МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Кафедра кібербезпеки та програмного забезпечення

На правах рукопису

Підлубний Олексій Васильович

Програмне забезпечення системи моніторингу продуктивності застосунків на базі APM

Спеціальність: 123 «Комп'ютерна інженерія»

Освітній ступінь: бакалавр

Науковий керівник:

Доренський Олександр Павлович _____

(підпис) (дата)

кандидат технічних наук

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри

_____ О.А. Смірнов

(підпис) ПІБ

«____»_____2021 р.

Кафедра_КбПЗ_2021рік

Кропивницький – 2021

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

Підлубному Олексію Васильовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи моніторингу продуктивності застосунків на базі APM*

керівник роботи *Доренський Олександр Павлович, канд. техн. наук*
(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 моніторингу продуктивності застосунків на базі APM*

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*

Кафедра КБПЗ 2021рік

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)

Кафедра_КОПЗ_2021рік

АНОТАЦІЯ

Підлубний О.В. Програмне забезпечення системи моніторингу продуктивності застосунків на базі АРМ. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній кваліфікаційній бакалаврській розроблено програмне забезпечення, яке призначено для системи моніторингу продуктивності застосунків на базі АРМ.

Метою розробки є програмне забезпечення системи моніторингу продуктивності застосунків на базі АРМ.

Результат роботи – програмна реалізація системи моніторингу продуктивності застосунків на базі АРМ.

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 Sydney.

Ключові слова: комп'ютерна інженерія, моніторинг, АРМ

КафеДра_КОПЗ_2021рік

ABSTRACT

Pidlubnyi O.V. APM-based application performance monitoring system software. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2021

In this bachelor's qualification the software which is in tended for system of monitoring of productivity of applications on the basis of APM is developed.

The purpose of the development is the software of the application performance monitoring system based on APM.

The result is a software implementation of an AP M-based application performance monitoring 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 environment of Delphi 10.4 Sydney.

Keywords: computer engineering, monitoring, APM

Кафедра_КПЗ_2021рік

ЗМІСТ

Кафедра_КбПЗ_2021рік

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ІС	–	інформаційні системи
ІТ	–	інформаційні технології
ПЗ	–	програмне забезпечення
ТВС	–	технології відкритих систем
APM	–	моніторинг продуктивності застосунків (Application Performance Monitoring)
DaaS	–	робочий стіл як послуга
MTTR	–	середній час відновлення
NPMD	–	Network Performance Monitoring and Diagnostics
VDI	–	інфраструктура віртуальних робочих столів
WaaS	–	робоче місце як послуга

Кафедра_КбПЗ_2021рік

ВСТУП

Актуальність теми. Моніторинг продуктивності застосунків (Application Performance Monitoring, APM) вирішує завдання контролю, керування доступністю й безпосередньо продуктивністю застосунків. Різні фахівці можуть трактувати це визначення по-різному, тому спробуємо описати, що це за рішення, із чого полягає й чому буває важливо для компанії.

Корпоративні застосунки міняються й стають багаторівневими, розподіленими між різними серверами або навіть континентами, переходячи в хмари. Тому такі складні розподілені застосунки вимагають контролю, тому що в деяких компаніях є основою бізнесу. Як і в будь-якого моніторингу рішення APM визначають базову продуктивність і ухвалюють це за норму. Далі будь -які відхилення від неї реєструються й вимагають ухвалення рішення з метою визначення причини відхилення в продуктивності. Таким образів на відміну від рішень NPMD (Network Performance Monitoring and Diagnostics), призначених для аналізу продуктивності IT-інфраструктури, тут фокус системи – застосунок, помилки в коді і їх вплив на його продуктивність.

За допомогою APM ми можемо оцінити:

- стан фізичного устаткування;
- стан віртуальної машини;
- стан віртуальному Java машини;
- стан контейнера;
- поведінка самого застосунка;
- стан допоміжної інфраструктури, баз даних, кеші, зовнішні веб сервіси.

Після того як ми одержуємо показники продуктивності із усіх цих джерел, рішення APM повинне інтерпретувати для нас і провести кореляцію між ними для оцінки впливу на бізнес транзакції. Це те саме місце, де магія рішень APM проявляється на повну силу.

Кафедра_КОПЗ_2021рік

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи моніторингу продуктивності застосунків на базі АРМ.

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

– Огляд існуючих систем моніторингу продуктивності застосунків на базі АРМ.

– Дослідження системи моніторингу продуктивності застосунків на базі АРМ.

– Програмна реалізація системи моніторингу продуктивності застосунків на базі АРМ.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі моніторингу продуктивності застосунків на базі АРМ.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи моніторингу продуктивності застосунків на базі АРМ, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

Кафедра_КБПЗ_2021рік

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Виробники рішень повинні розуміти на глибокому рівні поведінку різних застосунків, щоб видати вердикт – живий пацієнт або скоріше мертвий. Особливо це важливо робити швидко, тому що релізи застосунків міняються досить часто, а рівень програмістів у групі прикладників буває різний.

Наступний крок – це постійний моніторинг і порівняння поточної продуктивності з базової, щоб чуйно реагувати на відхилення. Наприклад, якщо ключові бізнес транзакції, як правило, виконуються протягом 4 секунд у п'ятницю, а в поточну п'ятницю – це становить 8 секунд, то виникає логічне запитання: «Що трапилося?». Далі система моніторингу застосунків, проаналізувавши всі внутрішні й зовнішні залежності, повинна визначити поведінка якого елемента не відповідає нормі й у зручному форматі повідомити відповідальних фахівців. Далі справа за малим – усунути виявлену проблему. Частина проблем через налаштування системи можуть усуватися автоматично, наприклад, шляхом виділення застосункових ресурсів при росту кількості користувачів або обсягу запитів.

1.2 Область застосування

Таким чином, ми можемо уточнити визначення моніторингу продуктивності застосунків APM – це система, яка виконує:

- збір показників продуктивності всього середовища застосунків;
- інтерпретацію цих показників у світлі виконуваних бізнес процесів і транзакцій;
- порівняння й аналіз поточних показників і базових;

Кафедра_КПЗ_2021рік

– захват відповідної контекстної інформації при виявленні відхилень;

– відправлення повідомлень про ненормальну поведінку;

– адаптацію середовища застосунків для усунення проблем із продуктивністю.

Отже, підіб'ємо підсумок, рішення для моніторингу продуктивності застосунків – необхідний інструмент, який дозволяє зрозуміти поведінку вашого застосунка, виявити проблеми, перш ніж ваші користувачі піддадуться негативному впливу й, по можливості, швидко розв'язати ці проблеми. З погляду бізнесу, рішення APM важливо, тому що воно зменшує середній час відновлення (MTTR). А це озна чає, що проблеми із продуктивністю вирішуються швидше й ефективніше, що впливає на продуктивність роботи співробітників і репутацію компанії.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи моніторингу продуктивності застосунків на базі APM, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

Кафедра_КбПЗ_2021рік

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми кваліфікаційної бакалаврської роботи

Radware APM

Модуль моніторингу продуктивності застосунків (APM – Application Performance Monitoring) дозволяє вимірювати продуктивність веб-застосунків і гарантувати відповідність очікуваному рівню надання послуг (SLA) для кожного застосунку, з наочним наданням інформації з застосунків, географічного положення й окремим транзакціям. Інструмент Radware APM призначений для швидкого виявлення й усунення несправностей і аналізу причин затримки в роботі застосунків.

Radware APM дозволяє у візуальній формі одержати інформацію про зниження якості роботи користувачів до того, як це привело до скарг із їхнього боку, проаналізувати причини зниження й вжити відповідних заходів.

Модуль Radware APM є частиною системи централізованого керування Radware Apsolute Vision.

У чому переваги рішення для моніторингу продуктивності застосунків Radware ATM?

Модуль Radware APM є частиною системи централізованого керування Radware Apsolute Vision. Він не вимагає якої-небудь інтеграції в інфраструктуру застосунків, тому не створює застосункових витрат і не вимагає створення скриптів для застосунка, імітації штучних транзакцій і т.д. Виміру продуктивності заснований на реальних користувацьких транзакціях, а також реальних помилках, які реєструються системою збору даних і оброблені аналітичним механізмом Radware Apsolute Vision. Механізм аналізу пропонує широкий спектр докладних звітів, а, що також настроюються звіти для

Кафедра_КБПЗ_2021рік

налаштування вимірів продуктивності відповідно до конкретних цілями й бізнесами-показниками.

Radware APM вимірює продуктивність усіх веб-застосунків і всіх серверів, трафік яких обробляє Alteon ADC. Крім того, такі виміри можуть бути зібрані з декількох пристроїв Alteon, забезпечуючи огляд продуктивності застосунка й QoE кінцевих користувачів між декількома ЦОД.

Рішення забезпечує:

– Наочність, необхідну для безперервної підтримки веб-застосунком комерційних умов SLA.

– Ефективний інструмент усунення збоїв, а також точної локалізації проблем із продуктивністю. Усе виконується в реальному часі.

– Спрощене впровадження, для якого не потрібно спеціально створених скриптів, що знижує загальні витрати.

Не вимагає інтеграції в інфраструктуру застосунків і обслуговування

Розгортання нових мережних пристроїв або установка програмних агентів на сервери застосунків не потрібно. Відразу після запуску нового сервера застосунків, Radware APM починає його моніторинг.

Моніторинг якості сервісу для реальних користувачів

Рішення Radware ATM для наскрізного моніторингу продуктивності застосунка фіксує фактичне QoE користувача при фактичній затримці на виконання транзакції й виявленні помилок у реальному часі.

Будь-які виміри транзакції розбиваються на триетапну ланцюжок доставки застосунка:

– Час відгуку між сервером і ADC – час, затрачуване в ЦОД.

– Мережний час – час, затрачуване на те, щоб відповідь сервера досяглася клієнта.

– Час на відображення в браузері – час, затрачуване браузером на завантаження й відображення сторінки.

Кафедра_КОПЗ_2021рік

Підтримка внутрішніх і зовнішніх застосунків

Рішення забезпечує моніторинг як зовнішніх користувацьких застосунків, так і внутрішніх корпоративних застосунків, доступних тільки з інтранету. Універсально підходить для більшості варіантів використання.

Micro Focus Application Performance Management

Рішення для керування продуктивністю застосунків Micro Focus Application Performance Management (APM) дозволяють у режимі реального часу усувати будь-які проблеми, пов'язані з локальними, хмарними й мобільними застосунками. Рішення Micro Focus APM доступні в різних варіантах – ви можете вибрати модель, яка оптимально підходить для ваших цілей. Підберіть локальний, SaaS- або гібридний варіант рішень згідно з наявними платформою й бізнес-моделі.

Синтетичний моніторинг

Моделювання віртуальних користувачів у цілодобовому режимі шляхом запуску сценаріїв через регулярні проміжки часу з декількох фізичних місць дозволяє попереджати про проблеми, пов'язані із працездатністю й продуктивністю, до того, як ці проблеми торкнуться ваших клієнтів.

Моніторинг реальних користувачів

Оцінюйте якість взаємодії з усіма кінцевими користувачами незалежно від місцезнаходження й часу. Автоматично визначайте базову інфраструктуру й класифікуйте дії користувачів.

Детальна діагностика

Одержуйте докладні дані про дії користувачів і роботі серверних застосунків у традиційному, віртуалізованому або хмарному середовищу, щоб виявити фактори, що знижують продуктивність. Оперативно визначайте й усувайте проблеми.

Кафедра_КОПЗ_2021рік

Моніторинг систем

Єдиний набір інструментів для моніторингу інфраструктури, застосунків і мережі з більш ніж 100 готовими до використання моніторами й шаблонами моніторів, що дозволяють легко усувати проблеми в системах.

Керування рівнем послуг

Панель моніторингу стану всіх настроєних угод про рівень обслуговування (SLA) і угод про рівень операційної підтримки (OLA) дозволяє знизити ризик порушень SLA.

Real User Monitoring

Засіб моніторингу застосунків, яке дозволяє вам спостерігати за активністю користувачів у системі й при необхідності вчасно й цілеспрямовано реагувати на проблеми.

Аналітичні дані про умови роботи користувача

Відслідковуйте роботу застосунків і те, як їх використовують користувачі в браузері, у хмарі й на мобільних пристроях. Ця інформація включає дані про всіх користувачів і всіх територіях.

Запис сесії користувацької взаємодії

Відтворення сесій взаємодії користувачів дозволяє визначати пріоритет окремих проблем і вирішувати їх більш оперативно.

Реалістичні тестові сценарії

Можливість створювати сценарії тестування за допомогою даних про сеанси реальних користувачів.

Відстеження транзакцій користувача

Поліпшене відстеження транзакцій кінцевих користувачів за рахунок автоматичного визначення рівнів застосунка дозволяє швидше виправляти складні ситуації.

Інтегрований результат

Більш ефективне управління якістю обслуговування з урахуванням даних про сеанси взаємодії кінцевих користувачів і продуктивності інфраструктури.

Кафедра_КБПЗ_2021рік

Контроль бізнес-процесів

ПЗ для моніторингу застосунків бізнес, що послідовно відслідковує, процеси для своєчасного визначення проблем продуктивності, перш ніж вони вплинуть на роботу користувачів.

Контроль якості взаємодії з користувачем

Стежите за споконвічним рівнем і тенденціями зміни продуктивності застосунків. ПЗ BPM надає достовірні, що піддаються оцінці виміру, за допомогою яких можна встановити взаємозв'язок між наслідками для діяльності компанії і їх першопричинами, а також переглянути відповідні угоди про рівень обслуговування.

Активний моніторинг

Формуйте прогнози відносно показників якості обслуговування для застосунків усіх типів, включаючи корпоративні, гібридні, хмарні й мобільні.

Раннє виявлення проблем

Виявляйте проблеми до того, як їх помітять користувачі. ПЗ BPM здійснює прогнозування й виявлення проблем навіть під час простою систем, і коли адміністратори застосунків не виконують перевірку їх стану.

Оперативне виявлення проблем

Визначайте й вирішуйте проблеми швидше. BPM сортує проблеми по розташуванню, застосунку, транзакції (наприклад вхід у систему або перевірка балансу), постачальникові хмарних послуг і іншим критеріям.

Зручні функції розгортання

Управляйте продуктивністю застосунків, розгорнувши відповідний рішення локально або використовуючи його по моделі «ПЗ як послуга».

Інтеграція рішень для поліпшення співробітництва

Визначайте першопричини й швидко усувайте проблеми. Ефективна інтеграція за допомогою Micro Focus Diagnostics і платформи APM забезпечує співробітництво між розроблювачами застосунків і службою підтримки.

Кафедра_КПЗ_2021рік

Diagnostics

Швидко ізолюйте й розв'яжете проблеми, пов'язані із продуктивністю застосунків для розробки й виробництва, забезпечивши видимість на рівні коду для рівня транзакцій.

Оцінка

Збирайте дані про продуктивність для застосунків, транзакцій, потоку користувачів, інфраструктури, кодових сегментів і SQL-інструкцій.

Установлення черговості

Визначите проблеми продуктивності й стабільності, що мають найбільший вплив.

Визначення

Співвідносите проблеми продуктивності транзакцій з компонентами застосунків, інфраструктурою, кодовими сегментами й SQL-інструкціями.

Рішення для моніторингу Business Service Level

Програмне забезпечення, яке дозволяє задавати цільові показники рівня IT-послуг, а потім відслідковувати їхнє досягнення й аналізувати звіти в контексті бізнес-цілей компанії.

Значення вимірності мети

Звіти по ефективності керування рівнем обслуговування дозволяють оцінити виконання вашими IT-службами поставлених перед ними завдань. Управляйте рівнем обслуговування на випередження з урахуванням комерційних мет, вимірюйте його ефективність у контексті поставлених перед компанією завдань, визначайте реалістичні й розрахункові цільові показники рівня обслуговування й відслідковуйте їхнє досягнення в режимі реального часу.

Sitescope

Просте в установці й налаштуванні безагентне програмне забезпечення для моніторингу застосунків, що забезпечує підтримку різнорідної й гібридної інфраструктури, а також швидку окупність.

Кафедра_КБПЗ_2021рік

Автоматизований моніторинг застосунків

Налаштовуйте й автоматизуйте моніторинг застосунків для сучасних динамічних хмарних середовищ, використовуючи інтерфейси API, потоки Operation Orchestration і Hybrid Cloud Management.

Інтуїтивно зрозуміле керування моніторингом

Робите динамічні відновлення, додаючи й видаляючи лічильники й граничні значення, коли віртуальні машини переміщаються з однієї хост-системи в іншу, і змінюйте граничні значення на основі історичних даних моніторингу.

Швидка окупність

Добийтеся швидкої окупності інвестицій (видимі результати вже через 60 хвилин) за рахунок прискореної установки, відновлення, моніторингу й розгортання з використанням шаблонів масового розгортання й функції «Публікація змін».

Більш 100 типів хостів і платформ застосунків

Відслідковуйте інтенсивність використання, час відгуку, коефіцієнт навантаження й доступність для хостів і прикладних платформ різних типів, включаючи Cisco, Citrix, Microsoft, Oracle, SAP, Siebel, Weblogic і багато інші.

Хмарні структури, віртуалізація – Docker і Kubernetes готові до цього

Об'єднаєте широкий спектр платформ віртуалізації й типів моніторингу, у тому числі моніторинг Microsoft Azure і Amazon Web Services і інтеграцію Amazon Cloud Watch, забезпечивши можливість автоматичного масштабування, оповіщення й ведення звітності.

Керування системами від декількох постачальників

Здійснюйте моніторинг продуктів від різних постачальників, усуваючи потенційні проблеми в критично важливих застосунках до того, як вони вплинуть на роботу інфраструктури.

Apppulse Active

Програмне забезпечення для штучного моніторингу.

Змінюйте свої процеси в режимі реального часу й зводите до мінімуму середній час відновлення в IT-застосунках.

Кафедра_КОПЗ_2021рік

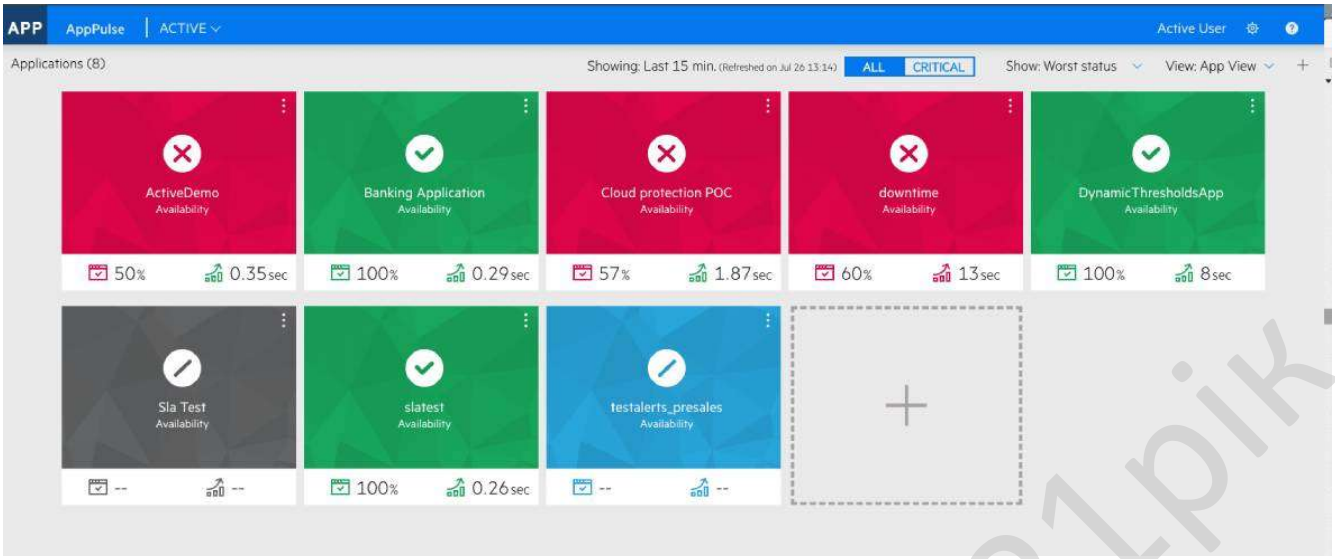

Рисунок 2.1 – Інтерфейс користувача Apppulse Active

Інформаційна панель застосунка дозволяє зосередитися на найважливіших аспектах. Об'єднання, групування й фільтрування найбільш важливих застосунків у режимі реального часу. Просто використовуйте перетаскування.

Контролюйте всі штучні транзакції й місця розташування в застосунку, а також їх стан за необхідний період часу.

Відслідковуйте час реагування на рівні кожної окремої транзакції в певному місці розташування.

Оглядовий звіт представляє зведену інформацію про доступність, час відгуку й загальній кількості збоїв для будь-якого застосунка.

Apppulse Mobile

Довідайтеся, наскільки користувачам зручно працювати з вашим мобільним застосунком, зосередьтеся на тому, що найбільше важливо для ваших користувачів.

Fundex

Fundex дозволяє визначити унікальний базовий індекс умов роботи користувачів мобільних застосунків. Це простій для розуміння показник, у якому враховуються негативні явища в роботі застосунка, такі як збої й уповільнена дія. Об'єднання всіх ключових показників дозволяє зрозуміти, яким образом можна

Кафедра_КБПЗ_2021рік

поліпшити продуктивність і наскільки користувачі задоволені вашим мобільним застосунком.

Аналітика збоїв

Докладний аналіз даних і збоїв дає можливість поліпшити стабільність застосунка й уникнути збоїв у майбутньому. Вам буде відомо все – від дій користувачів до використовуваної версії ОС, – і залишиться тільки простежити, щоб усі помилки були виправлені.

Комплексне відстеження транзакцій

Відстеження транзакцій на рівнях розподіленого застосунка для комплексного відстеження потоків транзакцій. Одержуйте інформацію на рівні коду для кожного випадку взаємодії користувача: ви зможете довідатися про той, скільки часу було витрачено на виконання методів, про інструкції SQL і виключеннях.

Ні додавання тегів, ні змін у коді

Впровадження Apppulse Mobile – елементарний процес. Вам не потрібно змінювати код або додавати теги, тому ви не зіб'єтеся із графіка випуску релізів.

Apppulse Web

ПЗ для моніторингу веб-застосунків.

Контролюйте продуктивність веб-застосунків у режимі реального часу, щоб користувачі залишилися задоволені.

Високий показник зручності використання мобільних застосунків Fun Index, який дає загальна вистава про проблеми користувачів, що негативно впливають на роботу.

Apppulse Web вимірює час завантаження сторінки, включаючи Spa-сторінки, з погляду користувача, дозволяючи виявити елементи, через які знижується швидкість завантаження, і команди AJAX, які негативно позначаються на взаємодії з користувачем.

Кафедра_КБПЗ_2021рік

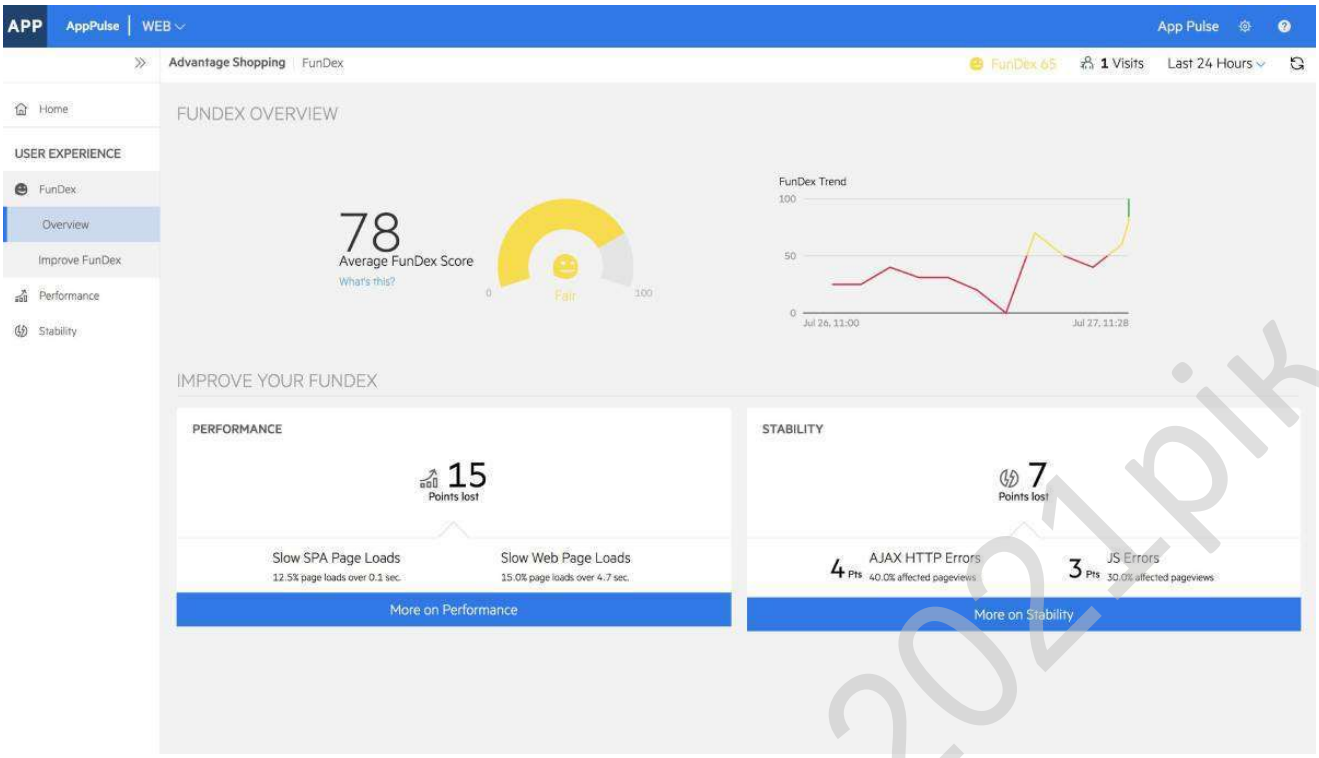

Рисунок 2.2– Інтерфейс користувача Apppulse Web

Підраховуйте кількість сторінок з помилками, концентруйтеся на найбільш складних проблемах і збирайте детальну інформацію про помилки й запитах AJAX роботи, що знижують ефективність.

Anturis – Хмарний моніторинг для серверів і веб-сайтів, моніторинг IT-інфраструктури

Це хмарна (SaaS) платформа, призначена для зовнішнього моніторингу веб-сервісів компанії й внутрішнього моніторингу IT-інфраструктури (серверів і застосунків). У команду цього проекту входять досвідчені фахівці й інженери, що працювали на великі компанії й відомі стартапи, у тому числі Parallels, Amdocs, Atempo, K7 Cloud і jnetx.

Вартість: Безкоштовний тарифний план, або від $9,50 щомісяця.

Appneta – Рішення для APM

Appneta здійснює комплексний контроль застосунків і продуктивності мережі, надає великі кошти моніторингу кінцевих користувачів. У рамках

Кафедра_КБПЗ_2021-рік

проекту доступні прості у використанні, ефективні сервіси, що дозволяють установити зв'язки між застосунками й мережами, а також між компаніями, керуючими ними.

Вартість: Безкоштовний тріал 90 днів, потім від $79 щомісяця.

Bigpanda – Автоматизація керування інцидентами

Bigpanda – це SaaS-платформа, що спрощує процес розв'язання конфліктів у складних веб-середовищах. Проект може допомогти вам розібратися з потоком даних і алертів, візуалізувати залежності у виробничих процесах. І якщо щось піде не так, ви зможете швидко виявити причину й усунути її.

Вартість: Тріал на 21 день, потім від $449 щомісяця.

Boundary – Моніторинг серверів і застосунків для розроблювачів

Boundary – це сервіс моніторингу інфраструктури застосунків. Не вимагає внесення яких-небудь змін у самі застосунки. Не залежить від використовуваних мов програмування, розташовується на кожній віртуальній машині. Boundary збирає великий обсяг даних про продуктивність, консолідує інформацію з інших джерел і формує комплексні карти застосунків, обновлювані в реальному часі.

Вартість: Безкоштовний тарифний план або від $12 щомісяця.

Datadog – Хмарний моніторинг у вигляді сервісу

Datadog – це моніторинговий сервіс, агрегуючий метрики й події із серверів, з баз даних, застосунків, інструментів і сервісів, на підставі яких формується комплексна вистава інфраструктури. Ці можливості надаються на базі аналітичної SaaS-платформи, що дозволяє координувати дії команд розроблювачів і адмінів, щоб уникати простоїв, швидко вирішувати проблеми із продуктивністю й вчасно завершувати цикли розробки й розгортання.

Вартість: Тріал 14 днів, потім від $15 щомісяця.

Dynatrace – APM-сервіс

Dynatrace – один з іноваторів і лідерів у сфері APM. Компанія пропонує потужну APM-систему, у якій реалізований превентивний підхід до керування продуктивністю застосунків. Це дозволяє в рази зменшити тривалість усунення

Кафедра_КОПЗ_2021рік

проблем, а також визволити купу ресурсів, раніше затрачуваних на рішення завдань по продуктивності.

Вартість: Тріал 30 днів, потім потрібно запросити ціни.

Gear5 – Система оповіщення й моніторингу продуктивності сайтів

Вимірює час завантаження й швидкість роботи вашого сайту з погляду реальних користувачів. Дані збираються прямо із браузерів користувачів, що дозволить вам одержати точну виставу про роботу сайту.

Вартість: Безкоштовний тарифний план або від $15,91 щомісяця.

Idera – Моніторинг IT-інфраструктури

Це проект по моніторингові серверів і веб-сайтів для хмарних інфраструктур. Idera надає простий, уніфікований, розумний і швидкий інструментарій для хмарних серверів AWS (EC2), веб-сайтів, веб-застосунків і сервісів. Проект допомагає оптимізувати продуктивність і процес усунення збоїв. Інформація в панелі керування Idera обновляється кожні кілька секунд (не хвилин), і з її допомогою ви зможете виявити такі сховані проблеми, як крадіжка процесорних ресурсів (CPU Steal), і вчасно вжити заходів.

Вартість: Тріал 14 днів, потім від $70 щомісяця.

Instrumental – Моніторинг застосунків у реальному часі

Дуже потужний моніторинговий сервіс, дозволяє знімати порядку 500 000 метрик/сек, і навіть не спітніє.

Вартість: Тріал 30 днів, потім від $150 щомісяця.

Logicmonitor – хостинговий моніторинг мереж, серверів, застосунків, сховищ і хмарних сервісів

Logicmonitor – це SaaS-сервіс для моніторингу фізичної, віртуальної й хмарної IT-інфраструктури. Тут ви зможете відслідковувати продуктивність, переглядати історію й звіти, налаштовувати оповіщення поштою й SMS, щоб попередити співробітників про потенційні проблеми, які потрібно розв'язати, поки вони не почали впливати на ваші бізнеси-процеси. Logicmonitor надає у вигляді однієї веб-консолі заздалегідь зконфігурований, готовий до роботи

Кафедра_КОПЗ_2021рік

моніторинговий сервіс для більшості вендорів свичей, роутерів, файрволів, балансувальників, серверів, застосунків, баз даних, Voip-систем і сховищ.

Вартість: Тріал 14 днів, потім від $249 щомісяця.

Munin – Програмний моніторинг системи, мережі й за стосунків

Munin – інструмент для моніторингу мережних ресурсів, що допомагає аналізувати динаміку їх споживання й вирішувати проблеми типу «що зараз убило нашу продуктивність?». Munin дуже простий у налаштуванні й використанні, за замовчуванням він надає купу графіків, майже не вимагаючи зайвих рухів.

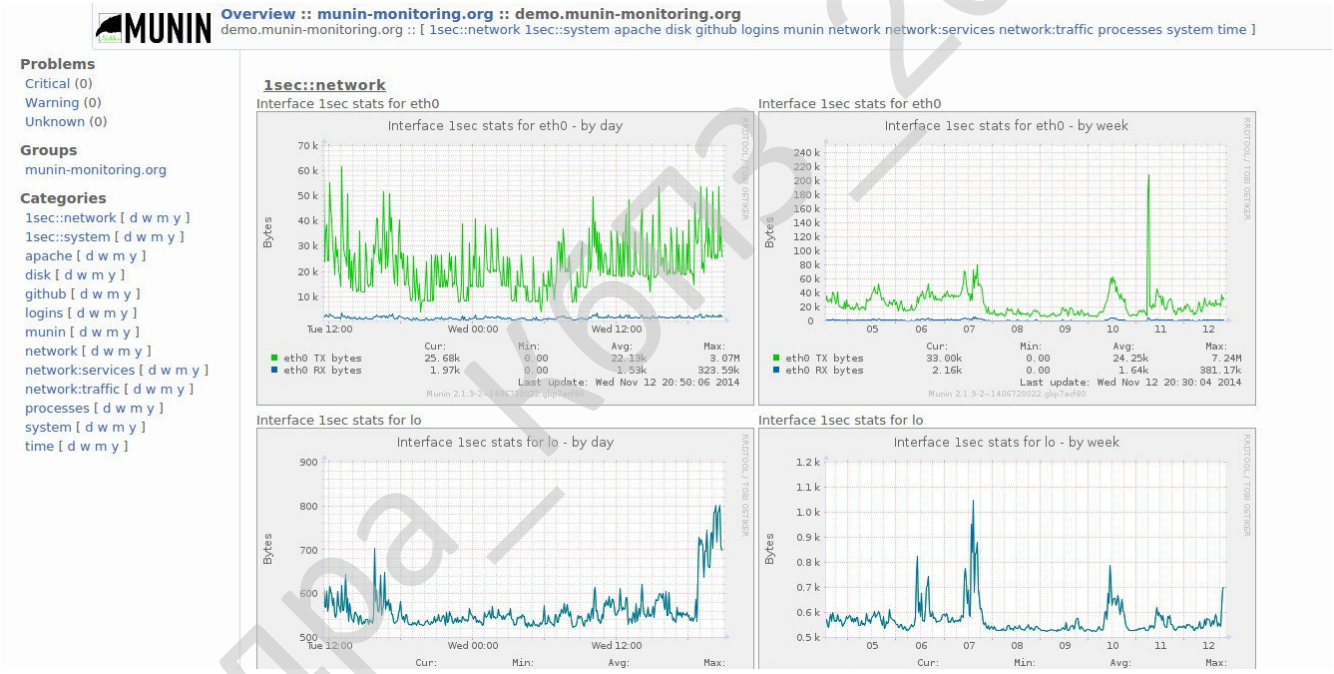

Рисунок 2.3 – Інтерфейс користувача Munin

Вартість: Безкоштовно, open source.

Кафедра – КБР – 2021рік

Nagios – Система моніторингу IT-інфраструктури й розсилання оповіщень

Nagios – потужна моніторингова система, що дозволяє визначати й вирішувати проблеми, пов'язані з IT-інфраструктурою, перш ніж вони вплинуть на важливі бізнеси-процеси. Це масштабований і гнучкий інструмент, завдяки якому можна спати спокійно – неждані збої не порушать хід роботи.

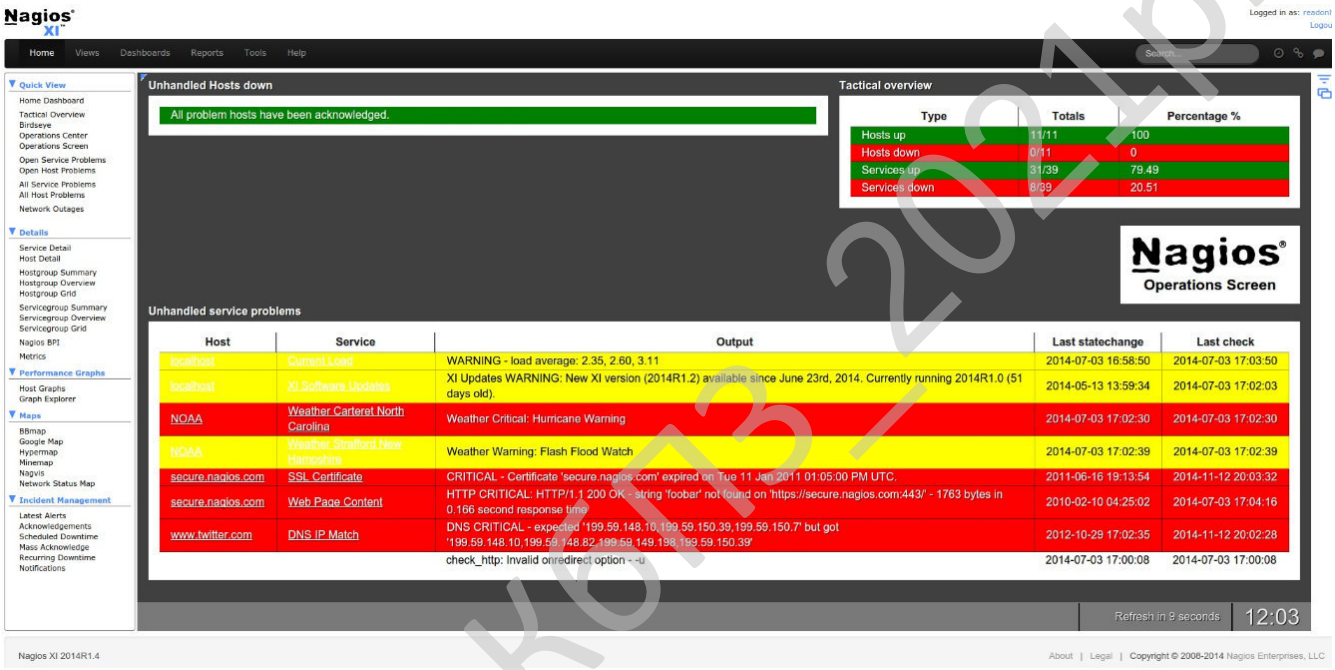

Рисунок 2.4 – Інтерфейс користувача Nagios

Вартість: Безкоштовний движок Nagios Core.

New Relic – Система моніторингу й APM

Компанія New Relic пропонує аналітичну SaaS-платформу, що дозволяє управляти продуктивністю застосунків і здійснювати моніторинг реальних користувачів веб-застосунків, розгорнутих у хмарних сервісах і дата-центрах Застосунки можуть бути написані на Ruby, Java, .NET, Python, PHP, Node.js. Також New Relic пропонують рішення для мобільного моніторингу під ios і Android.

Вартість: Lite-версія й тріал 14 днів, потім від $75 щомісяця.

Кафедра_КОПЗ_2021рік

Ruxit – Система моніторингу продуктивності веб-застосунків

Ruxit вивчає ваше середовище й проводить аналіз різних аномалій. І коли щось піде не так, ви одержите не тільки попередження, але й можливий рішення проблеми.

Вартість: Перші 1000 годин безкоштовно, потім від $0,2 за кожну годину, або 500 сесій, або 100 мережних перевірок.

Scoutapp – Хостинговий моніторинг серверів

Зручні графіки й система оповіщення, розгортається за п'ять хвилин. Більш 60 плагінів, просте конфігурування через веб-інтерфейс, без необхідності запам'ятовувати якісь команди.

Вартість: Тріал 14 днів, потім від $15 за кожний сервер щомісяця.

Sematext – Система моніторингу продуктивності, розсилання оповіщень і виявлення аномалій

Компанія Sematext пропонує ряд модульних масштабованих хмарних сервісів – SPM Performance Monitoring, Logsene Log Management & Analytics, Site Search Analytics і ряд інших. Деякі з них доступні в локальних версіях (on premise).

Вартість: Тріал 30 днів, потім від $0,035 за сервер у годину.

Solarwinds – Система моніторингу серверів і APM

Сотні тисяч системних адміністраторів по усьому світу користуються продуктами Solarwinds для керування середовищами, що включають від десяти до десятків тисяч мережних пристроїв. Це сімейство продуктів містить у собі інструменти для керування відказостійкістю й продуктивністю, для конфігурування й інжинірингу. Компанія заснована в 1999 році, штаб-квартира в Остине, Техас, а групи розроблювачів розкидані по різних країнах.

Вартість: Тріал 30 днів, вартість застосунків від $2000.

Stackify – Application Monitoring & performance for DevOps insight.

Stackify – це хмарна платформа для моніторингу й виявлення несправностей у веб-застосунках. Призначена для розроблювачів ПЗ, сисадмінів і

Кафедра_КБПЗ_2021рік

служб підтримки. За допомогою Stackify можна легко виявляти й вирішувати виникаючі в застосунках проблеми. Платформа пропонує інструменти для моніторингу, реєстрації помилок, балок і безпечного дистанційного доступу до даних, агрегуємих на сервісі.

Вартість: Тріал 14 днів, потім від $15 за сервер щомісяця.

Whatsupgold – ПЗ для моніторингу мереж і серверів

Дозволяє одержати точну виставу про продуктивність вашої IT-середовища. Використовує різні методики моніторингу для виміру продуктивності, визначення доступності й статусу вашої мережі, серверів і застосунків.

Вартість: Тріал 30 днів, потім від $1755.

Manageengine Opmanager – ПЗ для мережного керування й моніторингу

Компанія Manageengine створює застосунки рівня Enterprise для керування IT-ресурсами. Це більші мережні фреймворки для великих, у тому числі міжнародних компаній. На теперішній момент Manageengine обслуговує більш 45 000 клієнтів, у тому числі близько 60% компаній зі списку Fortune 500.

Вартість: Тріал 30 днів, потім від $595 за 25 пристроїв.

Count.ly – Мобільна аналітична платформа

Countly – застосунок для мобільної аналітики, що працює в реальному часі. Збирає дані з мобільних телефонів і візуалізує інформацію, дозволяючи проаналізувати використання застосунків і поведінка користувачів.

Вартість: Тріал 30 днів, потім потрібно запросити ціни.

CA Technologies

Компанія CA Tchnologies займається дизайном, розробкою, маркетингом, ліцензуванням і підтримкою програмних продуктів для керування в сфері IT, що працюють на всіляких платформах. У портфелі компанії є рішення для мейнфреймів і розподілених середовищ, для керування інфраструктурою,

Кафедра_КБПЗ_2021рік

проектами, безпекою, сервісом, продуктивністю, а також для автоматизації ЦОДів і віртуалізації.

Вартість: На сайті не зазначена

Riverbed – Керування продуктивністю

Компанія Riverbed розробила кілька продуктів класу Enterprise, що дозволяють вирішувати фундаментальні проблеми, пов'язані із продуктивністю в сфері WAN.

Вартість: На сайті не зазначена

Icinga – Open Source моніторинг

Icinga 2 – система мережного моніторингу, паралельна галузі розвитку Icinga 1. За замовчуванням не має користувацького інтерфейсу, його потрібно поставити окремо. Сумісна з Nagios, у якості його форка успадкувала його недоліки.

Вартість: Безкоштовно, open source.

Zabbix – Розподілене open source-рішення класу Enterprise для моніторингу

Система призначена для моніторингу й відстеження статусів різноманітних сервісів, серверів і мережного устаткування.

Вартість: Безкоштовно, open source.

Monit – Просте рішення для превентивного моніторингу

Monit – це маленька open source-утиліта для керування й моніторингу Unix-систем. Monit забезпечує автоматичний супровід і відновлення, уміє виявляти причини збоїв.

Вартість: Безкоштовно, open source.

Cacti

Повноцінний рішення для побудови графіків різних метрик, створене на базі Rrdtool.

Вартість: Безкоштовно, open source.

Кафедра_КБПЗ_2021рік

Sealion

Цей сервіс дозволяє швидко діагностувати проблеми на Linux-серверах.

Вартість: Тріал 14 днів, потім від $19 щомісяця.

Opennms

Вільна платформа класу Enterprise для мережного моніторингу й керування.

Вартість: Безкоштовно, open source.

Site24x7

Зовнішній моніторинговий сервіс для веб-сайтів, створений компанією ZOHO Corp. Website & Server Monitoring Service.

Вартість: Тріал 30 днів, потім від $9 щомісяця.

Serverdensity

Моніторингова система для серверів на Freebsd, Windows, OS X і Linux, з кастомними плагінами.

Вартість: Тріал 15 днів, потім від $9 щомісяця.

Keymetrics

PM2 менеджер процесів для Nodejs і моніторинг серверів.

Вартість: Є безкоштовна версія.

Nixstats

Nixstats – безкоштовний сервіс для моніторингу й відображення статистики сервера. Поки безкоштовний, досить зареєструватися на сайті.

Sensu

Сервіс моніторингу й телеметрії з відкритим ядром.

Вартість: Тріал 2 місяця, потім 50$ щомісяця.

Netdata

Real-time моніторинг і телеметрія. Посилання на гитхаб для установки демо-версії.

Кафедра_КБПЗ_2021рік

Zenoss

Гнучкий комплексний сервіс для моніторингу подій, продуктивності й доступності.

Є безкоштовна Core-версія.

Glowroot

Відкрита APM-система, написана на Java.

Вартість: Безкоштовно, open source.

Observium

Моніторингова система з дуже високої ступінь автоматизації. Підтримує дуже широкий спектр устаткування, платформ і операційних систем.

Вартість: Безкоштовний тарифний план або 150 фунтів стерлінгів у рік.

Hawkular

Набір Rest-сервісів для моніторингу, розвиток якого спонсорується Red Hat.

Вартість: Безкоштовно, open source.

Prometheus

Ще один відкритий рішення для моніторингу й розсилання оповіщень.

Вартість: Безкоштовно, open source.

Raintank

Компанія raintank пропонує open source рішення для моніторингу класу Enterprise – Grafana.

Вартість: Безкоштовно, open source.

Moskito

Цей інструмент багато в чому схожий з Newrelic і Appdynamics, але реалізує концепцію більш вузьконаправленого застосування.

Вартість: Безкоштовно, open source.

Cloudstats

Комплексна хмарна система моніторингу й резервного копіювання.

Кафедра_КбПЗ_2021рік

Spiceworks

Повністю безкоштовна (навіть підтримка) система моніторингу з багатими можливостями.

Вартість: Безкоштовно.

Visual Studio Application Insights

Дітище Microsoft, призначене для моніторингу й виявлення неполадок у веб-застосунках і службах.

Вартість: Залежить від обраних модулів і функцій.

System Center Operations Manager

Ще один продукт від творців Windows, призначений для моніторингу й керування в корпоративному сегменті.

Anodot

Система реального часу для виявлення аномалій і аналітики.

Вартість: Тріал 30 днів, потім запросите ціну.

Інструменти для моніторингу й діагностики HP

Вартість: Ряд інструментів безкоштовні, на інші запросите ціни.

IBM Smart Cloud Monitoring

Система для моніторингу приватних хмарних інфраструктур.

Вартість: Тріал, потім запросите ціни.

Monitor.us

Платформа для створення власного набору моніторингових онлайн-сервісів.

Вартість: Тріал 15 днів, потім усе залежить від ваших потреб.

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

Кафедра_КОПЗ_2021рік

Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Delphi 10.4 Sydney

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

Основні можливості Delphi 10.4.1:

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

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

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

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

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

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

Кафедра_КБПЗ_2021рік

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

− Підтримка Metal Driver GPU для macOS і iOS.

− Вбудований Fmxlinux.

− Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API. Реалізація компонента Media Player для macOS тепер використовує Avfoundation. Реалізований заново стилізуємий FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.

− 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 стосуться, 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 проект ах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з

Кафедра_КБПЗ_2021рік

підвищеною швидкодією. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

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

Істотне поліпшення Delphi Code Insight

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

Delphi Custom Managed Records

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

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

Єдине керування пам'яттю

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

Кафедра_КБПЗ_2021рік

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

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

Розширена підтримка бібліотек C++

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

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

Win 64-відладник і збирач для C++

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

Підвищення якості й швидкодії інструментів

- 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.

Кафедра_КОПЗ_2021рік

– Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.

– Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично масштабуться 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 компо нентів 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 використовуть 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 стилізуємий FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.

Кафедра_КБПЗ_2021рік

– Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

– Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

Універсальний інсталятор для установки Online і Offline

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

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

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на кваліфікаційну бакалаврську роботу, реалізації підлягає програмне забезпечення, яке призначено для системи моніторингу продуктивності застосунків на базі APM.

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

Кафедра_КПЗ_2021рік

технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

Кафедра_КОПЗ_2021рік

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Керування продуктивністю застосунків (APM) ставиться до моніторингу або керування продуктивністю вашого коду, залежностей застосунків, часу транзакцій і загальної взаємодії з користувачем.

APM звичайно містить у собі вимір декількох метрик, пов'язаних із продуктивністю застосунка, картами сервісів, користувацькими транзакціями в реальному часі і т.д. Ціль APM – перетворити продукт «чорного ящика» у щось більш прозоре, надаючи інтелектуальну інформацію про його метрики продуктивності. Більш детальна інформація може бути витягнута залежно від типу застосунка, який вимагає пильності.

Деякі люди називають APM керуванням продуктивністю застосунків, а деякі називають моніторинг продуктивності застосунків. Хоча керування – це моніторинг, що скоріше попереджає підхід і, з боку реагування, коли справа доходить до вирішення проблем. У кожному разі, інструменти APM мають вирішальне значення для здоров'я ваших застосунків. Коротше кажучи, керування продуктивністю застосунків полягає в тому, щоб якомога швидше зрозуміти причину будь-якої проблеми, т. Е. З'ясувати, чому транзакції застосунків сповільнилися або зазнають невдачі.

Термін моніторинг використовується щораз, коли ви збираєте неопрацьовані дані й демонструєте їхньому користувачеві. З іншого боку, керування використовується, коли інструмент дає вам можливість вживати правильних заходів відносно, що відслідковуються елементів. Завдання керування можуть містити в собі ідентифікацію надмірно використовуваних компонентів, внесення змін у конфігурацію мережі, створення звітів і багато чого іншого.

Кафедра_КОПЗ_2021рік

Ключові особливості APM:

– Моніторинг продуктивності для всіх транзакцій. Сама основна функція – відслідковувати продуктивність кожної транзакції, створеної вашим застосунком. Ці аналітичні дані з вашого інструмента APM можна використовувати, щоб зрозуміти, до яких запитів часто звертаються, які застосунки працюють повільно і який застосунок вимагає поліпшення.

– Сервісні карти. Карти сервісів дають вам підходящу платформу для пошуку першопричин помилок застосунків шляхом розуміння взаємозалежностей. Інструмент APM в ідеалі повинен виявляти всі мережні елементи, застосунки й сервери у вашій інфраструктурі й класифікувати їх. Інструменти звичайно надають своїм користувачам службові карти, які дають їм можливість візуалізувати взаємозалежності, що допомагає заощаджувати час і енергію.

– Моніторинг користувачів у режимі реального часу. Моніторинг користувачів у реальному часі (RUM) схожий на моніторинг продуктивності, який збирає й аналізує кожну користувацьку транзакцію в застосунку або на веб-сайті в реальному часі (як це відбувається). Це широко відомо як скорочення від RUM – моніторинг реального користувача, реальні показники користувача, вимір реального користувача.

– Синтетичний моніторинг. Синтетичний моніторинг –, що це попереджає моніторинг транзакцій, який допомагає користувачам виявляти проблеми й допомагає їм визначити, чи працює їх застосунок повільний або не працює. Для кінцевих користувачів / клієнтів-адміністраторів, які скаржаться на свій застосунок, це кошмар. Синтетичний моніторинг допомагає користувачам зм'якшити проблеми до того, як вони вплинуть на кінцевих користувачів, за рахунок оповіщення, що попереджає, і моніторингу в режимі реального часу.

– Оповіщення кореляції. Інструмент керування продуктивністю застосунків, такий як Motadata, забезпечує кореляцію попереджень. Це означає, що ви можете співвідносити свої дані Netops, DevOps і APM. Пороги

Кафедра_КБПЗ_2021рік

встановлюються грамотно для показників і застосунків. Кореляція попереджень сприяє більш швидкому дозволу проблем із продуктивністю застосунків.

– Розподілене трасування транзакцій. Надійний інструмент керування продуктивністю застосунків дозволяє користувачам відслідковувати запити транзакцій від наскрізних розподілених систем. Більшість інструментів АРМ забезпечують глибоку видимість аж до рівня коду, щоб допомогти користувачам виявити проблеми, які а якщо ні, то можуть залишитися без уваги. Користувачі повинні мати можливість відслідковувати окремі транзакції застосунків і допомагати їм переходити до запитів рівня бази даних для аналізу вповільнених екземплярів.

Основні переваги АРМ:

– Збільшення доходів і продажів. У кожній організації є кілька важливих застосунків, які прямо або побічно пропорційні іміджу або доходу бренда. Проблеми, пов'язані із цими застосунками, можуть вплинути на загальну продуктивність бізнесу. Інструменти керування продуктивністю застосунків можуть принести значні вигоди, пов'язані з доходом, за рахунок зниження середнього часу відновлення (MTTR) або середнього часу дозволу будь-якого інциденту. Швидке виявлення й вирішення проблем забезпечує краще обслуговування клієнтів, що, у свою чергу, збільшує продажі. Без інструмента АРМ компаніям може видатися, що їх досвід роботи із клієнтами зав'язаний на очі, а потім вони втратять дохід через низьку продуктивність застосунків без догляду.

– Забезпечити безперервність бізнесу. Гарне програмне забезпечення АРМ може знизити ризик збоїв у ваших бізнес-операціях / IT-інфраструктурі. Застосунок є основою для будь-якого бізнесу. Час простою критично важливих для бізнесу застосунків впливає на організацію з погляду втрати доходу, утопленої репутації, зниження продуктивності, впливу, пов'язаного з дотриманням нормативних вимог.

Кафедра_КБПЗ_2021рік

– Поліпшена взаємодія з кінцевим користувачем. Загальне підвищення якості застосунків не тільки поліпшує взаємодія з кінцевими користувачами, але й сприяє більш продуктивній взаємодії на бізнес-рівні як із внутрішніми, так і із зовнішніми користувачами.

Задоволення ваших кінцевих користувачів завжди є вашим пріоритетом. Якщо ваші кінцеві користувачі погано проводять час, вони можуть не сказати про цей заздалегідь. Вони просто перестануть використовувати ваш застосунок або портал і швидко перейдуть до ваших конкурентів. Це сувора реальність. Інструмент APM може тримати вас на крок спереду.

Якщо ви покладаєтеся на застосунки, то APM є для вас безцінним програмним забезпеченням. Якщо ви з команди DevOps, то це на 100% так, APM підійде. Компанії, що використовують інструменти керування продуктивністю застосунків, вважають це своєю перевагою перед конкурентами, оскільки вони швидше вирішують проблеми, вирішують більша кількість проблем за певний період часу, і мати інтелектуальне бізнес-розуміння своїх операцій.

Згідно Gartner, «Керування продуктивністю застосунків може допомогти компаніям побільшати свої доходи й допомогти їм задовольнити своїх клієнтів, що приведе до гарного прибутку».

У цей час майже кожна організація середнього й великого розміру використовує інструмент APM. Згідно В ікіпедії, «З 2013 року програмне забезпечення для керування продуктивністю застосунків увійшло в зону твердої конкуренції, пов'язаної з технічними інноваціями й стратегіями продуктів, у яких беруть участь кілька постачальників, а також їх різні точки зору. Така інтенсивна конкуренція привела до збоїв на ринку APM».

Кафедра_КБПЗ_2021рік

3.2 Розробка структурної схеми

Моніторинг продуктивності застосунків і стану IT інфраструктури (Application Performance Monitoring, APM) – це одна з найбільш перспективних послуг.

У часи пандемії, коли більшість співробітників працює віддалено, рішення APM стають особливо затребуваними. Так, результат опитування більш 1000 IT-фахівців із усього світу показав, що 80% респондентів не вистачає достовірної інформації про хід роботи сервісів і застосунків. Також відзначена важливість розуміння бізнес-процесів для ефективної роботи окремих команд і підприємства в цілому.

Що таке моніторинг продуктивності застосунків?

Моніторинг продуктивності застосунків – це набір інструментів, що дозволяють IT-фахівцям вашого підприємства одержувати повну й своєчасну інформацію про якість роботи всіх бізнес-застосунків і пов'язаних з ними процесів.

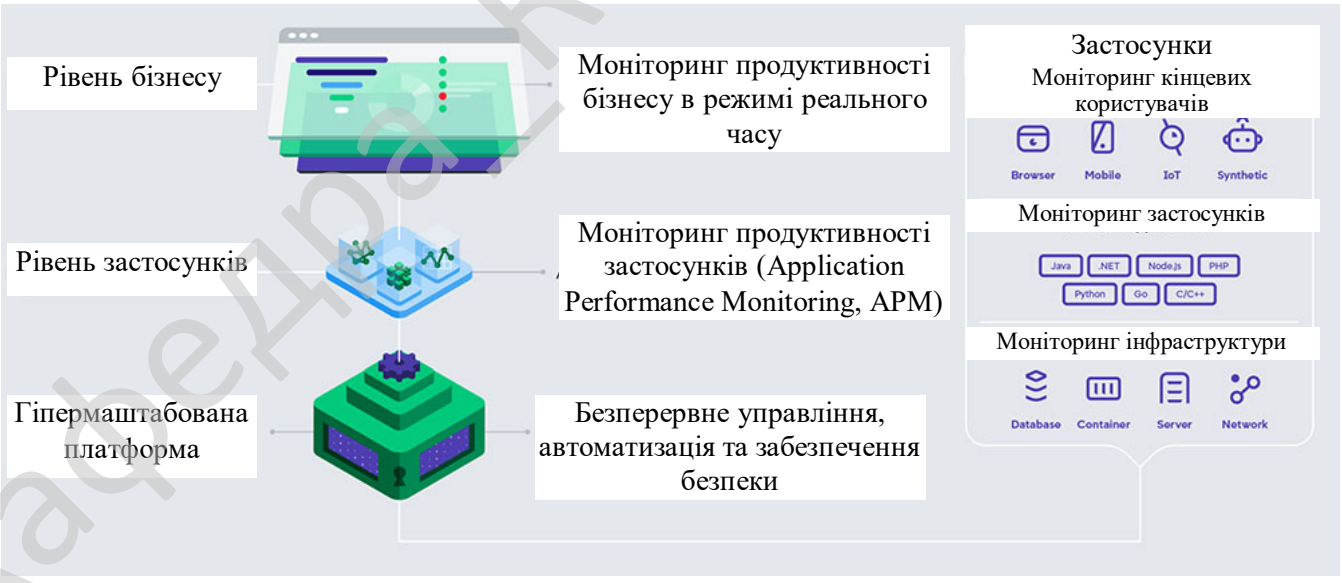

Рисунок 3.1 – Структурна схема системи

КафедраКБПЗ-2021рік

Системи АРМ гарантують, що застосунки, з якими працюють користувачі, відповідають стандартам продуктивності й функціональним вимогам, тим самим забезпечуючи необхідний рівень користувацької взаємодії (UX).

Структурна схема системи відображає наскрізний моніторинг продуктивності застосунків. На сьогодні програмне забезпечення усе більше впроваджується в наше повсякденне життя. Раніше багато процесів доводилося виконувати вручну, і для їхнього рішення було потрібно особиста присутність відповідного IT-фахівця. Зараз же більшість бізнес-завдань можуть бути виконані віддалено, шляхом простої взаємодії із ПК, планшетом або смартфоном. Усе більше компаній надають різні послуги за допомогою мобільних і десктопних застосунків.

Наприклад, уже нікого не дивують сервіси виклику таксі, доставки їжі, інтернет-банкингу і т.д. Слід зазначити, що лояльність клієнтів до бренда прямо залежить від якості бізнес-застосунків і досвіду взаємодії (UX, user experience). Якщо, приміром, користувач розв'язав замовити таксі за допомогою мобільного застосунка, але в нього застосунок мимовільний закрилося з помилкою, то з великою часткою ймовірності він скористається послугою конкурента. А якщо це буде повторюватися часто, те компанія назавжди втратить цього клієнта.

Тому основною метою АРМ-рішень є наскрізний моніторинг роботи бізнес-застосунків. Правильно підібраний інструмент АРМ допоможе виявити й указати на джерело проблеми, перейти з реактивного на проактивний підхід вирішення проблем, і показати користувачеві як проблеми з застосунком впливають на бізнес.

Основна перевага АРМ-рішень полягають у тому, що вони дають можливість користувачам побачити те, що відбувається усередині застосунка. Інструменти АРМ можуть відслідковувати кожний запит, починаючи від користувацького інтерфейсу (браузер або мобільний застосунок), закінчуючи класами/методами, викликами баз даних і сторонніх сервісів.

Кафедра_КБПЗ_2021рік

Інструменти моніторингу застосунків допомагають будувати інтерактивну карту взаємодій, яка завжди залишається актуальної й обновляється в реальному часі.

Крім цього, АРМ-рішення установлює базові показники (бейслайни) по кожній, що збирається або настроєній метриці, а також відслідковує відхилення від них. Це допомагає відслідковувати можливі проблеми з тим або іншим застосунком, які могли вплинути на якість роботи користувачів системи.

Характерною рисою рішень є функція аналізу аномалій, яка суттєво спрощує діагностику більшості виникаючих проблем.

У силу того, що агрегує інформацію, отриману з різних джерел (Бази Даних, Сервера застосунків, Логі, Браузери й Мобільні застосунки) – він є єдиним джерелом істини (SSOT) для всіх команд, які розробляють і обслуговують застосунка. У результаті, користувачі одержують ряд незаперечних переваг:

– Глибока деталізація інформації, із вказівкою самих повільних частин застосунка, дозволяє значно скоротити час необхідний на пошук причини проблеми, і дозволяє зменшити кількість персоналу, яка потрібно для діагностики.

– Фахівці з керування IT-службами бачать як інфраструктура впливає на застосунок, і можуть відразу зрозуміти яка саме команда розв'яже дану проблему, без залучення інших команд.

– Розроблювачі витрачають набагато менше часу на виявлення й усунення неполадок у роботі застосунків, одержуючи, таким чином, набагато більше вільного часу на розробку нового функціонала.

– Зручність використання аналітики для менеджменту застосунків, яка дозволяє корелювати бізнес-метрики KPI з урахуванням стану того або іншого застосунка.

Кафедра_КОПЗ_2021рік

– Порівняння різних метрик (як бізнес, так і технічних) у різних релізах, дозволяє зрозуміти наскільки покращилася робота застосунка з виходом нової версії.

Сьогодні пропонується два варіанти рішень по моніторингові продуктивності застосунків – у вигляді SaaS і On-Premises версій. У випадку з SaaS варіантом, усі питання відносно розгортання, обслуговування й масштабування платформи бере на себе сам вендор. При цьому ціни на ліцензії SaaS і On-Prem пропозицій абсолютно однакові для обох варіантів розгортання.

APM-рішення, які пропонуються на ринку IT-послуг, практично універсальні. Вони здатні забезпечити наскрізний моніторинг будь-яких бізнес-застосунків, які написані на мовах Java, .NET, Python, C/C++, Node.js і багатьох інших. Не може не радувати й платформна сумісність цих інструментів моніторингу. Це й мобільні застосунки для IOS і Android, і різні бази даних, і більшість сучасних браузерів.

В APM-рішень є можливість витягати необхідні показання роботи застосунків прямо з, що виконуються класів і методів. Це дає можливість будувати різні аналітичні панелі – дешборди, які показують стан продуктивності застосунків у реальному часі. Для кращого розуміння, приведу вам наступний приклад:

У компанії, що працює в сфері ecommerce, є власний застосунок для онлайн-покупок. За допомогою APM-рішення ми витягаємо передані або вертаються значення (інформацію) з коду застосунка по оплачених товарах, які були куплені користувачами за допомогою даного застосунка. На підставі отриманої інформації ми можемо побудувати зведену інформаційну панель, у якій буде показаний виторг за потрібний період часу. Далі, можна зв'язати й проаналізувати показники виторгу, зрівнявши їх з технічним станом застосунка, тієї або іншою версією релізу, певною порою року і т.д.

Кафедра_КБПЗ_2021рік

3.3 Розробка функціональної схеми

У нашій роботі систему моніторингу продуктивності застосунків на базі АРМ реалізуємо на прикладі банківської мережі. На початковому етапі дослідження було виконано структурно-функціональне моделювання розподілу елементів інформаційні системи (ІС) і інформаційні потоки між ними на різних рівнях. На першому етапі побудована модель розміщення елементів інформаційного середовища ІТ-інфраструктури системи моніторингу продуктивності застосунків на базі АРМ, що також ураховує взаємозв'язок рівнів ієрархії (центр, філія, відділення) і груп банківських систем (Back office, Middle office, Front office). Результати аналізу моделі відображені на функціональній схемі (рисунок 3.2)

На функціональній схемі наведені основні блоки системи, яка досліджується. Тобто системи з розподіленими ресурсами у вигляді застосунку середи ІТ-інфраструктури системи моніторингу продуктивності застосунків на базі АРМ, ефективність якої оцінюють. Така система наведена на прикладі філії банку.

Front-office – це комплекс програмно-апаратних засобів, що підвищують ефективність спілкування. Це спеціалізовані системи, що автоматизують роботу співробітників, які спілкуються з "зовнішнім миром", які допомагають їм у повсякденній діяльності. Іншою стороною взаємодії учасників «телефонних» відносин є системи, що автоматизують внутрішні взаємодії в компанії. Back-office, – це "каркас" підприємства. Для ефективної роботи всієї фірми обидві сторони (Front-office і Back-office) повинні взаємодіяти один з одним по оптимальних алгоритмах.

Front-office відповідає за відносини із клієнтами й ухвалює рішення щодо здійсненні угод, він не повинен мати можливість здійснювати інформаційний облік або аналіз результативності своїх операцій. Middle-office відповідає, зокрема, за моніторинг і аналіз ризику, а також розробку мер активного

Кафедра_КБПЗ_2021рік

керування. Back-office здійснює рутинні операції ведення внутрішнього й зовнішнього обліку, розрахунку податків і складання звітності. Незважаючи на необхідність строгого поділу функцій front-office, middle-office і back-office, першочерговою задачею є забезпечення доступу до інформації всім учасникам процесу керування ризиками.

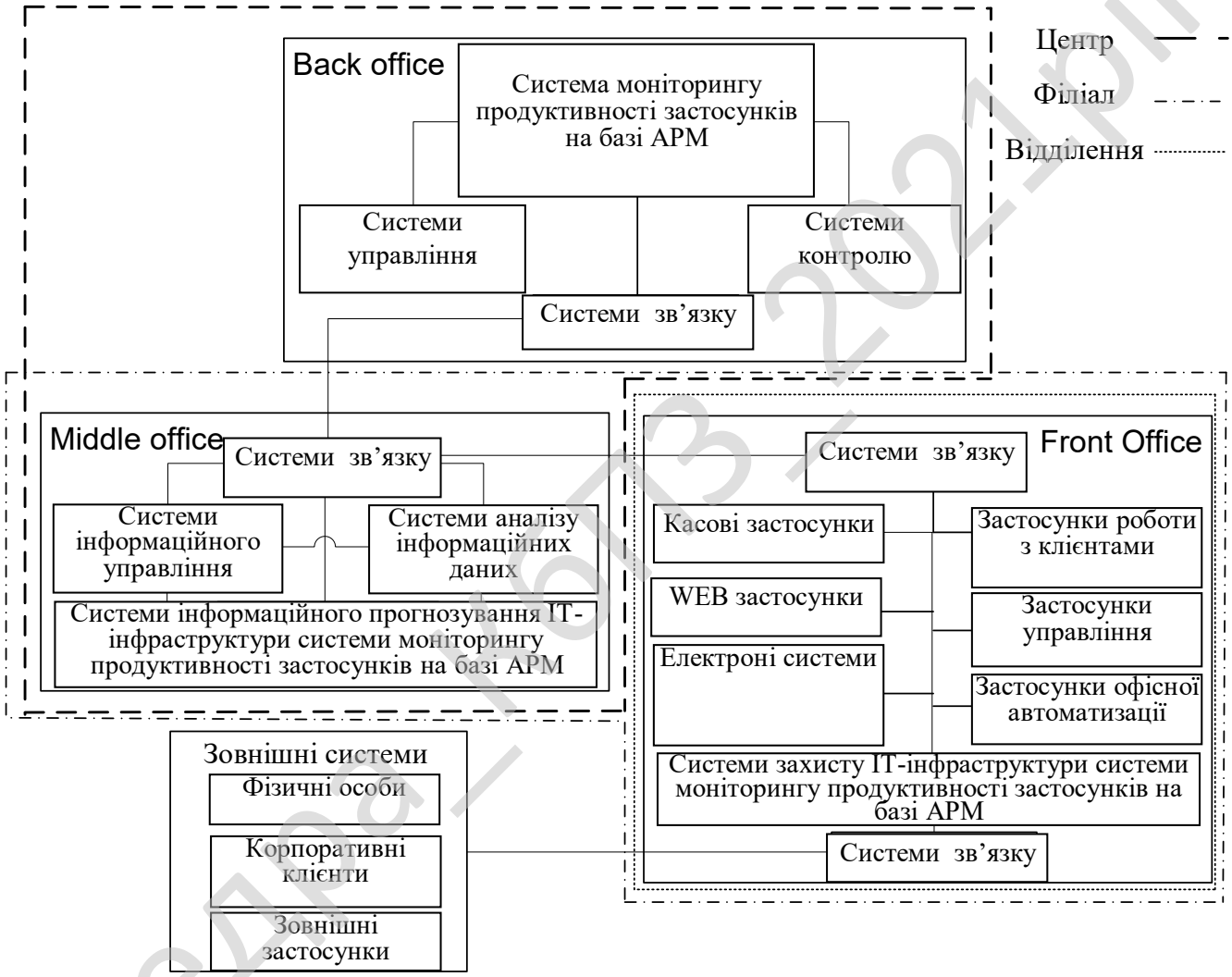

Рисунок 3.3 – Структурна схема розміщення елементів ІС

Автоматизація Front-офісних операцій

При автоматизації Front-офісних операцій сучасні системи дозволяють:

– Оптимізувати роботу з різними видами Front-офісних операцій. У системі передбачений готові бізнес процеси обробки й обліку різних операцій,

Кафедра – ІБТЗ – 2021 рік

наприклад валютні контракти, включаючи валютні опціони; операції по залученню/розміщенню ресурсів, включаючи короткострокове кредитування/фінансування, застави й так далі.

– Можливість використання відкритих інтерфейсів для автоматичного імпорту операцій з будь-якої зовнішньої системи – наявність відкритих інтерфейсів для кожного типу операції дають можливість імпортувати інформацію про угоду з будь-якого зовнішнього джерела, включаючи текстовий файл.

– Можливість починати обробку Front-офісної операції на підставі заявки.

– Можливість використовувати різні методи (документообіг) обробки заявки на операцію. Існує можливість організувати документообіг заявки в системі. Електронний документообіг має на увазі автоматичне формування повідомлень уповноваженим співробітникам, використання різної бізнес логіки.

– Ефективна інтеграція з функціями Middle-офісу і Back-офісу – передбачена можливість інтеграції Front-офісних функцій з функціями Middle-офісу й Back-офісу, наприклад відсутня необхідність повторного уведення інформації про угоду при формуванні платіжних документів.

– Формування різних операційних звітів – будь-які необхідні операційні звіти, наприклад, Trade Slips, журнали угод і так далі, можуть автоматично формуватися й роздруковуватися в системі.

– Формування різних аналітичних звітів у розрізі дилерів, портфелів, валюти, угод і так далі – у системі передбачена можливість використання різних екранних форм або звітів для проведення аналізу ведення Front-офісних операцій з використанням різних аналітичних зрізів, наприклад по контрагентах, по дилерах, та інше.

Автоматизація Middle-офісних операцій

На жаль, у більшості українських банків діяльність відд ілів Middle-офісу дотепер майже не автоматизована. Основна аналітична звітність складається за допомогою не призначених для цього додатків. Відсутність єдиної інтегрованої

Кафедра_КОПЗ_2021рік

системи для обліку всіх Front-офісних операцій значно обмежує можливості співробітників Middle-офісу по керуванню й аналізу діяльності банку.

Сучасні системи дозволяють:

– Оптимізувати внутришньоінформаційні потоки між центрами виникнення прибутку й витрат.

– Управляти параметризацією всіх банківських операцій, у тому числі існує можливість додавання необмеженого числа аналітичних параметрів по кожній угоді.

– Управляти й здійснювати моніторинг кредитно-інвестиційного портфеля банку в режимі реального часу.

– Становити різні аналітичні звіти для оцінки й прогнозування діяльності банку.

– Становити різні аналітичні звіти для оцінки й прогнозування діяльності банків-конкурентів.

– Проектувати й управляти необмеженим числом лімітів, нормативів, у тому числі існує можливість аналізувати використання лімітів і нормативів у режимі реального часу.

– Переоцінювати відкриті позиції.

– Управляти ризиками: VAR-аналіз, стратегії хеджування, інші звіти.

Автоматизація Back-офісу

Автоматизація Back-офісу дозволить уникнути рутинного дублювання введення інформації. Використання сучасної системи дозволить автоматизувати наступні функції:

– Автоматичне формування бухгалтерських проводок по кожній операції на підставі раніше створених шаблонів.

– Можливість проектувати різні типи документообігу для тверджень операцій.

– Автоматичне формування розрахункових інструкцій для генерації SWIFT файлів(підтримка множини SWIFT форматів).

Кафедра_КОПЗ_2021рік

– Проектування й формування пакетів документації по кожному типі операції.

– Автоматичне формування підтверджень по кожній угоді, облік отриманих підтверджень від контрагентів.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Відповідно до методичних рекомендацій розроблення графічної частини кваліфікаційної бакалаврської роботи розглянемо розроблену діаграму процесів яка зображена на рисунку 3.4

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, поток 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.

Кафедра_КоПЗ_2021рік

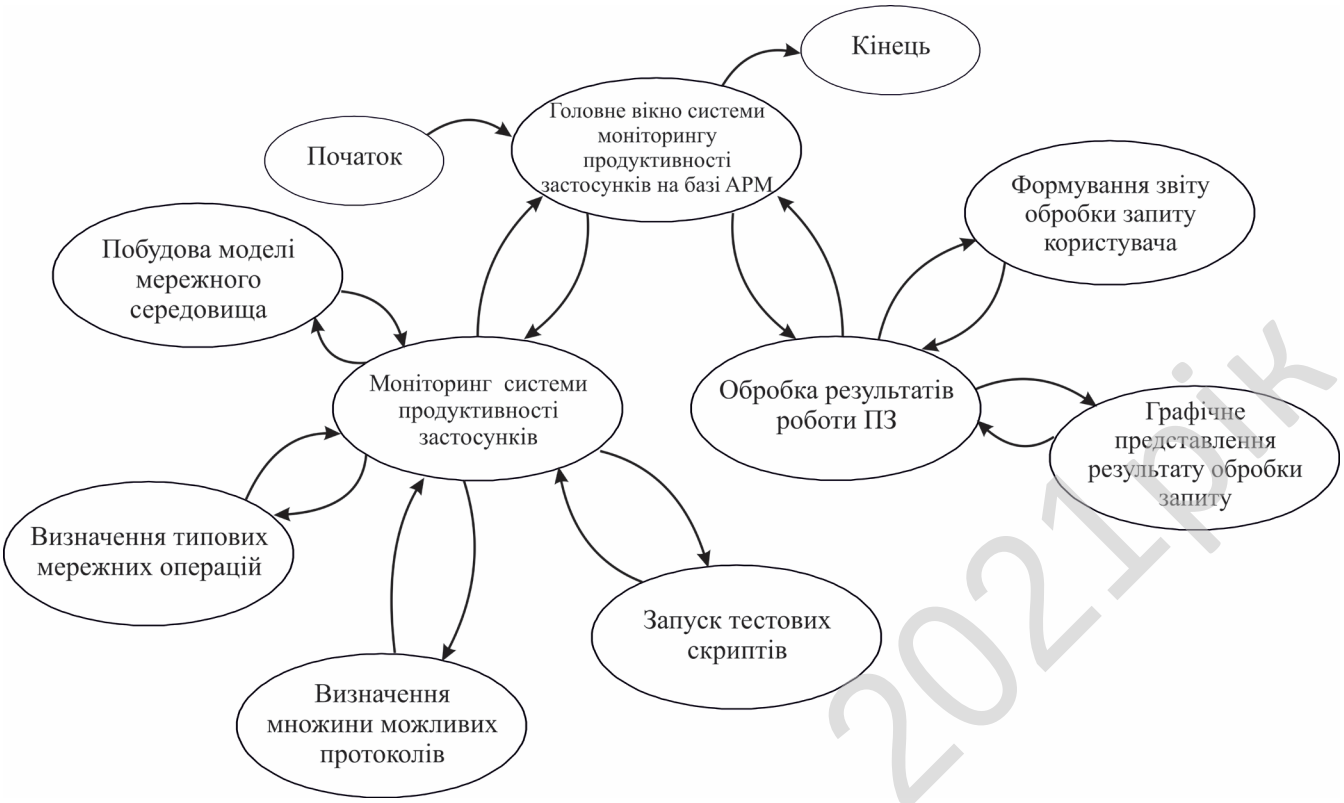

Рисунок 3.4– Діаграма взаємодії процесів

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.

Кафедра_КБПЗ_2021рік

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо алгоритм роботи основної програми. Його блок-схема зображена на рисунку 4.1. З рисунку видно, що після запуску програми спочатку відбувається ініціалізація динамічних структур даних та змінних, визначається множини та показники ефективності. вивід основного вікна програми.

Потім здійснюється визначення методики розрахунку з послідуючим проведенням розрахунку значень показників ефективності та перетворення показників до потрібної однорідності з обчисленням значень узагальненого показника ефективності.

Після виконаних дій проходе виклик підпрограми що зображено на рисунку 4.2 та детально розглянемо реалізацію обробки запитів. Далі проходе змінна обробка даних системи.

Розглянемо нижче функцію, яка дозволяє побудувати схему мережі додатку середи IT-інфраструктури з застосуванням технології АРМ.

```
function TMainfm.ADrawScheme(X: Integer) : integer;
begin
  Inc(X);
//додаємо наступний елемент
  if KolScheme < X then
  begin
    PaintScheme := false;//схема не побудована
    if Manager.ElementsCount > Manager.CountVakMest then
    begin
      ShowMessage('Некоректно з'єднані пристрої!');
      Manager.DeleteAllSchemea;
    end;
  end
```

Кафедра_КОПЗ_2021рік

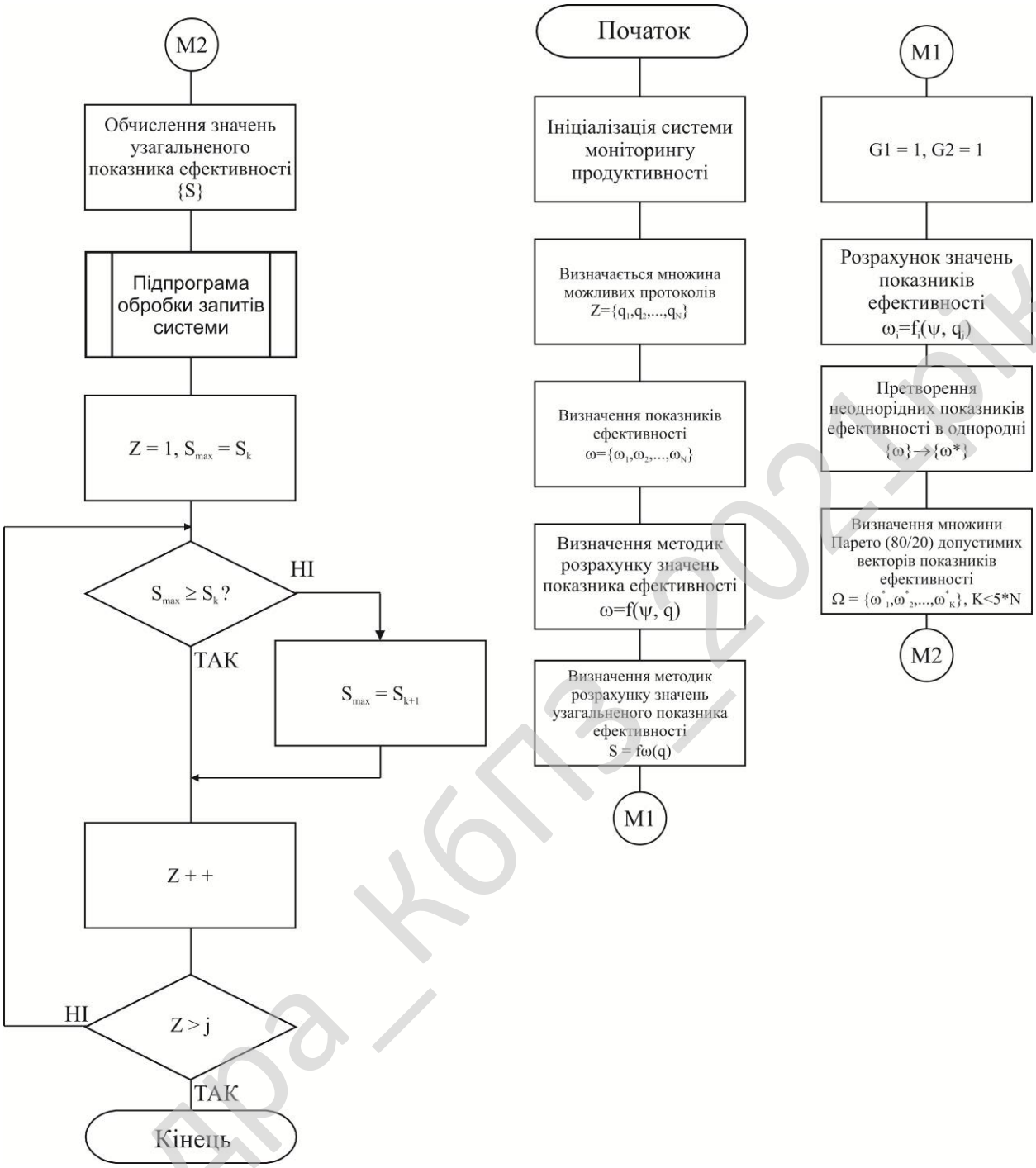

Рисунок 4.1 – Блок-схема основної програми

```
  else
  begin
       PaintScheme := true;// схема коректно побудована
  end;
  Status.Panels[3].Text := IntToStr(Manager.CountVakMest);
  Result := X;
end;
```

КафедРа_КБПЗ_2021ОК

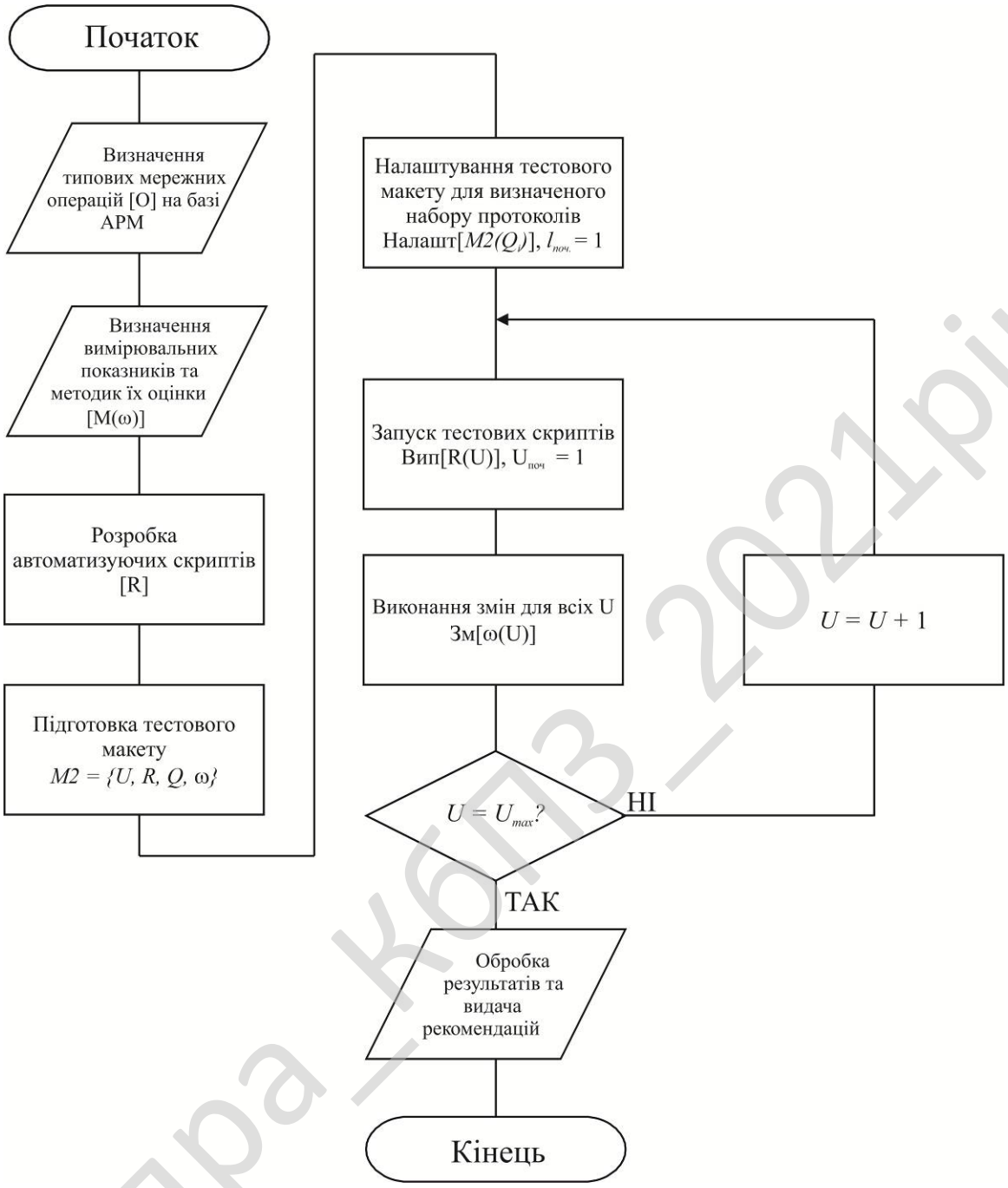

Рисунок 4.2 – Блок-схема роботи підпрограми

Наведемо функцію оцінки ефективності розподіленої мережі.

```
procedure TEfficForm.btnCloseSharesClick(Sender: TObject);
var
  OS:Boolean;
  FLibHandle : THandle;
  Name9x:array [0..12] of Char;
  NameXP:PWChar;
```

Кафедра_КБПЗ_2021рік

```
  i:Integer;
  ShareName: String;
begin
  if not IsXP(OS) then Close;
//Визначаємо тип системи
  if lbxShares.Items.Count = 0 then Exit;
  for i:= 0 to lbxShares.Items.Count -1 do
    if lbxShares.Selected[i] then Break;
//Шукаємо обраний елемент
  if not lbxShares.Selected[i] then Exit;
//Якщо не знайдений ідемо
  ShareName := lbxShares.Items.Strings[i];
  if OS then begin //Код
    FLibHandle := LoadLibrary('NETAPI32.DLL');
    if FLibHandle = 0 then Exit;
    @NetShareDelXP := GetProcAddress(FLibHandle,'NetShareDel');
    if not Assigned(NetShareDelXP) then //Перевірка
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    i:= SizeOf(WideChar)*256;
    GetMem(NameXP,i);  //Виділяємо пам'ять під змінну
    StringToWideChar(ShareName,NameXP,i);
//Перетворимо в PWideChar
    NetShareDelXP(nil,NameXP,0);
//Видаляємо ресурс
    FreeMem(NameXP);
//Звільняємо пам'ять
  end else begin
    FLibHandle := LoadLibrary('SVRAPI.DLL');
    if FLibHandle = 0 then Exit;
    @NetShareDel := GetProcAddress(FLibHandle,'NetShareDel');
    if not Assigned(NetShareDel) then //Перевірка
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    FillChar(Name9x, SizeOf(Name9x), #0); //Очищаємо масив
    move(ShareName[1],Name9x[0],Length(ShareName)); //Заповнюємо масив
    NetShareDel(nil,@Name9x,0); //Видаляємо ресурс
  end;
```

Кафедра_КбПЗ_2021рік

```
FreeLibrary(FLibHandle);end;
```

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

– набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;

– набір клієнтів, які використовують сервіси, що надаються серверами;

– мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

Кафедра_КбПЗ_2021рік

– рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;

– прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;

– рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

– модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

– модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузерами.

Кафедра_КОПЗ_2021рік

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка ПЗ. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення –звичайний браузер. Це позбавляє його необхідності завантажувати та інсталювати спеціальні програми (хоча інколи така необхідність все-таки виникає).

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних.

Кафедра_КОПЗ_2021рік

Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються. Слід згадати про технології EJB (Enterprise Java Beans), CORBA, а також про .NET – порівняно нову ініціативу компанії Microsoft. Для зберігання даних та їх передачі часто використовується так звана розширювана мова розмітки XML (Extensible Markup Language).

Подійно-орієнтована архітектура (Event-driven architecture, надалі EDA) – шаблон архітектури програмного забезпечення, який призначений для створення подій, їх виявлення, споживання і реагування на них.

Подія може бути визначена як значна зміна стану. Наприклад, коли споживач купує автомобіль, стан автомобіля змінюється з "на продаж" до "продано". Архітектура системи дилера автомобілів може трактувати цю зміну стану як подію, поява якої може стати відомою іншим програмам даної архітектури.

Кафедра_КПЗ_2021рік

З формальної точки зору, те, що виробляється, публікується, поширюється, виявляється і споживається (як правило, асинхронно) є повідомленням, яке називають сповіщенням про подію (або нотифікацією), а не самою подією, яка є зміною стану, що викликає появу повідомлення.

Події не подорожують, вони просто відбуваються. Проте термін подія часто використовується метонімічно для позначення самого нотифікаційного повідомлення, що може призвести до певної плутанини.

Цей архітектурний шаблон може застосовуватися при проектуванні і реалізації ПЗ і систем, які передають події між слабкозв'язаними компонентами програмного забезпечення і сервісами (службами).

Подійно-орієнтована система як правило складається з емітерів подій (або агентів) і споживачів подій (або стоків).

Стоки несуть відповідальність за здійснення реагування на появу події. Реакція не завжди може бути повністю забезпечена самим стоком. Наприклад, стік, може бути відповідальним лише за фільтрацію, трансформацію і відправку події до іншого компонента або він може забезпечити повністю самостійну реакцію на таку подію. Перша категорія стоків може бути заснована на традиційних компонентах, таких як проміжне програмне забезпечення, орієнтоване на обробку повідомлень (message oriented middleware, MOM), в той час, як друга категорія стоків (самостійна реакція в режимі он-лайн) може вимагати більш придатної платформи (фреймворку) для виконання транзакцій.

Розробка ПЗ і систем в подійно-орієнтованій архітектурі дозволяє їм бути сконструйованими способом, який більш відповідає вимогам до їх створення, оскільки такі системи в більшій мірі пристосовуються до непередбачуваних і асинхронних середовищ.

Подійно-орієнтована архітектура (EDA) може доповнювати сервісно-орієнтовану архітектуру (SOA), оскільки сервіси (служби) можуть бути активовані тригерами, які ініціюються при настанні подій.

Кафедра_КОПЗ_2021рік

Ця парадигма особливо корисна, коли стік не забезпечує власного виконання будь-яких дій.

Подійно-орієнтована SOA (SOA-2) розвиває архітектури SOA і EDA для забезпечення більш глибокого і надійного рівня сервісів за рахунок використання раніше невідомих причинно-наслідкових зв'язків, щоб сформувати новий шаблон подій. Цей новий шаблон бізнес-аналітики дає поштовх до небаченого раніше зростання рівня автоматизації підприємства за рахунок привнесення додаткової цінної інформації в описану раніше модель діяльності.

Обчислювальна техніка та сенсорні пристрої (сенсори, датчики, контролери) можуть виявляти зміни стану об'єктів або умов і створювати події, які потім можуть бути оброблені сервісом (службою) або системою.

Подія може складатися з двох частин: заголовка події та тіла події. Заголовок події може включати в себе інформацію таку як, наприклад, назва події, часова мітка події і тип події. Тіло події – це частина, яка описує факт, що стався в дійсності. Тіло події не слід плутати з шаблоном або логі кою, яка може бути застосована як реакція на саму події.

Архітектура, керована подіями, складається з чотирьох логічних рівнів (шарів). Вона починається з виявлення факту, його технічного подання у формі події і закінчується непустою множиною реакцій на цю подію.

Генератор подій.

Першим логічним шаром є генератор подій, який виявляє факт і представляє цей факт подією. Оскільки фактом може бути практично все, що може бути сприйнято, то ним може бути і генератор подій. Наприклад, генератором може бути клієнт електронної пошти, система електронної комерції або певний тип датчика.

Перетворення різних даних, отриманих від датчиків, в єдину стандартизовану форму даних, які можуть бути оцінені, є основною проблемою при розробці та реалізації цього шару. Однак, враховуючи, що подія є строго

Кафедра_КБПЗ_2021_рік

декларативною, можна легко застосовувати будь-які операції трансформації, тим самим усуваючи необхідність забезпечення високого рівня стандартизації.

Канал подій.

Канал подій - це механізм, через який інформація від генератора подій передається до обробника подій (event engine) або стоку.

Це може бути з'єднання TCP/IP або вхідний файл будь-якого типу (простий текст, формат XML, e-mail тощо). В один і той же час може бути відкрито кілька каналів подій. Як правило, оскільки обробник подій повинен працювати в режимі, наближеному до реального часу, канали подій зчитуються асинхронно. Події зберігаються в черзі, очікуючи наступної обробки механізмом обробки подій.

Механізм обробки подій.

Механізм обробки подій (event processing engine) є місцем, де подія ідентифікується і вибирається відповідна реакція на нього, яка потім виконується. Це також може призвести до породження ряду тверджень. Якщо подія, яка надійшла до механізму обробки подій, є наприклад такою «Запаси продукту ID досягли нижнього допустимого рівня», це може ініціювати, наприклад, такі реакції як «Замовити продукт ID» і «Сповістити персонал».

Наступна подійно-орієнтована дія (післядія).

Щодо того, як можуть проявлятися наслідки події, слід відмітити, що вони можуть проявитись багатьма різними способами і у різноманітних формах (наприклад, повідомлення електронної пошти, надіслане комусь, або ПЗ, що виводить деяке попередження на екран). Залежно від рівня автоматизації, який забезпечується стоком (механізмом обробки подій), ці дії можуть виявитись зайвими.

Є три основні стилі обробки подій: простий, потоковий і складний. Часто ці три стилі використовуються спільно у розвинутій подійно-орієнтованій архітектурі.

Кафедра_КПЗ_2021рік

Проста обробка подій.

Проста обробка подій стосується подій, які безпосередньо належать до специфічних вимірних змін умов. У випадку простої обробки подій, мають справу з появою відомих подій, що ініціюють післядію (післядії). Проста обробка подій зазвичай використовується для управління потоком робіт в реальному часі, скорочуючи тим самим час затримки і вартість робіт.

Наприклад, прості події можуть створюватись (породжуватись) датчиком, що виявляє зміну тиску в шині або температуру навколишнього середовища.

Обробка потоку подій.

При обробці потоку подій (event stream processing, далі ESP) відбуваються як звичайні, так і відомі події. Звичайні події (заявки, передачі RFID) перевіряються на те, чи є вони відомими, і передаються інформаційним передплатникам. Обробка потоку подій зазвичай використовується для управління потоком інформації в реальному часі і на рівні підприємства, що дозволяє своєчасно приймати рішення.

Обробка складних подій.

Обробка складних подій (Complex event processing (CEP)) дозволяє за шаблонами простих і звичайних подій проводити аналіз того, чи наступила складна подія. Обробка складних подій полягає в оцінюванні взаємного впливу подій і в наступному виконанні дій. При цьому, типи подій (відомих або звичайних) можуть перетинатись, а події можуть виникати протягом тривалого періоду часу.

Кореляція подій може бути причинною, тимчасовою або просторовою. CEP вимагає використання складних інтерпретаторів подій, визначення і підбору шаблонів подій, а також відповідних кореляційних методів. Обробка складних подій зазвичай використовується для виявлення і реагування на аномальну поведінку, загрози і можливості у бізнесі.

Кафедра_КБПЗ_2021рік

4.2 Захист розробленого програмного забезпечення

Дані в програмі захищаються за допомогою використання алгоритму шифрування ДСТ Р 34.10-2012, який використовує перетворення у групі точок на еліптичній кривій в простому полі Галуа. У системі шифрування/дешифрування як параметри розглядається еліптична крива E_p (a,b) і точка G на ній. Учасник В вибирає закритий ключ n і обчислює відкритий ключ $P_B = n \times G$. Щоб зашифрувати повідомлення P_m використовується відкритий ключ одержувача В P_B. Учасник А вибирає випадкове ціле позитивне число k і обчислює зашифроване повідомлення C_m, що є точкою на еліптичній кривій.

$$C_m = \{k \times G, P_m + k \times P_B\} \quad (4.1)$$

Щоб дешифрувати повідомлення, учасник В множить першу координату точки на свій закритий ключ і віднімає результат від другої координати:

$$P_m + k \times P_B - n_B \times (k \times G) = P_m + k \times (n_B \times G) - n_B \times (k \times G) = P_m \quad (4.2)$$

Учасник А зашифрував повідомлення P_m додаванням до нього $k \times P_B$. Ніхто не знає значення k, тому, хоча P_B і є відкритим ключем, ніхто не знає $k \times P_B$. Супротивнику для відновлення повідомлення доведеться обчислити k. Зробити це буде нелегко. Одержувач також не знає k, але йому як підказку посилається $k \times G$. Помноживши $k \times G$ на свій закритий ключ, одержувач одержить значення, що було додано відправником до незашифрованого повідомлення. Тим самим одержувач, не знаючи k, але маючи свій закритий ключ, може відновити незашифроване повідомлення.

$p>3$ є непарним числом. Еліптична крива GF(p) над F_p виражена формулою виду

$$y^2 = x^3 + a \cdot x + b \quad (4.3)$$

де $a, b \in F_p$ та $4 \cdot a^3 + 27 \cdot b^2 \neq (\mathrm{mod}\ p)$

Додавання точок.

$P = (x_1, y_1) \in E(F_p)$ та $Q = (x_2, y_2) \in E(F_p)$, де $P \neq \pm Q$.

Тоді $P + Q = (x_3, y_3)$, де:

Кафедра_КОПЗ_2021рік

$$x_3 = (\frac{y_2 - y_1}{x_2 - x_1})^2 - x_1 - x_2 \text{ та } y_3 = (\frac{y_2 - y_1}{x_2 - x_1}) \cdot (x_1 - x_3) - y_2 \quad (4.4)$$

Подвоєння точки.

Припустимо $P = (x_1, y_1) \in E(F_p)$, де $P \neq -P$. Тоді $2P = (x_3, y_3)$, де

$$x_3 = (\frac{3 \cdot x_1^2 + a}{2 \cdot y_1})^2 - 2 \cdot x_1 \text{ та } y_3 = (\frac{3 \cdot x_1^2 + a}{2 \cdot y_1}) \cdot (x_1 - x_3) - y_1 \quad (4.5)$$

Також використовується хеш функція H() визначена в алгоритмі ГОСТ 28147-89. числа p, q, a, y – є відкритими для усіх користувачів мережі , число x є секретним.

Щоб підписати деяке повідомлення m потрібні такі кроки :

1) Користувач А генерує випадкове число k, $k<q$.

2) Користувач А обчислює значення r, s за формулами :

$$r = (a^K \bmod p) \bmod q, \quad (4.6)$$

$$s = (x * r + k * H(m)) \bmod p. \quad (4.7)$$

Якщо $H(m)$ кратне q, то $H(m) \bmod q = 1$.

3) Якщо $r = 0$ то (беремо інше k) перехід до 1).

4) Цифровий підпис являє собою два числа $r \bmod 2^{256}$ та $s \bmod 2^{256}$.

Користувач А відправляє користувачу В повідомлення m із цим цифровим підписом.

Користувач В перевіряє отриманий підпис виконуючи наступні кроки :

1) $$V = H(m)^{q-2} \bmod q; \quad (4.8)$$

$$Z1 = (s * v) \bmod q; \quad (4.9)$$

$$Z2 = ((q - r) * v) \bmod q; \quad (4.10)$$

$$U = ((a^{Z1} * y^{Z2}) \bmod p) \bmod p. \quad (4.11)$$

2) Якщо $u = r \, 2^{256}$, то підпис істиний.

Кафедра_КОПЗ_2021рік

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програма має простий та інтуїтивно зрозумілий інтерфейс, який зображений на рисунку 5.1. З рисунку головного вікна можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

– Функціональних кнопок ПЗ: Запуску тестування; Збереження результатів; Виведення вікна мережних операцій; Виведення вікна перегляду мережних протоколів; Перегляду ефективності роботи.

– Верхнього меню: Файл; Тестування; Параметри; Довідка.

– Навігаційного меню.

– Розділу обрання групи мережних компонент.

– Виведення результату роботи системи у графічному вигляді (іконки).

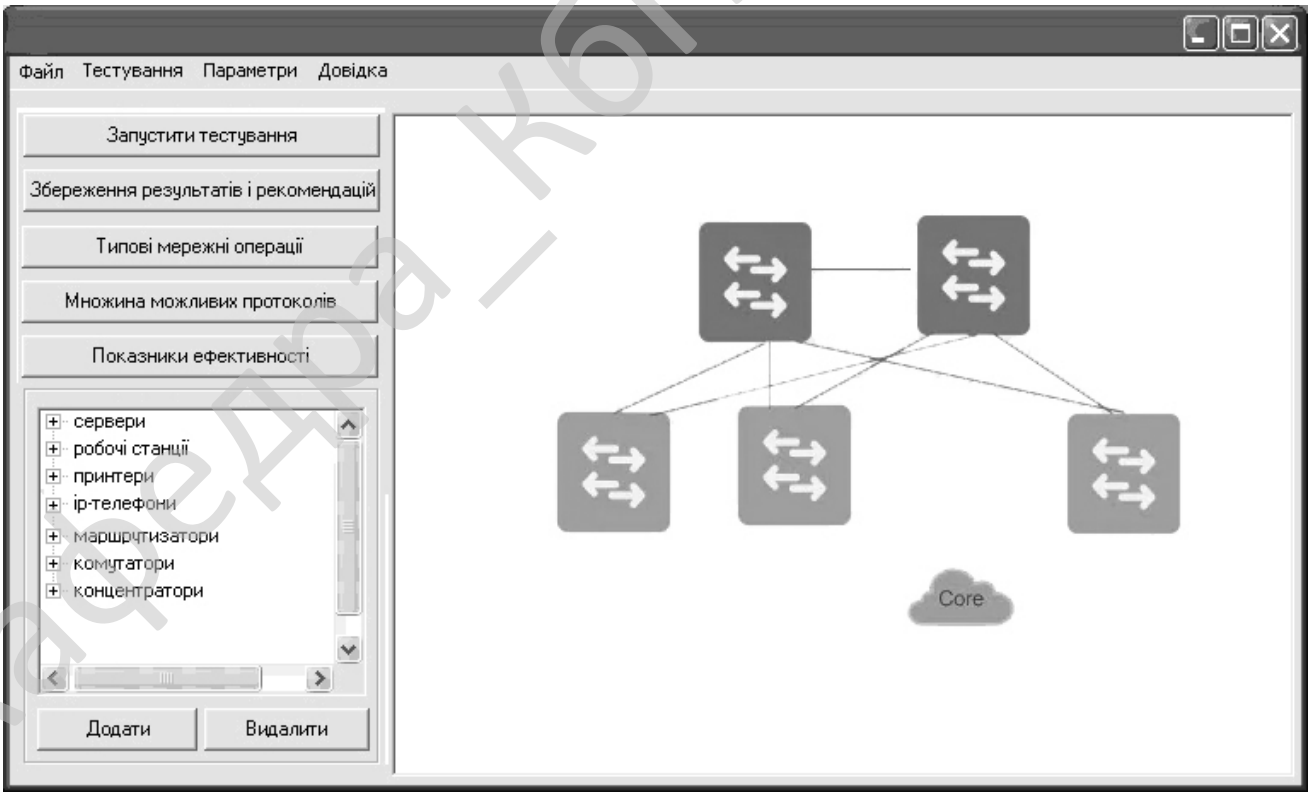

Рисунок 5.1 – Головне вікно ПЗ

Кафедра_КбПЗ_2021рік

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення. Розроблена програма має дуже простий і зрозумілий інтерфейс з 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.

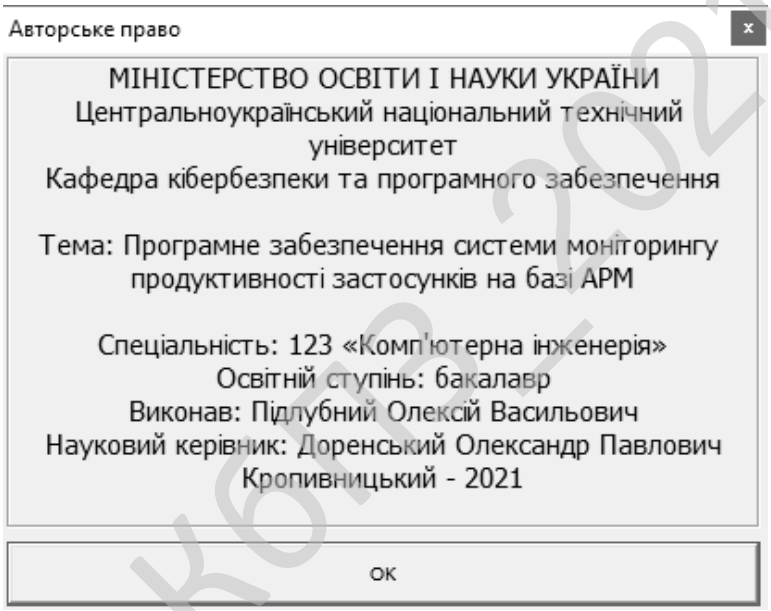

Рисунок 5.2 – Авторське право

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.

Кафедра_КБПР_2021рік

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання кваліфікаційної бакалаврської роботи, призначено для системи моніторингу продуктивності застосунків на базі APM.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем моніторингу продуктивності застосунків на базі APM.

– Досліджена система моніторингу продуктивності застосунків на базі APM.

– На основі отриманих результатів досліджень створена програмна реалізація системи моніторингу продуктивності застосунків на базі APM.

Розроблені під час виконання кваліфікаційної бакалаврської роботи алгоритми дозволяють успішно вирішувати завдання моніторингу продуктивності застосунків на базі APM.

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 створені 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 Sydney. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи моніторингу продуктивності застосунків на базі APM. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на

Кафедра_КБПЗ_2021рік

загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТ Р 34.10-2012.

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_2021рік

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інформаційні технології. Взаємозв'язок відкритих систем. Базова еталонна модель. Частина 1. Базова модель (ISO/IEC 7498-1:1994, IDT):ДСТУ ISO/IEC 7498-1:2004 – [Чинний від 2006–01–01]. – Київ: Держспоживстандарт України, 2007. – 67 с. – (Національний стандарт України).

2. Карманов И.Н. Измерения, испытания, контроль. Метрология и метрологическое обеспечение: учеб. пособ. / И.Н. Карманов, Н.А. Мещеряков, О.К. Ушаков. – Новосибирск: СГГА, 2006. – 184 с.

3. Каспина Т.В. Экономика и управление приборостроительным произвоздством: учебн. пособ. для высших учебных заведений / Т.В. Каспина, Н.Н. Лямина. – М.: ИЦ "Академия", 2008. – 240 с.

4. Клюев В.В. Неразрушающий контроль и диагностика. Справочник, 2-е изд., перераб. и доп. / В.В. Клюев – М: Машиностроение, 2003. – 656 с.

5. Клюня В.Л. Основы экономической теории / В.Л. Клюня, Н.В. Черченко. – Минск: Минск, 2006. – 238 с.

6. Коваленко А.С. Разработка структуры базы данных интегрированной информационной системы / А.С. Коваленко, А.В. Коваленко // Информационные технологии и защита информации в информационно-коммуникационных системах: монографія / Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2015. – С. 54-64.

7. Кожанова А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / О.А. Смірнов, А.С. Кожанова, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2013. – Вип. 6(113). – С. 255-257.

8. Коваленко А.С. Задачи распознавания ситуаций в ERP системах / А.В. Коваленко ., А.А. Смірнов, А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2014. – Вип. 4(120). – С. 161-164.

Кафедра_КБПЗ_2021рік

9. Коваленко А.С. Підсистема технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко , О.А.Смірнов, О.В. Коваленко // Системи озброєння і військова техніка.– Х.: ХУПС, 2014. – № 1(37). – С. 126-129.

10. Коваленко А.С. Анализ эффективности использования экспертной системы технической диагностики с традиционной структурой / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 2(38). – С. 106-108.

11. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.

12. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.

13. Коваленко А.С. Структура системи технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Кіровоградського національного технічного університету / техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кіровоград: Вид-во КНТУ, 2014. – Вип. 27. – С. 245-251.

14. Коваленко А.С. Дослідження будови інтегрованої інформаційної системи та її елементів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 4(40). – С. 85-88.

15. Коваленко А.С. Розробка структури бази даних для обліку технічного стану елементів інтегрованої інформаційної системи з урахуванням вимог споживачів інформації / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2015. – Вип. 1(126). – С. 75-79.

Кафедра_КБПЗ_2021рік

16. Коваленко А.С. Обґрунтування набору даних для оцінки технічного стану інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2015. – Вип. 1(42). – С.39-41.

17. Коваленко А.С. Експертна система технічного діагностування інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2015. – № 1(41). – С. 106-111.

18. Коваленко А.С. Удосконалення методу технічного обслуговування об'єктів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко, О.П. Доренський // Системи озброєння і військова техніка. – Х.: ХУПС, 2016. – № 2(46). – С. 109-114.

19. Коваленко А.С. Метод визначення оптимального комплексу робіт з відновлення працездатності інтегрованої системи технічної діагностики в умовах ресурсних обмежень / А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2016. – Вип. 3(140). – С. 69-72.

20. Kovalenko A.S. Information model and its element for displaying information on technical condition of objects of integrated information system / A.S. Kovalenko, A.A. Smirnov, A.V. Kovalenko, A.P. Dorensky // International Journal of Computational Engineering Research (IJCER). – India: Delhi, 2016. – Volume 6, Issue 1. – P. 21-27.

21. Кожанова А.С. Система технічної діагностики інтегрованих інформаційних систем – обґрунтування необхідності створення, визначення понятійного апарату та напрямів досліджень / А.С. Кожанова, О.А. Смірнов, М.П. Савченко, Д.М. Ізосімов, В.В. Мороз // Створення та модернізація озброєння і військової техніки в сучасних умовах: Тринадцята наук.-техн. конф., 5-6 вер. 2013 р., м. Феодосія: тези доп. – Феодосія: ДНВЦ, 2013. – С. 187-188.

22. Кожанова А.С. Визначення основних напрямків досліджень щодо створення системи технічної діагностики інтегрованих інформаційних систем

Кафедра_КБПЗ_2021рік

/ А.С. Кожанова, О.А. Смірнов, А.В. Челпанов // Проблемні питання розвитку озброєння та військової техніки Збройних Сил України: IV наук.-техн. конф., 16-20 груд. 2013 р., м. Київ: зб. тез. – Київ: ЦНДІ ОВТ ЗСУ, 2013. – С. 293.

23. Коваленко А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформатика та системні науки : V Всеукр. наук.-практ. конф., 13–15 бер. 2014 р., м. Полтава : зб. тез. – Полтава: ПУЕТ, 2014. – С. 292-294.

24. Коваленко А.С. Задачи распознавания ситуаций в системах организационной стратегии интеграции производства и операций / А.С. Коваленко, А.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVI міжнар. наук.-практ. сем., 11-12 квіт. 2014 р., м. Кіровоград: зб. тез. – Кіровоград: КНТУ, 2014. – С. 53-55.

25. Коваленко А.С. Створення систем технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VI між нар. наук.-практ. конф., 17-18 квіт. 2014 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2014. – С. 241.

26. Коваленко А.С. Визначення понятійного апарату та напрямів досліджень для синтезу систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комп'ютерне моделювання у наукоємних технологіях (КМНТ-2014): наук.-техн. конф. з міжнар. участю, 28-31 трав. 2014 р., м. Харків: зб. наук. праць. – Харків: ХНУ, 2014. – С. 190-193.

27. Коваленко А.С. Основні складові та функції системи технічної діагностики інтегрованих інформаційних систем / Коваленко А.С. // Інформаційні технології та комп'ютерна інженерія: наук.-практ. конф., 4 груд. 2014 р., м. Кіровоград: зб. тез доп. – Кіровоград: КНТУ, 2014. – С. 236.

Кафедра_КОПЗ_2021рік

28. Коваленко А.С. Розробка структури бази даних інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VII міжнар. наук.-практ. конф., 17-18 квіт. 2015 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2015. – С. 15.

29. Коваленко А.С. Дослідження елементів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVII між нар. наук.-практ. сем., 17-18 квіт. 2015 р., м. Кіровоград: зб. тез – Кіровоград: КНТУ, 2015. – С. 5.

30. Коваленко А.С. Метод автоматизованої перевірки результатів вимірювання параметрів об'єкті в інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Стратегія якості у промисловості і освіті: XI міжнар. конф., 1 – 5 черв. 2015 р., м. Варна, Болгарія.: зб. матер. – Варна: ТУВ, 2015. – С. 423-426.

31. Коваленко А.С. Обґрунтування необхідності створення розподіленої бази даних для забезпечення захисту рухомих повітряних об`єктів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Перспективні напрями захисту інформації: І всеукр. наук.-практ. конф., 07 вер. 2015 р., м. Одеса: зб. тез доп. – Одеса: ОНАЗ, 2015. – С. 35-39.

32. Коваленко А.С. Розробка інформаційної моделі автоматизованої оцінки технічного стану інтегральної інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформаційні технології та взаємодії (ІТ & І): ІІ між нар. наук.-практ. конф., 3-5 лист. 2015 р., м. Київ: тези доп. – Київ: КНУ ім. Т. Шевченка, 2015. – С. 41-42.

33. Коваленко А.С. Разработка метода усовершенствования технического обслуживания интегрированной информационной системы / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Информационные и телекоммуникационные технологии: образование, наука, практика: ІІ междунар. научн.-практ. конф., 3-4 дек. 2015 г., г. Алматы, Казахстан: сб. труд. – Алматы: КазНИТУ им. К.И. Сатпаева, 2015. – Т.2. – С. 423-427.

Кафедра_КБПЗ_2021рік

34. Королюк Н.А. Оценка временных интервалов работы лица, принимающего решение, на автоматизированном командном пункте / Н.А. Королюк, А.И. Тимочко // Системи обробки інформації. – Х.: ХУПС, 2005. – Вип. 8 (48). – С. 51-54.

35. Смірнова Т.В., Дрєєв 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.

36. 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.

37. Т.В. Смірнова, О.М. Дрєєв, 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.

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

39. Т.В. Смірнова, О.М. Дрєєв, 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.

40. Т. В. Смирнова, А. А. Смирнов, А. Н. Дреев, А. В. Дудан, «Оптимизация технологического процесса восстановления и упрочнения поверхностей с заданными характеристиками в виде облачного сервиса», Вестник Полоцкого государственного университета. Серия В, Промышленность.

Кафедра_КБПЗ_2021рік

Прикладные науки. **Республика Беларусь** - 2020. - № 3. - С. 50-61.Режим доступу: http://elib.psu.by:8080/handle/123456789/24988

41. 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)

42. 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)

43. T. Smirnova, I. Smirnov, A. Lopata, L. Lopata «Improvement of functional properties of gas-thermal coatings by electro-contact treatment», *Problems of Tribology*, Vol. 25, № 1/95, P. 41-48. 2020.

44. Т.В. Смірнова «Формування евристичних правил, бази знань та формалізація структури й правил технологічного процесу для оптимізаційної хмарної інформаційної системи», *Системи управління, навігації та зв'язку*, № 2 (60). с. 101-104, 2020. (Категорія Б)

45. Т.В. Ворона, Т.И. Ивченко, В.Я. Николайчук, «Повышение износотойкости стальных газотермических покрытий электроконтактной обработкой», *Сучасні енергетичні установки на транспорті, технології та обладнання для їх обслуговування,* Херсон 28-29 вересня 2017 року, Херсонська державна морська академія, 2017, с. 197-198**.**

46. В.Н. Лопата, М.С. Агеев, Т.В. Ворона, «Переход от гальванической технологии к газотермическим технологиям при получении антикоррозионных покрытий», «*Modern questions of production and repair in industry and in transport*» February 10–16, Brno, Czech Republic, 2018, pp. 245-249.

47. А.В. Дудан, М.С. Агеев, Т.В. Ворона, «Переход от гальванической технологии к газотермическим технологиям при получении антикоррозионных покрытий», «*Инновационные технологии в машиностроении*», ИннТехМаш (Новополоцк 19-20 апреля 2018), 2018, с. 186-192.

Кафедра_КОПЗ_2021рік

48. Т.В. Смірнова, О.А. Смірнов, О.М. Дрєєв, 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

49. Т.В. Смірнова, О.М. Дрєєв, 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.

50. Т.В. Смірнова, О.М. Дрєєв, 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

51. Т.В. Смірнова, О.М. Дрєєв, 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.

52. 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.

53. Т.В. Смірнова, О.М. Дрєєв, 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.

54. Т.В. Смірнова, О.М. Дрєєв, О.А. Смірнов, «Формування абстрактних експертних систем на основі досліджень відомих експертних систем», *XXI*

Кафедра_КБПЗ_2021рік

Міжнародний науково-практичний семінар «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 17-18 травня, 2019, с. 143-147.

55. 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

56. Т.В. Смирнова, А.Н. Дреев, 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.

57. Т.В. Смірнова, О.М. Дрєєв, 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

58. Т.В. Смірнова, О.М. Дрєєв, 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.

59. Т.В. Смірнова, О.М. Дрєєв, 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.

60. 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 questions of production and repair in industry and in transport»*, *Tbilisi, Georgia*, 23-28 *March* 2020, с. 201-205.

61. Т.В.Смірнова, Є.К. Солових, О.А.Смірнов, «Дослідження стандартів забезпечення кібербезпеки хмарних технологій як сервісів», *X міжнародна*

Кафедра_КБПЗ_2021рік

науково-технічна конференція «ITSEC», Єгипет, м. Шарм -ель-Шейх, 19 -24 березня, 2020. с. 36.

62. Т.В.Смірнова, Л.І. Поліщук, О.А.Смірнов, «Аналіз хмарних технологій як сервісів», *XIII всеукраїнська науково-практична конференція «Комп'ютерні інтелектуальні системи та мережі (КІСМ-2020)»*. м. Кривий Ріг. 24-26 березня, 2020, С. 210-212.

63. Т.В. Смірнова, Л.І. Поліщук, О.М. Дрєєв, «Застосування сервісу CAEaaS як системи інженерних розрахунків з використанням хмарних технологій», *II Міжнародна науково-практична конференція «Інформаційна безпека та інформаційні технології, Information Security and Information Technologies»*, м. Кропивницький, 2-3 квітня, 2020, с. 52.

64. Т.В. Смірнова, О.М. Дрєєв, О.А. Смірнов, Є.К. Солових, «Інформаційна структура технологічного процесу електродугового напилення». *Всеукраїнська науково-технічна конференція «Стан, досягнення та перспективи інформаційних систем і технологій»*, м. Одеса, 21-22 квітня, 2020, с. 184-186.

65. Т.В. Смірнова, О.М. Дрєєв, О.А. Смірнов, Є.К. Солових, «Реалізація інформаційної структури технологічного процесу електродугового напилення», *XXII Міжнародний науково-практичний семінар «Комбінаторні конфігурації та їх застосування»*, м. Кропивницький, 15-16 травня, 2020, с. 158-162.

66. Т.В. Смірнова, А.В. Щербань, Ю.Ю. Моторін, О.А. Смірнов, «Перспективні напрямки забезпечення кібербезпеки сервісу CAEaaS», *VI Всеукраїнська науково-практична конференція «Перспективні напрями захисту інформації»*, м. Одеса, 2-6 вересня 2020, с. 58-59

Кафедра_КОПЗ_2021рік

Додаток А
(обов'язковий)

Технічне завдання

Зміст

Кафедра_КОПЗ_2021рік

					КБР-123.21.0038.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив		Підлубний О.В.			Програмне забезпечення системи моніторингу продуктивності застосунків на базі АРМ	Літ.	Аркуш 1 / Аркушів 6
Перевірив		Доренський О.П.				Б	
Н. Контр.		Гермак В.С.				ЦНТУ КІ-18-3СК	
Затв.		Смірнов О.А.					

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи моніторингу продуктивності застосунків на базі APM.

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 моніторингу продуктивності застосунків на базі APM.

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;

Кафедра_КБПЗ_2021рік

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

– системи моніторингу продуктивності застосунків на базі АРМ;

– 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.

Кафедра_КбПЗ_2021рік

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 забезпечення.

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 Sydney.

Кафедра_КбПЗ_2021рік

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 – 75 аркушів.

Кафедра_КбПЗ_2021рік

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 22.05.2020 р.

11.2 Подання кваліфікаційної бакалаврської роботи на захист 3.06.2020 р.

Кафедра_КБПЗ_2021рік

Додаток Б

(обов’язковий)

Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник кваліфікаційної бакалаврської роботи

_____ Доренський О.П.

Програмне забезпечення системи моніторингу продуктивності застосунків на базі АРМ моніторингу продуктивності застосунків на базі АРМ

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 33

Літера: РП

Кропивницький – 2021 року

Кафедра_КбПЗ_2021рік

Файл fmMain.pas основної програми

```
unit fmMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Connector, Db, Search, Effic, About
  Dialogs, ActnList, Menus,inifiles, StdCtrls,Datamodule,ClassManager, ExtCtrls,
  ComCtrls,ClassInformation,ClassDraw, ToolWin,ClassGrid,ClassGenAlg,ClassTrass;

type
  TMainFm = class(TForm)
    ActionList1: TActionList;
    actExit: TAction;
    actFormReadFile: TAction;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    actAddElement: TAction;
    actAddElement1: TMenuItem;
    Panel1: TPanel;
    Panel2: TPanel;
    Tree: TTreeView;
    PaintBox: TImage;
    actOptionsOpen: TAction;
    N5: TMenuItem;
    N6: TMenuItem;
    GroupBox1: TGroupBox;
    actFormManagerBD: TAction;
    N7: TMenuItem;
    N8: TMenuItem;
    PopupMenu1: TPopupMenu;
    N9: TMenuItem;
    TreeUzel: TTreeView;
    PopupMenu2: TPopupMenu;
    N10: TMenuItem;
    Panel3: TPanel;
    Button1: TButton;
    N11: TMenuItem;
    N12: TMenuItem;
    Status: TStatusBar;
    btRazm: TButton;
    Time: TTimer;
    btRazmCont: TButton;
    Panel4: TPanel;
    Panel5: TPanel;
    Prog: TProgressBar;
    Button2: TButton;
    N13: TMenuItem;
    btKomp: TButton;
    Time2: TTimer;
    btKompCont: TButton;
    Button3: TButton;
    N14: TMenuItem;
    ScrollBox1: TScrollBox;
    N15: TMenuItem;
    Save: TSaveDialog;
    procedure N15Click(Sender: TObject);
    procedure N14Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure btKompContClick(Sender: TObject);
    procedure Time2Timer(Sender: TObject);
```

Кафедра_КбПЗ_2021рік

```
    procedure btKompClick(Sender: TObject);
    procedure N13Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure btRazmContClick(Sender: TObject);
    procedure TimeTimer(Sender: TObject);
    procedure btRazmClick(Sender: TObject);
    procedure N12Click(Sender: TObject);
    procedure N11Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure N10Click(Sender: TObject);
    procedure TreeClick(Sender: TObject);
    procedure N9Click(Sender: TObject);
    procedure PaintBoxMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure PaintBoxClick(Sender: TObject);
    procedure PaintBoxMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure N8Click(Sender: TObject);
    procedure actFormManagerBDExecute(Sender: TObject);
    procedure PaintBoxMouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure actOptionsOpenExecute(Sender: TObject);
    procedure actAddElementExecute(Sender: TObject);
    procedure actExitExecute(Sender: TObject);
    procedure actFormReadFileExecute(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    function DrawScheme (X : integer) : integer;
  private
    { Private declarations }
  public
    { Public declarations }
    PX,PY : integer;
    Index,KolScheme,isKolScheme : integer;
    Info : TInformation;
    PointScheme : Tpoint;
    AlgStart,IsStopAlg : boolean;
    CheckObj,Move,PaintScheme,isPaintScheme : boolean;
    Trass : TTrass;
    procedure CreateChildForm(form: TForm; TForms: TFormClass);
  protected
    //Draw : TDraw;
    GenAlg : TGenAlg;
    // Метод у якому необхідно створювати всі об'єкти
    procedure CreateObject;
    // Метод у якому необхідно знищувати всі об'єкти
    procedure FreeObject;
  end;


var
  MainFm: TMainFm;
  Draw : TDraw;


implementation

{$R *.dfm}

uses
FormReadFile,ClassMatr,FormAddElement,ClassOptions,FormOptions,FormMAnagerBD;

function TMainfm.DrawScheme(X: Integer) : integer;
begin
  Inc(X);//додаємо наступний елемент
  if KolScheme < X then
  begin
    PaintScheme := false;//схема не побудована
```

КафедраКбПЗ_2021рік

```
    if Manager.ElementsCount > Manager.CountVakMest then
    begin
      ShowMessage('Некоректно з'єднані пристрої!');
      Manager.DeleteAllSchemea;
    end;
  end
  else
  begin
       PaintScheme := true;// схема коректно побудована
  end;
  Status.Panels[3].Text := IntToStr(Manager.CountVakMest);
  Result := X;
end;

procedure TMainfm.CreateChildForm(form: TForm; TForms: TFormClass);
begin
  form:=TForms.Create(Application);
  form.ShowModal;
  form.Free;
  form:=nil;
end;

procedure TMainFm.FormCreate(Sender: TObject);
begin
  CreateObject;
end;

procedure TMainFm.FormDestroy(Sender: TObject);
begin
  FreeObject;
end;

procedure TMainfm.FreeObject;
begin
  FreeAndNil(Manager);
  FreeAndNil(Info);
  FreeAndnil(Draw);
end;

procedure TMainFm.N10Click(Sender: TObject);
begin
  if (Manager.UzelsCount > 0)and(TreeUzel.Selected.Text <> 'Вузли') then
  begin
    try
      Draw.DrawAll;
      Draw.DrawMoreZone(TreeUzel.Selected.Text);
    except

    end;

  end;
end;

procedure TMainFm.N11Click(Sender: TObject);
begin
  VisibleGrid := N11.Checked;
end;

procedure TMainFm.N12Click(Sender: TObject);
begin
  IsSnap := N12.Checked;
end;

procedure TMainFm.N13Click(Sender: TObject);
begin
  if Index > -1 then
    Manager.Elements[Index].ChangeZakr;
end;
```

КафеДра_КбПЗ_2021рік

```
procedure TMainFm.N14Click(Sender: TObject);
begin
if (Manager.UzelsCount > 0)and(TreeUzel.Selected.Text <> 'Вузли') then
  begin
    try
      Draw.DrawAllnoLine;
      Trass.ShowTrass(Manager.FindUzel(TreeUzel.Selected.Text));
    except

    end;

  end;
end;

procedure TMainFm.N15Click(Sender: TObject);
begin
  if Save.Execute then
  begin
    if Info.Save(Save.FileName) then ShowMessage('Файл '+Save.FileName+' успішно
збережений!');
  end;
end;

procedure TMainFm.N8Click(Sender: TObject);
begin
  if ShowMiniElements then
  begin
    N8.Checked := false;
  end;
  Panel1.Visible := N8.Checked;
  Draw.DrawGrid;
end;

procedure TMainFm.N9Click(Sender: TObject);
begin
  if Index > -1 then
  begin
    Manager.Elements[Index].ChangeRotate;
    Draw.DrawAll;
    Draw.DrawZone(Index);
  end;
end;

procedure TMainFm.PaintBoxClick(Sender: TObject);
begin
 { Index := Manager.FindByPoint(PX,PY);
  if Index > -1 then
  begin
    Draw.DrawZone(Index);
  end;              }
end;

procedure TMainFm.PaintBoxMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  ShowMessage(intToStr(PaintBox.Width));
 Draw.DrawAll;
  Index := -1;
  Index := Manager.FindByPoint(PX,PY);
  CheckObj := false;
  if PaintScheme then
  begin
    PointScheme.X := X;
    PointScheme.Y := Y;
    PaintScheme := false;
    isPaintScheme := true;
    Index := -1;
  end;
```

Кафедра_КОПЗ_2021рік

```
  if Index > -1 then
  begin
    Draw.DrawZone(Index);
    CheckObj := true;
  end;
end;

procedure TMainFm.PaintBoxMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  PX := X;
  PY := Y;
  Move := false;
  if isPaintScheme then
  begin
    Draw.DrawScheme(PointScheme.X,PointScheme.Y,X,Y);
  end;
  if CheckObj then
  begin
    Draw.ChangePoz(Index,X,Y);
    Move := true;
  end;
end;

procedure TMainFm.PaintBoxMouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
var P,P2 : Tpoint;
    TempKolScheme : integer;
begin
  P.X := PX;
  P.Y := PY;
  CheckObj := false;
  if isPaintScheme then
  begin
    P2.X := X;
    P2.Y := Y;
    Manager.AddSchemea(PointScheme,P2);
    isPaintScheme := false;
    isKolScheme := DrawScheme(isKolScheme);

  end;
  if (Index > -1)and(Move) then
  begin
    Manager.Elements[Index].VerhPoint := P;
    if IsSnap then Draw.SnapToGrid(Manager.Elements[Index]);
  end;
  if Manager.ElementsCount > 0 then Draw.DrawAll;
  if Index > -1 then  Draw.DrawZone(Index);
end;

procedure TMainFm.Time2Timer(Sender: TObject);
begin
  if GenAlg.CrossingKomp(genAlg.FHrom[0].GetGen,genAlg.FHrom[0].GetGen)then
  begin
    GenAlg.Iteration := 0;
    Prog.Position := 0;
  end
  else
  begin
    Inc(GenAlg.Iteration);
    Prog.StepIt;
  end;
  GenAlg.SetCelFunc;
  Status.Panels[5].Text := IntToStr(GenAlg.CelFunc);
  if IsAlgDraw then Draw.DrawAll;
  if (GenAlg.Iteration > genAlg.MinPop)or(isStopAlg) then
  Begin
    Draw.DrawAll;
```

Кафедра_КбПЗ_2021рік

```
    Time2.Enabled := false;
    Panel4.Visible := false;
    end;
end;

procedure TMainFm.TimeTimer(Sender: TObject);
begin
  if GenAlg.Crossing(genAlg.FHrom[0].GetGen,genAlg.FHrom[0].GetGen)then
  begin
    GenAlg.Iteration := 0;
    Prog.Position := 0;
  end
  else
  begin
    Inc(GenAlg.Iteration);
    Prog.StepIt;
  end;
  GenAlg.SetCelFunc;
  Status.Panels[5].Text := IntToStr(GenAlg.CelFunc);
  if IsAlgDraw then Draw.DrawAll;
  if (GenAlg.Iteration > genAlg.MinPop)or(isStopAlg) then
  Begin
    Draw.DrawAll;
    Time.Enabled := false;
    Panel4.Visible := false;
  end;
end;

procedure TMainFm.TreeClick(Sender: TObject);
begin
  if (Manager.ElementsCount > 0)and(Tree.Selected.Text <> 'Пристрої') then
  begin
    try
      Draw.DrawAll;
      Draw.DrawZone(Manager.FindElement(Tree.Selected.Text));
    except

    end;

  end;
end;

procedure TMainFm.actAddElementExecute(Sender: TObject);
begin
  CreateChildForm(fmAddElement,TfmAddElement);
end;

procedure TMainFm.actExitExecute(Sender: TObject);
begin
  Close;
end;

procedure TMainFm.actFormManagerBDExecute(Sender: TObject);
begin
  CreateChildForm(fmManagerBd,TfmManagerBd);
end;

procedure TMainFm.actFormReadFileExecute(Sender: TObject);
begin
  ShagSetk := 0;
  CreateChildForm(fmReadFile,TfmReadFile);
  if Manager.ElementsCount > 0 then
  begin
    if not ShowMiniElements then
    begin
      Panel1.Visible := true;
      n8.Checked := true;
    end;
    btRazmCont.Enabled := false;
```

Кафедра_КОПЗ_2021рік

```
    btKompCont.Enabled := false;
    Info.ShowElements(Tree,TreeUzel);
    Tree.Items[0].Expand(False);
    Tree.Items[0].Selected:=true;
    Tree.SetFocus;
    TreeUzel.Items[0].Expand(False);
    TreeUzel.Items[0].Selected:=true;
    TreeUzel.SetFocus;
    if ShowMiniElements then
    begin
      Panel1.Visible := false;
      n8.Checked := false;
    end;
    Manager.DeleteAllSchemea;
    Draw.DrawFirst;
    Draw.DrawAll;
    AlgStart := false;
  end;
  Status.Panels.Items[1].Text := IntTOStr(Manager.ElementsCount);
end;

procedure TMainFm.actOptionsOpenExecute(Sender: TObject);
begin
  CreateChildForm(fmOptions,TfmOptions);
end;

procedure TMainFm.Button1Click(Sender: TObject);
var D : string;
   // X : integer;
begin
  if Manager.ElementsCount > 0 then
  begin
    if InputQuery('Введіть кількість пристроїв','',D) then
    begin
      manager.DeleteAllSchemea;
      Draw.DrawAll;
      KolScheme := StrToInt(D);
      isKolScheme := DrawScheme(0);
    end;
  end
  else
  begin
    ShowMessage('Необхідно експортувати пристрої!');
  end;
end;

procedure TMainFm.Button2Click(Sender: TObject);
begin
  IsStopAlg := true;
end;

procedure TMainFm.Button3Click(Sender: TObject);
begin
  if manager.UzelsCount > 0 then
  begin
    Trass.Prepare;
  end;
end;

procedure TMainFm.btKompContClick(Sender: TObject);
begin
  if AlgStart then
  begin
    GenAlg.PrepareSecond;
    GenAlg.Invertion := optInvertion;
    GenAlg.MinPop := OptMinPop;
    GenAlg.Mutation := OptMutation;
    AlgStart := true;
    Draw.DrawAll;
```

Кафедра_КбПЗ_2021рік

```
    Panel4.Visible := true;
    isStopAlg := false;
    Prog.Max := GenAlg.MinPop;
    Prog.Position := 0;
    if isAlgDraw then
  begin
    Time2.Enabled := true;
  end
  else
  begin
    while GenAlg.Iteration < GenAlg.MinPop do
    begin
      if GenAlg.CrossingKomp(genAlg.FHrom[0].GetGen,genAlg.FHrom[0].GetGen)then
      begin
        GenAlg.Iteration := 0;
        Prog.Position := 0;
      end
      else
      begin
        Inc(GenAlg.Iteration);
        Prog.StepIt;
      end;
//  GenAlg.SetCelFunc;
      Status.Panels[5].Text := IntToStr(GenAlg.CelFunc);
  //if IsAlgDraw then Draw.DrawAll;
    end;
    Draw.DrawAll;
    Time2.Enabled := false;
    Panel4.Visible := false;
  end;
  end;
end;

procedure TMainFm.btKompClick(Sender: TObject);
begin
  if Manager.SchemesCount > 1 then
  begin
  GenAlg.PrepareSecond;
  IsStopAlg := false;
  GenAlg.Invertion := optInvertion;
  GenAlg.MinPop := OptMinPop;
  GenAlg.Mutation := OptMutation;
  AlgStart := true;
  Draw.DrawAll;
  Panel4.Visible := true;
  Prog.Max := GenAlg.MinPop;
  btKompCont.Enabled := true;
  Prog.Position := 0;
  if isAlgDraw then
  begin
    Time2.Enabled := true;
  end
  else
  begin
    while GenAlg.Iteration < GenAlg.MinPop do
    begin
      if GenAlg.CrossingKomp(genAlg.FHrom[0].GetGen,genAlg.FHrom[0].GetGen)then
      begin
        GenAlg.Iteration := 0;
        Prog.Position := 0;
      end
      else
      begin
        Inc(GenAlg.Iteration);
        Prog.StepIt;
      end;
//  GenAlg.SetCelFunc;
      Status.Panels[5].Text := IntToStr(GenAlg.CelFunc);
  //if IsAlgDraw then Draw.DrawAll;
```

Кафедра_КбПЗ_2021рік

```
    end;
    Draw.DrawAll;
    Time2.Enabled := false;
    Panel4.Visible := false;
  end;
  end
  else
  begin
    ShowMessage('Для прогнозування необхідно не менш 2 пристроїв!');
  end;
end;

procedure TMainFm.btRazmClick(Sender: TObject);
begin
  if Manager.SchemesCount > 0 then
  begin
  GenAlg.PrepareSecond;
  IsStopAlg := false;
  GenAlg.Invertion := optInvertion;
  GenAlg.MinPop := OptMinPop;
  GenAlg.Mutation := OptMutation;
  AlgStart := true;
  Draw.DrawAll;
  Panel4.Visible := true;
  Prog.Max := GenAlg.MinPop;
  btRazmCont.Enabled := true;
  Prog.Position := 0;
  if isAlgDraw then
  begin
    Time.Enabled := true;
  end
  else
  begin
    while GenAlg.Iteration < GenAlg.MinPop do
    begin
      if GenAlg.Crossing(genAlg.FHrom[0].GetGen,genAlg.FHrom[0].GetGen)then
      begin
        GenAlg.Iteration := 0;
        Prog.Position := 0;
      end
      else
      begin
        Inc(GenAlg.Iteration);
        Prog.StepIt;
      end;
//  GenAlg.SetCelFunc;
      Status.Panels[5].Text := IntToStr(GenAlg.CelFunc);
  //if IsAlgDraw then Draw.DrawAll;
    end;
    Draw.DrawAll;
    Time.Enabled := false;
    Panel4.Visible := false;
     end;
  end

end;

procedure TMainFm.btRazmContClick(Sender: TObject);
begin
  if AlgStart then
  begin
    GenAlg.PrepareSecond;
    GenAlg.Invertion := optInvertion;
    GenAlg.MinPop := OptMinPop;
    GenAlg.Mutation := OptMutation;
    AlgStart := true;
    Draw.DrawAll;
    Panel4.Visible := true;
    isStopAlg := false;
```

Кафедра_КбПЗ_2021рік

```
    Prog.Max := GenAlg.MinPop;
    Prog.Position := 0;
    if isAlgDraw then
  begin
    Time.Enabled := true;
  end
  else
  begin
    while GenAlg.Iteration < GenAlg.MinPop do
    begin
      if GenAlg.Crossing(genAlg.FHrom[0].GetGen,genAlg.FHrom[0].GetGen)then
      begin
        GenAlg.Iteration := 0;
        Prog.Position := 0;
      end
      else
      begin
        Inc(GenAlg.Iteration);
        Prog.StepIt;
      end;
//  GenAlg.SetCelFunc;
      Status.Panels[5].Text := IntToStr(GenAlg.CelFunc);
  //if IsAlgDraw then Draw.DrawAll;
    end;
    Draw.DrawAll;
    Time.Enabled := false;
    Panel4.Visible := false;
  end;
  end;
end;

procedure TMainFm.CreateObject;
var
server_name: string;
database_name: string;
begin
  Save.FileName := GetCurrentDir + '\SAVE';
  Trass := TTrass.Create;
  Manager := TManager.Create;
  Info := TInformation.Create;
  ShagSetk := 0;
  AlgStart := false;
  GenAlg := TgenAlg.Create;
  PaintScheme := false;
  isPaintScheme := false;
  Draw := TDraw.Create(PaintBox);
  try
    ini:=TIniFile.Create(ExtractFilePath(Application.ExeName)+'Prof.ini');
    server_name:=ini.ReadString('options','server_name','');
    database_name:=ini.readString('options','database_name','');
    if (server_name='') or (database_name='') then
    begin
      server_name:='localhost';
      if not InputQuery('Уведіть ім'я сервера','',server_name) then
      begin
        Application.Terminate;
        Exit;
      end;
      if not InputQuery('Введіть повний шлях до файлу БД','',database_name) then
      begin
        Application.Terminate;
        Exit;
      end;
      ini.WriteString('options','server_name',server_name);
      ini.WriteString('options','database_name',database_name);
    end;
    try
      Data.Database.DatabaseName := database_name;
      Data.DataBase.Connected := true;
```

Кафедра_КбПЗ_2021рік

```
      Data.Transaction.Active := true;
    except
      ShowMessage('Неправильні параметри підключення');
      Application.Terminate;
      Exit;
    end;
  finally
    LoadOptions(ini);
    //FreeAndNil(ini);
  end;
end;


end.
```

Кафедра_КбПЗ_2021рік

Файл Connector.pas – створення з'єднань між мережними пристроями

```
unit Connector;

interface

type
  TConnection = record
    CStart: record
              x,y: integer;
            end;
    CEnd:   record
              x,y: integer;
            end;
    Num: Integer;
  end;
  TField = array [0..200,0..200] of Integer;

var
  MField, Field: TField;

procedure Connect;

implementation

uses windows, SysUtils, math, Unit1, Unit3, Types;

procedure Connect;
var i,j,k,l,m: Integer;
    temp: TField;
    cur: Integer;
    x,y: Integer;
    napr: Integer;
    naprd: Integer;
    Och: array [1..1000] of record
                              x,y:byte;
                            end;
    uk1,uk2: integer;
    tocon: Integer;
    ppp: TConnection;

const dd: array [1..4,1..3] of record
                                 dx,dy: integer;
                               end =
          (((dx:0;dy:-1),(dx:-1;dy:0),(dx:1;dy:0)),
           ((dx:1;dy:0),(dx:0;dy:-1),(dx:0;dy:1)),
           ((dx:0;dy:1),(dx:1;dy:0),(dx:-1;dy:0)),
           ((dx:-1;dy:0),(dx:0;dy:1),(dx:0;dy:-1)));
{     tt: array [0..5,0..3] of byte =
          ((10,9,10,7),
           (8,10,6,10),
           (9,10,10,6),
           (10,10,9,8),
           (7,6,10,10),
           (10,8,7,10));}
  function Get(var x,y:integer):Boolean;
  begin
    Result:=true;
    if uk1 = uk2 then begin Result:=false; exit; end;
    x:=och[uk1].x;
    y:=och[uk1].y;
    uk1:=uk1+1;
    if uk1>1000 then uk1:=1;
  end;
  procedure Put(x,y:integer);
  begin
    och[uk2].x:=x;
```

Кафедра_КбПЗ_2021рік

```
    och[uk2].y:=y;
    uk2:=uk2+1;
    if uk2>1000 then uk2:=1;
  end;
  function Test(x,y: Integer):boolean;
  begin
    Result:=false;
    if (x<1) or (x>199) or (y<1) or (y>199) then exit;
    if (x=ppp.CStart.x) and (y=ppp.CStart.y) then
      begin Result:=true; temp[x,y]:=napr*100000+cur+1; exit; end;
    if (MField[x,y]=tocon) and (tocon<>-1) then
      begin Result:=true; temp[x,y]:=napr*100000+cur+1; exit; end;
    if temp[x,y]=9999999 then exit;
//    if (napr mod 2 = 1) and ((temp[ x-1,y]=9999999) or (temp[x+1,y]=9999999))
then exit;
    if (temp[x,y]>10001) and (temp[x,y]<10011) then exit;
    if (temp[x,y]=10000) and (napr mod 2 = 0) then exit;
    if (temp[x,y]=10001) and (napr mod 2 = 1) then exit;
    if (temp[x,y]<>0) and (temp[x,y] mod 100000 < cur) then exit;
    temp[x,y]:=napr*100000+cur+1;
    Put(x,y);
  end;
  function Find(var x,y: Integer): boolean;
  var t: integer;
      xx,yy: integer;
  begin
    Result:=true;
    xx:=x; yy:=y;
    t:=temp[x,y] mod 100000;
    napr:=temp[x,y] div 100000 + 2; if napr>4 then napr:= napr-4;
    if (temp[x+dd[napr,1].dx,y+dd[napr,1].dy] mod 100000 < t) and
(temp[x+dd[napr,1].dx,y+dd[napr,1].dy]<>0) then begin xx:=x+dd[napr,1].dx;
yy:=y++dd[napr,1].dy; t:=temp[xx,yy] mod 100000; end;
    if (temp[x+dd[napr,2].dx,y+dd[napr,2].dy] mod 100000 < t) and
(temp[x+dd[napr,2].dx,y+dd[napr,2].dy]<>0) then begin xx:=x+dd[napr,2].dx;
yy:=y++dd[napr,2].dy; t:=temp[xx,yy] mod 100000; end;
    if (temp[x+dd[napr,3].dx,y+dd[napr,3].dy] mod 100000 < t) and
(temp[x+dd[napr,3].dx,y+dd[napr,3].dy]<>0) then begin xx:=x+dd[napr,3].dx;
yy:=y++dd[napr,3].dy; t:=temp[xx,yy] mod 100000; end;
    if (x=xx) and (y=yy) then Result:=False;
    x:=xx;
    y:=yy;
    if (x=ppp.CEnd.x) and (y=ppp.CEnd.y) then Result:=false;
  end;
label 1,2,3,4;
begin
  for i:=0 to 200 do for j:=0 to 200 do begin Field[i,j]:=0; MField[i,j]:=-1;
end;
  for k:=1 to High(Elements) do
    for i:=0 to Elements[k].Size.x do
      for j:=0 to Elements[k].Size.y do
Field[Elements[k].Coord.x+i,Elements[k].Coord.y+j]:=9999999;
  for k:=0 to High(Cons) do
   for m:=1 to High(Cons[k]) do
    begin
     ppp.CStart.x:=Cons[k][0].X;
     ppp.CStart.y:=Cons[k][0].Y;
     ppp.CEnd.x:=Cons[k][m].X;
     ppp.CEnd.y:=Cons[k][m].Y;
     temp:=Field;
     cur:=1;
     x:=ppp.CEnd.x;
     y:=ppp.CEnd.y;
     tocon:=MField[x,y];
{     if tocon<>-1 then begin
                        ppp.CEnd.x:=ppp.CStart.x;
                        ppp.CEnd.y:=ppp.CStart.y;
                        ppp.CStart.x:=x;
                        ppp.CStart.y:=y;
```

Кафедра_КСПВ_2021рік

```
                        x:=ppp.CEnd.x;
                        y:=ppp.CEnd.y;
                        if MField[x,y]<>-1 then goto 2;
                      end;}
      if MField[ppp.CStart.x,ppp.CStart.y]<>-1 then
tocon:=MField[ppp.CStart.x,ppp.CStart.y];
      uk1:=1;
      uk2:=1;
      if (temp[x+1,y]=0) or (temp[x+1,y]=10000) or (temp[x+1,y]=10001) then
napr:=2;
      if (temp[ x-1,y]=0) or (temp[ x-1,y]=10000) or (temp[ x-1,y]=10001) then
napr:=4;
      if (temp[x,y+1]=0) or (temp[x,y+1]=10000) or (temp[x,y+1]=10001) then
napr:=3;
      if (temp[x, y-1]=0) or (temp[x, y-1]=10000) or (temp[x, y-1]=10001) then
napr:=1;
      temp[x,y]:=1+napr*100000;
      Put(x,y);
      while true do begin
        if Get(i,j) then
             begin
               cur:=temp[i,j] mod 100000;
               naprd:=temp[i,j] div 100000;
               napr:=naprd;
               if Test(i+dd[naprd,1].dx,j+dd[naprd,1].dy) then goto 1;
               if MField[i,j]=-1 then begin
                 cur:=cur+5;
                 napr:= naprd-1; if napr=0 then napr:=4;
                 if Test(i+dd[naprd,2].dx,j+dd[naprd,2].dy) then goto 1;
                 napr:=naprd+1; if napr=5 then napr:=1;
                 if Test(i+dd[naprd,3].dx,j+dd[naprd,3].dy) then goto 1;
                 cur:= cur-5;
               end;
             end else goto 2;
        MessageBox(0,pchar(inttostr(cur)),'',0);
        if cur>300 then goto 2;
      end;
1:
      if tocon=-1 then begin
        x:=ppp.CStart.x;
        y:=ppp.CStart.y;
        l:=k;
      end else begin
        x:=i;
        y:=j;
        l:=tocon;
        Field[x+dd[napr,1].dx,y+dd[napr,1].dy]:=10010;
        if (x=ppp.CEnd.x) and (y=ppp.CEnd.y) then goto 3;
        napr:=napr+2; if napr>4 then napr:= napr-4;
        naprd:=napr;
        goto 4;
      end;
      naprd:=temp[x,y] div 100000 + 2; if naprd>4 then naprd:= naprd-4;
      napr:=napr+2; if napr>4 then napr:= napr-4;
      while Find(x,y) do
        begin
4:
          if napr=naprd then
            begin
              if MField[ x-dd[napr,1].dx, y-dd[napr,1].dy]=-1 then
                Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10000+naprd mod 2
                  else if Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]<>10010 then
                    Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10006;
            end else begin
              if ((naprd=2) and (napr=1)) or ((naprd=3) and (napr=4)) then
                Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10002;
              if ((naprd=1) and (napr=4)) or ((naprd=2) and (napr=3)) then
                Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10003;
              if ((naprd=3) and (napr=2)) or ((naprd=4) and (napr=1)) then
```

Кафедра_КСПЗ_2021рік

```
                Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10004;
              if ((naprd=1) and (napr=2)) or ((naprd=4) and (napr=3)) then
                Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10005;
            end;
          naprd:=napr;
          MField[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=l;
        end;
      if napr=naprd then
        begin
          if MField[ x-dd[napr,1].dx, y-dd[napr,1].dy]=-1 then
            Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10000+naprd mod 2
              else Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10006;
        end else begin
          if ((naprd=2) and (napr=1)) or ((naprd=3) and (napr=4)) then
            Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10002;
          if ((naprd=1) and (napr=4)) or ((naprd=2) and (napr=3)) then
            Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10003;
          if ((naprd=3) and (napr=2)) or ((naprd=4) and (napr=1)) then
            Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10004;
          if ((naprd=1) and (napr=2)) or ((naprd=4) and (napr=3)) then
            Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10005;
        end;
      naprd:=napr;
      MField[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=l;
3:
      Field[x,y]:=10000+naprd mod 2;
      MField[x,y]:=l;
2:
    end;
end;

end.
```

Кафедра_КбПЗ_2021рік

Файл Db.pas - створення бази даних мережних пристроїв

```
unit Db;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, Grids, DBGrids, DB, DBTables, ExtCtrls, StdCtrls,
  Mask, DBCtrls;

type
  TForm3 = class(TForm)
    Panel1: TPanel;
    Table1: TTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    GroupBox1: TGroupBox;
    DBEdit1: TDBEdit;
    Label1: TLabel;
    Button1: TButton;
    DBEdit2: TDBEdit;
    Label2: TLabel;
    GroupBox2: TGroupBox;
    GroupBox3: TGroupBox;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    GroupBox4: TGroupBox;
    Panel2: TPanel;
    Image1: TImage;
    GroupBox5: TGroupBox;
    Label3: TLabel;
    Button2: TButton;
    DBEdit5: TDBEdit;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    DBEdit6: TDBEdit;
    Label4: TLabel;
    Label5: TLabel;
    Panel3: TPanel;
    TabSheet3: TTabSheet;
    procedure FormCreate(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure DBGrid1TitleClick(Column: TColumn);
    procedure Table1AfterScroll(DataSet: TDataSet);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

uses Main1, Main2, Db_elements;

{$R *.dfm}
```

Кафедра_КбПЗ_2021рік

```
procedure TForm3.FormCreate(Sender: TObject);
begin
  Table1.DatabaseName := ExtractFilePath(Application.ExeName)+'bases\';
  Table1.TableName := 'microchips.db';
  Table1.Active := true;
end;

procedure TForm3.Button5Click(Sender: TObject);
begin
Close;
Table1.Edit;
Table1.Post;
end;

procedure TForm3.Button4Click(Sender: TObject);
begin
  if Table1.RecordCount > 0 then
    begin
      if Application.MessageBox('Ви дійсно хочете видалити поточний мережний
пристрій', 'Увага', MB_YESNO) = IDYES then
        Table1.Delete;
    end;
end;

procedure TForm3.Button3Click(Sender: TObject);
begin
  form11.ShowModal;
end;

procedure TForm3.Button1Click(Sender: TObject);
begin
  form12.ShowModal;
end;

procedure TForm3.DBGrid1TitleClick(Column: TColumn);
begin
  if column.Index = 0 then begin Table1.IndexFieldNames := 'Mark';
DBGrid1.Columns[0].Title.Caption := 'Назва'; DBGrid1.Columns[1].Title.Caption :=
'Тип'; end;;
  if column.Index = 1 then begin Table1.IndexFieldNames := 'Type';
DBGrid1.Columns[0].Title.Caption := 'Назва'; DBGrid1.Columns[1].Title.Caption :=
'Тип*'; end;
end;

procedure TForm3.Table1AfterScroll(DataSet: TDataSet);
var i: integer;
begin
  Form7.Table1.First;
  for i := 0 to form7.Table1.RecordCount - 1 do
    begin
      if form7.Table1.fieldbyname('Name').AsString =
table1.FieldByName('Elem_name').AsString then break;
      form7.Table1.Next;
    end;
end;

end.
```

Кафедра_КбПЗ_2021рік

Файл Search.pas - вікно пошуку мережних пристроїв

```
unit Search;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls;

type
  TForm4 = class(TForm)
    LabeledEdit1: TLabeledEdit;
    Button1: TButton;
    Button2: TButton;
    Label1: TLabel;
    Image1: TImage;
    procedure FormActivate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var  Form4: TForm4;

implementation

uses Db;

{$R *.dfm}

procedure TForm4.FormActivate(Sender: TObject);
begin
  LabeledEdit1.SetFocus;
  LabeledEdit1.SelStart := 0;
  LabeledEdit1.SelLength :=  255;
end;

procedure TForm4.Button1Click(Sender: TObject);
var i : integer;
begin
  Form3.Table1.first;
  for i := 0 to Form3.Table1.RecordCount - 1 do
    begin
      if pos (LabeledEdit1.Text, Form3.Table1.fieldbyname('Mark').asstring) <> 0
then begin close; {Form3.DBGrid1.SetFocus;} exit end;
      Form3.Table1.Next;
    end;
  Form3.Table1.First;
  Application.MessageBox('Пристрій не знайдено', 'Увага!', MB_OK);
  Close;
  Exit; end; end.
```

Кафедра_КбПЗ_2021рік

Файл Effic.PAS - оцінка ефективності спроектованої розподіленої інформаційної системи моніторингу продуктивності застосунків на базі АРМ

```
unit Effic;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, ComCtrls,
  ShellAPI, ShlObj;

type
  TEfficForm = class(TForm)
    gbxShares: TGroupBox;
    lbxShares: TListBox;
    gbxSessions: TGroupBox;
    lvSessions: TListView;
    bvlSessions: TBevel;
    gbxFiles: TGroupBox;
    btnGetShares: TButton;
    btnCloseShares: TButton;
    btnAddShares: TButton;
    btnCloseSession: TButton;
    btnGetSessions: TButton;
    bvlTopSessions: TBevel;
    plButtonFiles: TPanel;
    btnGetFiles: TButton;
    btnCloseFile: TButton;
    bvlLeftFiles: TBevel;
    plFiles: TPanel;
    lvFiles: TListView;
    bvlTopFiles: TBevel;
    gbxTraffic: TGroupBox;
    lvTraffic: TListView;
    bvlTraffic: TBevel;
    tmrTraffic: TTimer;
    function  IsXP(var Value: Boolean): Boolean;
    procedure btnGetSharesClick(Sender: TObject);
    procedure btnCloseSharesClick(Sender: TObject);
    function  SelectDirectory: String;
    procedure btnAddSharesClick(Sender: TObject);
    function  CardinalToTimeStr(Value:Cardinal):String;
    procedure btnGetSessionsClick(Sender: TObject);
    procedure btnCloseSessionClick(Sender: TObject);
    procedure btnGetFilesClick(Sender: TObject);
    procedure btnCloseFileClick(Sender: TObject);
    procedure tmrTrafficTimer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    SessionCloseKey: array [0..512] of SmallInt;
  end;

type
 TShareInfo2 = packed record
  shi2_netname : PWChar;
  shi2_type: DWORD;
  shi2_remark :PWChar;
  shi2_permissions: DWORD;
  shi2_max_uses : DWORD;
  shi2_current_uses : DWORD;
  shi2_path : PWChar;
  shi2_passwd : PWChar;
 end;
```

Кафедра КбПЗ_2021рік

```
 PShareInfo2 = ^ TShareInfo2;
 TShareInfo2Array = array [0..512] of TShareInfo2;
 PShareInfo2Array = ^ TShareInfo2Array;

type
 TShareInfo50 = packed record
  shi50_netname : array [0..12] of Char;
  shi50_type : Byte;
  shi50_flags : Word;
  shi50_remark : PChar;
  shi50_path : PChar;
  shi50_rw_password : array [0..8] of Char;
  shi50_ro_password : array [0..8] of Char;
 end;

type
 TSessionInfo502 = packed record
   Sesi502_cname: PWideChar;
   Sesi502_username: PWideChar;
   Sesi502_num_opens: DWORD;
   Sesi502_time: DWORD;
   Sesi502_idle_time: DWORD;
   Sesi502_user_flags: DWORD;
   Sesi502_cltype_name: PWideChar;
   Sesi502_transport: PWideChar;
 End;
 PSessionInfo502 = ^TSessionInfo502;
 TSessionInfo502Array = array[0..512] of TSessionInfo502;
 PSessionInfo502Array = ^TSessionInfo502Array;

type
 TSessionInfo50 = packed record
   Sesi50_cname       : PChar;
   Sesi50_username    : PChar;
   sesi50_key         : Cardinal;
   sesi50_num_conns   : Word;
   sesi50_num_opens   : Word;
   sesi50_time        : Cardinal;
   sesi50_idle_time   : Cardinal;
   sesi50_protocol    : Byte;
   pad1               : Byte;
 end;

type
 TFileInfo3 = packed record
   fi3_id          : DWORD;
   fi3_permissions : DWORD;
   fi3_num_locks   : DWORD;
   fi3_pathname    : PWChar;
   fi3_username    : PWChar;
 end;
 PFileInfo3 = ^TFileInfo3;
 TFileInfo3Array = array[0..512] of TFileInfo3;
 PFileInfo3Array = ^TFileInfo3Array;

type
 TFileInfo50 = packed record
   fi50_id          : Cardinal;
   fi50_permissions : WORD;
   fi50_num_locks   : WORD;
   fi50_pathname    : PChar;
   fi50_username    : PChar;
   fi50_sharename   : PChar;
 end;

type
  TMibIfRow = packed record
    wszName            : array[0..255] of WideChar;
    dwIndex            : DWORD;
```

Кафедра_КбПЗ_2021рік

```
    dwType              : DWORD;
    dwMtu               : DWORD;
    dwSpeed             : DWORD;
    dwPhysAddrLen       : DWORD;
    bPhysAddr           : array[0..7] of Byte;
    dwAdminStatus       : DWORD;
    dwOperStatus        : DWORD;
    dwLastChange        : DWORD;
    dwInOctets          : DWORD;
    dwInUcastPkts       : DWORD;
    dwInNUCastPkts      : DWORD;
    dwInDiscards        : DWORD;
    dwInErrors          : DWORD;
    dwInUnknownProtos   : DWORD;
    dwOutOctets         : DWORD;
    dwOutUCastPkts      : DWORD;
    dwOutNUCastPkts     : DWORD;
    dwOutDiscards       : DWORD;
    dwOutErrors         : DWORD;
    dwOutQLen           : DWORD;
    dwDescrLen          : DWORD;
    bDescr              : array[0..255] of Char;
  end;
  TMibIfArray = array [0..512] of TMibIfRow;
  PMibIfRow = ^TMibIfRow;
  PMibIfArray = ^TMibIfArray;

type
 TMibIfTable = packed record
   dwNumEntries     : DWORD;
   Table            : TMibIfArray;
 end;
 PMibIfTable = ^TMibIfTable;
var
NetShareEnumXP:function (     servername:PWChar;
                          level:DWORD;
                          bufptr:Pointer;
                          prefmaxlen:DWORD;
                          entriesread,
                          totalentries,
                          resume_handle:LPDWORD): DWORD; stdcall;

var
NetShareEnum:function ( pszServer   : PChar;
                        sLevel           : Cardinal;
                        pbBuffer    : Pchar;
                        cbBuffer    : Cardinal;
                        pcEntriesRead,
                        pcTotalAvail: Pointer):DWORD; stdcall;

var
NetShareDelXP:function (servername: PWideChar;
                        netname: PWideChar;
                        reserved: DWORD): LongInt; stdcall;

var
NetShareDel:function (  pszServer,
                        pszNetName:PChar;
                        usReserved:Word): DWORD; stdcall;

var
NetShareAddXP: function(servername: PWideChar;
                        level: DWORD;
                        buf: Pointer;
                        parm_err: LPDWORD): DWORD; stdcall;

var
NetShareAdd: function ( pszServer:Pchar;
                        sLevel:Cardinal;
```

Кафедра_КБПЗ_2021рік

```
                        pbBuffer:PChar;
                        cbBuffer:Word):DWORD; stdcall;

Var
NetSessionEnumXP:function(servername,
                        UncClientName,
                        username:PWChar;
                        level:DWORD;
                        bufptr:Pointer;
                        prefmaxlen:DWORD;
                        entriesread,
                        totalentries,
                        resume_handle:LPDWORD):DWORD; stdcall;

var
NetSessionEnum:function(pszServer:PChar;
                        sLevel: DWORD;
                        pbBuffer:Pointer;
                        cbBuffer:DWORD;
                        pcEntriesRead,
                        pcTotalAvial:Pointer):integer; stdcall;

var
NetSessionDelXP:function(ServerName,
                        UncClientName,
                        username:PWChar):DWORD; stdcall;

var
NetSessionDel:function( pszServer:PChar;
                        pszClientName: PChar;
                        sReserved: SmallInt):DWORD; stdcall;

var
NetFileEnumXP:function( servername,
                        basepath,
                        username:PWChar;
                        level:DWORD;
                        bufptr:Pointer;
                        prefmaxlen:DWORD;
                        entriesread,
                        totalentries,
                        resume_handle:LPDWORD):DWORD; stdcall;

var
NetFileEnum:function(    pszServer,
                        pszBasePath:PChar;
                        sLevel:DWORD;
                        pbBuffer:Pointer;
                        cbBuffer:DWORD;
                        pcEntriesRead,
                        pcTotalAvail:pointer):integer; stdcall;

var
NetFileClose:function(  ServerName:PWideChar;
                        FileId:DWORD):DWORD; stdcall;

var
NetFileClose2:function( pszServer:PChar;
                        ulFileId:LongWord):DWORD; stdcall;

var
GetIfTable:function(    pIfTable    : PMibIfTable;
                        pdwSize     : PULONG;
                        bOrder      : Boolean ): DWORD; stdcall;

var
  EfficForm: TEfficForm;

implementation
```

Кафедра_КБПЗ_2021рік

```
{$R *.dfm}

{ TEfficForm }

//
//  Спочатку нам потрібно визначиться, під якою системою ми працюємо,
// щоб довідатися яку частину коду (для XP чи ні) використовувати в цей момент.
// Для цього напишемо невелику функцію, що і буде визначати тип системи
моніторингу продуктивності застосунків на базі APM.
//

function TEfficForm.IsXP(var Value: Boolean): Boolean;
var Ver: TOSVersionInfo;
    BRes: Boolean;
begin
  Ver.dwOSVersionInfoSize := SizeOf(TOSVersionInfo);
  BRes := GetVersionEx(Ver);
  if not BRes then   //Перевірка
  begin
    Result := False; //Інформація не отримана
    Exit;            //ідемо
  end else
    Result := True;  //Інформація отримана

  case Ver.dwSchemeformId of //визначаємося
    VER_SCHEMEFORM_WIN32_XP       : Value := True;  //Windows XP - підходить
    VER_SCHEMEFORM_WIN32_WINDOWS  : Value := False; //Windows 9x - підходить
    VER_SCHEMEFORM_WIN32s         : Result := False //Windows 3.x - не підходить
  end;
end;

//
//  Одержання всіх відкритих загальних ресурсів
//

procedure TEfficForm.btnGetSharesClick(Sender: TObject);
var
  i:Integer;
  FLibHandle : THandle;
  ShareXP : PShareInfo2Array;     //<= Змінні
  entriesread,totalentries:DWORD; //<= для Windows XP
  Share : array [0..512] of TShareInfo50; //<= Змінні
  pcEntriesRead,pcTotalAvail:Word;        //<= для Windows 9x
  OS: Boolean;
begin
  lbxShares.Items.Clear;
  if not IsXP(OS) then Close; //Визначаємо тип системи моніторингу
продуктивності застосунків на базі APM

  if OS then begin  //Код для XP
    FLibHandle := LoadLibrary('NETAPI32.DLL'); //Завантажуємо бібліотеку
    if FLibHandle = 0 then Exit;
    //Зв'язуємо функцію
    @NetShareEnumXP := GetProcAddress(FLibHandle,'NetShareEnum');
    if not Assigned(NetShareEnumXP) then //Перевірка
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    ShareXP := nil; //Очищаємо покажчик на масив структур
    //Виклик функції
    if NetShareEnumXP(nil,2,@ShareXP,DWORD(-1),
                      @entriesread,@totalentries,nil) <> 0 then
    begin  //Якщо виклик невдалий вивантажуємо бібліотеку
      FreeLibrary(FLibHandle);
      Exit;
    end;
    if entriesread > 0 then //Обробка результатів
```

Кафедра КбПЗ 2021рік

```
    for i:= 0 to entriesread - 1 do
      lbxShares.Items.Add(String(ShareXP^[i].shi2_netname));
  end else begin //Код для 9x
    FLibHandle := LoadLibrary('SVRAPI.DLL'); //Завантажуємо бібліотеку
    if FLibHandle = 0 then Exit;
    //Зв'язуємо функцію
    @NetShareEnum := GetProcAddress(FLibHandle,'NetShareEnum');
    if not Assigned(NetShareEnum) then //Перевірка
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    //Виклик функції
    if NetShareEnum(nil,50,@Share,SizeOf(Share),
                @pcEntriesRead,@pcTotalAvail)<> 0 then
    begin  //Якщо виклик невдалий вивантажуємо бібліотеку
      FreeLibrary(FLibHandle);
      Exit;
    end;
    if pcEntriesRead > 0 then //Обробка результатів
    for i:= 0 to pcEntriesRead - 1 do
      lbxShares.Items.Add(String(Share[i].shi50_netname));
  end;
  FreeLibrary(FLibHandle); //Не забуваємо вивантажити бібліотеку
end;

//
//  Закриття загального ресурсу
//

procedure TEfficForm.btnCloseSharesClick(Sender: TObject);
var
  OS:Boolean;
  FLibHandle : THandle;
  Name9x:array [0..12] of Char;
  NameXP:PWChar;
  i:Integer;
  ShareName: String;
begin
  if not IsXP(OS) then Close; //Визначаємо тип системи моніторингу
продуктивності застосунків на базі АРМ

  if lbxShares.Items.Count = 0 then Exit;
  for i:= 0 to lbxShares.Items.Count -1 do
    if lbxShares.Selected[i] then Break;  //Шукаємо обраний елемент
  if not lbxShares.Selected[i] then Exit; //Якщо не знайдений ідемо
  ShareName := lbxShares.Items.Strings[i];

  if OS then begin //Код для XP
    FLibHandle := LoadLibrary('NETAPI32.DLL');
    if FLibHandle = 0 then Exit;
    @NetShareDelXP := GetProcAddress(FLibHandle,'NetShareDel');
    if not Assigned(NetShareDelXP) then //Перевірка
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    i:= SizeOf(WideChar)*256;
    GetMem(NameXP,i);  //Виділяємо пам'ять під змінну
    StringToWideChar(ShareName,NameXP,i); //Перетворимо в PWideChar
    NetShareDelXP(nil,NameXP,0);   //Видаляємо ресурс
    FreeMem(NameXP);  //Звільняємо пам'ять
  end else begin  //Код для 9x
    FLibHandle := LoadLibrary('SVRAPI.DLL');
    if FLibHandle = 0 then Exit;
    @NetShareDel := GetProcAddress(FLibHandle,'NetShareDel');
    if not Assigned(NetShareDel) then //Перевірка
    begin
      FreeLibrary(FLibHandle);
```

Кафедра_КбПЗ_2021рік

```
      Exit;
    end;
    FillChar(Name9x, SizeOf(Name9x), #0); //Очищаємо масив
    move(ShareName[1],Name9x[0],Length(ShareName)); //Заповнюємо масив
    NetShareDel(nil,@Name9x,0); //Видаляємо ресурс
  end;
  FreeLibrary(FLibHandle);
end;

//
//  Показу діалогу вибору директорії
//

function TEfficForm.SelectDirectory: String;
var
  lpItemID : PItemIDList;
  BrowseInfo : TBrowseInfo;
  DisplayName : array[0..MAX_PATH] of Char;
  TempPath : array[0..MAX_PATH] of Char;
begin
  FillChar(BrowseInfo, sizeof(TBrowseInfo), #0);
  BrowseInfo.hwndOwner := Handle;
  BrowseInfo.pszDisplayName := @DisplayName;
  BrowseInfo.lpszTitle := 'Specify a directory';
  BrowseInfo.ulFlags := BIF_RETURNONLYFSDIRS;
  lpItemID := SHBrowseForFolder(BrowseInfo);
  if Assigned(lpItemId) then begin
    SHGetPathFromIDList(lpItemID, TempPath);
    GlobalFreePtr(lpItemID);
  end else Result := '';
  Result := String(TempPath);
end;

//
//  Додавання загального ресурсу
//

procedure TEfficForm.btnAddSharesClick(Sender: TObject);
const
  STYPE_DISKTREE = 0;
  ACCESS_ALL =  258;
  SHI50F_FULL = 258;
var
  FLibHandle : THandle;
  Share9x : TShareInfo50;
  ShareXP : TShareInfo2;
  TmpDir, TmpName: String;
  TmpDirXP, TmpNameXP: PWChar;
  OS: Boolean;
  TmpLength: Integer;
begin
  TmpDir := SelectDirectory; //Визначаємо шлях до майбутнього ресурсу
  TmpName := InputBox('Share name','Enter name','Test'); //Визначаємо ім'я під
яким він буде видний у мережі
  if TmpDir = '' then Exit;

  if not IsXP(OS) then Close; //З'ясовуємо тип системи моніторингу
продуктивності застосунків на базі АРМ

  if OS then begin //Код для XP
    FLibHandle := LoadLibrary('NETAPI32.DLL');
    if FLibHandle = 0 then Exit;
    @NetShareAddXP := GetProcAddress(FLibHandle,'NetShareAdd');
    if not Assigned(NetShareAddXP) then
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    TmpLength := SizeOF(WideChar)*256; //Визначаємо необхідний розмір
```

Кафедра_КбПЗ_2021рік

```
    GetMem(TmpNameXP, TmpLength); //Конвертуємо в PWChar
    StringToWideChar(TmpName, TmpNameXP, TmpLength);
    ShareXP.shi2_netname := TmpNameXP; //Ім'я

    ShareXP.shi2_type := STYPE_DISKTREE; //Тип ресурсу
    ShareXP.shi2_remark := ''; //Коментар
    ShareXP.shi2_permissions := ACCESS_ALL; //Доступ
    ShareXP.shi2_max_uses := DWORD(-1); // Кількість максим. підключень
    ShareXP.shi2_current_uses := 0; // Кількість поточних підключень

    GetMem(TmpDirXP, TmpLength);
    StringToWideChar(TmpDir, TmpDirXP, TmpLength);
    ShareXP.shi2_path := TmpDirXP; //Шлях до ресурсу

    ShareXP.shi2_passwd := ''; //Пароль

    NetShareAddXP(nil,2,@ShareXP, nil); //Додаємо ресурс
    FreeMem (TmpNameXP); //звільняємо пам'ять
    FreeMem (TmpDirXP);
  end else begin
    FLibHandle := LoadLibrary('SVRAPI.DLL');
    if FLibHandle = 0 then Exit;
    @NetShareAdd := GetProcAddress(FLibHandle,'NetShareAdd');
    if not Assigned(NetShareAdd) then
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    FillChar(Share9x.shi50_netname, SizeOf(Share9x.shi50_netname), #0);
    move(TmpName[1],Share9x.shi50_netname[0],Length(TmpName)); //Ім'я
    Share9x.shi50_type := STYPE_DISKTREE; //Тип ресурсу
    Share9x.shi50_flags := SHI50F_FULL; //Доступ
    FillChar(Share9x.shi50_remark,
      SizeOf(Share9x.shi50_remark), #0); //Коментар
    FillChar(Share9x.shi50_path,
      SizeOf(Share9x.shi50_path), #0);
    Share9x.shi50_path := PAnsiChar(TmpDir); //Шлях до ресурсу
    FillChar(Share9x.shi50_rw_password,
      SizeOf(Share9x.shi50_rw_password), #0); //Пароль повного доступу
    FillChar(Share9x.shi50_ro_password,
      SizeOf(Share9x.shi50_ro_password), #0); //Пароль для читання
    NetShareAdd(nil,50,@Share9x,SizeOf(Share9x));
  end;
  FreeLibrary(FLibHandle);
end;

//
//  Помітьте що активний і неактивний час сесій буде даватися нам
//  у вигляді кількості секунд (тип Cardinal). Пропоную написати невелику
//  функцію, завдання якої буде перетворювати кількість секунд у більше
//  звичну форму відображення.
//

function TEfficForm.CardinalToTimeStr(Value: Cardinal): String;
var d,h,m,s: Real;
begin
    d:=0;
    h:=0;
    m:=0;
    s:=Value;
    if s > 59 then begin
        m:=int(s / 60);
        s:= s-s-(m*60);
    end;
    if m > 59 then begin
        h:=int(m/60);
        m:= m-m-(h*60);
    end;
```

Кафедра КСПЗ - 2021рік

```
    if h > 23 then begin
        d:=int(h/24);
        h:= h-h-(d*24);
    end;
    Result:='';
    if (d>0) then Result:=Result+floattostr(d)+' d. ';
    if (h<9) then Result:=Result+'0'+floattostr(h)+':' else
Result:=Result+floattostr(h)+':';
    if (m<9) then Result:=Result+'0'+floattostr(m)+':' else
Result:=Result+floattostr(m)+':';
    if (s<9) then Result:=Result+'0'+floattostr(s)     else
Result:=Result+floattostr(s);
end;

//
//  Одержання списку сесій
//

procedure TEfficForm.btnGetSessionsClick(Sender: TObject);
var
  OS: Boolean;
  FLibHandle : THandle;
  SessionInfo50: array [0..512] of TSessionInfo50;
  SessionInfo502 : PSessionInfo502Array;
  TotalEntries,EntriesReadXP: DWORD;
  EntriesRead,TotalAvial: Word;
  i:integer;
begin
  lvSessions.Items.Clear;

  if not IsXP(OS) then Close; //З'ясовуємо тип системи моніторингу
продуктивності застосунків на базі АРМ

  if OS then begin  //Код для XP
    FLibHandle := LoadLibrary('NETAPI32.DLL');
    if FLibHandle = 0 then Exit;
    @NetSessionEnumXP := GetProcAddress(FLibHandle, 'NetSessionEnum');
    if not Assigned(NetSessionEnumXP) then
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    SessionInfo502 := nil;
    if NetSessionEnumXP(nil,nil,nil,502,@SessionInfo502,DWORD(-
1),@entriesreadXP, @totalentries, nil)=0 then
    for i:=0 to EntriesReadXP-1 do
    begin
      with lvSessions.Items.Add do  //Заповнення даними зі структури
      begin
        Caption := string(SessionInfo502^[i].sesi502_cname); //Ім'я комп'ютера
        SubItems.Add(SessionInfo502^[i].sesi502_username);   //Ім'я користувача
        SubItems.Add(IntToStr(SessionInfo502^[i].sesi502_num_opens));
//Відкритих ресурсів
        SubItems.Add(CardinalToTimeStr(SessionInfo502^[i].Sesi502_Time)); //Час
активне
        SubItems.Add(CardinalToTimeStr(SessionInfo502^[i].sesi502_idle_time));
//Час не активне
      end;
    end;
  end else begin //Код для Windows 9x
   FLibHandle := LoadLibrary('SVRAPI.DLL');
   if FLibHandle = 0 then Exit;
   @NetSessionEnum := GetProcAddress(FLibHandle, 'NetSessionEnum');
    if not Assigned(NetSessionEnum) then
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
```

Кафедра_КБПЗ_2021рік

```
   if NetSessionEnum
(nil,50,@SessionInfo50,SizeOf(SessionInfo50),@EntriesRead,@TotalAvial) = 0 then
   for i:=0 to EntriesRead-1 do
   begin
      with lvSessions.Items.Add do  //Заповнення даними зі структури
      begin
        Caption := string(SessionInfo50[i].Sesi50_cname); //Ім'я комп'ютера
        SubItems.Add(SessionInfo50[i].Sesi50_username);   //Ім'я користувача
        SubItems.Add(IntToStr(SessionInfo50[i].sesi50_num_opens)); //Відкритих
ресурсів
        SubItems.Add(CardinalToTimeStr(SessionInfo50[i].Sesi50_Time)); //Час
активне
        SubItems.Add(CardinalToTimeStr(SessionInfo50[i].sesi50_idle_time));
//Час не активне
        SessionCloseKey[i]:= SessionInfo50[i].sesi50_key; //Унікальний
ідентифікатор для закриття
      end;
   end;
 end;
 FreeLibrary(FLibHandle);
end;

//
//  Завершення обраної сесії
//

procedure TEfficForm.btnCloseSessionClick(Sender: TObject);
var
  OS: Boolean;
  FLibHandle : THandle;
  CNameXP: PWideChar;
  CName9x: PAnsiChar;
  Key:SmallInt;
  i: Integer;
begin
  if not IsXP(OS) then Close; //З'ясовуємо тип системи моніторингу
продуктивності застосунків на базі APM

  if not Assigned(lvSessions.Selected) then Exit;
  i:= lvSessions.Selected.Index; //Визначаємо номер обраної сесії

  if OS then begin
    FLibHandle := LoadLibrary('NETAPI32.DLL');
    if FLibHandle = 0 then Exit;
    @NetSessionDelXP := GetProcAddress(FLibHandle, 'NetSessionDel');
    if not Assigned(NetSessionDelXP) then
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    //Перетворимо дані в необхідний вид
    CNameXP := PWChar(WideString('\\'+lvSessions.Items.Item[i].Caption));
    NetSessionDelXP(nil,CNameXP,nil);
  end else begin
    FLibHandle := LoadLibrary('SVRAPI.DLL');
    if FLibHandle = 0 then Exit;
    @NetSessionDel := GetProcAddress(FLibHandle, 'NetSessionDel');
    if not Assigned(NetSessionDel) then
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    //Перетворимо дані в необхідний вид
    CName9x := PAnsiChar(lvSessions.Items.Item[i].Caption);
    key := SessionCloseKey[i]; //Беремо ключ із масиву
    NetSessionDel(nil,CName9x,Key);
  end;
  FreeLibrary(FLibHandle);
end;
```

Кафедра_КбПЗ_2021рік

```
//
//  Одержання списку відкритих файлів
//

procedure TEfficForm.btnGetFilesClick(Sender: TObject);
var
  OS: Boolean;
  FLibHandle : THandle;
  FileInfoXP: PFileInfo3Array;
  FileInfo9x: array [0..512] of TFileInfo50;
  TotalEntries,EntriesReadXP: DWORD;
  EntriesRead,TotalAvial: Word;
  i:integer;
begin
  lvfiles.Items.Clear;

  if not IsXP(OS) then Close; //З'ясовуємо тип системи моніторингу
продуктивності застосунків на базі APM

  if OS then begin  //Код для XP
    FLibHandle := LoadLibrary('NETAPI32.DLL');
    if FLibHandle = 0 then Exit;
    @NetFileEnumXP := GetProcAddress(FLibHandle, 'NetFileEnum');
    if not Assigned(NetFileEnumXP) then
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    FileInfoXP := nil;
    if NetFileEnumXP(nil,nil,nil,3,@FileInfoXP,DWORD(-1),@entriesreadXP,
@totalentries, nil)=0 then
    for i:=0 to EntriesReadXP-1 do
    begin
      with lvFiles.Items.Add do  //Заповнення даними зі структури
      begin
        Caption := string(IntToStr(FileInfoXP^[i].fi3_id)); //Ідентифікатор
        SubItems.Add(FileInfoXP^[i].fi3_pathname);   //Шлях до файлу
        SubItems.Add(FileInfoXP^[i].fi3_username);   //Ім'я користувача
      end;
    end;
  end else begin //Код для Windows 9x
   FLibHandle := LoadLibrary('SVRAPI.DLL');
   if FLibHandle = 0 then Exit;
   @NetFileEnum := GetProcAddress(FLibHandle, 'NetFileEnum');
    if not Assigned(NetFileEnum) then
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
   if NetFileEnum (nil,
nil,50,@FileInfo9x,SizeOf(FileInfo9x),@EntriesRead,@TotalAvial)= 0 then
   for i:=0 to EntriesRead-1 do
   begin
      with lvFiles.Items.Add do  //Заповнення даними зі структури
      begin
        Caption := string(IntToStr(FileInfo9x[i].fi50_id)); //Ідентифікатор
        SubItems.Add(FileInfo9x[i].fi50_pathname); //Шлях до файлу
        SubItems.Add(FileInfo9x[i].fi50_username); //Ім'я користувача
      end;
   end;
 end;
 FreeLibrary(FLibHandle);
end;

//
//  Закриття файлу
//
```

Кафедра КбПЗ_2021рік

```
procedure TEfficForm.btnCloseFileClick(Sender: TObject);
var
  OS: Boolean;
  FLibHandle : THandle;
  i: Integer;
begin
  if not IsXP(OS) then Close; //З'ясовуємо тип системи моніторингу
продуктивності застосунків на базі АРМ

  if not Assigned(lvFiles.Selected) then Exit;
  i:= lvFiles.Selected.Index; //Визначаємо номер обраного файлу

  if OS then begin //Код для XP
    FLibHandle := LoadLibrary('NETAPI32.DLL');
    if FLibHandle = 0 then Exit;
    @NetFileClose := GetProcAddress(FLibHandle, 'NetFileClose');
    if not Assigned(NetFileClose) then
    begin
      FreeLibrary(FLibHandle);
      Close;
    end;
    NetFileClose(nil,StrToInt(lvFiles.Items.Item[i].Caption));//Закриваємо файл
  end else begin //Код для Windows 9x
    FLibHandle := LoadLibrary('SVRAPI.DLL');
    if FLibHandle = 0 then Exit;
    @NetFileClose2 := GetProcAddress(FLibHandle, 'NetFileClose2');
    if not Assigned(NetFileClose2) then
    begin
      FreeLibrary(FLibHandle);
      Close;
    end;
    NetFileClose2(nil,StrToInt(lvFiles.Items.Item[i].Caption));
  end;
  FreeLibrary(FLibHandle);
end;

//
//  Визначаємо вхідний - вихідний трафік
//

procedure TEfficForm.tmrTrafficTimer(Sender: TObject);
  // Допоміжна функція, що перетворить MAC адресу до "нормального" виду
  //Визначаємо спеціальний тип, щоб можна було передати у функцію масив
  type TMAC = array [0..7] of Byte;
  //Як перше значення масив, друге значення, розмір даних у масиві
  function GetMAC(Value: TMAC; Length: DWORD): String;
  var
    i: Integer;
  begin
    if Length = 0 then Result := ' 00-00-00' else
    begin
      Result := '';
      for i:= 0 to Length -2 do
        Result := Result + IntToHex(Value[i],2)+'-';
      Result := Result + IntToHex(Value[ Length-1],2);
    end;
  end;

//Сама процедура
var
  FLibHandle : THandle;
  Table: TMibIfTable;
  i : integer;
  Size   : integer;
begin
  tmrTraffic.Enabled := false; //Припиняємо про всякий випадок таймер
  lvTraffic.Items.BeginUpdate;
  lvTraffic.Items.Clear;  //Очищаємо список
  FLibHandle := LoadLibrary('IPHLPAPI.DLL'); //Завантажуємо бібліотеку
```

Кафедра КБПЗ_2021рік

```
  if FLibHandle = 0 then Exit;
  @GetIfTable := GetProcAddress(FLibHandle, 'GetIfTable');
  if not Assigned(GetIfTable) then
  begin
    FreeLibrary(FLibHandle);
    Close;
  end;

  Size := SizeOf(Table);
  if GetIfTable(@Table, @Size, false ) = 0 then //Виконуємо функцію
    for i:= 0 to Table.dwNumEntries-1 do begin
      with lvTraffic.Items.Add do begin //Виводимо результати
        Caption := String(Table.Table[i].bDescr); //Найменування інтерфейсу
        SubItems.Add(GetMAC(TMAC(Table.Table[i].bPhysAddr),
          Table.Table[i].dwPhysAddrLen)); //MAC адреса
        SubItems.Add(IntToStr(Table.Table[i].dwInOctets)); //Усього прийнято
байт
        SubItems.Add(IntToStr(Table.Table[i].dwOutOctets)); //Усього відправлено
байт
      end;
    end;
  lvTraffic.Items.EndUpdate;
  FreeLibrary(FLibHandle);
  tmrTraffic.Enabled := true; //Не забуваємо активувати таймер
end;

end.
```

Кафедра_КбПЗ_2021рік

Файл About.PAS - довідка про програму

```
unit About;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls;

type
  TForm2 = class(TForm)
    Image1: TImage;
    Memo1: TMemo;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var  Form2: TForm2;

implementation

{$R *.dfm}

procedure TForm2.Button1Click(Sender: TObject);
begin
Form2.Close;
end;

procedure TForm2.FormCreate(Sender: TObject);
begin
Memo1.Clear;
Memo1.Lines.Add(' БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА ');
Memo1.Lines.Add('');
Memo1.Lines.Add('на  тему:');
Memo1.Lines.Add('');
Memo1.Lines.Add(' Системи моніторингу продуктивності застосунків на базі АРМ');
Memo1.Lines.Add('');
Memo1.Lines.Add('');
Memo1.Lines.Add('Розробив: студент Підлубний Олексій Васильович');
Memo1.Lines.Add('                    гр. КІ-18-3СК');
Memo1.Lines.Add('');
Memo1.Lines.Add('Керівник: Доренський О.П ');
Memo1.Lines.Add('');
Memo1.Lines.Add('');
Memo1.Lines.Add('м. Кропивницкий 2021');
end;
end.
```

Кафедра_КбПЗ_2021рік