

Центральноукраїнський національний технічний університет
Центр заочної та дистанційної освіти
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2024 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи проектування
на основі Building Information Model”

Виконав здобувач вищої освіти
II курсу, групи КН-23Мз
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Черноліс Е.Р.
« ____ » _____ 2024 р.

Керівник проекту
кандидат технічних наук
_____ Смірнова Т.В.
« ____ » _____ 2024 р.
Рецензент _____

Центральноукраїнський національний технічний університет

Центр *Заочної та дистанційної освіти*

Кафедра *Кібербезпеки та програмного забезпечення*

Рівень вищої освіти *магістр*

Галузь знань 12 *“Інформаційні технології”*

Спеціальність 122 *“Комп’ютерні науки”*

Освітньо-професійна (освітньо-наукова) програма *“Комп’ютерні науки”*

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2024 року

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

Чернолісу Едуарду Руслановичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи проектування на основі Building Information Model*

2. Керівник роботи *Смірнова Тетяна Віталіївна, канд. техн. наук*

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

затверджені наказом вищого навчального закладу № 20-13 від 07.08.2024 року

3. Строк подання студентом роботи до захисту *2.12.2024 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи проектування на основі Building Information Model*

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Маркетингове та економічне обґрунтування ІТ-проєкту.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Доренська А.О.	05.10.2024	14.11.2024
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2024	16.11.2024

7. Дата видачі завдання « 6 » вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2024 р.	
3.	Розробка моделі компонента	20.10.2024 р.	
4.	Розробка структур даних	25.10.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2024 р.	
6.	Програмування алгоритмів	10.11.2024 р.	
7.	Розрахунок економічної ефективності	13.11.2024 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2024 р.	
9.	Оформлення ПЗ	17.11.2024 р.	
10.	Попередній захист роботи	2.12.2024 р.	

Дата видачі завдання
« 6 » вересня 2024 р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2024 р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Черноліс Е.Р. Дослідження та програмна реалізація системи проектування на основі Building Information Model. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи проектування на основі Building Information Model.

Метою розробки є дослідження та програмна реалізація системи проектування на основі Building Information Model.

Об'єктом дослідження є процес проектування на основі Building Information Model.

Предметом дослідження є методи проектування на основі Building Information Model.

Методи дослідження базуються на методах проектування Інтернету речей, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи проектування на основі Building Information Model.

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 з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: Комп'ютерні науки, Building Information Model

ABSTRACT

Chernolis E.R. Research and software implementation of the design system based on the Building Information Model. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this final qualification work for the second (master's) level of higher education, software was developed, which is intended for the design system based on the Building Information Model.

The purpose of the development is the research and software implementation of the design system based on the Building Information Model.

The object of research is the design process based on the Building Information Model.

The subject of research is design methods based on the Building Information Model.

Research methods are based on Internet of Things design methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the design system based on the Building Information Model.

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

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

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the Python environment.

Keywords: Computer science, Building Information Model

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	24
2.3 Розгорнута постановка завдання	25
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	27
3.1 Опис функціонування системи	27
3.2 Розробка структурної схеми.....	30
3.3 Розробка функціональної схеми	44
3.4 Розробка діаграми процесів.....	47
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	49
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	49
4.2 Захист розробленого програмного забезпечення.....	67
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	72
6 НАУКОВА НОВИЗНА	78

						ВКРМ-122.24.0017.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Черноліс Е.Р.				Дослідження та програмна реалізація системи проектування на основі Building Information Model	Літ.	Аркуш	Аркушів
Перев.	Смірнова Т.В.					М	1	106
Н.контр.	Коваленко А.С.					ЦНТУ КН-23Мз		
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	79
7.1	Визначення цільової аудиторії кінцевого готового продукту	79
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	79
7.3	Вибір методу оцінки вартості ПЗ	81
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	82
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	84
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	84
7.7	Визначення ключових факторів успіху конкретного проєкту.....	86
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	88
8.1	Вступ.....	88
8.2	Аналіз умов праці на робочому місці програміста.....	90
8.3	Розробка заходів з умов поліпшення охорони праці.....	93
8.4	Розрахункова частина	94
8.5	Висновки до розділу.....	96
9	ОСНОВНІ ВИСНОВКИ.....	98
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	100

КБПЗ-2024

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АРМ	–	автоматизоване робоче місце
АСУ	–	автоматизована система управління
ДБЖ	–	джерело безперебійного живлення
ДКС	–	домашня кабельна мережа
ДУ	–	дистанційне управління
ЕОМ	–	електронно-обчислювальна машина
ЕФ	–	екранна форма
ЗТО	–	звукова трансляція й оповіщення
ІБ	–	інтелектуальний будинок
ІЧ	–	інфрачервоний
ОВК	–	управління опаленням, вентиляцією й кондиціонуванням
ОДС	–	оперативна диспетчерська система
ОПС	–	охоронно-пожежна сигналізація
ПДУ	–	пульти дистанційного управління
ПЗ	–	програмне забезпечення
ПЛК	–	програмувальні логічні контролери
ПМО	–	програмно-математичного забезпечення
РК	–	рідкокристалічний
САПР	–	система автоматизованого проектування
СКК	–	система кабельних комунікацій
ТЗ	–	технічне завдання
ВІМ	–	Building Information Model
ЕІВ	–	європейська інсталяційна шина

ВСТУП

Актуальність теми. При проектуванні з використанням єдиної інформаційної моделі будинку (Building Information Model, BIM) замість безлічі розрізних креслень і кошторисів створюється відразу повна віртуальна модель майбутнього об'єкта. Всі інженери різних спеціальностей – архітектори, конструктори, фахівці зі слабкоstromових систем, електрики, опалення, вентиляції, кондиціонування, пожежогасіння – вносять свої зміни в єдиний документ. У цей час стандартом де-факто в галузі є проектування в традиційних САПР, скажемо Autodesk AutoCAD. У проектну команду звичайно входить від 10 до 50 чоловік, кожному інженерові виділяється своя ділянка робіт. У підсумку виходить 100-500 різних креслень, які необхідно звести воедино. Як правило, кожний проектувальник використовує свої звичні шаблони й сімейства, а креслення представлені у версіях від декількох дат. У такій ситуації головному інженерові проекту доводиться несолодко – завдання яке стоїть перед ним нагадує дуже складний, головоломний і досить дорогий тетріс. Вентиляційний короб цілком може накластися на трасу фреонопроводу й перетнути шинопровід. Електрики обов'язково спробують поставити ДГУ на місце капітальної стіни, тому що архітектори-конструктори не оновили вчасно свою версію документа. Головний інженер проекту й головний архітектор повинні все це побачити й скорегувати. Виходить не в усіх. А потім відповідно до підготовленого комплексу креслень приступають до будівництва, і отут починається трагікомедія – архітектурні шедеври начебто сходів, що впираються в капітальні стіни, або дверей, розташованих у метрі від підлоги, виявляються ввічнені в бетоні, якщо якісь ляпи при узгодженні залишилися непоміченими.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи проектування на основі Building Information Model.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

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

- Огляд існуючих систем проектування на основі Building Information Model.
- Дослідження системи проектування на основі Building Information Model.
- Програмна реалізація системи проектування на основі Building Information Model.

Об'єктом дослідження є процес проектування на основі Building Information Model.

Предметом дослідження є методи проектування на основі Building Information Model.

Методи дослідження базуються на методах проектування Інтернету речей, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод проектування на основі Building Information Model.
- Розроблено вітчизняний продукт проектування на основі Building Information Model, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі проектування на основі Building Information Model.

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

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2024 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи проектування на основі Building Information Model, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ_2024

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Замовники завжди незадоволені співвідношенням проектного кошторису й підсумкової вартості будівництва будинку. Мова не про те, що підрядник вантажівками вивозить матеріали із площадки. Більша частина цих сум проходить через графу «додаткові роботи», необхідність у яких виникла безпосередньо на будівельному майданчику, тобто це ціна переробок.

З'ясувати, хто винуватий у роздуванні бюджету – будівельники або проектувальники, дуже складно. Перші помиляються так само, як і другі, а часом, у силу специфіки, навіть більше. Однак об'єкт повинен бути уведений у лад, тому помилки усуваються вже в процесі будівництва, що позначається на підсумковій вартості об'єкта.

Необхідно відзначити, що найчастіше помилки зв'язані саме з недосконалістю методів, використовуваних при підготовці документації.

1.2 Область застосування

Областю застосування програмного забезпечення, що розробляється є розумний будинок. Розумний будинок – це інтелектуальна система, що поєднує всі комунікації й дозволяє централізовано ними управляти. Технологія дозволяє об'єднати системи опалення, водопроводу, освітлення, кондиціонування, сигналізації, відеоспостереження. Головна ідея інновації – це забезпечити безпека й енергозбереження. Як же на практиці йдуть справи зі зменшенням витрати ресурсів? Розберемося на прикладі окремо взятого столичного будинку. Вартість установки системи «Розумний будинок» під ключ для трикімнатної квартири – від 4000 €. До 40% цієї суми – витрати на необхідне встаткування. Цікаве

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

питання: скільки потрібно часу, щоб витрати окупилися і яка буде економія на ресурсах? Для підрахунків узяті ідеальні умови й наведені зразкові показники: контроль освітлення за допомогою датчиків руху й вимикання світла в 23.00; підтримка температури термостатами на рівні 22°C, відсутність одночасного включення холоду й обігріву.

Опалення

Централізоване керування над опаленням має на увазі контроль над температурою й за графіком роботи нагрівачів, усунення одночасної роботи опалювальних приладів і кондиціонерів. Такий розумний похід заощадить ще 5 кВт у день. Підсумкова сума економії – близько 5 тис грн – показує, що в реальності значимих результатів енергозбереження не існує. Строк окупності вкладень на установку системи «розумний будинок» при такому розкладі – 80 років, навіть при використанні дешевого встаткування – цей строк не буде менш 15 років. Перспектива, кінчено, не райдужна й з'являється питання про доцільність використання системи з метою енергозбереження. Але є більше позитивні приклади. Вартість системи на 1 кв. метр і строк окупності менше для об'єктів з великою площею. Технологія розумного будинку актуальна для заміських будинків. Підраховані нами раніше 5 кВт економії можуть скласти значну суму, що компенсує основну частину витрат на установку системи. «Розумний будинок» у більшості випадків – технологія не для економії, а для зручності. Про це свідчить той факт, що користувачі системи встановлюють його для керування побутовою технікою й освітленням. У таких випадках можна говорити тільки про емоційну задоволеність, чим про економічну доцільність.

В «Розумному будинку» залучається найбільше можливість контролю над енергоресурсами й роботою електроприладів, систем опалення й водопостачання. Система, керована штучним інтелектом, відключить подачу гарячої води, якщо хазяї відсутні, або прогріє повітря в приміщення до їхнього приходу. Відключення електроприладів у випадку їхньої непотрібності й коли хазяїв немає будинку, поради по оптимальній експлуатації побутової техніки, наприклад,

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

підказка запуснути пральну машину на нічний режим, тому що в цей час нараховуються витрати по зниженому тарифі. «Розумний будинок» не замінимо, коли мова йде про безпеку й форс-мажорних обставинах. У випадку задимлення, пожежі, затоплення, спрацьовуванні сигналізації система сповістить власників передбаченим для цього способом. Прибуття вчасно додому, перекриття газу, відключення води за допомогою дистанційного керування допоможуть уникнути небажаних наслідків і забезпечать збереження майна. У таких ситуаціях технологія окупає себе повністю, тому що вцілілий ремонт, цілісність електроприладів перевищують по сумі витрати на установку системи «розумний будинок». Зручність застосування технології оцінено споживачами, саме тому попит на неї росте. Розробки, нововведення в цієї сфери передбачають, що «розумний будинок» у майбутньому принесе власникам житла значну економію.

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи проектування на основі Building Information Model, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Огляд програми "ArchiCAD"

ArchiCAD – це професійний набір інструментів для проектування в області архітектури, будівництва й дизайну. Програмний продукт орієнтований на професійних архітекторів, дизайнерів і проектувальників. ArchiCAD – це не просто програма тривимірного моделювання, це – цілий комплекс рішень для професіоналів, що дозволяє вести всю документацію по будівництву від поетажних планів і розрізів будинків до специфікації матеріалів і будівельно-технічної документації.

Основні можливості програми "ArchiCAD":

– "ArchiCAD" професійний САПР для архітекторів, розроблений компанією Graphisoft. Призначений для проектування архітектурних рішень, а також елементів ландшафту, меблів і т.п.

– При роботі в пакеті використовується концепція віртуального будинку. Суть її полягає в тому, що проект ArchiCAD являє собою виконану в натуральну величину об'ємну модель реального будинку, що існує в пам'яті комп'ютера. Для її виконання проектувальник на початкових етапах роботи із проектом фактично будує будинок, використовуючи при цьому інструменти, що мають свої повні аналоги в реальності: стіни, перекриття, вікна, сходи, різноманітні об'єкти й т.д.

– Після завершення робіт над віртуальним будинком, проектувальник одержує можливість витягати різноманітну інформацію про спроектований об'єкт: поетажні плани, фасади, розрізи, експлікації, специфікації, презентаційні матеріали та ін. Підтримує взаємодія з різними програмами.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Огляд програми "КЗ-котедж"

"КЗ-котедж" – комплекс комп'ютерних програм для проектування дерев'яних будинків з оциліндрованого колоди й профільованого бруса. Комплекс "КЗ-котедж" призначений для розробки архітектурно-будівельної й виробничої документації, необхідної при виготовленні й складанні будинків. "КЗ-котедж" дозволяє прискорити випуск проектної документації й підвищити якість підготовки виробничих завдань по виготовленню будинків з оциліндрованих колод і профільованого бруса на базі комп'ютерних технологій.

Основні можливості програми "КЗ-котедж":

– Проектування дерев'яних будинків у програмі "КЗ-котедж" засновано на принципі «Від комп'ютерної моделі до повного комплекту документів». Конструкторові необхідно тільки створити модель майбутнього виробу, а всі креслення, специфікації, що управляють програми для ЧПУ програмний комплекс КЗ-котедж підготує в автоматичному режимі.

– Автоматичне одержання всього пакета документів виключає виникнення помилок, пов'язаних з «людським фактором».

– Створення проектів будинків, проектування котеджів, автоматичне одержання «разбревновки», розгорнень стін – це лише невелика частина завдань, на рішення яких орієнтований комплекс "КЗ-котедж".

– Програмний комплекс "КЗ-котедж" рекомендується для використання в проектних організаціях і підприємствах, що спеціалізуються на виробництві й монтажі будинків з оциліндрованих колод і профільованого бруса.

– "КЗ-котедж" – комплекс комп'ютерних програм для проектування дерев'яних будинків з оциліндрованого колоди й профільованого бруса.

– Комплекс "КЗ-котедж" призначений для розробки архітектурно-будівельної й виробничої документації, необхідної при виготовленні й складанні будинків. "КЗ-котедж" дозволяє прискорити випуск проектної документації й підвищити якість підготовки виробничих завдань по виготовленню будинків з оциліндрованих колод і профільованого бруса на базі комп'ютерних технологій.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

– Програма "КЗ-котедж" призначена для проектування дерев'яних будинків, лазень, альтанок і будь-яких конструкцій з колоди, профільованого бруса або лафета. Ви й ваш клієнт бачите реальну 3D-модель у кольорі, що допомагає зрозуміти особливості конструкції й одержати в підсумку якісний проект.

– Програма для проектування КЗ-котедж дуже простий в освоєнні, щоб розібратися з нею, у середньому, потрібно 3-4 дні, до того ж, із всіх аналогічних пропозицій "КЗ-котедж" сама доступна. Основним завданням програми є підготовка повного пакета документів для клієнта, виробництва й складання дерев'яних конструкцій (альтанки, лазні, котеджу).

– Програма розроблена спеціально для тих, хто будує будинки з оциліндрованого колоди й профільованого бруса, при цьому враховувався досвід більше 50 підприємств.

– Програма створена українською мовою, немає мовного бар'єра.

– Легко освоюється навіть тими, хто з комп'ютером знаком недавно.

– Виключається можливість помилки при передачі інформації на виробництво

– Скорочуються часові й трудові витрати на підготовку конструкторсько-технологічної документації.

– При внесенні змін у проект вам не прийде заново переробляти робочу документацію – вона буде автоматично відредагована програмою протягом декількох хвилин.

Огляд програми "Home and Landscape Design"

“Home and Landscape Design” – це програма для створення дизайну. Вона має ряд переваг: легке перетаскування створеного дизайну на інші предмети й техніки й 3d планування, потужний інструмент програм дозволяє створювати читацькі 3d об'єкти, і створювати розрахунок і оцінку витратних матеріалів. Одна з основних ізюминок, це те що вона дозволяє трудитися з нею як новачкам і професіоналам, якимось можливо будуть потрібні більше тонкі налагодження. В

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

“Home and Landscape Design” ви можете ввозити 3d об'єкти й ін, це випускає вам ще тисячі варіантів для ваших рішень.

Основні властивості програми "Home and Landscape Design":

– Дане програмне забезпечення має естетичну привабливість. Вона дозволяє застосовувати 2d лінійку й авоську й могутнішу свою 2d і 3d авоськи, це дає більше краще знання про дизайн.

– Дивіться як на малюнку добре переходять кольори при різних видах освітлення, ви так само зможете помітити блиск підлоги на кухні, глянути який розслаблюючої буде ванна кімната, виконана у всяких тонах. 3d деталі заявляються одними із самими прекрасними й техніки поліпшується з кожним роком. І це все на тлі легкості навігації в 3d світі програми. Home and Landscape Design володіє ізюминками інструментів. Вони добре реалізовані для дизайну, що робить їх потрібними. У вкладці “Головна” всі загальні типів будови й фундаменти й техніка поміщені на очах.

– Більше того, якщо вам дратує екстер'єр програми, ви можете поміняти її під свій смак, додавши для достатку наприклад посудомиєчну машину або шафу. Ви можете легко поміняти колір, і навіть розмір об'єктів у бібліотеці. І якщо вам дратує який або стандартний типу, можете засновувати свої на основі бібліотечних.

– Ще однією ізюминкою “Home and Landscape Design” є Realmodel. Вона дозволяє друкувати дизайн на папері або картоні й етикетками, якісь чітко встановлюють черга типів для легкого складання. Ви можете створити фактичну модель вашого будинку. Побачити на екрані це одне, але на паперовому або картонному макеті це виглядає більш реалістично так це звичайно доходить ідеально для презентацій.

– Програма Home and Landscape Design Professional має гарну опору й докладну тих документацію. Створено особливі відео уроки, що б допомогти вам уміть навчитися використовувати програмою. Інструкція в паперовому виді так само детальна й безмежна, у ній змальовані всі кроки, починаючи від баз, і закінчуючи добутком власних типів меблів.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

– Великий гарнітур функцій у сполученні з першокласними інструментами, і велика бібліотека об'єктів майструють “Home and Landscape Design” однією із кращих програм у своєму сегменті. Це відмінний інструмент для надання допомоги в реконструкції будинку й навіть ландшафтних ізюминок.

Огляд програми "VisiCon"

"VisiCon"це програма яка дозволяє оперативно й просто розробляти дизайн проект приміщення створена для людей, що не мають спеціальної технічної підготовки.

Основне призначення програми "VisiCon":

– Первинне проектування й перепланування квартири (оперативне пророблення ідеології майбутніх приміщень без докладної деталізацій проекту).

– Підбор найкращих варіантів організації простору й розміщення меблів у режимі реального часу на основі вже готових бібліотек тривимірних об'єктів.

– Підбор найкращих колірних рішень і використовуваних матеріалів.

– Точне створення дизайн проекту приміщення (одна кімната, кілька кімнат, план квартири) за допомогою "Майстра проектів" за кілька секунд. Розміщення меблів і предметів інтер'єра в заданому просторі, у т.ч. дизайн, оформлення й проектування кухні, ванної кімнати, спальні, вітальні, кабінету й інших функціональних приміщень на основі спеціалізованих бібліотек 3D елементів максимально наближених до реальних форм і розмірів.

– Використання матеріалів і текстур у дизайн проектах інтер'єрів приміщень. Перегляд отриманих результатів у кольоровому об'ємному зображенні й внесення всіх необхідних змін у дизайн проект. Одержання кількісної статистичної інформації з дизайн проекту (найменування й кількість використаних елементів, розміри, площі проектованого приміщення й т.п.).

– Одержання різних роздруківок проекту інтер'єра й збереження тривимірних зображень приміщення в різних ракурсах у файлах графічного формату.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

оформлення й перепланування квартири, підбирати найкращі варіанти організації простору квартири, кімнати, ванною, вітальною, кухні, спальні, офісу й т.п., розставляти меблі й предмети інтер'єра в режимі реального часу на основі вже готових бібліотек тривимірних об'єктів, підбирати найкращі колірні рішення в оформленні інтер'єра, а також переглядати створені дизайн-проекти в 3D-виді з різних ракурсів.

Існує дві версії програми "VisiCon":

– Базова версія "VisiCon 1.1 se" розроблена спеціально для домашнього використання й дозволяє створити точний дизайн-проект приміщення (одна кімната, кілька кімнат, план квартири). По кожному дизайн-проекту надається вся необхідна інформація – найменування й кількість використаних елементів, розміри, площі проєктованого приміщення й т.п. Створений дизайн-проект ви можете також зберігати, редагувати або роздруковувати на принтері.

– Професійна версія "VisiCon Pro 1.1" призначена для розробки дизайн-проектів офісних приміщень, ванних кімнат (у т.ч. Розрахунок кахельної плитки), кухні, проєктування торговельних площ і магазинів, а також для простий і оперативний дизайн будь-яких приміщень.

– Всі елементи створені на основі каталогів російських компаній і максимально наближені до реальних зразків і розмірів (більше 700 об'єктів). Крім використання стандартних елементів, є можливість замовити необхідні додаткові бібліотеки.

Системні вимоги програми "VisiCon":

- Потужність процесора від ~ Celeron 300 мГц.
- Windows 98 і вище.
- Пам'ять – 128 Мб і більше.
- Відеоконтролер SVGA 800?600, High Color (16 bit).
- USB, CD-ROM, інтернет, електронна пошта.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

3d maletti

Maletti Group представляє революційну програму 3D моделювання інтер'єрів салонів краси. Нові можливості дозволяють використовувати тривимірні зображення крісел, мийок, дзеркал, кушеток і інших перукарських меблів для створення дизайну приміщення вашого майбутнього салону краси. Передбачено функції вибору кольору оббивок, різні комплектації моделей, різноманітний вибір оздоблювальних матеріалів: від керамічної плитки до настінних картин і барвистих постерів. Простий і зрозумілий інтерфейс програми дозволяє з легкістю створювати приміщення, навіть якщо у вас немає досвіду роботи з 3d графікою.

Основні можливості програми Maletti:

- Програма працює в on-line режимі. Зареєстрованим користувачам сервіс надається безкоштовно. Процедура реєстрації займає всього пари хвилин.
- Основні можливості програми 3D дизайни. Створення об'ємної реалістичної моделі інтер'єра салону краси. 3D програма дозволяє вам за кілька хвилин створити свій власний дизайн-проект салону краси абсолютно безкоштовно! Крім того, програма дає чіткі подання про масштаби. Допустимо, у вас є приміщення 40 квадратних метрів. Ви плануєте відкрити салон, але точно не знаєте скільки робітників місць може розміститися на цій площі. Не знаєте як буде сполучатися модель, що сподобалася, крісла з обраною раніше мийкою. Завдяки нашій програмі, ви зможете зорієнтуватися в моделях, квітах, стилістиці й, головне, зрозуміти на яке завантаження й прохідність салону розраховувати. А це першочергова інформація для складання кошторису й бізнес-плану.
- Формування кошторису витрат на встаткування й оздоблювальні матеріали. Розміщення замовлення на меблі. У будь-який момент ви можете автоматично розрахувати вартість вашого проекту, з урахуванням витрат на перукарські меблі, ремонт і обробку.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Огляд програми "KitchenDraw"

"KitchenDraw" – програмне забезпечення для створення дизайну кухні й ванною кімнати. Це програмне забезпечення французького розроблювача призначено як фахівців, так і для починаючих користувачів. Програма дозволяє швидко й ефективно проектувати кухні й ванні кімнати, створювати каталоги продукції, видавати звіти й організовувати процес продажів. У ній також можна створювати складні приміщення (включаючи, наприклад, закруглені стіни, балки, сходи, стовпи й т.д.), вільно організовувати простір завдяки потужним функціям і розмаїтості параметричних об'єктів.

Основні можливості програми "KitchenDraw":

– Програма дозволяє проектувати одне або кілька приміщень довільної форми з установкою стін, вікон, дверей та ін., здійснювати контроль при установці стін з використанням різних стилів.

– Способи візуалізації: план, фасади, перспектива (сітчаста, згладжена, реалістична, фотореалістична), анімація, відео. Є конфігурація розрахунку цін на ті або інші матеріали й меблі, а також керування будь-якою кількістю валют. У програмний комплекс входить: дизайн інтер'єра, вибір стилів і автоматична заміна меблів з урахуванням стилю, розміщення меблів з інтегрованою технікою, мийки, витяжки та інше, автоматичне й ручне оброзмірювання, створення кошторисів і специфікацій.

– Програма "KitchenDraw" дозволяє швидко й ефективно проектувати інтер'єр внутрішнього приміщення. Це досить зручний продукт для приватних студій дизайну інтер'єра. У ньому передбачене створення свого каталогу об'єктів. Крім того, істи можливість створювати різні звіти, кошториси й організовувати процес продажів (ціноутворення, знижки, ПДВ, керування валютами й т.д.). Зручний і простий інтерфейс, велика кількість об'єктів дозволять у досить короткий термін створити закінчений вид інтер'єра в 3D.

– Оплата програми дуже своєрідна: скачавши програму із сайту розроблювача, їй можна користуватися безкоштовно 30 годин. Потім за кожен

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

годину використання стягується від €3 (максимально). При інтенсивному використанні передбачені численні бонуси й знижки.

Мінімальні системні вимоги програми "KitchenDraw":

- процесор Pentium 3 і вище;
- операційна система Windows 95 і вище;
- оперативна пам'ять – від 256 Мб;
- відеокарта SVGA у режимі 800x600 256 кольорів.

Огляд програми "Astron dizayn"

Програма "Astron dizayn" дозволить Вам змоделювати прямокутне приміщення, розфарбувати підлогу, стіни й потовк, установити двері й вікна, розставити меблі й вибрати підходящі аксесуари. Вам більше не загрожують стомлюючі експерименти, пов'язані з експериментами по перестановці меблів, коли з кімнати в кімнату доводиться переносити м'який куточок, дубова шафа й холодильник.

– В опціях програми "Astron dizayn" представлений найширший асортименти корпусних і модульних меблів: передпокої, вітальні, дитячі, спальні, шафи, бібліотеки, журнальні столики, малі форми й навіть вітрина.

– Усе, що вам необхідно тепер: запустити планувальник, правильно задати параметри приміщення, розташування дверей і вікон, вибрати меблі й можна сміли пускатися в самі сміливі експерименти. "Astron dizayn" не тільки не тільки заощадить ваші фізичні сили й час, але й зробить приємність широтою можливостей. "Astron dizayn" Придасться аматорам усього революційного в житті.

Огляд програми "Google SketchUp"

"Google SketchUp" – Цей програмний продукт був розроблений компанією "Google" спеціально для новачків в 3D-графіці. При цьому "SketchUp" більш ніж функціональний і з легкістю потягає з різними професійними пакетами 3D-графіки. "Google SketchUp" буває платна й безкоштовна, платна версія називається "Google Sketchup Pro".

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Безкоштовна "SketchUp" постачена гідним і цілком достатнім набором функції для повноцінної роботи із практичним будь-яким завданням по 3D моделюванню.

Основні можливості програми "Google Sketchup":

– Програма "SketchUp" володіє практичний самим інтуїтивним інтерфейсом, із всіх 3D програм що я тільки бачив. Цей унікальний спосіб побудови 3D моделей дозволяє миттєво побудувати прямокутники, окружності й інші примітиви, при цьому все це робиться прямо в 3D саме так як зручно й зрозуміло користувачеві. Програма дозволяє однаково легко працювати, як по точних розмірах, так і на око.

– Стандартний набір об'єктів програми невеликий, але окремі об'єкти можна скачати з інтернету, скориставшись можливостями самої програми. Крім цього в мережі можна знайти достатню кількість готових робіт з обробки інтер'єра, що, безумовно, буде гарною підмогою в освоєнні програми.

Ребра і граней: Це все є на нього:

– Кожна модель SketchUp складається тільки із двох речей: ребра й грані. Краю являють собою прямі лінії, і особи 2D форм, які створюються, коли кілька країв формують плоский контур. Наприклад, прямокутна особа зв'язані чотири ребра, які з'єднані між собою під прямим кутом. Для побудови моделей в SketchUp, ви малюєте ребра й грані, використовуючи кілька простих інструментів, які ви можете довідатися в невелику кількість часу. Це так просто.

Тягни/Штовхай швидко перейти від 2D до 3D:

– Видавите будь-яку плоску поверхню в тривимірному виді із запатентованої Push / Pull інструмента "SketchUp". Просто натисніть кнопку, щоб почати екструзії, наведіть курсор миші й натисніть знову, щоб зупинити. Ви можете Push / Pull прямокутник у коробку. Або намалювати контур сходів і Push / Pull його в 3D. Хочете зробити вікно? Push / Pull отвір, через стіну. SketchUp відома тим, що просто у використанні, і Push / Pull є причиною, чому.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Точні виміри: Робота з точністю:

– "SketchUp" є більшим для робітників швидко й вільно в 3D, але це більше, ніж просто модний електронний олівець. Оскільки ви працюєте на комп'ютері, усе, що ви створюєте в "SketchUp" має точний вимір. Коли ви будете готові, ви можете створювати моделі, які є точними, як вам потрібно, щоб вони були. Якщо ви хочете, ви можете роздрукувати масштабні види вашої моделі, і якщо у вас є "SketchUp Pro", ви навіть можете експортувати геометрію в інші програми, як "AutoCAD" і "3ds Max".

Створення складних профілів і форм:

– Ви можете використовувати інноваційні, "SketchUp", зроби всі Follow Me інструмент для створення 3D-форми шляхом видавлювання 2D-поверхонь уздовж заданого шляху. Змоделювати вигнутих труб шляхом видавлювання окружності уздовж L-образної лінії. Створіть пляшку, намалювавши половину її контуру, потім за допомогою Випливайте за мною, щоб змести його по колу. Ви навіть можете використовувати Follow Me, щоб закруглити (філе) краю на такі речі, як поручні, меблі й електронні гаджети.

Заливання: Застосування квітів і текстур:

– Ви можете використовувати Paint Bucket SketchUp, інструмент для малювання моделі з матеріалами, такими як кольори й текстури.

Груп і компонентів: створення моделей розумніше:

– Під "злипання" частини геометрії моделі, щоб зробити групи, можна створювати вкладені об'єкти, які легше переміщати, копіювати й приховувати. Компоненти багато в чому схожі на групи, але зі зручною особливістю: копії компоненти зв'язані один з одним, так що зміни, внесені в один автоматично відбиваються на всіх інших. Вікна, двері, дерева, стільці й мільйони інших речей вигоду від такого поводження.

Тіні: Виконати тіні досліджень і додати реалізму:

– Потужний, "SketchUp", тіней у реальному часі 3D движок дозволяє виконувати точні дослідження тінь на вашій моделі.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Розділи: Дивитися у вашій моделі:

– Ви можете використовувати інтерактивні розділи особливості SketchUp, щоб часово зрізати частину вашого проекту, дозволяючи заглянути усередину. Ви можете використовувати розділи для створення ортогональних видів (наприклад, плани поверхів), на експорт геометрії CAD програм з використанням SketchUp Pro, або просто, щоб одержати краще подання про вашу модель, поки ви працюєте на ньому. Розділ Літаки можна переміщати, обертати й навіть анімувати використанням Сцени функції SketchUp.

Сцени: Збереження видів і створення анімації:

– Ми створили сцени, щоб ви могли легко зберегти точний вид вашої моделі, щоб ви могли повернутися до них пізніше. Необхідно створити анімацію? Просто створіть кілька сцен і натисніть кнопку.

Подивитися навколо й Walk: Досліджуйте свої утвори з перших рук:

– SketchUp дозволяє проникнути усередину вашої моделі з набором простих інструментів навігації розроблений, щоб дати вам вид від першої особи. Натисніть, з позиції камери на "стояти" у будь-якому місці вашої моделі. Використовуйте подивитися навколо, щоб перетворити вашу віртуальну голову. Нарешті, перемкнетеся на Walk, щоб вивчити ваші утвори на ноги, ви можете навіть підніматися й спускатися сходи й пандуси, як ви граєте у відео грі.

Розміри й мітки: Додати інформацію ваших проектів:

– Ви можете використовувати супер-інтуїтивний виміри й етикетку інструменту, щоб додати розміри, анотації й інші славні подробиці вашої роботи.

Викладач: Піймати на швидко:

– Інструктор SketchUp у діалоговому вікні, що ви можете активувати в будь-який час, надає контекстно-контекстно-залежної довідки.

Шари й Outliner: Будьте організовані:

– Коли ви будете великі, складні моделі, все може заплутатися дуже швидко. "SketchUp" надає два корисні способи зберегти вашу геометрію керованим:

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

"Google Earth": Дивитесь ваші моделі в контексті:

– "SketchUp" і "Google Earth" є частиною того самого сімейства продуктів, тобто ви можете обмінюватися інформацією між ними легко. Потрібна будівельний майданчик для вашого проекту? Імпорт масштабних аерофотознімків, у тому числі топографію, прямо з "Google Earth" в "SketchUp", нажавши одну кнопку. Хочете бачити свій "SketchUp" моделі в контексті, в "Google Earth"? Натисніть іншу кнопку, і ви можете. Кожної може використовувати "SketchUp" для створення моделей, які можна побачити когонебудь в "Google Earth".

Інструменти пісочниці: Робота на місцевості:

– "SketchUp" у пісочниці інструменти дозволяють створювати, оптимізувати й змінювати 3D місцевості. Ви можете створити гладку пейзаж з безліччю імпортованих ліній контуру, додати уступи й долини для стоків, а також створити площадку будинку й дороги.

3D-моделі: знайти моделі майже все, що вам потрібно:

– Бібліотека 3D-моделей, онлайн-сховище 3D-моделей, які ви можете знайти через коли тобі щось потрібно. Навіщо будувати щось, коли ви можете скачати його безкоштовно?

Імпорт 3DS: Одержати перевагу на вашій моделі:

– Ви можете імпортувати файли 3DS безпосередньо в моделі "SketchUp". У меблів у форматі 3DS, що ви хотіли б використовувати? Імпорт його, а потім зберегти на Truckin.

Імпорт зображень: красити стіни з фотографіями:

– З SketchUp, ви можете імпортувати файли зображень як JPG, TIFF, PNG файлів і PDF-файлів. Ви можете використовувати їх самі по собі (щось начебто плакатів), але ви також можете прикріпити їх до поверхні, щоб створити фотореалістичні моделі будинків, конструкцій упакувань і багато чого іншого.

Експорт TIFF, JPEG і PNG:

– SketchUp дозволяє експортувати растрові зображення до 10000 пікселів площі, так створенні образа, що ви можете відправити по електронній пошті, публікувати в документ або проект на стіні також легко, як вибрати кілька варіантів і натиснути кнопку Export.

PRO імпорту й експорту DXF і DWG: 2D-креслень і 3D-моделей:

– SketchUp Pro дозволяє імпортувати й експортувати DXFs і DWGS, що дає вам простий спосіб переміщення плани, розрізи, фасади або навіть всієї вашої моделі в (і з) Ваші улюблені програми CAD. Імпортувати й експортувати геометрію залишається в масштабі 1:1, і шари зберігаються.

PRO Експорт PDF і EPS: 2D векторних зображень:

– З Pro версія SketchUp, ви можете експортувати видом своїх моделей у форматі PDF і EPS, що дозволяє продовжувати працювати з ними, у векторній редагування програм, як Illustrator і Freehand. Для 2D-зображень, які повинні бути залежать від дозволу, ніщо не зрівняється експорту в ці формати.

PRO Експорт 3DS, OBJ, XSI, FBX, VRML і DAE:

– Якщо ви використовуєте SketchUp Pro, ви можете експортувати моделі для ряду корисних 3D-форматів. Професіонали використовують кілька різних інструментів, і це дозволить експортерам SketchUp приєднатися до найбільш професійні робочі процеси, пропонуючи сумісність практично з кожної популярного додатка 3D моделювання в існуванні.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – це потужна мова програмування, яка проста у вивченні. Він має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис і динамічна типізація Python разом з його інтерпретованим характером роблять його

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

в роботі технологічного обладнання;

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

КБПЗ – 2024

					VKPM-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

САПР рівня AutoCAD з'явилися більше сорока років тому, що після кульманів і креслярських дощок сприймалася як революція. Завдяки цим програмам продуктивність праці інженерів збільшилася в десятки разів, однак по суті в них була реалізована та ж ідеологія креслярських інструментів, тільки в цифровому форматі.

У програмах для BIM, наприклад в Revit від Autodesk, використовується інший підхід. Його можна зрівняти з віртуалізацією роботи скульптора або дизайнера. Як макетують автомобілі? Група дизайнерів покриває дерев'яний каркас шаром пластиліну й за допомогою різців і шпательів вирізує з нього модель, надаючи їй бажаний вигляд. Кожний член групи бачить результати роботи своїх колег, тому на такій моделі два різних контури однієї й тих же дверей точно не виявляться накладені один на іншій. А тепер представте, що крім дизайнерів у той же самий час над моделлю працюють фахівці із двигунів, трансмісії й т.д. Всі вони одночасно створюють у віртуальному середовищі різні вузли – це, по суті, і є BIM.

Зміну САПР на BIM можна зрівняти зі зміною релігії. Міняти переваги в області ПЗ завжди важко. З BIM ситуація ще більш складна. Основи роботи в традиційних САПР викладають в інститутах і на курсах, а навчальних матеріалів по BIM поки мало, тим більше українською мовою, тому й фахівців із впровадження в Україні ще менше. При цьому не слід забувати, що будь-яка річ зростає в ціні пропорційно своїй дефіцитності.

ПЗ BIM відрізняється від AutoCAD тим, що воно на порядок складніше для освоєння й вимагає зміни як мислення інженера, так і ідеології роботи всієї проектної команди. Я не зря згадав про зміні релігії. BIM – це не аркуш ватману,

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

на якому рисується креслення: у даному програмному продукті кожний параметр елемента будинку задається у властивостях самого елемента. При його зміні в розрахунок включаються вже оновлені властивості цього елемента.

Перерахунок у програмі відбувається автоматично, протягом декількох митей. Для кожного елемента інфраструктури створюється набір найрізноманітніших параметрів: розміри, фізичні характеристики конструктивних елементів і комунікацій, положення щодо інших об'єктів, правила взаємодії з ними й т.д. Таких параметрів дуже багато – сотні й навіть тисячі. Таким чином, будинок стає параметричним.

При впровадженні ВІМ може виявитися, що співробітники пройшли навчання, а ніхто так нічого й не зрозумів. Без усяких жартів. Подібне нововведення завжди ризик. Воно може не прижитися, саботуватися колективом або не сподобатися замовникам, і тоді гроші виявляться витраченими впусту.

Що це дає клієнтам? На перший погляд, для них це обертається подорожчанням проектних робіт. На перевірку ж – більше скромним загальним бюджетом об'єкта при кращій якості й меншій тривалості будівництва. Інакше кажучи, за допомогою ВІМ об'єкт планується ретельніше й, відповідно, якісніше, а будівництво здійснюється швидше.

Як конкретний приклад розглянемо спроектований нами модульний розумний будинок. Понесені ним витрати будемо природно порівнювати з вартістю об'єкта, спроектованого за допомогою традиційної САПР. Звичайно, паралельного проектування ніхто не робив, але при наявності досвіду оцінку його вартості зробити неважко.

Підготовка документації по стадії «П» (проектна документація) і «Р» (робоча документація) зажадала приблизно на 20% більше трудозатрат, чим при використанні традиційних методів проектування. Однак документація була здана протягом п'яти місяців замість звичайних шести. Один місяць нам заощадив головний інженер проекту, що завдяки ПЗ був урятований від необхідності виловлювати взаємопересічні комунікації та інші «міни».

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Узгодження із замовником і переробки проекту в BIM пройшли істотно швидше – приблизно на 50%, або за 15 днів замість місяця. Крім всіх креслень, замовникові була надана 3D-модель його будинку, де він міг наочно побачити, а потім швидко ідентифікувати компоненти, які вимагали змін.

Зміни теж вносилися більш оперативно, тому що BIM сама робить розрахунок і модифікує елементи систем відповідно до нових параметрів (пам'ятаєте, що говорилося про параметричну модель?). Наприклад, місце розташування колектора системи холодопостачання було вирішено змінити. Програма сама вирізала нові отвори в стінах, поміняла місця кріплень, перенесла трасу трубопроводів системи пожежогасіння щоб уникнути перетинань, а також перерахувала гідравлічні характеристики трубопроводів. Правда, щоб усе відбувалося так красиво й швидко, необхідно провести колосальну роботу з настроювання системи, розробці шаблонів і сімейств, вибудовуванню логічних зв'язків і багато іншому. Специфікація матеріалів теж розраховується автоматично, а от кошторис на роботи – як і раніше вручну.

Отже, проектна документація погоджена, наступає стадія будівництва. У випадку з нашим розумний будинком на це теж піде менше часу. Віртуальну модель будинку можна завантажити на будь-який планшет і показати на будь-якому комп'ютері.

По своїй реалізації й наочності віртуальна модель нагадує ігри на зразок Minecraft, де легко орієнтуються навіть діти. Вибравши вид від першої особи, можна ходити по віртуальному будинку. Всі інфраструктурні системи підписані й підсвічені різними кольорами. Завдяки такій тривимірній візуалізації монтажники легко зрозуміють, де повинен проходити, скажемо, трубопровід пожежної системи, під яким кутом він обгинає труби системи кондиціонування і як простіше змонтувати насадки системи газового пожежогасіння. Крім того, технологія BIM дозволяє змодельювати весь процес будівництва й при необхідності оцінити весь життєвий цикл об'єкта, починаючи від заливання фундаменту й закінчуючи демонтажем будівельних конструкцій і їхньою утилізацією.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Найважливіше, що в документації немає помилок з вини проєктувальників, тому глобальних переробок у процесі будівництва не буде потрібно. Традиційно на випадок переробок у бюджет завжди заставляється ледве більша сума. Від цієї завищеної вартості ми можемо укластися в 85%, тим самим заощадивши замовникові 15% від вартості будівництва. Витрати скорочуються й завдяки більше швидкому зведенню об'єкта.

Таким чином, повністю готовий об'єкт виявляється на 10-15% дешевше аналогічного, створеного із застосуванням традиційних САПР.

В 2014 році Європейський союз прийняв закон, відповідно до якого всі будівельні компанії, що працюють із держпідрядами, будуть зобов'язані створювати проєкти й кошториси винятково в BIM. Норма набуде чинності в 2016 році. Це рішення обґрунтовується тим, що системи BIM дозволяють більш точно прогнозувати бюджет будівництва, що для єврочиновників критично важливо. Таким чином, у будівельній галузі Європи BIM стає стандартом де-факто. У нас, як звичайно, свій шлях, але чому б не скористатися позитивним досвідом наших сусідів?

3.2 Розробка структурної схеми

Ще якихось 30-50 років тому з корисними й незвичайними функціями в житловому приміщенні люди зіштовхувалися тільки в кінофільмах. Люди вважали технічні нововведення й можливості систем вигадкою режисерів і незбутньою фантастикою. Сьогодні таке стало можливим за рахунок технологій «розумного будинку», які складаються зі спеціальних електродатчиків, що управляють компонентів і виконують пристосувань і пристроїв.

Можливості «розумного будинку»

Вважається, що в кожному будинку є свій стиль і характер, своя специфіка, що виділяють його серед інших. Це ставиться до габаритів і архітектури, дизайну й оформленню. До індивідуальностей будинку можна

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

віднести його можливість виконувати різні корисні й охоронні функції. Якщо приміщення обладнане високотехнологічними електронними системами, які дозволяють з'єднати всі комунікаційні системи в одну під керуванням електронного розуму, то його називають «розумний будинок». Нові технології розумний будинок можуть контролювати різні функції:

- локального й віддаленого відеоспостереження;
- охоронної й пожежної сигналізації;
- вуличного освітлення;
- відкривати ширму, висувати й вмикати телевізор;
- закривати або відкривати віконні штори або жалюзі;
- час від часу включати й виключати світло, імітуючи те, що в приміщенні хтось є;
- перемикає яскравість світла;
- дистанційно включати обігрівач у приміщенні;
- автоматично відкривати ворота, шлагбауми;
- включати вентиляцію;
- змінювати режими використання встаткування у випадку аварійних ситуацій;
- віддаленого контролю й керування всіма елементами приміщень і прибудинкової території через мережу інтернет.

Особливості системи нових технологій розумний будинок

Механізми в «Розумному будинку» спрямовані на централізований контроль над всіма системами життєзабезпечення й комфорту. Бажано попередньо закладати в проектну документацію установку нові технології розумний будинок. Такі нововведення привітають багато проектних організацій і враховують їх при створенні дизайн-проекту. Однак і вони не реалізують свої знання й уміння на 100%. Тому в деяких житлових приміщеннях встановлено тільки кілька корисних пристроїв. Інше житло укомплектоване безліччю пристосувань, який користуються постійно або час від часу. До приладів і

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

функцій «Розумного будинку» кожний хазяїн ставиться також по-різному. Для когось це – тільки звична необхідна функція, а хтось не зупиняється на наявному матеріалі, виношуючи цікаві ідеї. Його охоплює своєрідний азарт по встаткуванню свого житла всякими цікавими й корисними штучками. У цьому випадку садиба буде еволюціонувати разом з потребами хазяїв.

Переваги системи

Подібно тому, як у свій час, ми обожнювали телевізор з пультом, так сьогодні багато громадян ставляться до програмувального й настроєного під будь-які потреби внутрибудинкового й зовнішнього встаткуванню.

Переваги нових технологій розумного будинку:

- корисне виконання різних функцій.
- різноманітні асортименти послуг.
- упорядкована, координована й взаємозалежна робота всіх пристосувань.
- відмова від штату прислуги й охоронців.
- економія ресурсів при відсутності хазяїв.
- максимальний комфорт і безпека.

Економічність і економія

Високотехнологічна система, що поєднує всі комунікації в одну мережу, установлюється не дуже часто у звичайних апартаментах. Це доступно великогабаритним дорогим віллам, вимагає істотних витрат як в установці, так і у випадку виникнення необхідності проведення ремонтних робіт, усуненні неполадок у мережах. Але останнім часом є ощадливі варіанти облаштуваності садиби високотехнологічними системами. Їхня вартість можна зрівняти із ціною монтажу звичайного домофона. Сама система також може бути ощадливою. З її допомогою можна заощаджувати воду, тепло, електроенергію. Система може контролювати опалення, зменшуючи подачу газу по мінімуму в той час, коли в обителі немає людей. Розумні функції садиби по охороні виключають витрати на додатковий зміст охоронців або підключення сигналізації до міської охоронної мережі.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

Як лінія передачі система LSN використовує додаткову жилу стандартного кабелю марки NYM. Це дозволяє заощаджувати на прокладці додаткового шинного кабелю й не вимагає проведення додаткової мережі. Передача даних відбувається в первинній смузі частот.

Принцип передачі даних

Для реалізації процесу обміну даними необхідна наявність середовища передачі. Дотепер розглядалися такі лінії передачі, як кручена пара, радіочастоти або модульований сигнал несучої частоти, переданий по електромережі напругою 220 В. У системі LCN дані передаються по додатковій жилі стандартного проведення NYM. Тобто при монтажі системи використовується пятижильний кабель NYM. Якщо він є в наявності, виходить, будинок підготовлений до установки сучасного шинного встаткування. Передача даних відбувається в первинній смузі частот. При цьому відмовляються від використання несучої частоти, а швидкість передачі даних становить 9600 бод. При такій швидкості передачі система LCN може відправляти до 100 телеграм у секунду. Радіус дії системи становить 1000 метрів. Цього цілком достатньо для обслуговування великого житлового будинку. Максимальне число LCN модулів, які можуть бути з'єднані один з одним таким способом, становить 254.

Уже із цього можна зробити вивід, що система LCN підходить для застосування не тільки в житловому будинку, але й у великих об'єктах цільового призначення з більшим числом поверхів і кількістю приміщень до 100.

Кожний модуль LCN черпає енергію із власного блока живлення. Це значить, що модулі є взаємно незалежними. Таким чином, при бажанні можна приступитися до установки шини з одним-єдиним модулем. Існує можливість за допомогою високої щільності інтеграції помістити сенсорику, акторику й блок живлення в один модуль.

Практична інсталяція

Така висока інтеграція дозволяє фахівцеві проводити інсталяцію тим способом, до якого він звик. У тому місці, де раніше встановлювався вимикач,

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

тепер розташовується модуль LCN. Однак при будівництві необхідно передбачити глибину розеток для убудованого монтажу, щоб вони були сумісні з модулем діаметром 50 мм і глибиною 20 мм.

Старі види вимикачів замінюються клавішами. Якщо зі зворотної сторони стіни в прихожій розташована клавіша, вона також приєднується до убудованого в кімнаті модулю.

Живильні проведення світильників (максимум дволампових) підключаються безпосередньо до двох 230-вольтових виходів LCN-модуля.

Таким чином, без зміни монтажу системи клієнт одержує ряд нових функцій:

- Плавне регулювання світла із двох незалежних виходів з будь-яким часом диммування.
- Наявність входу для ІЧ-дистанційного керування з $16 \times 3 = 48$ функціями.
- Динамічне групове формування.
- Рахункові й арифметичні функції керування.
- Система візуалізації (вивід інформації на табло).
- Миготлива лампа як датчик сигналу для людей з ушкодженням слуху, наприклад замість дверного дзвінка.
- Облік і обробка даних.

У приватному будинку напевно затребувані не всі можливості, які пропонує система LCN. Проте, фахівець із монтажу увесь час одержує замовлення від клієнтів на нестандартне виконання роботи. Це веде до збільшення зобов'язань перед клієнтом і, крім того, демонструє професіоналізм і компетенцію фахівця.

Технологія LCN надає можливості підвищення кваліфікації й освоєння нових областей. Наприклад, якщо електрик може приєднати до убудованого модуля тільки маленький температурний сенсор розміром 10x20 мм, те йому не важко буде розібратися в технології регулювання температури й освоїти область створення штучного мікроклімату.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

Принцип функціонування шини LCN

Шина LCN є командно-орієнтованою системою. Відправлена клавішею телеграма-команда повністю описує виконавчу функцію. Ця команда могла б звучати, наприклад, у такий спосіб: «активізувати другий вихід через 5 секунд на 90% освітленості».

Цей процес відрізняється від колишньої практики: програмування функції в активаторі й вимоги її виконання за допомогою подачі команди, наприклад перемиканням клавіші. LCN використовує для передачі даних мінімум 24 біта корисної інформації, що перебуває в сенсорі, а не в активаторі. Ця відносно висока маса даних приводить до більшої гнучкості програмування.

Таким чином, 10 різних клавіш відповідними символами описують 10 різних часів, збільшення й зниження яскравості освітлення або зміни світлових сцен. Це відбувається завдяки простому збереженню команди у встановленому клавішному вимикачі.

Зв'язок з іншими системами

Сполучення системи LCN з іншими системами легко здійснено завдяки командно-орієнтованій роботі.

Установники обох шин не повинні домовлятися, щоб призначити функцію, наприклад, для активатора. Кожний може розпорядитися відповідним активатором по власному розсуді. Необхідно тільки знати, які є модулі в системі LCN. Цю інформацію можна одержати безпосередньо із шини. При цьому монтери одержує не тільки інформацію про функції відповідного приладу, але й точний опис завдань модулів у текстовій формі.

Повідомлення про поточний робочий стан можна одержати із шини в будь-який момент. Тому система LCN сумісна з іншими системами.

Не тільки в об'єктах цільового призначення, але й в Smart Home бажано постійно або по запити мати подання про стан роботи сенсорів і активаторів. У цьому випадку шинна система має незаперечну перевагу перед звичайною кабельною системою. У будь-якому місці шини можуть бути встановлені табло

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

або так звана Home-Management-System. Сигнальне проведення, необхідний у випадку звичайного кабельного з'єднання, є тут зайвим. Це забезпечує скорочення інсталяційних витрат, а разом з тим і вартості системи.

Також важливі ролі грають механізми індикації статусу й квітування. У випадку деяких систем створюється враження, що про це просто забули. Іноді пропонується спрощений метод контролю: сигнальна лампочка показує, що ввімкнувся активатор. Чи відбулося це насправді, залишається невідомим.

LCN має у своєму розпорядженні триступінчасту систему індикації статусу системи: функціональне квітування, індикація статусу, команди статусу.

1. Функціональне квітування. Необхідно запитувати функціональне квітування для кожної телеграми. У ході цього процесу повинне виконуватися квітування не тільки прийому повідомлень, але й виконання заданих функцій. Наприклад, прийнята команда «опустити жалюзі». Через застопорення механізму або іншого робочого стану вона не може бути виконана. У такому випадку не досить відіслати назад квитанцію «команда прийнята». Система LCN посилає службове повідомлення «команда не виконана» і, більше того, указує причину відмови. Подальшим важливим аспектом є безпосередній зв'язок передавача із приймачем і той факт, що LCN-приймач може самостійно створювати телеграми квітування. Якщо в проміжках включені маршрутизатори або з'єднувачі, то вони автоматично передають функціональну квитанцію далі. Причина, по якій механізм квітування не функціонує в системі EIB, полягає в тому, що формування квитанції відбувається вже на першому з'єднувачі, хоча приймач ще взагалі не одержав ніякої команди.

2. Індикація статусу. Статус всіх входів і виходів модулів системи LCN постійно перевіряється спеціальною програмою-монітором. Тому кожна команда, кожний вхідний сигнал, а також сигнал про несправність моментально надходять у шину. LCN має у своєму розпорядженні спеціальний канал повідомлення, що видає інформацію про кожне повідомлення в режимі реального часу від місця виникнення події. Світлові табло або Home-Management-System постійно

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

опитують канал. Внаслідок цього налагодження режиму візуалізації не викличе в майстра ніяких складностей. Майстер повинен вибрати зі списку всіх модулів модуль індикації й визначити параметри відображення: статус входів, виходів, реле або показання інших табло або установок візуалізації. Стан лампочок на світловому табло може підтримувати чотири режими: «вмк/вимк/мигає» і «мерехтить». Лампочка може мигати, якщо в групі світильників який-небудь із них не працює. При наявності системи Home-Management-System можливо навіть визначення точного місця розташування неробочого світильника. Також можна створювати запит на те, щоб одночасно відображалися перше й останнє значиме повідомлення. За допомогою реєстра сум можна зв'язувати й обробляти повідомлення. Обшир функцією формування повідомлення керує модуль системи. У якості табло використовується 4+1-діодна стандартна клавішна панель EIB.

3. Команди статусу. Команда статусу – це третій щабель системи індикації LCN. Вона дає можливість компоненту відправляти команду на шину завжди, коли що-небудь діє на вході або виході модуля. Наприклад, вихід модуля, відповідальний за освітлення, міг би також управляти вентилятором у ванню. Тому що працюючий вентилятор створює сильний шум, то під час користування душем можна змусити його працювати тільки на 25%. Після того як людина покине ванну кімнату й виключить світло, вентилятор увімкнеться на повну потужність і зробить ґрунтовне очищення повітря. Для цього необхідно виконати тільки дві LCN-команди.

Інсталяція системи LCN не викличе в електрика ніяких труднощів, тому що вона не вимагає попередніх робіт, що підготовляють будинок до установки сучасного встаткування. Наявність вільної жили кабелю NYM є достатньою умовою для початку установки системи. Система не має потреби в блоці живлення, тому витрати на її інсталяцію значно знижуються. Завдяки великій кількості функцій, які можуть одночасно виконувати модулі, потрібна менша кількість спеціальних компонентів. Висока потужність шини LCN

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

забезпечує навіть роботу великого встаткування. LCN пересилає не тільки окремі біти, а докладні функціональні описи. Це забезпечує у свою чергу високу гнучкість у параметруванні встаткування. Для навчання параметруванню, згідно даним виробника, досить одного дня.

Спеціалізоване програмне забезпечення (ПЗ) на комп'ютері загального призначення можна розділити на кілька груп:

– Функціонально закінчене ПЗ, орієнтоване на виконання конкретних споживчих завдань. Прикладом може служити програма Home Assistant компанії Siemens. У рамках цієї програми вже створені користувацькі інтерфейси для керування освітленням, опаленням, електроспоживанням й іншим устаткуванням. Залишається тільки настроїти програму, тобто прив'язати її до реальних датчиків і виконавчих механізмів. Такі програми мають убудовану підтримку EIB-протоколу й забезпечують зв'язок з інсталяцією по послідовному інтерфейсу RS-232 та USB.

– Інструментарій для створення користувацьких інтерфейсів. Багато постачальників EIB компонент (Siemens, Merten, ABB, Gira і ін.) пропонують так звані програми візуалізації, які дозволяють інсталятору самостійно формувати користувацький інтерфейс із бібліотеки елементів і зв'язувати їх із установленим устаткуванням. Як і попередня група програм, вони також мають убудовану підтримку EIB-протоколу.

– Системне програмне забезпечення, що забезпечує доступ до об'єктом EIB-інсталяції з інших програм з підтримкою сучасних стандартів обміну даними по локальних і глобальних комп'ютерних мережах. Гарним прикладом такого ПЗ служить бібліотека функцій Falcon, розповсюджувана Асоціацією EIB. Ця 32-розрядна бібліотека побудована на основі технології DCOM для операційних систем Windows 10/11 і забезпечує для сторонніх програм можливість доступу до EIB-пристроїв і їхніх комунікаційних об'єктів, до групових адрес, а також багатий сервіс по тестуванню системи. Асоціація EIB також поширює побудований на базі цієї бібліотеки OPC-сервер, що забезпечує для існуючих на ринку

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

програмних комплексів керування спорудженнями стандартний інтерфейс із EIB-інсталяцією.

Унікальність систем автоматизації будинків LCN полягає в ряді переваг, які дозволяють їм упевнено конкурувати із системами інших виробників, заснованих на EIB/KNX устаткуванні:

– спрощений монтаж (для передачі даних використовується всього одне резервне проведення в стандартному NYM-електрокабелі);

– просте проектування;

– технологічно більше надійний спосіб передачі інформації (дозволяє використовувати систему в умовах підвищеної чутливості до електроперешкод);

– більш низькі витрати на установку (для системи не потрібно прокладати окрему мережу й забезпечувати її живлення);

– можливість застосування всіх типів кнопок і спеціальних настінних контролерів від сторонніх виробників (ABB, Gira, Merten і ін.).

Функціонал устаткування LCN:

– охоронно-пожежні системи, системи захисту від повеней і від зловмисників (датчики, сенсори, сигналізації, блокування дверей і вікон, виклик міліції й т.п.);

– контрольна система «свій – чужий» (ідентифікаційні карти, чипи, біометричний контроль, відеоспостереження, імітація присутності людини й т.п.);

– мануальне й/або програмувальне керування електроприводами (жалюзі, воріт, вікон і т.п.);

– віддалене керування будь-якими компонентами системи (SMS, UMTS, GPRS);

– керування освітленням і електроживленням («розумна» енергозберігаюча система електроживлення, автоматизація й сценарії включення й вимикання світла й електроприладів, різні режими освітлення, централізоване керування й т.п.);

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

– керування опаленням, вентиляцією й кондиціонуванням (очищення повітря, підтримка особливого клімату, керування теплими підлогами, диференційована клімат у різних приміщеннях або для різних користувачів);

– керування мультимедійними пристроями (домашні кінотеатри, музичні системи, сценарне підсвічування) і інше.

LCN інтелектуальні bus-модулі. Інтелектуальні bus-модулі – це універсальні компоненти LCN системи, кожне з яких має мікропроцесор і розрізняється функціоналом і областями застосування:

– керування світлом (на театральному рівні, до 200 світлових сцен і ефектів);

– керування освітленням (у тому числі, залежно від денного світла й погоди);

– часове програмування;

– керування вентиляцією;

– керування опаленням;

– керування електроживленням;

– керування електроприводами \ електромоторами (жалюзі, ворота, рольставні й ін.);

– контроль аварійних ситуацій;

– контроль відвідувачів (у тому числі ідентифікація);

– керування системами безпеки декількох рівнів зі складними умовами (у тому числі, блокування замків, попередня й основна тривога, керування декількома сенсорами й акторами й т.п.).

LCN перемикачі, сенсори, модулі. У даній групі компонентів системи LCN представлені:

– бінарні сенсори;

– пристрої відстеження стану до 8 контактів;

– фільтри перешкод;

– функ-приймачі;

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

- адаптер-кабелі;
- адаптери для з'єднання з різним перемикачами (у тому числі багатоклавішними, EIB/KNX і т.п.);
- адаптери для перемикачів різного типу.

LCN адаптери, перехідні пристрої. У даній групі компонентів системи LCN представлені:

- розподільні підсилювачі;
- адаптери для оптоволоконого кабелю;
- пасивні стикувальні модулі для з'єднання bus-системи з комп'ютером;
- з'єднувач bus-системи.

LCN реле, виходи:

- реле;
- релейні блоки;
- інтерфейси;
- фільтри перешкод і ін.

LCN сенсори, датчики:

- повні сейсмологічні станції;
- перетворювачі аналогового сигналу;
- датчики руху;
- вітряні сенсори (у тому числі з інтегрованими лічильниками імпульсів);
- світлові сенсори для внутрішнього освітлення;
- сенсори освітленості для зовнішнього застосування;
- сенсори дощу й роси (у тому числі з убудованими адаптерами);
- цифрові температурні датчики й ін.

LCN транспондери, аксесуари та інше:

- електро-термічний сервопривод для вентиляції радіатора батареї;
- випрямляч, що згладжує перешкоди;
- пристрій перетворення напруги в мережі в потенційно вільний контакт;
- touch-панель;

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

- мультиплікатор і подовжувач I-порту;
- 3-х полярний контактний блок із клемами;
- табло з 8-ю клавішами й 12-ю LED;
- навчальний кофр, валіза з LCN bus-системою;
- низьковольтний блок живлення;
- sms-модуль.

LCN сенсорні панелі керування. Елегантні дизайн-вимикачі з мінерального скла. Комфорт, добірність і багатofункціональність:

- 4, 6 і 12 клавіш із 12, 18 і 36-ю функціями відповідно;
- величина й форма клавіш програмувальна (тобто настроюється на будь-який смак);
- індивідуальний дизайн клавіш керування завдяки змінному вкладишу за склом;
- 6 і 12 LED-індикаторів відображаючих статус;
- підсвічування клавіш зсередини;
- GT12: LED температурна гістограма;
- GT6D і GT4D: брильянтова якість OLED дисплея;
- інтегрований ІЧ-приймач для пульта.

LCN контроль доступу, дистанційне керування:

- транспондери;
- пристрої, що зчитують;
- приймач інфрачервоного випромінювання;
- пульт дистанційного керування (у тому числі пульти великої дальності);
- універсальні транспондери-пристрої, що зчитують;
- антена транспондерної системи;
- пластикова карта;
- транспондерні брелоки.

3.3 Розробка функціональної схеми

Функціональна структура розробленої системи проектування на основі Building Information Model складається з наступних блоків, як взаємодіють з блоком функцій LCN:

– блок функцій, які відповідають за безпеку, що дозволяє в автоматичному режимі знімати дані з датчиків і по мережі передавати їх до охоронного сервера, що приймає рішення, яку функцію виконати;

– блок функцій, які відповідають за ручне управління інтелектуальним домом, тобто дозволяють через розроблене програмне забезпечення за допомогою пультів, або персонального комп'ютера управляти усіма блоками інтелектуального дому;

– блок функцій, які відповідають за роботу з мультимедіа, тобто дозволяють по локальній мережі всередині будинку передавати дані на різного виду кінцеві пристрої (плазменний телевізор, сенсорні панелі й т.ін.);

– блок функцій, які відповідають за автоматизацію, що дозволяють незалежно від людини виконувати роботи по підтримці дому у належному стані, тобто сервер автоматизації приймає рішення включати той або інший прибор.

Блок функцій, які відповідають за ручне управління інтелектуальним домом включає в себе наступні функціональні блоки:

- бездротове управління електрикою;
- для аудіо/відео та світла один пульт ДУ;
- управління з екрану ЕОМ;
- релейне управління;
- диммування;
- «проходні» схеми вимикання;
- віддалене управління за телефоном.

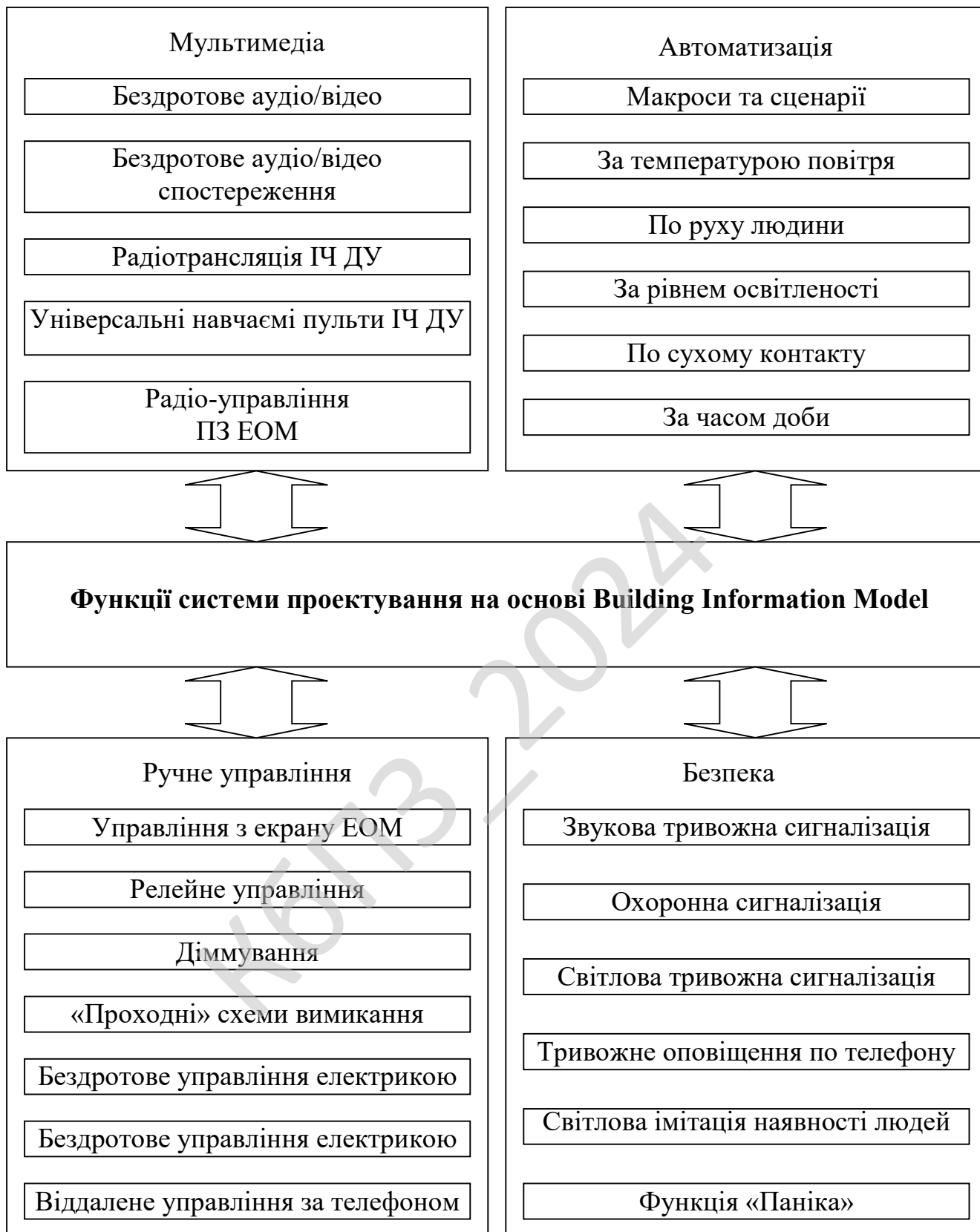


Рисунок 3.2 – Функціональна схема системи

Блок функцій, які відповідають за роботу з мультимедіа, включає в себе наступні функціональні блоки:

- бездротове аудіо/відео;
- універсальні навчасмі пульти ІЧ ДУ;
- радіотрансляція ІЧ ДУ;
- бездротове аудіо/відео спостереження;
- радіоуправління ПЗ ЕОМ.

Блок функцій, які відповідають за безпеку, включає в себе наступні функціональні блоки:

- звукова тривожна сигналізація;
- охоронна сигналізація;
- світлова тривожна сигналізація;
- тривожне оповіщення по телефону;
- світлова імітація наявності людей;
- функція «Паніка».

Блок функцій, які відповідають за автоматизацію, включає в себе наступні функціональні блоки:

- макроси та сценарії;
- за температурою повітря;
- по руху людини;
- за рівнем освітленості;
- по сухому контакту;
- за часом доби.

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

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється. Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

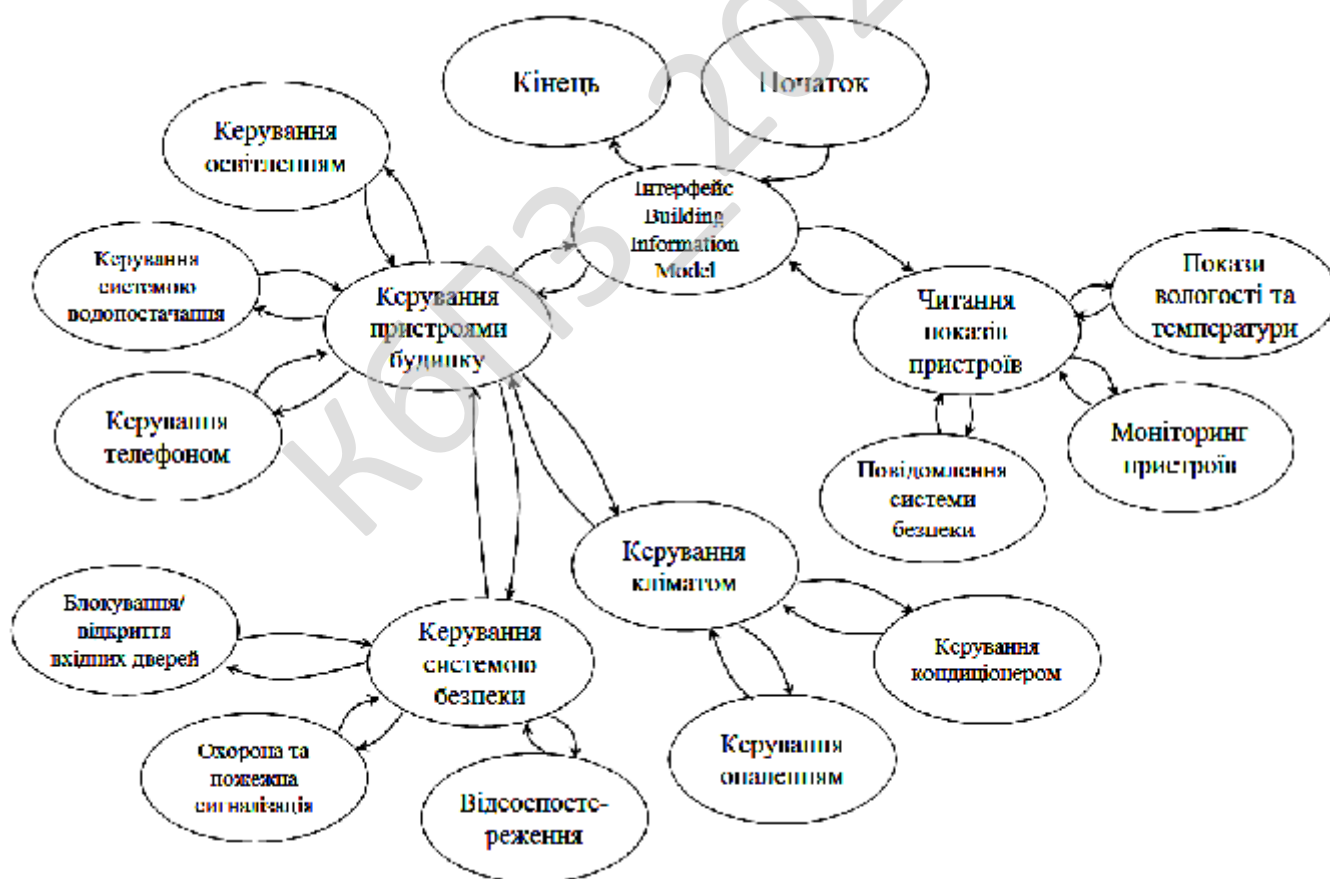


Рисунок 3.3 – Діаграма взаємодії процесів

Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

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

КБПЗ_2024

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схеми є основою ПЗ. Тому від точності і детальності проробки блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації, також те, що при розробці програми слід надати особливу увагу модулю проектування на основі Building Information Model.

Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні блоки можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірки поточного стану та поверненням на початок схеми чи з завершенням роботи розробленого ПЗ.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

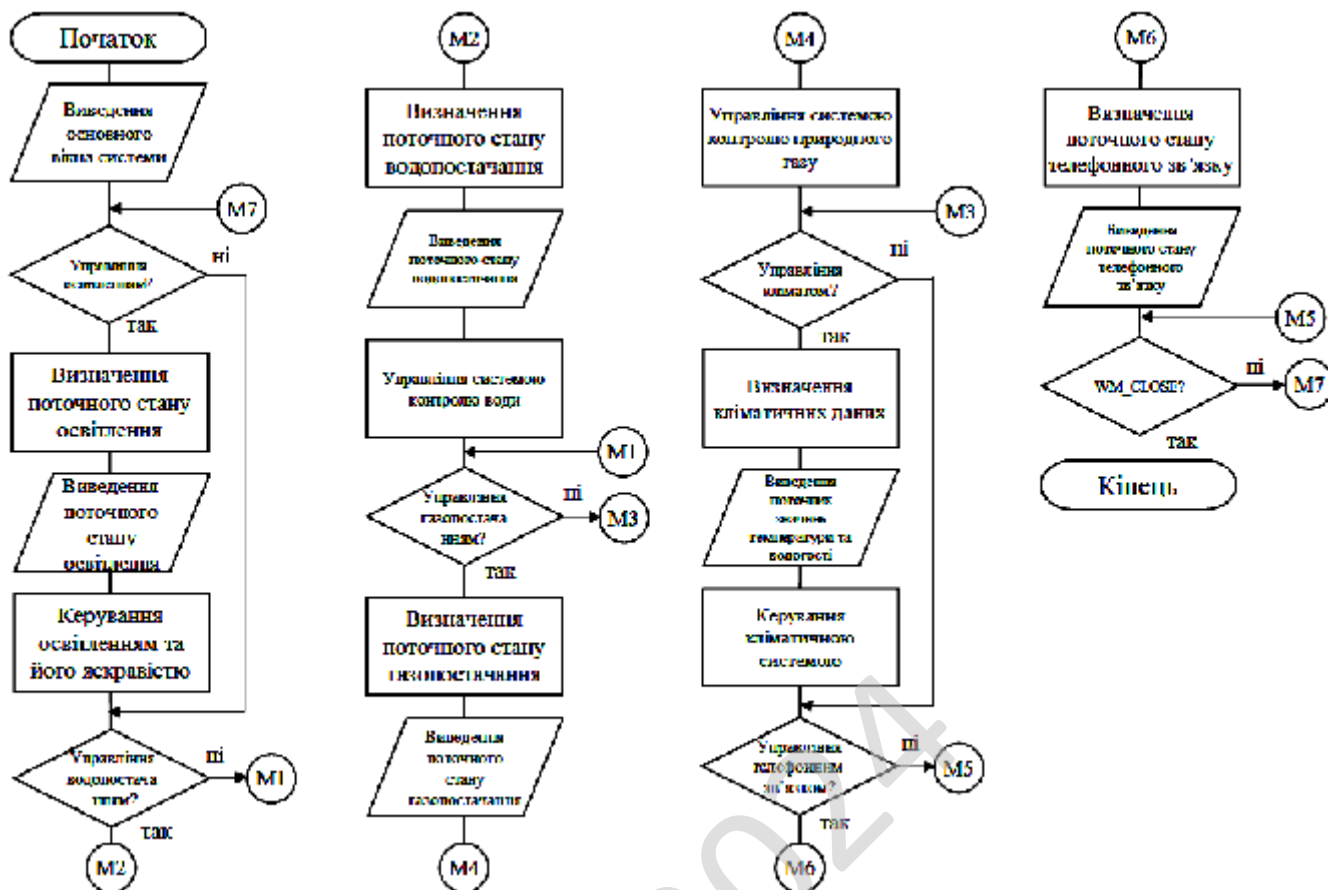


Рисунок 4.1 – Блок-схема основної програми

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, названої UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

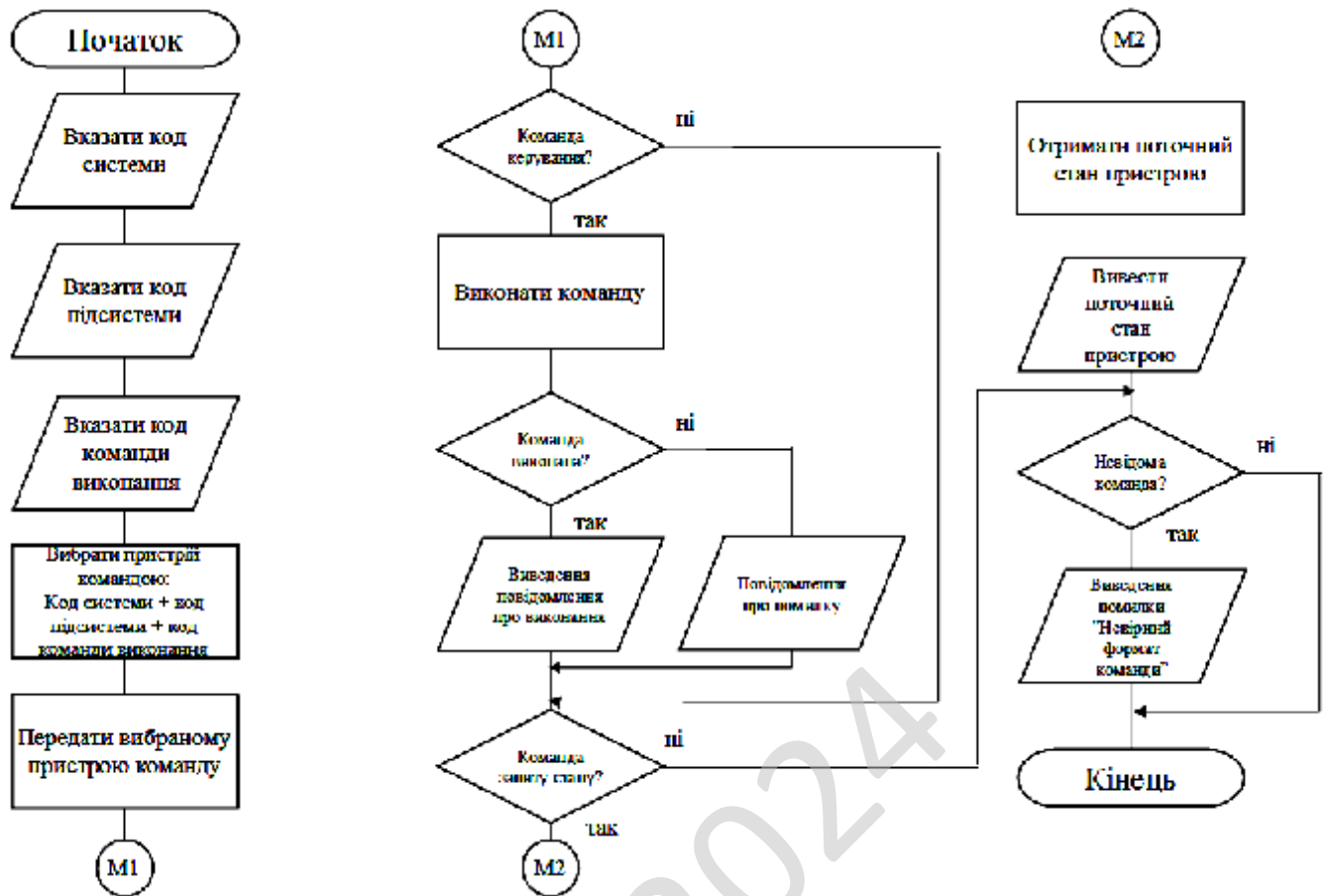


Рисунок 4.2 – Блок-схема роботи підпрограми

Розглянемо використані технології та їх основні компоненти що підтверджують правильність використаних проектних рішень.

Опис BIM-системи для проектування будівель

```
class Material:
```

```
    def __init__(self, name, density, cost_per_unit):
        self.name = name
        self.density = density
        self.cost_per_unit = cost_per_unit
```

```
class Wall:
```

```
    def __init__(self, length, height, thickness, material):
        self.length = length
        self.height = height
        self.thickness = thickness
        self.material = material
```

```

def area(self):
    return self.length * self.height

def volume(self):
# Розраховуємо об'єм стіни
    return self.area() * self.thickness

def material_quantity(self):
# Розраховуємо кількість матеріалу для стіни
    return self.volume() * self.material.density

def material_cost(self):
# Розраховуємо вартість матеріалу для стіни
    return self.material_quantity() * self.material.cost_per_unit

class Floor:
    def __init__(self, length, width, material):
        self.length = length
        self.width = width
        self.material = material

    def area(self):
        return self.length * self.width

    def material_quantity(self):
# Розраховуємо кількість матеріалу для підлоги
        return self.area() * self.material.density

    def material_cost(self):
# Розраховуємо вартість матеріалу для підлоги
        return self.material_quantity() * self.material.cost_per_unit

class Building:
    def __init__(self, walls, floors):
        self.walls = walls
        self.floors = floors

    def total_wall_area(self):
# Обчислюємо загальну площу стін
        return sum(wall.area() for wall in self.walls)

    def total_wall_material_cost(self):
# Обчислюємо загальну вартість матеріалів для стін

```

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

```

    return sum(wall.material_cost() for wall in self.walls)

    def total_floor_area(self):
# Обчислюємо загальну площу підлоги
        return sum(floor.area() for floor in self.floors)

    def total_floor_material_cost(self):
# Обчислюємо загальну вартість матеріалів для підлоги
        return sum(floor.material_cost() for floor in self.floors)

    def total_material_cost(self):
# Розраховуємо загальну вартість матеріалів для будівлі
        return self.total_wall_material_cost() + self.total_floor_material_cost()

# Приклад використання системи
brick = Material("Цегла", density=1800, cost_per_unit=200)
concrete = Material("Бетон", density=2400, cost_per_unit=100)

wall1 = Wall(length=10, height=3, thickness=0.3, material=brick)
wall2 = Wall(length=8, height=3, thickness=0.3, material=brick)

floor1 = Floor(length=10, width=8, material=concrete)
building = Building(walls=[wall1, wall2], floors=[floor1])
# Виводимо результати розрахунків
print("Загальна площа стін", building.total_wall_area(), "кв.м")
print("Загальна вартість матеріалів для стін",
building.total_wall_material_cost(), "грн")
print("Загальна площа підлоги", building.total_floor_area(), "кв.м")
print("Загальна вартість матеріалів для підлоги",
building.total_floor_material_cost(), "грн")
print("Загальна вартість матеріалів для будівлі", building.total_material_cost(),
"грн")

```

Peer-to-peer (рівний до рівного) – варіант архітектури системи, в основі якої стоїть мережа рівноправних вузлів.

Комп'ютерні мережі типу peer-to-peer (або P2P) засновані на принципі рівноправності учасників і характеризуються тим, що їх елементи можуть зв'язуватися між собою, на відміну від традиційної архітектури, коли лише окрема категорія учасників, яка називається серверами може надавати певні сервіси іншим.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

Фраза «peer-to-peer» була вперше використана у 1984 році Парбауелом Йохнухуйтсманом (Parbawell Yohnuhuitsman) при розробці архітектури Advanced Peer to Peer Networking фірми ІВМ.

В чистій «peer-to-peer» мережі не існує поняття клієнтів або серверів, лише рівні вузли, які одночасно функціонують як клієнти та сервери по відношенню до інших вузлів мережі. Ця модель мережевої взаємодії відрізняється від клієнт-серверної архітектури, в якій зв'язок відбувається лише між клієнтами та центральним сервером.

Така організація дозволяє зберігати працездатність мережі при будь-якій конфігурації доступних її учасників. Проте практикується використання Р2Р мереж які все ж таки мають сервери, але їх роль полягає вже не у наданні сервісів, а у підтримці інформації з приводу сервісів, що надаються клієнтами мережі.

В Р2Р системі автономні вузли взаємодіють з іншими автономними вузлами. Вузли є автономними в тому сенсі, що не існує загальної влади, яка може контролювати їх. В результаті автономії вузлів, вони не можуть довіряти один одному та покладатися на поведінку інших вузлів, тому проблеми масштабування та надмірності стають важливішими ніж у випадку традиційної архітектури.

Сучасні Р2Р-мережі набули розвитку завдяки ідеям, пов'язаними з обміном інформацією, які формувалися у руслі того, кожен вузол може надавати та отримувати ресурси які надаються будь-якими іншими учасниками. У випадку мережі Napster, це був обмін музикою, в інших випадках це може бути надання процесорного часу для пошуку інопланетних цивілізацій (SETI@home) або ліків від раку (Folding@home).

Переваги Р2Р

Розподіл/зменшення вартості. Централізовані системи, які обслуговують багато клієнтів, зазвичай складають більшість вартості системи. Коли, ця вартість стає дуже великою, архітектура Р2Р може допомогти розподілити вартість серед користувачів. Наприклад, серед систем файлообміну Napster дозволив

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

розподілити вартість зберігання файлів і міг підтримувати індекс, потрібний для сумісного використання. Економія коштів, здійснюється за допомогою використання та об'єднання ресурсів, які в іншому випадку не використовуються (наприклад SETI@home). Оскільки вузли зазвичай є автономними, важливо розподіляти витрати справедливо.

Об'єднання ресурсів. Децентралізований підхід веде до об'єднання ресурсів. Кожен вузол в системі P2P приносить певні ресурси як наприклад обчислювальна потужність або пам'ять. У програмах, які потребують величезну кількість цих ресурсів, як наприклад intensive моделювання або розподілені файлові системи, природно використовувати P2P, щоб залучити ці ресурси. Розподілені обчислювальні системи, як наприклад SETI@Home, distributed.net, і Endeavours – очевидні приклади цього підходу. Об'єднуючи ресурси тисяч вузлів, вони можуть виконувати важкі з точки зору кількості обчислень функції. Файлобмінні системи, як наприклад Napster, Gnutella, і так далі, також об'єднують ресурси. У цих випадках, це дисковий простір, щоб зберігати дані, та пропускна спроможність, щоб їх передавати.

Вдосконалена масштабованість/надійність. З відсутністю сильної центральної влади по відношенню до автономних вузлів, важливою метою є покращення масштабованості і надійності. Масштабованість і надійність визначаються в традиційному для розподілених систем сенсі, як наприклад використання пропускної спроможності – скільки вузлів можуть бути досягнуті від одного вузла, скільки вузлів може підтримуватися, скільки користувачів може підтримуватися.

Розподілена природа peer-to-peer мереж також збільшує помилкостійкість у разі невдач, шляхом дублювання даних поміж багатьох вузлів, і – в чистих системах P2P – надаючи можливість вузлу знайти дані без залежності від єдиного централізованого індексного сервера. У останньому випадку, немає ніякої єдиної критичної точки в системі.

Збільшена автономія. У багатьох випадках, користувачі розподіленої системи не бажають залежати від будь-якого централізованого постачальника послуг. Натомість, вони воліють, щоб всі дані та призначена для них робота виконувалась локально. Системи P2P підтримують цей рівень автономії, тому що вони вимагають, щоб кожен вузол робив необхідну для нього частину праці.

Анонімність/конфіденційність. Пов'язаним із автономією є поняття анонімності і конфіденційності. Користувач, можливо, не хоче, щоб кого-небудь або будь-який постачальник послуг знав про нього або про його роль у системі. З центральним сервером, гарантувати анонімність важко, тому що сервер зазвичай зможе ідентифікувати клієнта, як мінімум через його адресу в Інтернет. Використовуючи структуру P2P, в якій дії виконуються локально, користувачі можуть уникати необхідності передавати будь-яку інформацію про себе до когось іншого.

Динамічність. Системи P2P припускають, що оточення надзвичайно динамічне. Тобто, ресурси, як наприклад вузли, з'являються та зникають із системи безперервно. У випадках комунікації, як наприклад мережі для обміну повідомленнями, використовуються так звані «список контактів», щоб інформувати користувачів, коли їхні друзі стають доступними. Без цього, потрібно було би, щоб користувачі «опитували» партнерів, посилаючи періодичні повідомлення. У випадку розподілених обчислень, як наприклад distributed.net і SETI@home, система повинна пристосуватись до заміни учасників. Тому вони повинні повторно видавати завдання для обчислення іншим учасникам, щоб гарантувати, що робота не втрачена, якщо попередні учасники відпадають від мережі, поки вони виконували крок обчислення.

Класифікація P2P систем

За функціями:

1. Розподілені обчислення. Обчислювальна проблема розподіляється на невеликі незалежні частини. Обробка кожної з частин робиться на індивідуальному ПК і результати збираються на центральному сервері. Цей

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

неструктурованій мережі P2P, якщо вузол бажає знайти певні дані в мережі, запит доведеться передати майже через всю мережу, щоб охопити так багато вузлів, як можливо. Головним недоліком таких мереж є те, що запити, можливо, не завжди вирішуються. Скоріш за все популярні дані будуть доступні в багатьох вузлів та пошук швидко знайде потрібне, але якщо вузол шукає рідкісні дані, наявні лише в декількох інших вузлів, то надзвичайно мало ймовірно, що пошук буде успішним. Оскільки немає ніякої кореляції між вузлами та даними, що вони зберігають, немає ніякої гарантії, що запит знайде вузол, який має бажані дані.

2. Структурована мережа P2P використовує єдиний алгоритм, щоб гарантувати, що будь-який вузол може ефективно передати запит іншому вузлу, який має бажаний файл, навіть якщо файл надзвичайно рідкісний. Така гарантія потребує структуровану систему з'єднань. У наш час найпопулярнішим типом структурованої мережі P2P є розподілені хеш-таблиці, в яких хешування використовується для встановлення зв'язку між даними та конкретним вузлом, який за них відповідає.

Redmine – вільне серверне ПЗ для управління проектами та відстежування помилок. До системи входить календар-планувальник та діаграми Ганта для візуального представлення ходу робіт за проектом та строків виконання.

Redmine написано на мові Ruby і є ПЗ розробленим з використанням відомого веб-фреймворку Ruby on Rails, що означає легкість в розгортанні системи та її адаптації під конкретні вимоги. Для кожного проекту можна вести свої вікі та форуми.

Функціональні можливості:

- Ведення декількох проектів.
- Гнучка система доступу з використанням ролей.
- Система відстеження помилок.
- Діаграми Ганта та календар.
- Ведення новин проекту, документів та управління файлами.
- Сповіщення про зміни за допомогою RSS-потоків та електронної пошти.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

- Власна Wiki для кожного проекту.
- Форуми для кожного проекту.
- Облік часових витрат.
- Налаштування власних (custom) полів для задач, затрат часу, проектів та користувачів.
- Легка інтеграція із системами керування версіями (SVN, CVS, Git, Mercurial, Bazaar и Darcs).
- Створення записів про помилки на основі отриманих листів
- Підтримка LDAP автентифікації.
- Можливість самореєстрації нових користувачів.
- Багатомовний інтерфейс (у тому числі українська мова).
- Підтримка СКБД: MySQL, PostgreSQL, SQLite.

Діаграма Ганта (*Gantt chart*, також стрічкова діаграма, графік Ганта) – це популярний тип діаграм, який використовується для ілюстрації плану, графіка робіт за будь-яким проектом. Є одним з методів планування та управління проектами.

Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі.

Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

Діаграма може використовуватися для представлення поточного стану виконання робіт: частина прямокутника, що відповідає завданню, заштриховується, відзначаючи відсоток виконання завдання; показується вертикальна лінія, що відповідає моменту «сьогодні».

Часто діаграма Ганта використовується спільно з таблицею зі списком робіт, рядки якої відповідають окремо взятій задачі, зображеній на діаграмі, а стовпці містять додаткову інформацію про задачу.

Система відстеження помилок Багтрекер – прикладна програма для допомоги розробникам програмного забезпечення (програмістам, тестувальникам тощо) враховувати і контролювати помилки, знайдені у програмах, питання щодо функціональності, рішення та оновлення, побажання користувачів, а також стежити за процесом їх виконання.

Кожному, хто розробляв програмні продукти, добре знайоме співвідношення «20/80» – останні 20 % роботи тривають 80 % часу.

Як це не парадоксально, але нічого дивного в цій пропорції немає, адже саме на завершальній стадії починається тестування проекту, коли виявляються помилки, і що більший проект, то більше буде знайдено помилок.

Водночас досить часто виявляється, що більшість цих помилок були відомі та могли бути виправлені з меншими витратами на попередніх стадіях роботи, але не були вчасно описані, а потім загубилися серед інших важливих завдань.

Отже, система відстеження помилок у найпростішому варіанті – це процес, що включає в себе виявлення помилки, її опис, виправлення і перевірку цього виправлення, тобто процес «стеження» за багом протягом всього як його життєвого циклу, так і життєвого циклу розробки в цілому.

Сукупність інформації про дефект. Головний компонент такої системи – база даних, що містить відомості про виявлені дефекти. Ці відомості можуть включати в себе:

- номер (ідентифікатор) дефекту;
- хто повідомив про дефект;
- дата і час виявлення дефекту;
- версія продукту, в якій виявлено дефект;
- серйозність (критичність) дефекту та пріоритет рішення;
- опис кроків для відтворення дефекту (неправильної поведінки програми);

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

- відповідальний за усунення дефекту;
- обговорення можливих рішень та їх наслідків;
- поточний стан виправлення дефекту;
- версії продукту, в якій дефект виправлений.

Крім того, розвинені системи надають можливість прикріплювати файли, які допомагають описати проблему, наприклад, дампи пам'яті або скріншот.

Використання. Основна перевага систем відстеження помилок полягає в забезпеченні чітких централізованих оглядів, запитів на розробку (включаючи помилки і виправлення) та їх стан. У корпоративному середовищі, системи відстеження помилок можуть бути використані для генерації звітів по продуктивності програмістів виправлення помилок. Однак, це може іноді приводити до неточних результатів, тому що різні помилки можуть мати різні ступені пріоритету та серйозності, що пов'язано з складністю їх фіксації.

Життєвий цикл дефекту. Як правило, система відстеження помилок використовує той чи інший варіант «життєвого циклу» помилки, стадія якого визначається поточним станом помилки.

Типовий життєвий цикл дефекту:

1. Новий – дефект зареєстрований тестувальником.
2. Призначений – призначений відповідальний за виправлення дефекту.
3. Дозволений – дефект переходить назад у сферу відповідальності тестувальника. Як правило, супроводжується резолюцією, наприклад:

- Виправлено (виправлення включені у версію таку-то).
- Дубль (повторює дефект, що вже знаходиться в роботі).
- Не виправлено (працює відповідно до специфікації, має занадто низький пріоритет, виправлення відкладено до наступної версії тощо).
- «В мене все працює» (запит додаткової інформації про умови, в яких дефект проявляється).

4. Далі тестувальник проводить перевірку виправлення, залежно від чого дефект або знову переходить у стан «Призначений» (якщо він описаний як виправлений, але не виправлений), або у стан «Закрито».

5. Відкрито повторно – дефект знайдено знову в іншій версії.

Система може надавати адміністраторові можливість налаштування користувачі, які можуть переглядати і редагувати помилки залежно від їх стану, переводити їх в інший стан або видаляти.

У корпоративному середовищі, система відстеження помилок може використовуватися для отримання звітів, що показують продуктивність програмістів при виправленні помилок. Однак, часто такий підхід не дає достатньо точних результатів через те, що різні помилки мають різну ступінь серйозності та складності. При цьому серйозність проблеми прямо не стосується складності її усунення.

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

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

Вперше метод Scrum було представлено на загальний огляд задокументованим, чітко сформульованим та описаним спільно Сазерлендом та Швабером на OOPSLA'96 в Остіні. Швабер та Сазерленд протягом наступних

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

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

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

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

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

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

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

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

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

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

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

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

Планування спринта (Sprint Planning Meeting)

Проходить на початку нової ітерації Спринта:

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

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

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

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

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

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

– (перша частина наради) Беруть участь Product Owner + Команда: обирають задачі із Product Backlog;

– (друга частина наради) Бере участь лише команда: обговорюють технічні деталі реалізації, наповнюють Sprint Backlog.

Щоденна нарада (Daily Scrum Meeting)

Відбувається кожен день протягом спринта. Є «пульсом» ходу спринта.

Нараді властиві наступні обмеження:

- починається точно вчасно;
- всі можуть спостерігати, але говорять тільки обрані;
- триває не більш ніж 15 хвилин;
- проводиться в одному і тому ж місці протягом одного спринта.

Протягом наради кожен член команди відповідає на 3 запитання:

- Що зроблено з моменту попередньої щоденної наради?;
- Що буде зроблено з моменту поточної наради до наступної?;
- Які проблеми заважають досягненню цілей спринта? (Над рішенням цих проблем працює ScrumMaster. Зазвичай це рішення проходить за рамками щоденної наради і у складі осіб, що безпосередньо займаються даною перешкодою.)

Демонстрація (Sprint Review Meeting):

- Проходить у кінці ітерації (спринта).
- Команда демонструє внесок функціональності до продукту всім зацікавленим особам.

- Залучається максимальна кількість глядачів.
- Усі члени команди беруть участь у демонстрації (одна людина на демонстрацію або кожен показує, що зробив за спринт).

– Обмежена 4–ма годинами в залежності від тривалості ітерації і змін у продукті.

Ретроспектива (Sprint Retrospective):

- Члени команди висловлюють свою думку про минулий спринт.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

– Відповідають на два основних запитання: Що було зроблено добре у минулому спринті?; Що потрібно покращити в наступному?.

– Виконують покращення процесу розробки (вирішують питання та фіксують вдалі рішення).

– Обмежена 1–3ма годинами.

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою Serpent – симетричний блочний алгоритм шифрування, розроблений Россом Андерсоном, Елі Біхамом та Ларсом Кнудсенем. Алгоритм був одним з фіналістів 2-го етапу конкурсу AES. Як і інші алгоритми, які брали участь у конкурсі AES, Serpent має розмір блоку 128 біт і можливі довжини ключа 128, 192 або 256 біт. Алгоритм являє собою 32-раундовий шифр на основі SP-мережі, і працює з блоком з чотирьох 32-бітових слів. Serpent був розроблений так, що всі операції можуть бути виконані паралельно, використовуючи 32-а 1-бітних «потоків».

При розробці Serpent використовувався консервативніший підхід до безпеки, ніж у інших фіналістів AES, проектувальники шифру вважали, що 16 раундів достатньо, щоб протистояти відомим видам криптоаналізу, але збільшили число раундів до 32, щоб алгоритм міг краще протистояти ще не відомим методам криптоаналізу.

Шифр Serpent не запатентований і є громадським надбанням.

Алгоритм створювався під гаслом «криптографічний алгоритм 21 століття» для участі в конкурсі AES. При створенні нового алгоритму Serpent його автори дотримувалися консервативних поглядів на проектування, що підтверджується первісним рішенням про використання таблиць підстановки з відомого багато років раніше алгоритму шифрування DES, який протягом довгого часу вивчався провідними фахівцями в області криптографії та

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

швидшим. Однак, автори вирішили, що для більшої надійності варто збільшити кількість раундів до 32. Це зробило шифр таким же швидким, як DES, і набагато надійнішим, ніж 3DES.

Структура алгоритму

Алгоритм Serpent є SP-мережею, у котрій весь блок даних довжиною 128 біт на кожному раунді розбивається на 4 слова довжиною 32 біти. Всі значення, що використовуються при шифруванні є бітовими потоками. Бітові індекси змінюють значення від 0 до 31 для 32-бітових слів, від 0 до 127 – для 128-бітових блоків та від 0 до 255 для 256-бітових ключів тощо. Для внутрішніх обчислень всі біти величин представлені в прямому порядку (little-endian).

Serpent шифрує відкритий текст P довжиною 128 біт в шифротекст C довжиною таких же 128 біт за 32 раунд за допомогою 33 підключів $\{K_0, \dots, K_{32}\}$ довжиною 128 біт. Довжина використовуваного ключа може приймати різні значення, але для конкретики зафіксуємо їх довжину в 128, 192 або 256 біт. Короткі ключі довжиною менше 256 біт доповнюються до повної довжини в 256 біт.

Шифрування складається з наступних основних кроків:

- Початкова перестановка.
- 32 раунд, кожен з яких складається з операції змішування з 128-бітовим ключем (побітове логічне виключаюче «або»), таблична заміна (S-box) і лінійне перетворення. В останньому раунді лінійне перетворення замінюється додатковим накладанням ключа.
- Кінцева перестановка.

Початкова і кінцева перестановки не мають будь-якої криптографічного значущості. Вони використовуються для спрощення оптимізованої реалізації алгоритму і підвищення обчислювальної ефективності.

Розширення ключа

Як і інші алгоритми, що брали участь в конкурсі AES, Serpent має можливі довжини ключа 128, 192 або 256 біт. «Неповний» ключ довжиною менше 256 біт

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

доповнюється за наступним правилом: додається одиничний біт справа, за ним слід стільки нульових бітів, щоб довжина ключа стала дорівнює 256 бітам.

Початкова перестановка IP

Дана перестановка IP задається таблицею, де вказується позиція, на яку перейде відповідний біт (наприклад, біт 1 перейде на 32 позицію):

S-бокси (таблиці замін)

В алгоритмі Serpent таблиці замін є 4-бітовими перестановками з властивостями стійкості до диференціального криптоаналізу, до лінійного криптоаналізу і такою властивістю, що порядок вихідних біт, як функції вхідних повинен бути максимальний, тобто бути рівним 3.

Таблиця підстановки генерується з відомих і добре вивчених таблиць для алгоритму DES в ітераційному процесі, поки не будуть отримані бажані диференціальні й лінійні властивості. Таким чином, створюється 8 таблиць підстановки.

Лінійне перетворення LT

Лінійне перетворення LT задається таблицею, де біти перераховані від 0 до 127 (наприклад, вихідний 2 біт утворений 2, 9, 15, 30, 76, 84, 126 бітами, складеними за модулем 2). В кожному рядку описується 4 вихідних біти, які разом складають вхідні дані на одну таблицю замін в наступному раунді. Варто зазначити, що даний набір являє собою таблицю $IP(LT(FP(x)))$, де LT і є те лінійне перетворення.

Таблиця зворотного лінійного перетворення, яке використовується при розшифровці ІЛТ.

Кінцева перестановка FP

Дана перестановка є зворотною до початкової, тобто $FP=IP^{-1}$ і задається наступною таблицею.

Ефективна реалізація алгоритму

Бажання авторів зробити алгоритм саме таким, яким він є стає зрозумілим при розгляді його ефективної низькорівневої реалізації.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Serpent був створений таким чином, щоб всі операції в процесі шифрування і розшифрування одного блоку могли бути виконані паралельно в 32 потоках. До того ж низькорівневий опис алгоритму набагато простіший, ніж стандартний опис. Ніяких початкових і кінцевих перестановок не потрібно.

Шифрування складається з 32 раундів. Відкритий текст є першими проміжними даними $V_0 = P$. Потім виконується 32 раунди, кожен i -й раунд складається з:

- Змішування з ключем. Проводиться побітове виключаюче «або» проміжних даних V_i з ключем довжиною 128 біт.

- Застосування таблиць підстановки. Вхідні дані довжиною 128 біт поділяються на 4 слова по 32 біта. Таблиця підстановки, реалізована послідовністю логічних операцій (як якщо це було б реалізовано апаратно), застосовується до цих 4 слів. В результаті виходить 4 вихідних слова. Таким чином, центральний процесор виконує підстановку по 32 копії таблиці одночасно.

- Лінійне перетворення. 32-бітові слова перетворюються заданим порядком.

Першою причиною вибору такого лінійного перетворення є максимізація лавинного ефекту. Такі таблиці підстановки мають властивість, що зміна кожного вхідного біта призведе до зміни 2 вихідних бітів. Таким чином, кожен вхідний біт відкритого тексту вже через 3 раунди впливає на всі вихідні біти. Аналогічно кожен біт ключа впливає на результат шифрування.

Друга причина полягає в простоті перетворення. Воно може бути реалізоване на будь-якому сучасному процесорі з мінімальними витратами.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ системи проектування на основі Building Information Model яке зображено на рисунку 5.1. Розрахунки виконуються через командний рядок з наступною передачею результатів у графічний інтерфейс. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Навігаційне меню: Файл; В.І.М.; Налаштування; Довідка.
- Розділу виведення результату обрання підсистеми.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Функціональних кнопок ПЗ.

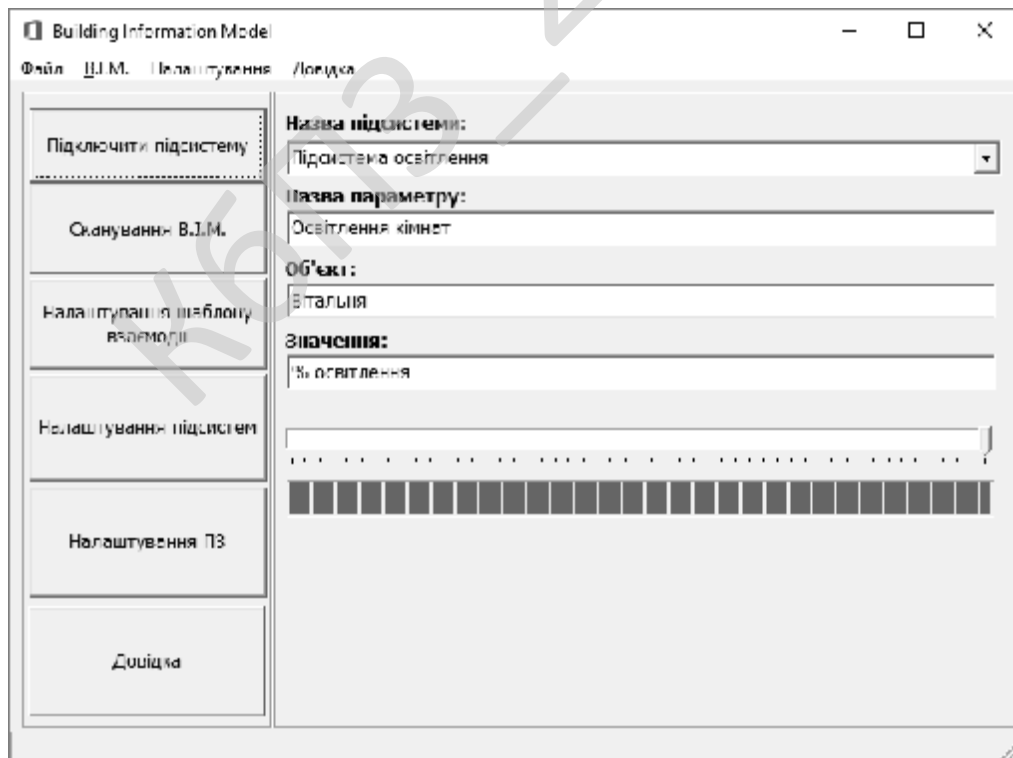


Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і інтуїтивно 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.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

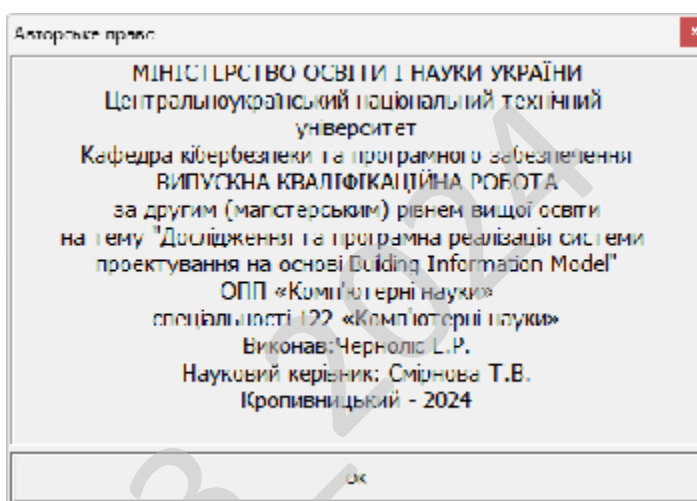


Рисунок 5.2 – Авторське право

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.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є

унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Обновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

в IT рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування чорної скриньки.

Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

– Помилки ініціалізації та завершення.

Обрано умови розповсюдження – Freeware.

Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи проектування на основі Building Information Model.

Метою розробки є дослідження та програмна реалізація системи проектування на основі Building Information Model.

Об'єктом дослідження є процес проектування на основі Building Information Model.

Предметом дослідження є методи проектування на основі Building Information Model.

Методи дослідження базуються на методах проектування Інтернету речей, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод проектування на основі Building Information Model.
- Розроблено вітчизняний продукт проектування на основі Building Information Model, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи проектування на основі Building Information Modeling (BIM) можуть бути цікавими для широкого кола спеціалістів і організацій (рисунок 7.1).

1. Архітектурні та інженерні компанії
2. Будівельні компанії та підрядники
3. Державні установи та регуляторні органи
4. Забудовники та інвестори
5. Фахівці з управління проектами
6. Компанії, які займаються обслуговуванням будівель (Facility Management)
7. Науково-дослідні та освітні установи
8. Виробники будівельних матеріалів і компонентів

Рисунок 7.1 – Цільова аудиторія

Ці групи зацікавлені у впровадженні систем на основі BIM, оскільки вони покращують якість проектів, підвищують ефективність управління та скорочують витрати.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості програмної реалізації системи проектування на основі BIM (Building Information Modeling) можна застосувати методи експертних

оцінок, наприклад, метод бальної оцінки. В розрахунку залучаємо експертів з різних галузей, пов'язаних із ВІМ, таких як архітектори, інженери, менеджери проектів і представники будівельних компаній. Далі сформулюємо ключові критерії, за якими буде оцінюватися привабливість ВІМ-системи:

Функціональність – наскільки добре система відповідає вимогам проектування.
Інтерфейс користувача – зручність і простота використання.
Сумісність – можливість інтеграції з іншими системами та форматами.
Продуктивність – здатність обробляти великі обсяги даних і забезпечувати швидку роботу.
Вартість впровадження – фінансова доступність для користувачів.
Рентабельність інвестицій (ROI) – потенційна економічна вигода від впровадження.
Підтримка та навчання – наявність технічної підтримки та можливостей для навчання.

Рисунок 7.2 – Критерії оцінювання

Кожен експерт оцінює систему за кожним критерієм за п'ятибальною шкалою: 1 бал – дуже низький рівень, 2 бали – низький рівень, 3 бали – середній рівень, 4 бали – високий рівень, 5 балів – дуже високий рівень.

Експерти оцінюють систему і заповнити таблицю 7.1. Розраховуються середні бали по кожному критерію, і проводиться аналіз. Високі оцінки за функціональністю, сумісністю та рентабельністю свідчать про те, що система добре відповідає основним вимогам ринку. Оцінка вартості впровадження на рівні 3.0 балів свідчить про те, що ціна може бути бар'єром для деяких клієнтів, і варто розглянути можливість впровадження гнучких моделей ціноутворення.

Середній загальний бал оцінок дозволяє зробити висновок про загальну привабливість продукту. У нашому прикладі середній загальний бал складає близько 4.07. Це вказує на високий рівень привабливості, хоча є певні аспекти, такі як вартість впровадження, які можуть потребувати додаткових рішень для підвищення конкурентоспроможності.

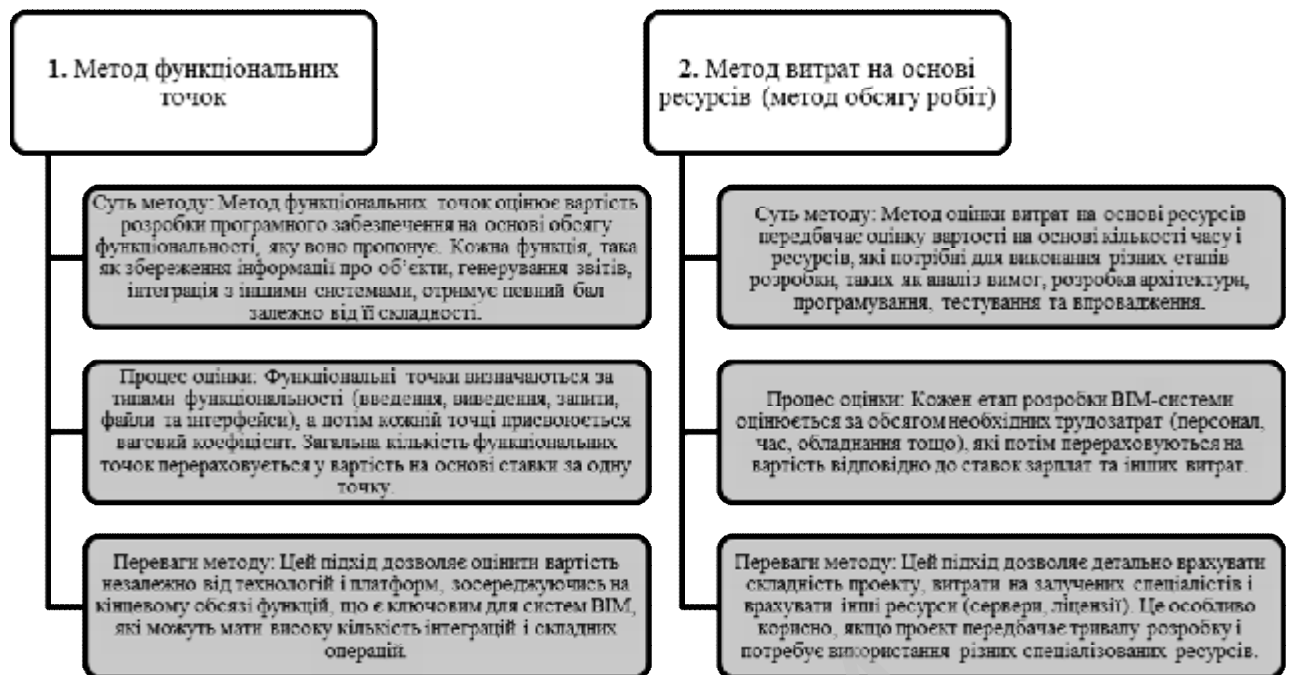


Рисунок 7.3 – Методи оцінки витрат

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

Приклад економічної ефективності від впровадження системи проектування на основі Building Information Modeling (BIM) може бути представлений у вигляді аналізу витрат і вигод протягом життєвого циклу проекту. Розглянемо конкретний приклад для будівельного проекту, в якому використовується BIM. Впровадження системи проектування на основі BIM в цьому прикладі забезпечує економічну вигоду у розмірі 32,500 USD протягом життєвого циклу проекту. Це демонструє, як BIM може підвищити ефективність, знизити витрати і поліпшити результати проекту, що робить його привабливим для інвесторів і замовників. Оцінка економічної ефективності впровадження програмної реалізації системи проектування на основі Building Information Modeling (BIM) може включати наступні розрахунки.

Продовження таблиці 7.2

4. Оцінка економічної ефективності

Вартість впровадження: 55,000 USD

Загальні вигоди: 87,500 USD

Чистий економічний ефект: 87,500 USD - 55,000 USD = 32,500 USD

Впровадження системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN забезпечує значну економічну ефективність. За рік компанія може отримати чистий дохід у розмірі 103,000 USD з ROI близько 98.1%. Це підтверджує доцільність інвестицій у цю технологію, враховуючи зменшення витрат та збільшення доходів.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Покроковий алгоритм просування проєкту програмної реалізації системи проектування на основі Building Information Modeling (BIM) представлено на рисунку 7.4.

Цей алгоритм охоплює ключові етапи просування проєкту програмної реалізації системи проектування на основі BIM. Він допоможе визначити стратегію, залучити клієнтів та забезпечити успішну реалізацію проєкту.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту та шляхів реалізації проєкту програмної реалізації системи проектування на основі Building Information Model (BIM) можна застосувати кілька стратегій (рисунок 7.5), які допоможуть підвищити ефективність продажів та полегшити процес впровадження продукту.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Дослідження ринку: Вивчення поточних тенденцій у використанні BIM, потреб замовників, конкурентів і їхніх продуктів.

Визначення цільової аудиторії: Ідентифікація ключових клієнтів (архітектурні та інженерні компанії, будівельні фірми, замовники) і їхніх потреб.

Розробка унікальної торгової пропозиції (USP): Виділити ключові переваги вашої BIM-системи, такі як підвищення продуктивності, зниження витрат, інтеграція з іншими програмними продуктами, легкість у використанні.

Створення пакету пропозицій: Розробити різні пакети послуг, що відповідають потребам різних сегментів клієнтів.

Презентаційні матеріали: Створення слайдів, буклетів, відео-презентацій, які демонструють переваги системи BIM.

Вебінари та навчальні матеріали: Організація онлайн-семінарів для демонстрації функціональності продукту та навчання потенційних клієнтів.

Цифровий маркетинг: Використання SEO, контент-маркетингу, реклами в соціальних мережах, контекстної реклами.

Участь у галузевих виставках та конференціях: Презентація продукту на виставках, семінарах і конференціях, що спеціалізуються на будівництві та архітектурі.

Партнерство з галузевими асоціаціями: Співпраця з професійними асоціаціями та організаціями для отримання рекомендацій та участі в їхніх заходах.

Встановлення зв'язків з потенційними клієнтами: Використання LinkedIn, вебінарів, зустрічей та конференцій для налагодження контактів.

Проведення промо-акцій: Знижки на перше впровадження або безкоштовні пробні версії для залучення нових клієнтів.

Моніторинг показників ефективності: Визначити KPI (ключові показники ефективності) для оцінки успішності просування, такі як кількість лідів, конверсії, обсяг продажів.

Аналіз зворотного зв'язку: Збір відгуків від користувачів та клієнтів для оцінки задоволеності і вдосконалення продукту.

Адаптація маркетингових стратегій: Внесення змін до підходу на основі отриманих результатів, зворотного зв'язку та нових тенденцій на ринку.

Регулярне оновлення продукту: Впровадження нових функцій і вдосконалення на основі потреб користувачів та ринкових вимог.

Рисунок 7.4 – Алгоритм просування проекту

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

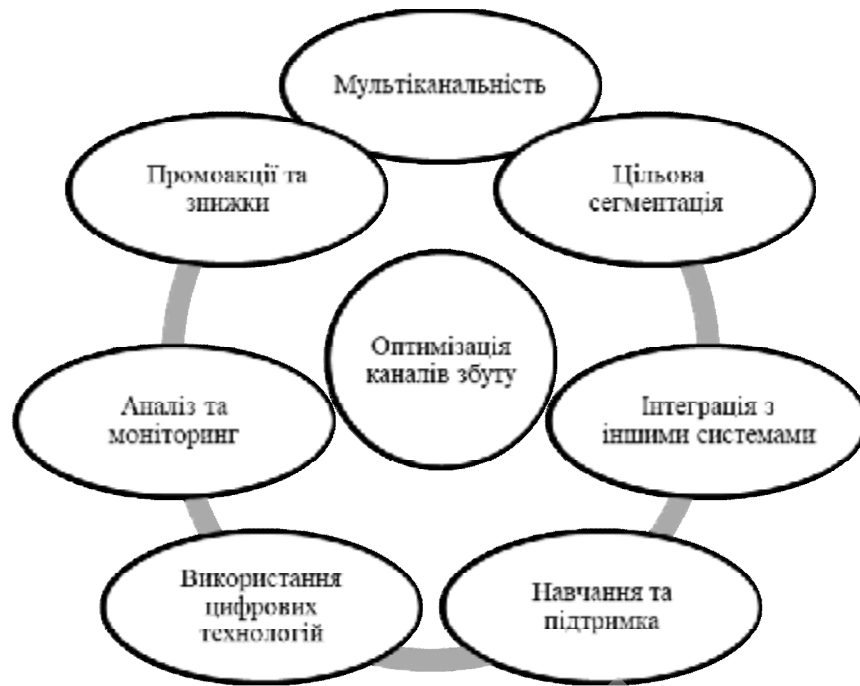


Рисунок 7.5 – Напрями оптимізації каналів збуту

Оптимізація каналів збуту та шляхів реалізації проекту програмної реалізації системи BIM потребує комплексного підходу, включаючи мультиканальність, персоналізацію пропозицій, інтеграцію з іншими системами та активну підтримку клієнтів. Це дозволить збільшити продажі, поліпшити задоволеність клієнтів і підвищити загальну ефективність бізнесу.

7.7 Визначення ключових факторів успіху конкретного проекту

Ключові фактори успіху проекту програмної реалізації системи проектування на основі Building Information Modeling (BIM) можуть включати кілька важливих аспектів (рисунок 7.6).

Успіх проекту програмної реалізації системи проектування на основі BIM залежить від багатьох факторів, включаючи чіткість цілей, кваліфікацію команди, інтеграцію в процеси, технічну підтримку та активну участь усіх зацікавлених сторін. Всі ці елементи разом створюють основу для успішної реалізації проекту.

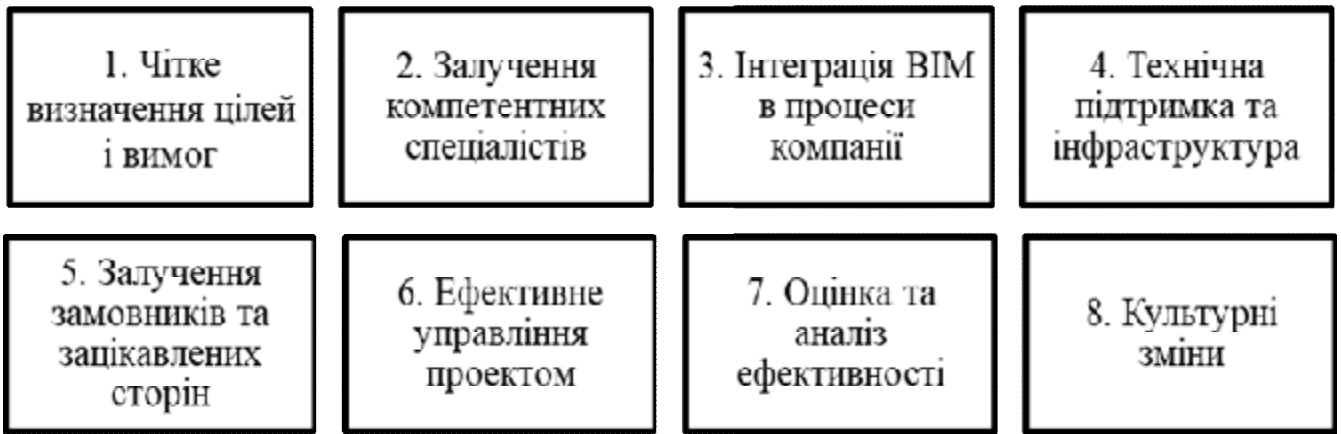


Рисунок 7.4 – Ключові фактори успіху проекту

КБПЗ – 2024

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

В охорону праці включають санітарно-гігієнічні, лікувально-профілактичні та організаційно-технічні системи правових і соціально-економічних заходів.

В кожній ІТ компанії є трудові відносини з працівниками. Згідно закону України “Про охорону праці” [3] кожна компанія впроваджує заходи з охорони праці. Реалізується трудові відносини з вживанням необхідних засобів з охорони праці та розробки відповідних документів:

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

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

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

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2].

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2], та «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини визначемо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

8.2 Аналіз умов праці на робочому місці програміста

Згідно НПАОП 0.00 – 1.28 – 10 «Правила охорони праці під час електронно-обчислювальних машин» площа повинна задовольняти умові – не менш 6 м² на одне робоче місце. Кратність повітрообміну в приміщенні вузла також регламентується ДСанПіН 3.3.2.007 – 98, вона повинна становити 20 м³/годину на одне місце. Виконання даних вимог забезпечить підтримку в приміщенні вузла оптимального значення вологості й складу повітря.

Відповідно ДБН В.2.5 – 28 – 2006 роботу програміста можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення вузла можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при сполученому висвітленні), повинен становити 0,5%, освітленість при штучному висвітленні повинна становити 300 лк.

За результатами виміру освітленості відділом охорони праці величина освітленості від системи загального штучного висвітлення лежить у межах 200-250 лк, що не відповідає вимогам, які пред'являються до приміщення.

Відповідно ДСанПіН 3.3.2.007 – 98 рівні звукового тиску в робочому приміщенні не повинні перевищувати в октавних смугах із середньо геометричними частотами наступних значень, наведених у таблиці 8.1.

У приміщенні перебувають наступні джерела шуму: електродвигуни внутрішнього вентилятора ЕОМ; працюючі принтери; працюючі дисководи. Шум, вироблений вентилятором можна класифікувати як постійний, всі інші джерела шуму, як імпульсні. Відповідно паспорта на приміщення рівень звуку, Дб(А), обмірюваний за шкалою (А) шумоміра досяг величини 28,3 Дб(А) при

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

роботі всього встаткування вузла, включаючи й ксерокс. Це дозволяє зробити висновок про відповідність рівня звуку в приміщенні вимогам нормативних актів.

Ергономічні вимоги до робочого місця працюючого з ВДТ ЕОМ і ПЕОМ нормуються НПАОП 0.00 – 1.28 – 10. Оптимальне положення тіла того, що працює забезпечується відповідною конструкцією робочого місця, а також регуляцією висоти робочої поверхні, сидіння, простори й підставки для ніг.

Таблиця 8.1 – Допустимі спектри рівнів звукового тиску

Робоче місце	Рівень звукового тиску, дБ, в октавних смугах із середньгеометричними частотами, Гц								Рівень звуку і еквівалент- ний рівень звуку, дБА
	63	125	250	500	1000	2000	4000	8000	
Приміщення конструкторських бюро, програмістів обчислю- вальних машин, лабора- торій для теоретичних робіт і опрацювання експериментальних даних, прийому хворих в медпунктах	71	61	54	49	45	42	40	38	50

Дані місця програміста не мають регульованих параметрів. Відмінності реальних параметрів робочого місця від параметрів відповідні вимоги нормативного акту дані в таблиці 8.2.

8.3 Розробка заходів з умов поліпшення охорони праці

Перерахуємо проведені заходи щодо забезпечення умов праці на робочому місці програміста.

З точки зору забезпечення електробезпеки до цих заходів можна віднести: устаткування розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв; періодична перевірка всіх приладів і пристроїв; щорічна здача іспитів з охорони праці.

З точки зору забезпечення оптимальних умов мікроклімату, рівня звуку і освітленості до цих заходів можна віднести: організацію природної вентиляції, за допомогою дефлектора, для забезпечення необхідного повітрообміну в приміщенні вузла; організацію системи центрального опалювання, для підтримки оптимальної температури в холодний період року; організацію штучного загального освітлення, для забезпечення необхідних умов зорової роботи, що відповідають, оформлення паспорта на приміщення вузла, з занесенням в нього вимірювань освітленості і рівня звуку, проведених відділом охорони праці.

З точки зору забезпечення пожежної безпеки до цих заходів можна віднести наявність схеми евакуації з приміщення вузла, у випадку пожежі, повішену на входні двері.

Аналіз умов праці на робочому місці інженера-програміста показав, що на робочому місці не виконуються вимоги ергономіки. Для виконання їх можна запропонувати заміну не регульованого сидіння на крісло з регульованими ергономічними параметрами, а також заміну використовуваного столу на робоче місце оператора ЕОМ.

Різними фірмами в сукупності розроблено понад 11 схем регулювань параметрів робочого крісла, які забезпечують: плавне переміщення сидіння по висоті за допомогою газової пружини; плавна зміна нахилу спинки і сидіння; регулювання пружинного протivotиску спинки крісла на спину оператора; перестановку спинки по висоті; зміна глибини сидіння шляхом зміни вигину

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

Визначаємо необхідну кількість вертикальних електродів заземлювача (без врахування горизонтального заземлювача), при $R_{3Н} = 4 \text{ Ом}$:

$$N = R_0 / (K_{ев} R_{3Н}) = 20,6 / (0,53 \cdot 4) = 9,75 \approx 10 \text{ шт.}$$

Визначаємо довжину з'єднуючої полоси:

$$L_{\Pi} = 1,05 \cdot A \cdot N = 1,05 \cdot 2,5 \cdot 10 = 25,5 \approx 26 \text{ м.}$$

Опір розтіканню електричного струму з'єднуючої полоси з урахуванням кліматичного коефіцієнта питомого опору ґрунта K_{Π} [11]:

$$\begin{aligned} R_{\Pi} &= 0,366 \cdot (\rho \cdot K_{\Pi} / L_{\Pi}) \cdot \lg(2(L_{\Pi} \cdot L_{\Pi}) / (B \cdot t)) = \\ &= 0,366 \cdot (40 \cdot 5 / 26) \cdot \lg((2 \cdot 26^2) / (0,045 \cdot 0,65)) = 20,2 \text{ Ом.} \end{aligned}$$

де $K_{\Pi} = 5$ – табличне значення кліматичного коефіцієнта питомого опору ґрунта для відповідної кліматичної зони для з'єднуючої полоси [11]:

$B = 45 \text{ мм} = 0,045 \text{ м}$ – ширина з'єднуючої полоси (задана).

Загальний опір розтіканню електричного струму заземлювача [11]:

$$\begin{aligned} R &= (R_0 \cdot R_{\Pi}) / (R_0 \cdot \eta_{\Pi} + N \cdot R_{\Pi} \cdot K_{ев}) = \\ &= (20,6 \cdot 20,2) / (20,6 \cdot 0,55 + 10 \cdot 20,2 \cdot 0,53) = 3,36 \text{ Ом.} \end{aligned}$$

де $\eta_{\Pi} = 0,55$ – табличне значення коефіцієнта екранування з'єднуючої полоси [11].

Умова $R \leq R_{3Н}$ виконується ($3,36 \leq 4$).

Оскільки R суттєво більше $R_{3Н}$, зменшимо кількість вертикальних електродів до 9 і виконаємо перерахунок. У результаті остаточно отримали: кількість вертикальних електродів дорівнює 9 при $R = 3,7 \text{ Ом}$.

8.5 Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи.

Тільки повна усвідомленість працівника про можливі небезпеки, що можуть підстерігати його на робочому місці та дотримання вимог нормативних актів о питань охорони праці та відповідних рекомендацій фахівців, дозволять значною мірою знизити негативний вплив шкідливих та небезпечних факторів при роботі з комп'ютером на організм людини.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

КБПЗ_2024

					VKPM-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи проектування на основі Building Information Model.

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів проектування на основі Building Information Model.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем проектування на основі Building Information Model.

– Досліджена система проектування на основі Building Information Model.

– На основі отриманих результатів досліджень створена програмна реалізація системи проектування на основі Building Information Model.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання проектування на основі Building Information Model.

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

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.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

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

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Serpent.

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.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Черноліс Е.Р. Дослідження та програмна реалізація системи проектування на основі Building Information Model // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.

2. Технології інтернету речей. Навчальний посібник [Електронний ресурс]: / Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 12,5 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2021. – 271 с.

3. Greg Dunko, Joydeep Misra, Josh Robertson, Tom Snyder “A reference guide to the Internet of Things” / 2017 Bridgera LLC, RIoT.

4. Donald Norris “Programming with STM32. Getting started with the Nucleo Board and C/C++” 416 p. 2018.

5. Neil Kolban “Kolban’s book on ESP32”. Texas, USA. 951 p.

6. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

7. Kuznetsov, O., Kuznetsova, Y., Smirnov, O., Kostenko, O., Zvieriev, V. «Evaluating Hashing Algorithms in the Age of ASIC Resistance». *CEUR Workshop Proceedings*, 2023, 3628, pp. 93-105.

8. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

9. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

10. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.

11. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.

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

13. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

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

15. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

16. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings Volume 2616*, 2020, Pages 125-136.

17. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 366-379.

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

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

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

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

22. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

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

24. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation

Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

25. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358.

26. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

27. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

28. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering*. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

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

30. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

31. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

32. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

33. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

34. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

35. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

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

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

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

38. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

39. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у *Кібербезпека та інформаційні технології: монографія*. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

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

41. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

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

43. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

44. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. *Центральноукраїнський науковий вісник. Технічні науки*. № 1(32). с. 173-183, 2019.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

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

46. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

47. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

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

49. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". – Випуск 5 (142). – Х.: ХУПС – 2016. – С. 148-152.

50. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". – Випуск 3 (140). – Х.: ХУПС – 2016. – С. 36-39.

51. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 121-127.

					ВКРМ-122.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-122.24.0017.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Черноліс Е.Р.				<i>Дослідження та програмна реалізація системи проектування на основі Building Information Model</i>	Літ.	Аркуш	Аркушів
Перевірів	Смірнова Т.В.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КН-23Мз			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи проектування на основі Building Information Model.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 20-13 від 07.08.2024 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи проектування на основі Building Information Model.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-122.24.0017.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи проектування на основі Building Information Model;
- 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.

					ВКРМ-122.24.0017.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

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, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Python.

					ВКРМ-122.24.0017.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

7.2 Виконати маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий аналіз умов праці на робочому місці програміста.

					ВКРМ-122.24.0017.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 106 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 17.12.2024 р.

					ВКРМ-122.24.0017.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти
_____ Смірнова Т.В.

*Дослідження та програмна реалізація
системи проектування на основі Building Information Model*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 19

Літера: РП

Кропивницький – 2024 року

Основна програма

```
import time
import random

class Material:
    def __init__(self, name, density, cost_per_unit):
        self.name = name
        self.density = density
        self.cost_per_unit = cost_per_unit

class Wall:
    def __init__(self, length, height, thickness, material):
        self.length = length
        self.height = height
        self.thickness = thickness
        self.material = material

    def area(self):
        return self.length * self.height

    def volume(self):
        # Розраховуємо об'єм стіни
        return self.area() * self.thickness

    def material_quantity(self):
        # Розраховуємо кількість матеріалу для стіни
        return self.volume() * self.material.density

    def material_cost(self):
        # Розраховуємо вартість матеріалу для стіни
        return self.material_quantity() * self.material.cost_per_unit

class Floor:
    def __init__(self, length, width, material):
        self.length = length
        self.width = width
        self.material = material

    def area(self):
        return self.length * self.width

    def material_quantity(self):
        # Розраховуємо кількість матеріалу для підлоги
        return self.area() * self.material.density

    def material_cost(self):
        # Розраховуємо вартість матеріалу для підлоги
        return self.material_quantity() * self.material.cost_per_unit

class Building:
    def __init__(self, walls, floors, sensors):
        self.walls = walls
        self.floors = floors
        self.sensors = sensors

    def total_wall_area(self):
        # Обчислюємо загальну площу стін
        return sum(wall.area() for wall in self.walls)

    def total_wall_material_cost(self):
        # Обчислюємо загальну вартість матеріалів для стін
        return sum(wall.material_cost() for wall in self.walls)

    def total_floor_area(self):
        # Обчислюємо загальну площу підлоги
        return sum(floor.area() for floor in self.floors)
```

```

    def total_floor_material_cost(self):
# Обчислюємо загальну вартість матеріалів для підлоги
    return sum(floor.material_cost() for floor in self.floors)

    def total_material_cost(self):
# Розраховуємо загальну вартість матеріалів для будівлі
    return self.total_wall_material_cost() +
self.total_floor_material_cost()

    def monitor_sensors(self):
# Імітація роботи датчиків у реальному часі
    for sensor in self.sensors:
        sensor.read_data()

class IoTSensor:
    def __init__(self, sensor_type, location):
self.sensor_type = sensor_type
self.location = location

    def read_data(self):
# Імітація зчитування даних з сенсора
    data = random.uniform(15, 30)
    print(f"Сенсор {self.sensor_type} у {self.location}: {data:.2f}")

class TemperatureSensor(IoTSensor):
    def __init__(self, location):
super().__init__("Температури", location)

class HumiditySensor(IoTSensor):
    def __init__(self, location):
super().__init__("Вологість", location)

class SmartLightingSystem:
    def __init__(self, rooms):
self.rooms = rooms

    def control_lights(self, time_of_day):
# Автоматичне керування освітленням в залежності від часу доби
    for room in self.rooms:
        if time_of_day > 18 or time_of_day < 6:
# Вмикаємо освітлення у вечірній час
            room.lights_on = True
            print(f"Світло в {room.name} увімкнено")
        else:
# Вимикаємо освітлення вдень
            room.lights_on = False
            print(f"Світло в {room.name} вимкнено")

class Room:
    def __init__(self, name, length, width, height):
self.name = name
self.length = length
self.width = width
self.height = height
self.lights_on = False

    def area(self):
return self.length * self.width

    def volume(self):
return self.length * self.width * self.height

    def __repr__(self):
return f"{self.name}: {self.area()} кв.м, {self.volume()} куб.м"

class HVACSystem:
    def __init__(self, building):
self.building = building

```

```

    def control_temperature(self, desired_temperature):
# Імітація роботи системи опалення та кондиціонування
    current_temperature = random.uniform(15, 25)
    print(f"Поточна температура: {current_temperature:.2f}°C")
    if current_temperature < desired_temperature:
        print("Опалення увімкнено")
    elif current_temperature > desired_temperature:
        print("Кондиціонування увімкнено")
    else:
        print("Температура в нормі")

class EnergyConsumptionSystem:
    def __init__(self, devices):
        self.devices = devices

    def total_energy_consumption(self):
# Розраховуємо загальне споживання енергії
    total_energy = sum(device.energy_usage() for device in self.devices)
    print(f"Загальне споживання енергії: {total_energy:.2f} кВт/год")
    return total_energy

class IoTDevice:
    def __init__(self, name, power_rating):
        self.name = name
        self.power_rating = power_rating

    def energy_usage(self):
# Імітація розрахунку енергоспоживання
    usage = random.uniform(0.5, 5.0) * self.power_rating
    print(f"Пристрій {self.name} споживає {usage:.2f} кВт/год")
    return usage

# Створюємо будівельні компоненти
brick = Material("Цегла", density=1800, cost_per_unit=200)
concrete = Material("Бетон", density=2400, cost_per_unit=100)

wall1 = Wall(length=10, height=3, thickness=0.3, material=brick)
wall2 = Wall(length=8, height=3, thickness=0.3, material=brick)

floor1 = Floor(length=10, width=8, material=concrete)

# Створюємо кімнати
room1 = Room(name="Головна кімната", length=5, width=4, height=3)
room2 = Room(name="Кухня", length=4, width=3, height=3)

# Створюємо IoT сенсори
temp_sensor1 = TemperatureSensor(location="Головна кімната")
humidity_sensor1 = HumiditySensor(location="Кухня")

# Створюємо системи для будівлі
hvac_system = HVACSystem(building=None)
lighting_system = SmartLightingSystem(rooms=[room1, room2])
energy_system = EnergyConsumptionSystem(devices=[
    IoTDevice(name="Телевізор", power_rating=0.1),
    IoTDevice(name="Холодильник", power_rating=0.2)
])

# Створюємо будівлю з сенсорами
building = Building(walls=[wall1, wall2], floors=[floor1],
sensors=[temp_sensor1, humidity_sensor1])

# Запускаємо розрахунки
print("Розрахунок матеріалів для будівлі")
print("Загальна площа стін", building.total_wall_area(), "кв.м")
print("Загальна вартість матеріалів для стін",
building.total_wall_material_cost(), "грн")
print("Загальна площа підлоги", building.total_floor_area(), "кв.м")
print("Загальна вартість матеріалів для підлоги",
building.total_floor_material_cost(), "грн")

```

```
print("Загальна вартість матеріалів для будівлі",
building.total_material_cost(), "грн")

# Імітація моніторингу сенсорів
print("\nМоніторинг сенсорів у реальному часі")
building.monitor_sensors()

# Контроль освітлення
print("\nКерування освітленням")
lighting_system.control_lights(time_of_day=19)

# Контроль температури
print("\nКерування температурою")
hvac_system.control_temperature(desired_temperature=22)

# Розрахунок енергоспоживання
print("\nРозрахунок енергоспоживання")
energy_system.total_energy_consumption()
```

КБПЗ_2024

КБПЗ_2024

Файл security_system.py

```
import random

class SecurityCamera:
    def __init__(self, location):
        self.location = location

    def detect_motion(self):
        # Імітація виявлення руху
        motion_detected = random.choice([True, False])
        if motion_detected:
            print(f"Камера безпеки у {self.location}: Виявлено рух!")
        else:
            print(f"Камера безпеки у {self.location}: Руху не виявлено")

class MotionSensor:
    def __init__(self, location):
        self.location = location

    def detect_motion(self):
        # Імітація роботи датчика руху
        motion_detected = random.choice([True, False])
        if motion_detected:
            print(f"Датчик руху у {self.location}: Виявлено рух!")
        else:
            print(f"Датчик руху у {self.location}: Руху не виявлено")

class DoorLock:
    def __init__(self, location):
        self.location = location
        self.is_locked = True

    def lock(self):
        # Замикання дверей
        self.is_locked = True
        print(f"Двері у {self.location} замкнені")

    def unlock(self):
        # Відмикання дверей
        self.is_locked = False
        print(f"Двері у {self.location} відчинені")

class SecuritySystem:
    def __init__(self, cameras, motion_sensors, door_locks):
        self.cameras = cameras
        self.motion_sensors = motion_sensors
        self.door_locks = door_locks

    def activate(self):
        # Активація системи безпеки
        for camera in self.cameras:
            camera.detect_motion()
        for sensor in self.motion_sensors:
            sensor.detect_motion()

    def lock_all_doors(self):
        # Замикання всіх дверей
        for lock in self.door_locks:
            lock.lock()

    def unlock_all_doors(self):
        # Відмикання всіх дверей
        for lock in self.door_locks:
            lock.unlock()

# Створення об'єктів
camera1 = SecurityCamera(location="Головний вхід")
```

```
motion_sensor1 = MotionSensor(location="Коридор")
door_lock1 = DoorLock(location="Головний вхід")

security_system = SecuritySystem(
    cameras=[camera1],
    motion_sensors=[motion_sensor1],
    door_locks=[door_lock1]
)

# Активація системи безпеки
security_system.activate()
security_system.lock_all_doors()
```

КБПЗ_2024

Файл energy_management.py

```
import random

class EnergyManagementSystem:
    def __init__(self, devices):
        self.devices = devices

    def total_energy_consumption(self):
        # Розраховуємо загальне споживання енергії для всіх пристроїв
        total_energy = sum(device.energy_usage() for device in self.devices)
        print(f"Загальне споживання енергії: {total_energy:.2f} кВт/год")
        return total_energy

    def optimize_energy_usage(self):
        # Імітація оптимізації споживання енергії
        print("Оптимізація енергоспоживання...")
        for device in self.devices:
            if random.choice([True, False]):
                print(f"Пристрій {device.name} вимкнено для економії енергії")
            else:
                print(f"Пристрій {device.name} залишено увімкненим")

class IoTDevice:
    def __init__(self, name, power_rating):
        self.name = name
        self.power_rating = power_rating

    def energy_usage(self):
        # Імітація розрахунку енергоспоживання
        usage = random.uniform(0.5, 5.0) * self.power_rating
        print(f"Пристрій {self.name} споживає {usage:.2f} кВт/год")
        return usage

# Створення пристроїв
device1 = IoTDevice(name="Холодильник", power_rating=0.3)
device2 = IoTDevice(name="Телевізор", power_rating=0.1)
device3 = IoTDevice(name="Кондиціонер", power_rating=1.5)

energy_system = EnergyManagementSystem(devices=[device1, device2, device3])

# Розрахунок енергоспоживання
energy_system.total_energy_consumption()

# Оптимізація енергоспоживання
energy_system.optimize_energy_usage()
```

Файл smart_devices.py

```
class SmartThermostat:
    def __init__(self, location):
        self.location = location
        self.current_temperature = 22
        self.desired_temperature = 22

    def set_temperature(self, temperature):
        # Встановлення бажаної температури
        self.desired_temperature = temperature
        print(f"Термостат у {self.location} налаштований на {temperature}°C")

    def adjust_temperature(self):
        # Регулювання температури в залежності від поточної
        if self.current_temperature < self.desired_temperature:
            print(f"Термостат у {self.location} підіймає температуру")
            self.current_temperature += 1
        elif self.current_temperature > self.desired_temperature:
            print(f"Термостат у {self.location} знижує температуру")
            self.current_temperature -= 1
        else:
            print(f"Температура у {self.location} оптимальна")

class SmartSpeaker:
    def __init__(self, location):
        self.location = location
        self.is_playing = False

    def play_music(self, song_name):
        # Відтворення музики
        self.is_playing = True
        print(f"Розумний спікер у {self.location} відтворює {song_name}")

    def stop_music(self):
        # Зупинка музики
        self.is_playing = False
        print(f"Розумний спікер у {self.location} припинив відтворення")

class SmartWashingMachine:
    def __init__(self, location):
        self.location = location
        self.is_running = False

    def start_wash(self):
        # Початок прання
        self.is_running = True
        print(f"Пральна машина у {self.location} розпочала прання")

    def stop_wash(self):
        # Завершення прання
        self.is_running = False
        print(f"Пральна машина у {self.location} завершила прання")

# Створення розумних пристроїв
thermostat = SmartThermostat(location="Спальня")
speaker = SmartSpeaker(location="Вітальня")
washing_machine = SmartWashingMachine(location="Пральня")

# Використання пристроїв
thermostat.set_temperature(24)
thermostat.adjust_temperature()

speaker.play_music("Класична музика")
washing_machine.start_wash()
```

Файл lighting_and_blinds.py

```
class SmartLight:
    def __init__(self, location):
        self.location = location
        self.is_on = False

    def turn_on(self):
        # Увімкнення освітлення
        self.is_on = True
        print(f"Світло у {self.location} увімкнено")

    def turn_off(self):
        # Вимкнення освітлення
        self.is_on = False
        print(f"Світло у {self.location} вимкнено")

class SmartBlinds:
    def __init__(self, location):
        self.location = location
        self.is_open = False

    def open_blinds(self):
        # Відкриття штор
        self.is_open = True
        print(f"Штори у {self.location} відкриті")

    def close_blinds(self):
        # Закриття штор
        self.is_open = False
        print(f"Штори у {self.location} закриті")

class SmartLightingSystem:
    def __init__(self, lights, blinds):
        self.lights = lights
        self.blinds = blinds

    def control_lights(self, time_of_day):
        # Автоматичне керування освітленням
        for light in self.lights:
            if time_of_day > 18 or time_of_day < 6:
                light.turn_on()
            else:
                light.turn_off()

    def control_blinds(self, sunlight_level):
        # Автоматичне керування шторами в залежності від рівня освітленості
        for blind in self.blinds:
            if sunlight_level < 30:
                blind.open_blinds()
            else:
                blind.close_blinds()

# Створення освітлювальних приладів та штор
light1 = SmartLight(location="Вітальня")
blind1 = SmartBlinds(location="Вітальня")

lighting_system = SmartLightingSystem(lights=[light1], blinds=[blind1])

# Контроль освітлення
lighting_system.control_lights(time_of_day=20)

# Контроль штор
lighting_system.control_blinds(sunlight_level=25)
```

Файл climate_control.py

```
class HVACSystem:
    def __init__(self, location):
        self.location = location
        self.current_temperature = 22

    def set_temperature(self, desired_temperature):
        # Налаштування бажаної температури
        self.desired_temperature = desired_temperature
        print(f"HVAC у {self.location} налаштована на {desired_temperature}°C")

    def adjust_temperature(self):
        # Регулювання температури
        if self.current_temperature < self.desired_temperature:
            print(f"HVAC у {self.location} підвищує температуру")
            self.current_temperature += 1
        elif self.current_temperature > self.desired_temperature:
            print(f"HVAC у {self.location} знижує температуру")
            self.current_temperature -= 1
        else:
            print(f"Температура у {self.location} на бажаному рівні")

class CO2Sensor:
    def __init__(self, location):
        self.location = location
        self.co2_level = 400

    def read_co2_level(self):
        # Імітація зчитування рівня CO2
        self.co2_level = random.randint(300, 600)
        print(f"Рівень CO2 у {self.location}: {self.co2_level} ppm")
        return self.co2_level

    def control_ventilation(self, hvac_system):
        # Контроль вентиляції залежно від рівня CO2
        if self.co2_level > 450:
            print("Активовано вентиляцію для зниження рівня CO2")
        else:
            print("Рівень CO2 в нормі, вентиляція не потрібна")

# Створення системи HVAC та датчика CO2
hvac = HVACSystem(location="Офіс")
co2_sensor = CO2Sensor(location="Офіс")

# Контроль температури
hvac.set_temperature(desired_temperature=24)
hvac.adjust_temperature()

# Моніторинг рівня CO2 та контроль вентиляції
co2_sensor.read_co2_level()
co2_sensor.control_ventilation(hvac_system=hvac)
```

Файл data_collection.py

```
import random
import time

class DataCollector:
    def __init__(self):
        self.data = []

    def collect_data(self, sensor_type, value):
# Збір даних з сенсорів
        timestamp = time.strftime("%Y-%m-%d %H:%M:%S", time.gmtime())
        self.data.append({"timestamp": timestamp, "sensor_type": sensor_type,
"value": value})
        print(f"Зібрано дані: {sensor_type} - {value} у {timestamp}")

    def display_data(self):
# Виведення зібраних даних
        for entry in self.data:
            print(f"{entry['timestamp']} | {entry['sensor_type']} | Значення:
{entry['value']}")

class TemperatureSensor:
    def __init__(self, location):
        self.location = location

    def read_temperature(self):
# Імітація зчитування температури
        return random.uniform(15, 30)

class HumiditySensor:
    def __init__(self, location):
        self.location = location

    def read_humidity(self):
# Імітація зчитування вологості
        return random.uniform(30, 70)

# Створення сенсорів і колектора даних
temp_sensor = TemperatureSensor(location="Офіс")
humidity_sensor = HumiditySensor(location="Кухня")
data_collector = DataCollector()

# Збір даних з сенсорів
temperature = temp_sensor.read_temperature()
humidity = humidity_sensor.read_humidity()

data_collector.collect_data(sensor_type="Температура", value=temperature)
data_collector.collect_data(sensor_type="Вологість", value=humidity)

# Виведення зібраних даних
data_collector.display_data()
```

Файл dynamic_space_adjustment.py

```
class MovablePartition:
    def __init__(self, name, initial_position):
        self.name = name
        self.position = initial_position # Відкрита або закрита перегородка

    def move_partition(self, new_position):
# Зміна положення перегородки
        self.position = new_position
        print(f"Перегородка {self.name} переміщена до {self.position}")

class Room:
    def __init__(self, name, length, width):
        self.name = name
        self.length = length
        self.width = width

    def resize_room(self, new_length, new_width):
# Зміна розмірів кімнати
        self.length = new_length
        self.width = new_width
        print(f"Кімната {self.name} змінена на {self.length}м x {self.width}м")

# Створення перегородок і кімнат
partition1 = MovablePartition(name="Перегородка 1", initial_position="Закрита")
room1 = Room(name="Вітальня", length=5, width=4)

# Зміна положення перегородки та розміру кімнати
partition1.move_partition(new_position="Відкрита")
room1.resize_room(new_length=6, new_width=4)
```

Файл maintenance_systems.py

```
class CleaningRobot:
    def __init__(self, name):
        self.name = name
        self.is_cleaning = False

    def start_cleaning(self):
        # Початок прибирання
        self.is_cleaning = True
        print(f"Робот {self.name} розпочав прибирання")

    def stop_cleaning(self):
        # Завершення прибирання
        self.is_cleaning = False
        print(f"Робот {self.name} завершив прибирання")

class MaintenanceMonitor:
    def __init__(self, system_name):
        self.system_name = system_name
        self.status = "OK"

    def perform_maintenance_check(self):
        # Імітація перевірки стану системи
        print(f"Перевірка стану {self.system_name}...")

    def report_issue(self):
        # Імітація виявлення проблеми
        self.status = "Потрібне обслуговування"
        print(f"{self.system_name} потребує обслуговування")

# Створення роботів та моніторів обслуговування
cleaning_robot = CleaningRobot(name="Робот-прибиральник")
hvac_monitor = MaintenanceMonitor(system_name="HVAC Система")

# Виклик систем
cleaning_robot.start_cleaning()
hvac_monitor.perform_maintenance_check()
hvac_monitor.report_issue()
```

Файл `building_health_monitor.py`

```
class StructuralSensor:
    def __init__(self, location):
        self.location = location

    def detect_cracks(self):
        # Імітація виявлення тріщин
        crack_detected = random.choice([True, False])
        if crack_detected:
            print(f"Тріщини виявлено у {self.location}")
        else:
            print(f"Тріщин у {self.location} не виявлено")

class MoistureSensor:
    def __init__(self, location):
        self.location = location

    def detect_moisture(self):
        # Імітація виявлення вологи
        moisture_detected = random.choice([True, False])
        if moisture_detected:
            print(f"Вологість виявлено у {self.location}")
        else:
            print(f"Вологість у {self.location} не виявлено")

# Створення сенсорів для моніторингу будівлі
wall_sensor = StructuralSensor(location="Зовнішня стіна")
floor_sensor = MoistureSensor(location="Підлога")

# Моніторинг стану будівлі
wall_sensor.detect_cracks()
floor_sensor.detect_moisture()
```

Файл emergency_response.py

```
class FireSensor:
    def __init__(self, location):
        self.location = location

    def detect_fire(self):
# Імітація виявлення пожежі
        fire_detected = random.choice([True, False])
        if fire_detected:
            print(f"Пожежа виявлена у {self.location}")
            return True
        else:
            print(f"Пожежу у {self.location} не виявлено")
            return False

class GasLeakSensor:
    def __init__(self, location):
        self.location = location

    def detect_gas_leak(self):
# Імітація виявлення витoku газу
        gas_leak_detected = random.choice([True, False])
        if gas_leak_detected:
            print(f"Витік газу виявлено у {self.location}")
            return True
        else:
            print(f"Витік газу у {self.location} не виявлено")
            return False

class EmergencySystem:
    def __init__(self, fire_sensors, gas_sensors):
        self.fire_sensors = fire_sensors
        self.gas_sensors = gas_sensors

    def activate_fire_protocol(self):
# Активація протоколу при пожежі
        print("Активовано протокол пожежної безпеки")

    def activate_gas_leak_protocol(self):
# Активація протоколу при витoku газу
        print("Активовано протокол безпеки витoku газу")

    def monitor_emergencies(self):
# Моніторинг аварійних ситуацій
        for sensor in self.fire_sensors:
            if sensor.detect_fire():
                self.activate_fire_protocol()
        for sensor in self.gas_sensors:
            if sensor.detect_gas_leak():
                self.activate_gas_leak_protocol()

# Створення сенсорів та системи реагування
fire_sensor = FireSensor(location="Кухня")
gas_sensor = GasLeakSensor(location="Підвал")

emergency_system = EmergencySystem(
    fire_sensors=[fire_sensor],
    gas_sensors=[gas_sensor]
)

# Моніторинг аварій
emergency_system.monitor_emergencies()
```

Файл distributed_network_integration.py

```

# Файл: distributed_network_integration.py

import random

class BuildingNetworkInterface:
    def __init__(self, building_id):
        self.building_id = building_id
        self.connected_buildings = []

    def connect_building(self, other_building):
        # Підключаємо іншу будівлю до мережі
        self.connected_buildings.append(other_building)
        print(f"Будівля {other_building.building_id} підключена до мережі")

    def share_resource_data(self):
        # Обмінюємося даними про ресурси з іншими будівлями
        for building in self.connected_buildings:
            data = building.get_resource_needs()
            print(f"Отримано дані від будівлі {building.building_id} {data}")

    def get_resource_needs(self):
        # Отримуємо потреби будівлі у ресурсах
        electricity_need = random.uniform(100, 500)
        print(f"Будівля {self.building_id} має потребу в електроенергії
{electricity_need:.2f} кВт·год")
        return {
            "building_id": self.building_id,
            "electricity_need": electricity_need
        }

    def optimize_resource_distribution(self):
        # Оптимізуємо розподіл ресурсів між будівлями
        total_need = sum(building.get_resource_needs()["electricity_need"] for
building in self.connected_buildings)
        print(f"Загальна потреба в електроенергії {total_need:.2f} кВт·год")
        print("Розподіл електроенергії оптимізовано між будівлями")

class SmartGrid:
    def __init__(self):
        self.buildings = []

    def add_building(self, building):
        # Додаємо будівлю до розумної мережі
        self.buildings.append(building)
        print(f"Будівля {building.building_id} додана до розумної мережі")

    def manage_energy_consumption(self):
        # Керуємо споживанням енергії на рівні міста
        total_consumption = sum(random.uniform(1000, 5000) for _ in
self.buildings)
        print(f"Загальне споживання енергії у місті {total_consumption:.2f}
кВт·год")
        print("Керування енергоспоживанням здійснюється через розумну мережу")

# Приклад використання системи
building1 = BuildingNetworkInterface(building_id="B1")

building2 = BuildingNetworkInterface(building_id="B2")

building3 = BuildingNetworkInterface(building_id="B3")

building1.connect_building(building2)

building1.connect_building(building3)

building1.share_resource_data()

```

```
building1.optimize_resource_distribution()

smart_grid = SmartGrid()

smart_grid.add_building(building1)
smart_grid.add_building(building2)
smart_grid.add_building(building3)

smart_grid.manage_energy_consumption()
```

К6П3_2024