin
            FMouseInControl:=False;

```

```

    if Enabled then
        Repaint;
    end;
end;

procedure TUKraineEB.ActionChange(Sender: TObject; CheckDefaults: Boolean);

    procedure CopyImage(ImageList: TCustomImageList; Index: Integer);
    begin
        with Glyph do
            begin
                Width:=ImageList.Width;
                Height:=ImageList.Height;
                Canvas.Brush.Color:=clFuchsia;
                Canvas.FillRect(Rect(0,0, Width, Height));
                ImageList.Draw(Canvas, 0, 0, Index);
            end;
        end;
    end;

begin
    inherited ActionChange(Sender, CheckDefaults);
    if Sender is TCustomAction then
        with TCustomAction(Sender) do
            begin
                if CheckDefaults or (Self.GroupIndex = 0) then
                    Self.GroupIndex:=GroupIndex;
                if (Glyph.Empty) and (ActionList <> nil) and (ActionList.Images <> nil)
                    and
                    (ImageIndex >= 0) and (ImageIndex < ActionList.Images.Count) then
                        CopyImage(ActionList.Images, ImageIndex);
                end;
            end;
        end;
    end;

constructor TBitBtn.Create(AOwner: TComponent);
begin
    FGlyph:=TButtonGlyph.Create;
    TButtonGlyph(FGlyph).OnChange:=GlyphChanged;
    inherited Create(AOwner);
    FCanvas:=TCanvas.Create;
    FStyle:=bsAutoDetect;
    FKind:=bkCustom;
    FLayout:=blGlyphLeft;
    FSpacing:=4;
    FMargin:=-1;
    ControlStyle:=ControlStyle + [csReflector];
    DoubleBuffered:=True;
end;

destructor TBitBtn.Destroy;
begin
    inherited Destroy;
    TButtonGlyph(FGlyph).Free;
    FCanvas.Free;
end;

procedure TBitBtn.CreateHandle;
var
    State: TButtonState;
begin
    if Enabled then
        State:=bsUp
    else
        State:=bsDisabled;
    inherited CreateHandle;
    TButtonGlyph(FGlyph).CreateButtonGlyph(State);
end;

```

```

end;

procedure TBitBtn.CreateParams (var Params: TCreateParams);
begin
  inherited CreateParams (Params);
  with Params do Style:=Style or BS_OWNERDRAW;
end;

procedure TBitBtn.SetButtonStyle (ADefault: Boolean);
begin
  if ADefault <> IsFocused then
  begin
    IsFocused:=ADefault;
    Refresh;
  end;
end;

procedure TBitBtn.Click;
var
  Form: TCustomForm;
  Control: TWinControl;
begin
  case FKind of
    bkClose:
      begin
        Form:=GetParentForm (Self);
        if Form <> nil then Form.Close
        else inherited Click;
      end;
    bkHelp:
      begin
        Control:=Self;
        while (Control <> nil) and (Control.HelpContext = 0) do
          Control:=Control.Parent;
        if Control <> nil then Application.HelpContext (Control.HelpContext)
        else inherited Click;
      end;
    else
      inherited Click;
  end;
end;

procedure TBitBtn.CNMeasureItem (var Message: TWMMMeasureItem);
begin
  with Message.MeasureItemStruct^ do
  begin
    itemWidth:=Width;
    itemHeight:=Height;
  end;
end;

procedure TBitBtn.CNDrawItem (var Message: TWMDrawItem);
begin
  DrawItem (Message.DrawItemStruct^);
end;

procedure TBitBtn.CMFontChanged (var Message: TMessage);
begin
  inherited;
  Invalidate;
end;

procedure TBitBtn.CMEnabledChanged (var Message: TMessage);
begin
  inherited;

```

```

    Invalidate;
end;

procedure TBitBtn.WMLButtonDblClk(var Message: TWMLButtonDblClk);
begin
    Perform(WM_LBUTTONDOWN, Message.Keys, Longint(Message.Pos));
end;

function TBitBtn.GetPalette: HPALETTE;
begin
    Result:=Glyph.Palette;
end;

procedure TBitBtn.SetGlyph(Value: TBitmap);
begin
    TButtonGlyph(FGlyph).Glyph:=Value as TBitmap;
    FModifiedGlyph:=True;
    Invalidate;
end;

function TBitBtn.GetGlyph: TBitmap;
begin
    Result:=TButtonGlyph(FGlyph).Glyph;
end;

procedure TBitBtn.GlyphChanged(Sender: TObject);
begin
    Invalidate;
end;

function TBitBtn.IsCustom: Boolean;
begin
    Result:=Kind = bkCustom;
end;

procedure TBitBtn.SetStyle(Value: TButtonStyle);
begin
    if Value <> FStyle then
    begin
        FStyle:=Value;
        Invalidate;
    end;
end;

procedure TBitBtn.SetKind(Value: TBitBtnKind);
begin
    if Value <> FKind then
    begin
        if Value <> bkCustom then
        begin
            Default:=Value in [bkOK, bkYes];
            Cancel:=Value in [bkCancel, bkNo];

            if ((csLoading in ComponentState) and (Caption = '')) or
                (not (csLoading in ComponentState)) then
            begin
                if BitBtnCaptions[Value] <> nil then
                    Caption:=LoadResString(BitBtnCaptions[Value]);
                end;

                ModalResult:=BitBtnModalResults[Value];
                TButtonGlyph(FGlyph).Glyph:=GetBitBtnGlyph(Value);
                NumGlyphs:=2;
                FModifiedGlyph:=False;
            end;
        end;
    end;
end;

```

```

    FKind:=Value;
    Invalidate;
end;
end;

function TBitBtn.IsCustomCaption: Boolean;
begin
    Result:=AnsiCompareStr(Caption, LoadResString(BitBtnCaptions[FKind])) <> 0;
end;

function TBitBtn.GetKind: TBitBtnKind;
begin
    if FKind <> bkCustom then
        if ((FKind in [bkOK, bkYes]) xor Default) or
            ((FKind in [bkCancel, bkNo]) xor Cancel) or
            (ModalResult <> BitBtnModalResults[FKind]) or
            FModifiedGlyph then
            FKind:=bkCustom;
        Result:=FKind;
    end;
end;

procedure TBitBtn.SetLayout(Value: TButtonLayout);
begin
    if FLayout <> Value then
        begin
            FLayout:=Value;
            Invalidate;
        end;
end;

function TBitBtn.GetNumGlyphs: TNumGlyphs;
begin
    Result:=TButtonGlyph(FGlyph).NumGlyphs;
end;

procedure TBitBtn.SetNumGlyphs(Value: TNumGlyphs);
begin
    if Value < 0 then Value:=1
    else if Value > 4 then Value:=4;
    if Value <> TButtonGlyph(FGlyph).NumGlyphs then
        begin
            TButtonGlyph(FGlyph).NumGlyphs:=Value;
            Invalidate;
        end;
end;

procedure TBitBtn.SetSpacing(Value: Integer);
begin
    if FSpacing <> Value then
        begin
            FSpacing:=Value;
            Invalidate;
        end;
end;

procedure TBitBtn.SetMargin(Value: Integer);
begin
    if (Value <> FMargin) and (Value >= - 1) then
        begin
            FMargin:=Value;
            Invalidate;
        end;
end;

procedure DestroyLocals; far;

```

```
var
  I: TBitBtnKind;
begin
  for I:=Low(TBitBtnKind) to High(TBitBtnKind) do
    BitBtnGlyphs[I].Free;
end;

procedure TBitBtn.CMMouseEnter(var Message: TMessage);
begin
  inherited;
  if ThemeServices.ThemesEnabled and not FMouseInControl and not (csDesigning in
ComponentState) then
  begin
    FMouseInControl:=True;
    Repaint;
  end;
end;

procedure TBitBtn.CMMouseLeave(var Message: TMessage);
begin
  inherited;
  if ThemeServices.ThemesEnabled and FMouseInControl then
  begin
    FMouseInControl:=False;
    Repaint;
  end;
end;

end.
```

Кафедра КБПЗ – 2021 рік