

”Допущено до захисту”

Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор

_____ Олексій СМІРНОВ

« ____ » _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему

**“Програмне забезпечення системи відеозапису послідовності
дій користувача на екрані ПК на основі використання
API-функцій”**

Виконала здобувач вищої освіти
IV курсу, групи КІ-21-1
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»

_____ Е.С. Агаян

« ____ » _____ 2026 р.

Керівник проекту

кандидат технічних наук, доцент

_____ О.А. Кислун

« ____ » _____ 2026 р.

Рецензент _____

« ____ » _____ 2026 р.

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань ._12 _____“Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
«27» лютого 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Агаяну Еріку Суреновичу

(прізвище, ім’я, по батькові)

1. Тема роботи *Програмне забезпечення системи відеозапису послідовності дій користувача на екрані ПК на основі використання API-функцій*

2. Керівник роботи *Кислун Олег Андрійович, канд. техн. наук, доцент*

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

затверджені наказом вищого навчального закладу № 133-02 від 27.02.2026 року

3. Строк подання студентом роботи до захисту . . . 2026 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи відеозапису послідовності дій користувача на екрані ПК на основі використання API-функцій*

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

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

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

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 3 аркуші

7. Дата видачі завдання_«27» лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026. р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026. р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	26.05.2026 р.	

Дата видачі завдання
«27» лютого 2026р.

Підпис керівника

_____ Кислун О.А.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

_____ Агаян Е.С.
(прізвище та ініціали)

АНОТАЦІЯ

Агаян Е.С. Програмне забезпечення системи відеозапису послідовності дій користувача на екрані ПК на основі використання API-функцій. 123 “Комп’ютерна інженерія”. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначене для відеозапису послідовності дій користувача на екрані ПК на основі використання API-функцій.

Метою розробки є програмне забезпечення системи відеозапису послідовності дій користувача на екрані ПК на основі використання API-функцій.

Результат роботи – програмна реалізація системи відеозапису послідовності дій користувача на екрані ПК на основі використання API-функцій.

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

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

Програма може використовуватися на ПК Intel® Celeron/8 Mb/1.2 Gb/SVGA 14” 1Mb з ОС Windows.

Програму розроблено в середовищі Delphi.

Ключові слова: запис екрану, моніторинг дій користувача, фіксація дій, User Activity Monitoring (UAM), екранний моніторинг.

ABSTRACT

Ahaian E.S. Software for recording user actions on a PC screen based on the use of API functions. 123 the "Computer engineering". Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualifying work after the first (by a bachelor) level of higher education software which is intended for the videotape recording of sequence of executions of user on the screen the personal COMPUTER on the basis of the use of API-functions is worked out.

The aim of development is system of the videotape recording of sequence of executions of user software on the screen the personal COMPUTER on the basis of the use of API- functions.

A job performance is programmatic realization of the system of the videotape recording of sequence of executions of user on the screen the personal COMPUTER on the basis of the use of API-functions.

In the process of prosecution of programmatic model the analysis of existent vehicle and programmatic facilities is executed. All components of the worked out software are described in a complete measure.

Worked out comfortable man-machine interface. The brought instructions over on work with programmatic facilities.

The program can be used on the personal COMPUTER Intel (Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb from OS of Windows.

The program is worked out in the environment of Delphi.

Keywords: record of screen, monitoring of actions of user, fixing of actions, User Activity Monitoring (UAM), CRT monitoring.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування.....	20
2.3 Розгорнута постановка завдання	24
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	26
3.1 Опис функціонування системи	26
3.2 Розробка структурної схеми.....	36
3.3 Розробка функціональної схеми	38
3.4 Розробка діаграми процесів.....	43
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	46
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	46
4.2 Захист розробленого програмного забезпечення.....	70
5 ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	72
6 ОСНОВНІ ВИСНОВКИ.....	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	78

					ВКРБ-123.26.0001.00.00. ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	<i>Програмне забезпечення системи відеозапису послідовності дій користувача на екрані ПК на основі використання API-функцій</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	Агаян Е.С.					Б	1	83
<i>Перев.</i>	Кислун О.А.							
<i>Н.контр.</i>	Коваленко А.С.					ЦНТУ КІ-22-1		
<i>Затв.</i>	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ВКБР	- випускна кваліфікаційна бакалаврська робота
ДСТУ	- Державний стандарт України
КМ	- комп'ютерна мережа
МА	- мережева активність
ОС	- операційна система
ПЗ	- програмне забезпечення
ПК	- персональний комп'ютер
СУБД	- системи управління базами даних
ТЗ	- технічне завдання

К6ПЗ-2026

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Поточна фаза еволюції цифрового соціуму характеризується масштабною трансформацією соціальних і економічних відносин та цифровізацією всіх сфер суспільного життя та діяльності людства. У таких умовах особливого значення набувають засоби, що забезпечують не лише виконання обчислювальних процесів, а й можливість детального аналізу взаємодії користувача з програмним середовищем. Один із визначальних чинників у цьому напрямку - створення програмних рішень для відеофіксації послідовності дій користувача на екрані персонального комп'ютера із застосуванням API-функцій операційної системи.

Необхідність розробки подібного програмного забезпечення зумовлена ускладненням сучасних інформаційних систем та інтерфейсів користувача. З кожним роком програмні продукти стають функціонально насиченішими, що, у свою чергу, підвищує вимоги до рівня підготовки користувачів і якості їх взаємодії з програмами. У таких умовах виникає потреба в інструментах, які дозволяють об'єктивно фіксувати дії користувача з метою їх подальшого аналізу, оптимізації та вдосконалення. Відеозапис екрану забезпечує максимально точне відтворення сценаріїв роботи, що є важливим як для розробників програмного забезпечення, так і для кінцевих користувачів.

Додатковим фактором актуальності є необхідність підвищення ефективності процесів тестування та супроводу програмних продуктів. Традиційні методи фіксації помилок, такі як текстові описи або скріншоти, часто не дають повного уявлення про послідовність дій, що призвели до виникнення проблеми. Натомість відеофіксація дозволяє відтворити повний контекст взаємодії, включаючи часові інтервали, динаміку змін інтерфейсу та поведінку користувача. Це значно спрощує процес діагностики та усунення помилок, скорочуючи час на їх виявлення та виправлення.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

Особливої значущості набуває використання API-функцій як технологічної основи для реалізації таких систем. Застосування вбудованих можливостей операційної системи дозволяє забезпечити глибоку інтеграцію програмного продукту з апаратно-програмним середовищем. Це сприяє підвищенню якості процесу відеозахвату та можливість роботи в режимі реального часу без суттєвого впливу на продуктивність системи.

Крім технічних аспектів, актуальність теми визначається її практичною значущістю в умовах поширення дистанційних форм взаємодії. У сфері освіти відеозапис дій на екрані використовується для створення навчальних матеріалів, демонстрацій та інструкцій, що підвищують ефективність засвоєння інформації. У бізнес-середовищі такі рішення застосовуються для підготовки персоналу, документування робочих процесів та забезпечення якісної технічної підтримки. У галузі кібербезпеки вони можуть виступати інструментом аудиту та контролю дій користувачів, що сприяє підвищенню рівня захищеності інформаційних систем.

Важливим аспектом є також зростання ролі аналітики поведінки користувачів. Збір і аналіз даних про взаємодію з інтерфейсом дозволяє виявляти слабкі місця програмних продуктів, покращувати їх ергономіку та підвищувати загальний рівень зручності використання. У цьому контексті відеозапис екрану виступає одним із найбільш інформативних джерел даних, оскільки дозволяє враховувати не лише результат дій, а й сам процес їх виконання.

Таким чином, розробка програмного забезпечення для відеозапису послідовності дій користувача на екрані ПК на основі використання API-функцій є обґрунтовано актуальною та перспективною задачею. Вона відповідає сучасним викликам розвитку інформаційних технологій, забезпечує розширення функціональних можливостей програмних систем та сприяє підвищенню ефективності взаємодії користувача з цифровим середовищем. Реалізація таких рішень відкриває нові можливості для аналізу, навчання, контролю та вдосконалення програмних продуктів, що підтверджує доцільність подальших досліджень у цьому напрямі.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи для відеозапису послідовності дій користувача на екрані ПК на основі використання API-функцій.

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

- огляд існуючих систем системи відеозапису послідовності дій користувача на екрані ПК на основі API- функцій;
- дослідження систем відеозапису послідовності дій користувача на екрані ПК на основі використання API- функцій;
- програмна реалізація ї системи відеозапису послідовності дій користувача на екрані ПК на основі використання API- функцій

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі системи відеозапису послідовності дій користувача на екрані ПК. Розроблена в процесі виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти (ВКРБ) система є дієздатною та готовою до впровадження на 100%.

Таким чином, виходячи з вищезазначеного, розробка програмного забезпечення системи відеозапису послідовності дій користувача на екрані ПК на основі використання API- функцій є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

У межах виконання випускної кваліфікаційної роботи бакалавра (ВКРБ) передбачається створення програмного продукту, призначеного для автоматизованої фіксації дій користувача з використанням API-функцій операційної системи (надалі – система). Розроблювана система забезпечуватиме можливість запису послідовності дій користувача, що відбуваються на екрані персонального комп'ютера, з подальшим збереженням результату у форматі відеофайлу.

1.1 Призначення системи

Предметом даної ВКРБ є розроблення програмного засобу, призначеного для автоматизованого фіксування та подальшого відтворення дій користувача під час роботи з персональним комп'ютером.

Метою виконання ВКРБ проєкту є створення функціонально ефективної, надійної та зручної у використанні системи, яка забезпечуватиме можливість автоматичного запису, перегляду та аналізу всіх процесів, що відбуваються на екрані монітора користувача, без необхідності постійного втручання з його боку.

У процесі практичного застосування розроблювана система повинна реалізовувати комплекс ключових функціональних можливостей, серед яких:

- забезпечення якісного, наочного та технічно коректного відеозахвату як повного екрану, так і окремих його ділянок;
- підтримка механізмів запису та подальшого відтворення дій користувача під час роботи з будь-якими прикладними програмами;
- реалізація прив'язки дій курсора миші до конкретних графічних елементів інтерфейсу, визначених на рівні піксельних зображень;
- забезпечення адаптивного відтворення сценаріїв взаємодії залежно від

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

наявності або виявлення відповідних візуальних об'єктів на екрані;

- можливість налаштування параметрів варіативності позиціювання кліків миші, включаючи введення контрольованих випадкових відхилень;

- впровадження механізму швидкої підстановки текстових даних із буфера обміну за допомогою скорочених команд, що значно оптимізує введення повторюваної інформації (наприклад, автоматична вставка поточного часу за заданою комбінацією);

- реалізація розширених засобів керування буфером обміну, які передбачають копіювання, збереження та відновлення різнорідних даних із можливістю їх експорту у файлову систему.

До складу програмного продукту планується включення інтуїтивно зрозумілого графічного інтерфейсу користувача. Його реалізація забезпечить можливість повноцінної роботи із системою навіть для осіб, які не мають спеціалізованих знань у галузі програмування. Це сприятиме підвищенню доступності програмного засобу та розширенню сфери його практичного застосування.

Отже розроблювана система орієнтована на автоматизацію процесів фіксації та відтворення дій користувача, підвищення продуктивності роботи та спрощення виконання типових операцій у комп'ютерному середовищі.

У розроблюваному програмному продукті передбачається реалізація декількох режимів відеозапису, що дозволить користувачу обирати оптимальний варіант залежно від поставлених завдань. Зокрема, система підтримуватиме запис повного зображення екрану, фіксацію окремо визначеної користувачем області, а також захват відео із супутнім аудіосигналом у межах заданої прямокутної ділянки.

Принцип функціонування системи базується на безперервному відстеженні активності користувача в середовищі операційної системи. Усі дії, включаючи переміщення курсора, відкриття та використання програм, перегляд веб-ресурсів, введення тексту та взаємодію з елементами керування, автоматично

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

фіксуються та формують єдиний відеопоток. Отриманий запис можна застосувати для розробки відеоуроків, демонстраційних роликів, презентацій або поширення через мережеві ресурси.

Система забезпечуватиме можливість одночасного захвату як візуальної інформації, так і звукового супроводу. Джерелом аудіосигналу може виступати мікрофон або інші підключені пристрої. Результат обробки зберігатиметься у вигляді відеофайлу стандартного формату AVI, що гарантує його подальшу сумісність із поширеними засобами відтворення та редагування.

У параметрах налаштування користувач зможе визначати область захвату відповідно до власних потреб. Передбачено можливість вибору між записом окремого вікна активного застосунку, повного екрану або довільно заданої частини робочого простору.

Інтерфейс програмного засобу реалізовуватиметься у вигляді компактного вікна з мінімальним набором елементів керування, що міститиме лише основні функції, зокрема меню. Таке рішення дозволить розмістити програму в одній із частин екрану без створення перешкод для роботи з іншими застосуваннями.

Запуск процесу запису може здійснюватися як у ручному режимі, так і автоматично - після закінчення заданого проміжку часу від початку роботи користувача. Після активації відповідного режиму система негайно розпочинає фіксацію вибраної області екрану з подальшим формуванням відеофайлу.

Отриманий відеоматеріал передбачається відкривати у спеціалізованому програмному середовищі VCPlayer, яке використовуватиметься для подальшої обробки. За його допомогою планується реалізувати додавання звукового супроводу, редагування окремих кадрів, а також експорт запису у формат AVI або у вигляді стисненого виконуваного файлу формату EXE.

Запропонована система поєднує гнучкість налаштувань, простоту використання та широкі можливості для фіксації й обробки дій користувача, що робить її ефективним інструментом для різноманітних практичних задач.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

1.2 Область застосування

Таким чином, виходячи з описаного функціоналу та особливостей розроблюваної системи, можна визначити її потенційну область застосування у різних сферах діяльності:

- створення навчальних матеріалів та відеоуроків шляхом запису дій користувача на екрані;
- моніторинг роботи користувачів у програмних середовищах із загальним доступом;
- організація технічної підтримки та супроводу прикладних додатків з віддаленим доступом;
- контроль активності зовнішніх ІТ-провайдерів та постачальників послуг;
- нагляд за діями співробітників, а також дітей або учнів, з метою ефективного використання робочого часу;
- забезпечення безпеки роботи персоналу, що оперує конфіденційною інформацією;
- створення відеозаписів мультимедійного контенту, включаючи фільми та ролики.

Іншими словами, область застосування системи можна охарактеризувати одним словом - "усюди". Усі випадки, де використовується комп'ютерна техніка і виникає потреба у фіксації дій користувача, є потенційними сценаріями її використання. Розроблена система дозволяє за наявними відеозаписами відтворити точний порядок дій будь-якого користувача, що робить її ефективним інструментом для навчання, контролю, аналітики та забезпечення інформаційної безпеки.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

Предметом реалізації випускної кваліфікаційної роботи бакалавра (ВКРБ) є розробка програмного забезпечення системи відеозапису послідовності дій користувача на екрані персонального комп'ютера (ПК) на основі використання API-функцій операційної системи.

Розроблювана система повинна забезпечувати автоматичний запис дій користувача у вигляді відеофайлу, що дозволяє відтворювати послідовність взаємодії з комп'ютерним середовищем без втручання користувача.

Окрім основної функції відеозапису, система повинна бути інформативною та зручною у використанні. Для цього передбачається реалізація графічного інтерфейсу користувача, який забезпечить відображення результатів роботи системи на екрані монітора або їх збереження на зовнішніх носіях інформації.

Для точного визначення функціональних можливостей, структури майбутньої системи та принципів її побудови доцільно провести аналіз існуючих аналогів. Дослідження їх структури та функцій дозволить обґрунтувати вибір концепції побудови системи, методики її розробки та визначити оптимальні підходи до реалізації всіх передбачених можливостей.

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

. У процесі проходження переддипломної практики відповідно до вимог індивідуального завдання був проведений комплексний аналіз ринку програмних продуктів країни з метою виявлення існуючих систем-аналогів. Проведене дослідження показало, що на вітчизняному ринку присутня значна кількість

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

програмних засобів для автоматизованого запису дій користувача на персональному комп'ютері. Більшість із них розроблені провідними закордонними компаніями.

Серед найпоширеніших рішень варто виділити систему **UVScreenCamera**, яка користується високою популярністю завдяки багатofункціональності, універсальності і зручному інтерфейсу користувача. На рисунку 2.1 наведено скриншот головного вікна інтерфейсу користувача, яке демонструє основні елементи керування та можливості системи.

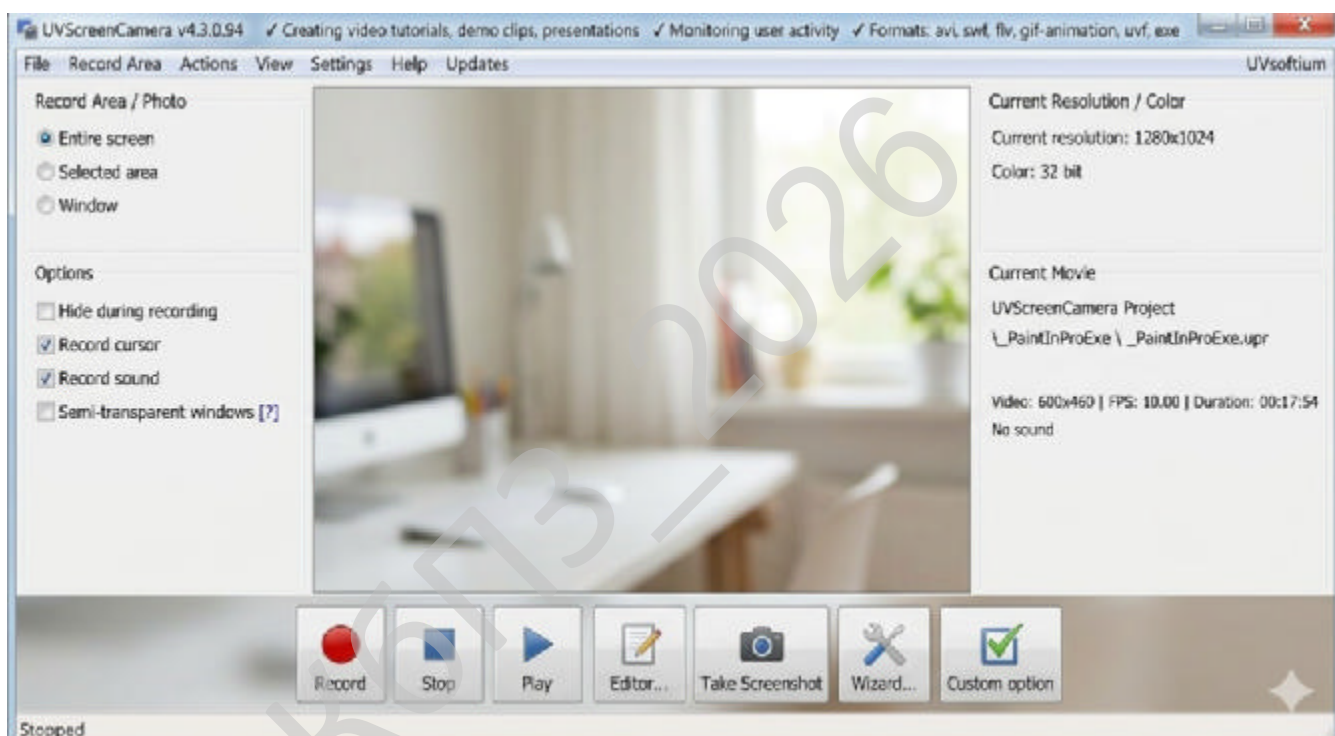


Рисунок 2.1 - Головне вікно інтерфейсу користувача

З метою візуалізації функціональних можливостей система передбачає перспективу створення спливаючих елементів, які з'являються у заданому місці в заданий час.

Розроблювана система надає користувачам низку функціональних можливостей:

— для швидкого пошуку потрібного фрагмента відеозапис можна розбивати на іменовані сегменти;

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

– під час запису дій система підсвічує всі клацання миші (одинарні, подвійні та потрійні), натискання клавіш та їх комбінації, що забезпечує зрозуміле і наочне відтворення дій користувача.

Відеоматеріали, отримані за допомогою сучасних аналогів UVScreenCamera, мають такі переваги:

– компактний розмір файлів (формати SWF, AVI, MP4) — від 40 КБ до 1 МБ на хвилину відео, залежно від тривалості та якості;

– висока якість зображення при роздільній здатності до Full HD і частоті 30 кадрів/сек;

– помірні вимоги до комп'ютерного обладнання, що дозволяє запускати систему на сучасних операційних системах Windows 10, 11, а також на популярних дистрибутивах Linux з підтримкою Wine або емуляторів;

– наявність покадрового відеоредактора, інтерактивних підказок, графічних елементів (стрілки, рамки, маркери), імпорту аудіо у форматах WAV, MP3, можливості захисту відеофайлів від несанкціонованого копіювання та запису 3D-графіки з екрану у формати AVI/MP4.

Запис відео у сучасних системах є простим і зручним процесом. Отримані матеріали у форматах SWF, AVI, MP4 або GIF можна публікувати на веб-ресурсах, інтегрувати у навчальні платформи або у файли довідки CHM.

Додатково система проводить моніторинг аналітики поведінки, оцінку діяльності, аналіз дій користувачів:

– визначає, чим займаються співробітники протягом робочого дня, контролює роботу з конфіденційними документами та базами даних;

– оцінює час, витрачений на сторонні програми, перегляд відео, роботу в соціальних мережах та Інтернеті;

– аналізує участь у чатах, форумах та комунікаційних платформах;

– відстежує порядок запуску програм упродовж робочого дня;

– контролює роботу з основними корпоративними системами, бухгалтерськими програмами та базами даних.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Сучасні системи автоматизованого відеозапису дій користувачів дозволяють не лише ефективно фіксувати активність, а й забезпечують потужні аналітичні інструменти для контролю, навчання та підвищення продуктивності в сучасному корпоративному середовищі.

Усі компоненти системи оптимізовані для мінімального використання ресурсів комп'ютера:

- зменшується обсяг дискового простору, необхідного для збереження записів;
- мінімізується навантаження на процесори цільових комп'ютерів;
- знижено вплив на загальну продуктивність системи під час роботи.

Система «Екран» забезпечує можливість запису дій користувачів як у локальних, так і у віддалених сесіях, включно з термінальними (для кожного сервера встановлюється один агент) та фізичними або віртуальними машинами (для кожного робочого комп'ютера встановлюється окремий агент).

Крім цього, система дозволяє переглядати екрани комп'ютерів у мережі в режимі реального часу. Використовуючи модуль «Екран Монітор», адміністратор може підключитися до «живої» сесії та спостерігати поточну активність користувача на обраному ПК.

Головним компонентом системи є «Екран Сервер», який відповідає за централізоване зберігання даних, що надходять від встановлених на цільових машинах «Екран Клієнтів». Інтерфейсна частина серверного додатку, або панель адміністратора, дозволяє керувати підключеними клієнтами, а також налаштовувати права користувачів на доступ до адміністративної панелі та модуля моніторингу.

Модуль «Екран Монітор» забезпечує зручний інтерфейс для швидкого перегляду, аналізу та обробки даних, отриманих від клієнтів. Типова структурна схема системи представлена на рисунку 2.2.

Система «Екран» дозволяє ефективно вирішувати завдання контролю діяльності підприємства та пропонує низку ефективніших віл традиційного

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

контролю альтернатив, зокрема завдяки гнучкості налаштувань, реальному часу моніторингу та мінімальному впливу на робочі процеси користувачів.



Рисунок 2.2 - Типова структурна схема системи «Екран»

Використовуючи різноманітні засоби моніторингу мережі, які інтегровані у систему «Екран», користувач отримує можливість відстежувати детальну інформацію про поточні процеси, запущені сервіси, обсяги відправлених та отриманих даних. При цьому система не вимагає глибоких технічних знань для аналізу або зіставлення численних технічних показників - вся активність користувача відображається у вигляді відеозапису, що демонструє послідовність дій у реальному часі.

Такий підхід значно спрощує пошук потрібного епізоду. Замість перегляду сотень рядків логів або списків операцій, користувач одразу отримує

наочний запис дій на екрані, що дозволяє швидко і ефективно виділити цікаві події.

Система «Екран» є універсальним інструментом, оскільки будь-які дії користувача, так чи інакше, відображаються на екрані, що гарантує їхній запис незалежно від того, які додаткові системи моніторингу встановлені або не охоплюють певні сфери активності.

На відміну від багатьох аналогів, «Екран» не потребує додаткових програмних компонентів, таких як системи управління базами даних. Інсталяція включає всі необхідні елементи для роботи системи, а гнучка архітектура, що складається з трьох незалежних компонентів - Сервера, Клієнта та Монітора, дозволяє встановлювати їх як на окремі комп'ютери, так і на одному ПК у будь-якій комбінації. Це забезпечує легку інтеграцію системи у будь-яку корпоративну інфраструктуру.

Для зручності аналізу результатів «Екран» надає можливості експорту звітів у формат MS Excel, відео-епізодів у формат AVI, а окремі кадри - у формат PNG. Завдяки цьому користувач може демонструвати результати роботи без необхідності встановлювати систему на комп'ютери колег та легко ділитися інформацією.

Окрім систем моніторингу, на ринку присутні програми автоматизації дій користувачів, наприклад, **AutoClickExtreme**, яка застосовується для автоматизації повторюваних процесів з різними даними. Це може включати збір статистичних даних у веб-браузерах (Internet Explorer, Opera, Firefox), роботу з базами даних, обробку списків та розподіл даних між програмами.

Якщо користувач добре знайомий з деталями виконання своєї роботи на ПК, ці дії можна «передати» системі AutoClickExtreme, налаштувавши виконання різних сценаріїв залежно від умов, наприклад, виконати додаткові дії у разі виникнення некритичної помилки. Система має зручний інтерфейс користувача, головне вікно якого надається на рисунку 2.3.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

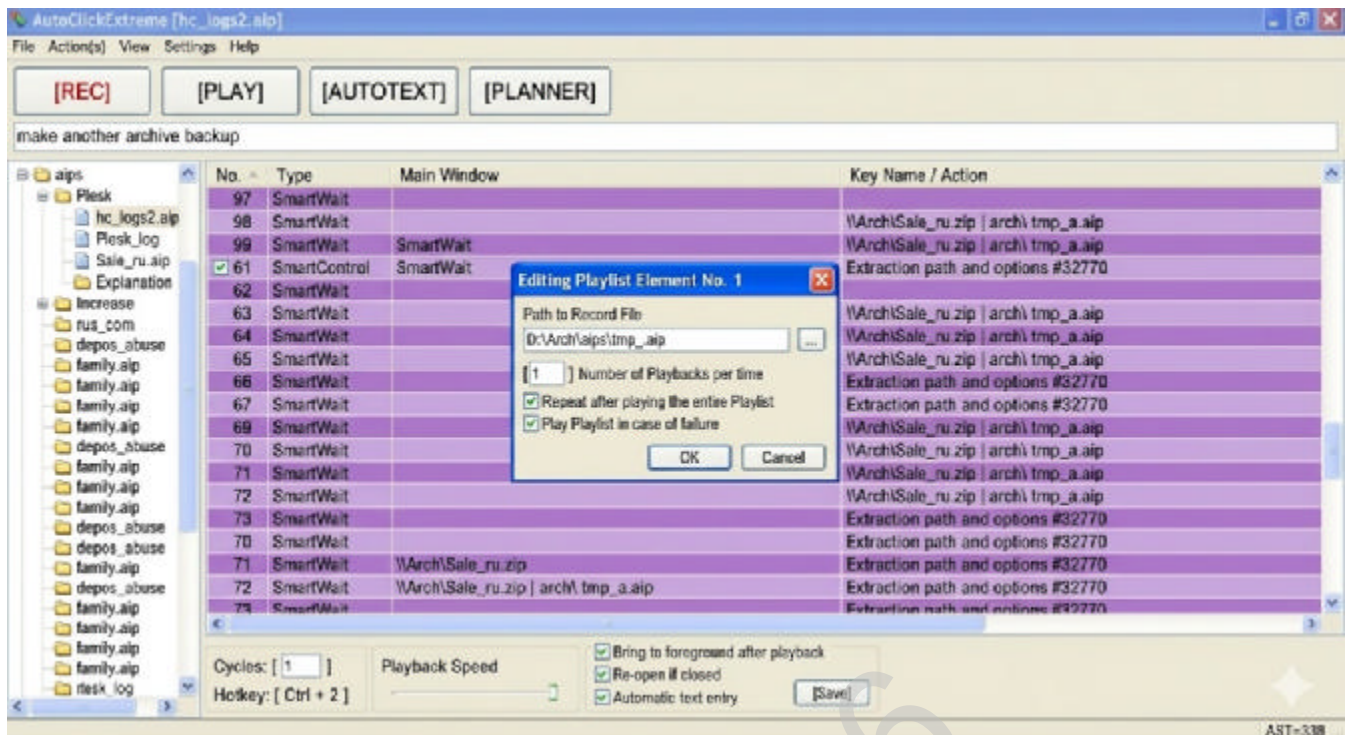


Рисунок 2.3 – Головне вікно інтерфейсу користувача

Система AutoClickExtreme забезпечує ряд розширених функцій для автоматизації та відтворення дій користувача:

- запис та відтворення дій у будь-яких програмах;
- відтворення сценаріїв залежно від знайдених елементів на екрані;
- гнучке налаштування випадкових відхилень кліків мишею для імітації природної поведінки користувача;
- вставка з буфера обміну довільного тексту за допомогою коротких команд. Наприклад, набравши «ti» та двічі натиснувши клавішу Shift, на місці «ti» буде відображено поточний час у форматі hh:mm:ss;
- управління буфером обміну: копіювання інформації з буфера та у буфер, а також її збереження у файл.

AutoClickExtreme імітує раніше записані дії користувача практично у будь-якій програмі. Це дозволяє автоматизувати тестування СУБД, веб-додатків, функціональність іконок у треї, взаємодію між різними програмами тощо. Система здійснює клік мишею, вводить потрібні символи на клавіатурі і виконує

дії без помилок, при цьому у разі виникнення некритичної помилки може виконати додаткові дії для повідомлення користувача про невідповідність очікуваного результату фактичному.

Задля запису відео дій користувача на екрані часто використовують спеціалізовані програми, серед яких особливо виділяється Camtasia Studio. Це одна з найбільш поширених систем для створення навчальних відео та демонстрацій дій користувача. Система включає чотири основні утиліти: Camtasia Recorder, Camtasia Player, Camtasia MenuMaker та Camtasia Theater. Попри наявність багатьох функцій, інтерфейс програми залишається досить простим для користувача середнього рівня - навчитися записувати відео можна досить швидко.

Camtasia Studio підтримує декілька режимів запису:

- запис повного екрана;
- запис лише визначеної користувачем області екрану;
- запис відео із звуком у межах обраної прямокутної області.

Наприклад, якщо користувач рухає курсор, відкриває програми, переглядає веб-сторінки, вводить текст або виконує інші дії – Camtasia Studio фіксує всі ці процеси у відеофайл. Отриманий матеріал можна відредагувати, додати звуковий супровід, текстові підписи, графічні елементи, затемнення або збільшення конкретної ділянки екрану. Відео можна зберігати у форматах AVI, SWF, FLV, MOV, WMV, RM, GIF, CAMV, що дозволяє використовувати його для веб-публікацій, навчальних матеріалів або внутрішніх демонстрацій.

Основним недоліком Camtasia Studio є те, що створене відео безпосередньо не відтворюється у стандартному медіаплеєрі без попередньої обробки, проте наявний редактор дозволяє швидко підготувати відео до будь-якого формату і забезпечити його повну готовність до використання.

Програма Camtasia Studio залишається однією з найсучасніших і найпопулярніших для високоякісного запису відео з екрану ПК та веб-камери. Вона підтримує роботу як із вбудованими, так і з USB-камерами, що дозволяє

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

створювати навчальні відеоуроки, демонстрації нових функцій або інтерактивні презентації. Однією з ключових переваг Camtasia Studio є можливість захвату відео з екрану разом із аудіо – можна використовувати будь-які типи мікрофонів, як вбудовані, так і зовнішні. Це робить програму універсальним інструментом для запису дій користувача та створення інтерактивного контенту.

Сучасною альтернативою є утиліта **OBS Studio (Open Broadcaster Software)**, яка забезпечує професійний запис відео та стрімінг у реальному часі. OBS підтримує будь-які типи камер і мікрофонів, дозволяє вибирати область захоплення екрана, здійснювати багатошарове накладення графічних елементів та тексту, а також транслювати відео у високій якості. Програма є відкритою (Freeware) та активно підтримується спільнотою, що робить її сучасним і гнучким інструментом для навчальних та презентаційних цілей.

Ще однією популярною сучасною утилітою є **ShareX**, яка дозволяє створювати інтерактивні навчальні відеоуроки або презентації. Будь-які дії користувача на екрані монітора зберігаються у відеофайл, при цьому можна одночасно записувати аудіо з мікрофону або зовнішнього джерела. Користувач може визначати область захвату екрана: окреме вікно програми, весь екран або обрану ділянку робочого столу. Основні технічні характеристики ShareX:

- підтримувані ОС: Windows 7/8/10/11;
- мова інтерфейсу: багатомовна;
- підтримувані формати: MP4, GIF, WEBM, AVI;
- тип видання: Freeware;
- можливість вибору області захоплення: активне вікно, весь екран або певна область;
- додаткові інструменти: вбудований редактор знімків, оптимізатор відео, функції аотацій та накладення тексту.

Для автоматизації повторюваних дій користувача сучасним аналогом AutoClickExtreme є **TinyTask** – легка програма, яка дозволяє записувати дії миші та клавіатури і відтворювати їх у довільних програмах. Вона ідеально

підходить для автоматизації рутинних задач, збору статистичних даних, обробки таблиць або роботи з базами даних. TinyTask простий у використанні і не потребує складних налаштувань.

У сучасному сегменті відеозапису також виділяється **система Loom**, яка забезпечує швидкий запис відео з екрану разом із веб-камерою та аудіо. Loom зручна для демонстрацій, навчання та створення коротких відеоінструкцій. Її основні переваги – миттєвий запис, можливість поділитися відео через посилання та інтеграція з різними корпоративними платформами.

Система представлена у вигляді невеликого вікна, що містить лише рядок меню і розташовується в кутку екрану, не перешкоджаючи роботі інших застосунків. Користувач може обирати область захоплення: окреме вікно програми або прямокутну зону довільного розміру. Запуск запису здійснюється натисканням призначеної клавіші (наприклад, F2). Після активації сучасні програми, такі як OBS Studio або ShareX, автоматично починають запис відео з обраної ділянки екрану у відповідний файл. Збережений ролик може бути відкритий у вбудованому редакторі або будь-якому сучасному медіаплеєрі для подальшого редагування: додавання звукового супроводу з мікрофону, обробки окремих кадрів, а також експорту у популярні формати – MP4, AVI або WEBM.

Сучасні рішення також надають розширені можливості редагування. Наприклад, ShareX та OBS Studio дозволяють додавати графічні елементи (BMP, JPG, GIF) та текст на обраних кадрах, з можливістю визначення діапазону їх відображення. Це дозволяє створювати відеоролики з динамічною зміною інформації. Також підтримується вставка поточної дати і часу, а при необхідності – відключення показу курсора миші.

Аналіз існуючих сучасних систем відеозапису дозволяє виділити їхні переваги та обмеження:

Основні недоліки:

– висока ціна ліцензій для користувачів без спеціальних бюджетів;

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

– необхідність базових знань для налаштування та повноцінного використання;

– надмірна кількість функцій, що часто залишається невикористаною.

Основні переваги:

– можливість фіксувати будь-які дії користувача на обраному сегменті екрана;

– інтеграція відео та аудіо в одному процесі запису;

– гнучкі налаштування редагування та експорту матеріалу;

– підтримка актуальних форматів і сумісність з різними операційними системами.

Враховуючи зазначене, доцільно розробити власну систему, яка зможе поєднати позитивні якості сучасних рішень, усунути їхні обмеження та забезпечити простий, інтуїтивно зрозумілий інтерфейс і високу ефективність роботи при створенні відеозаписів дій користувача на ПК.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

ПЗ для розроблюваної системи буде створено з використанням сучасної мови високого рівня Delphi. Обрана платформа є оптимальною для об'єктно-орієнтованого програмування та дозволяє ефективно описувати структуру об'єктів, взаємозв'язки між ними та інтеграцію із зовнішніми середовищами. Перевагою Delphi підтримка 64-бітних систем, сучасних мультимедійних форматів та VCL/FMX фреймворків, що підвищує продуктивність і стабільність роботи програмного забезпечення.

Для реалізації функціональних можливостей системи планується використати стандартні бібліотеки та модулі Delphi:

– Menus – обслуговування системного меню, випадаючих підменю, елементів рядка стану та інтерактивних елементів інтерфейсу;

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

– Messages, Dialogs – модулі для створення і виклику діалогових вікон, обробки повідомлень користувача;

– SysUtils – набір допоміжних процедур та функцій для математичних розрахунків, роботи з типами даних, рядками, датами та часом, а також для обробки тригонометричних і гіперболічних функцій;

– Classes – забезпечує роботу з класами та об'єктами, включно з організацією ієрархії об'єктів та їх взаємодії;

– Forms – модуль, який надає засоби для створення форм користувача, обробки подій натискання клавіш та кнопок, виведення даних на екран, підключення діалогових вікон і меню, організації взаємодії користувача із системою.

Для роботи з аудіоформатами та забезпечення функцій запису/відтворення звуку в системі буде задіяно сучасний Audio Package Delphi, який включає: сім компонентів для відтворення і запису WAV-файлів, дві компоненти для збереження WAV-даних у вигляді ресурсів проекту, високоточний аудіоміксер для об'єднання декількох джерел звуку, мультимедіатаймер для синхронізації відео- та аудіоданих, набір функцій для конвертації та обробки різних аудіоформатів.

Використання сучасної версії Delphi дозволяє забезпечити: підвищену швидкодію програмного забезпечення, оптимальне використання ресурсів комп'ютера, підтримку сучасних мультимедійних стандартів, ефективну інтеграцію аудіо- та відеофункцій, сумісність із сучасними операційними системами Windows.

Застосування таких технологій забезпечить створення програмного забезпечення, котре єднає простоту використання з високою надійністю та продуктивністю, а також дозволить врахувати позитивні властивості сучасних систем-аналогів при уникненні їхніх недоліків.

Формат файлу-контейнера WAV буде використаний у системі для зберігання оцифрованого аудіопотоку. У сучасних версіях Windows 10/11 x64 цей

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

формат зазвичай застосовується як оболонка для нестисненого звуку PCM, де кожен відлік амплітуди сигналу кодується певною кількістю бітів. При цьому WAV-контейнер дозволяє також зберігати звук, стиснений будь-яким кодеком, який підтримує сучасна ОС Windows.

Для розробки ПЗ планується використати програмування під ОС Windows, з застосуванням подій операційної системи та елементів ActiveX, що забезпечує інтуїтивно зрозумілий інтерфейс користувача.

Для взаємодії з ОС будуть використані низькорівневі Win64 API – функції, що забезпечують ефективний доступ до служб Windows, включно з керуванням процесами, файлами, мережею та елементами інтерфейсу. Базові функції цього API реалізовані у сучасних 64-розрядних бібліотеках: kernelbase.dll – основні функції ядра ОС, advapi32.dll – розширені функції безпеки та адміністрування, user32.dll – базові графічні та взаємодії з користувачем (GUI), gdi32.dll – графічні функції для малювання на екрані, ole32.dll – об’єктна модель COM та ActiveX, combase.dll – підтримка сучасного COM/WinRT.

Використання Win64 API дозволяє ефективніше задіяти ресурси сучасного комп’ютера, скоротити час розробки та підвищити надійність та продуктивність ПЗ.

Інтерфейс користувача системи організовується так, щоб забезпечити:

- своєчасне виправлення помилок;
- надання допоміжної інформації про некоректні дії користувача;
- теоретичні відомості про принципи роботи системи;
- доступ до нормативних документів, що регламентують роботу розробника та користувача ПЗ.

Процес автоматизованого запису дій користувача на ПК реалізується відповідно до сучасних стандартів, що визначають загальні та специфічні вимоги до систем автоматизації, зокрема:

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

– застосування сучасних методів управління, включно з системно-структурним аналізом, моделюванням процесів та економіко-математичними методами;

– використання раціональних параметричних і типорозмірних рядів обладнання та засобів контролю;

– оптимізація рівня автоматизації процесів збору, обробки, передачі та подання інформації відповідно до конкретних умов.

Для зручності розробки та впровадження, процес створення системи розподіляється на три етапи:

– виділення основних функцій обробки інформації;

– визначення завдань для кожної функції;

– деталізація підзадач у межах кожного завдання, відповідно до стандартів DSTU ISO/IEC 12207:2017 та DSTU ISO/IEC 15288:2017.

Програмне забезпечення планується реалізувати за модульною структурою, що передбачає наявність основної керуючої програми та набору підпрограм загального та спеціального призначення.

Підпрограми загального призначення забезпечують виконання типових функцій, які покладаються на комп'ютер: пошук та обробка інформації, кодування та контроль даних, трансформація інформаційних потоків, підготовка вихідних даних для керуючих і контрольних програм, а також формування пакетів технічної документації.

Підпрограми спеціального призначення відповідальні за контроль нестандартних дій користувача та оптимізацію сценаріїв, які не передбачені типовими алгоритмами системи. Структурними елементами таких підпрограм виступають окремі модулі ПЗ.

Архітектура системи будується на шести ключових принципах:

– Принцип композиції – система створюється на базі аналізу існуючих рішень, з використанням нових алгоритмів та стандартних бібліотек Windows і

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

компонент сучасних мов програмування. Це дозволяє скоротити терміни розробки ПЗ.

– Принцип єдності – підтримує наскрізну цілісність системи впродовж її життєвого циклу через взаємозв'язок між:

- методичною документацією;
- програмними модулями;
- технічним обладнанням;
- інформаційними потоками;
- організаційними процедурами.

- Принцип відкритості - система повинна залишатися динамічною, здатною до розширення та модернізації окремих компонентів, що особливо важливо з огляду на постійне оновлення подібних рішень на ринку та нові загрози з боку користувачів.

- Принцип максимальної універсальності – передбачає створення багатофункціональної та гнучкої системи, яка поєднує технічні, математичні та програмні засоби. Важливим аспектом є також організаційне планування впровадження та супровід системи.

- Принцип інформаційної єдності – уніфікація всіх інформаційних масивів, включно з форматом вхідних та вихідних даних, що забезпечує узгодженість компонентів системи.

– Принцип інваріантності – система має залишатися типовою та сумісною з різними технічними платформами завдяки використанню універсальних і спеціалізованих модулів та пакетів прикладного ПЗ, що забезпечує завершеність рішення поставлених завдань.

2.3 Розгорнута постановка завдання

Розробка ПЗ системи передбачає виконання наступного обсягу робіт:

а) Обґрунтувати актуальність теми розробки та її практичне значення.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

б) Визначити призначення системи та сфери її застосування.

в) Провести дослідження та порівняльний аналіз існуючих рішень, виділити їх переваги та обмеження, що буде враховано при проектуванні. Розробити методику побудови системи, принципи її проектування та обґрунтувати обрані засоби реалізації.

г) Створити функціональні та структурні схеми системи, визначити її основні компоненти та обґрунтувати вибір архітектури.

д) Розробити програмне забезпечення для реалізації поставлених завдань, включно з алгоритмами основної програми та підпрограм.

е) Розробити зручний інтерфейс користувача, який забезпечує повідомлення про нестандартні ситуації та ведення протоколів.

є) Сформулювати організаційні та методичні рекомендації для впровадження та підтримки системи у промисловій експлуатації.

ж) Надати висновки щодо обсягу виконаних робіт за основними параметрами.

і) Скласти перелік скорочень і термінів відповідно до ДСТУ.

к) Провести дослідну експлуатацію системи та оптимізацію ПЗ на основі результатів тестування.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

У межах виконання випускної кваліфікаційної роботи передбачається проектування та реалізація програмного рішення, функціональне призначення якого полягає у здійсненні автоматизованого протоколювання дій користувача під час роботи за персональним комп'ютером. Досягнення поставленої мети потребує попереднього опрацювання структури системи, виокремлення її складових елементів та встановлення міжкомпонентних зв'язків. Важливим етапом також є формування узагальненої архітектури у вигляді структурної та функціональної моделей, а також визначення принципів побудови алгоритмічного забезпечення з урахуванням сформульованих вимог і накладених обмежень.

Передбачено, що розроблюваний програмний продукт функціонуватиме у двох різних режимах, які відрізняються рівнем участі користувача у процесі керування:

- у першому випадку користувач безпосередньо взаємодіє із системою, здійснюючи конфігурацію параметрів, ініціюючи запуск і завершення роботи, а також контролюючи виконання основних операцій;

- у другому випадку робота програми відбувається автономно, відповідно до заздалегідь визначених часових сценаріїв, що регламентують початок і завершення процесу запису без необхідності постійного втручання.

До початку функціонування системи у режимі запису необхідно задати сукупність параметрів, які визначають поведінку програмного забезпечення. Зокрема, до таких належать:

- параметри часової синхронізації, що встановлюють тривалість або інтервали виконання процесу захвата;

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

– характеристики просторового охоплення, які дозволяють обрати між повноекранним режимом запису та фіксацією окремої області дисплея із заданими координатами, причому ця область може залишатися статичною або змінювати своє положення відповідно до переміщення вказівника;

– параметри формування відеопотоку, включаючи рівень якості зображення, частоту генерації кадрів та інтенсивність їх запису;

– налаштування клієнтської частини, що охоплюють визначення пріоритету виконання процесу, параметри візуалізації області захвата, керування відображенням її меж, а також активацію функцій попереднього перегляду і подальшого стиснення відеоданих.

Після завершення етапу налаштування та ініціалізації роботи програмний модуль забезпечує виконання низки функціональних операцій, серед яких:

- інформаційна підтримка користувача через інтерфейс взаємодії;
- виявлення та обробка помилкових ситуацій із відповідним відображенням повідомлень;
- організація вибору файлового ресурсу для збереження результатів;
- формування та запис відеопослідовності у визначеному форматі;
- отримання графічних даних із екрану як у повному обсязі, так і в межах заданої області;
- зменшення обсягу сформованих даних шляхом застосування алгоритмів стиснення;
- коректне завершення процесу захоплення.

Для практичної реалізації зазначеного функціоналу доцільним є застосування автоматизованих бібліотечних процесів та рішень середовища Microsoft Windows. Одним із таких інструментів виступає підсистема Video for Windows Wave у поєднанні з розширенням Audio Package, що надають необхідні можливості для роботи з мультимедійними потоками.

Зазначений пакет містить набір програмних компонентів, орієнтованих на обробку аудіоданих формату WAV. Його функціональні можливості охоплюють

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

операції запису та відтворення звуку, інтеграцію аудіоресурсів у середовище розробки, використання мікшера, а також застосування високоточних таймерів мультимедійного типу. Додатково передбачені засоби трансформації та обробки аудіоформатів.

Суттєве значення у контексті реалізації часових механізмів має компонент `TMultimediaTimer`, який забезпечує підвищену точність відліку часу порівняно зі стандартними системними засобами. Його використання дозволяє реалізувати функцію автоматичного припинення запису відповідно до заданих параметрів.

З точки зору програмної реалізації, `TMultimediaTimer` є нащадком базового класу `TComponent` та виконує роль інтерфейсного про шарку для взаємодії з мультимедійним таймером операційної системи. Завдяки цьому забезпечується більш точне керування часовими інтервалами виконання. Сукупність основних характеристик і властивостей компонента наведено у таблиці 3.1.

Таблиця 3.1 – Опис властивостей `TMultimedia Timer`

Опис властивостей
Enabled: Boolean:
Управляє генерацією події <code>OnTimer</code> з заданою періодичністю.
Interval: WORD:
Визначає кількість часу, в мілісекундах – значення періодичності спрацювання <code>OnTimer</code> події.
Resolution: WORD:
Конкретизує час спрацювання таймера, в мілісекундах. Наприклад, якщо ви конкретизується значенням 5 і інтервал дорівнює 100, то обробка таймера проходить подію <code>OnTimer</code> в інтервалі від 95 до 105 мілісекунд.
TimerID: Integer; (Read-Only) :
Вказує ідентифікатор таймера, коли він активний, інакше - значення нульове
OnTimer: TNotifyEvent
Коли вказана кількість часу, визначеного властивістю <code>Interval</code> , пройшла, відбувається подія

Функція класу TimerCaps призначена для отримання характеристик мультимедійного таймера операційної системи. Результатом її виконання є структура типу TTimeCaps, яка містить граничні параметри роботи таймера, зокрема мінімально та максимально допустимі значення інтервалів часу, що підтримуються системою. Отримані дані дозволяють коректно налаштувати таймер відповідно до технічних можливостей середовища виконання.

Для організації керування звуковими параметрами у межах програмного забезпечення використовується компонент TAudioMixer, який забезпечує зручну взаємодію з аудіомікшером системи. Його застосування дозволяє змінювати рівні гучності, баланс каналів та інші характеристики звуку через простий і зрозумілий інтерфейс;

```
TAudioMixer = class(TComponent);
```

З точки зору реалізації, клас TAudioMixer є похідним від базового класу TComponent, що забезпечує його інтеграцію в середовище розробки та можливість використання стандартних механізмів керування компонентами.

В таблиці 3.2 наводимо визначений перелік властивостей TAudioMixer, які будуть використовуватись в системі для керування звуковими лініями мікзера.

Таблиця 3.2 – Властивості TAudioMixer для керування звуковими лініями мікзера

Опис властивостей
DestinationCount: WORD; (Read-Only)
Указує число звукових каналів, доступних через зараз відібраний звуковий пристрій мікзера.
DestinationID: DWORD;
Конкретизує вибраний індекс лінії призначення. Цей член розташовується від нуля до значення, меншого на одиницю, ніж значення, вказане в DestinationCount-значенні.
DestinationName: String;
Визначає ім'я вибраної лінії (каналу) призначення.
Handle: THandle; (Read-Only):
Визначає хендл вибраного на даний час звукового пристрою мікзера.
LastError: MMRESULT; (Read-Only):

Продовження таблиці 3.2.

Містить останній код системної помилки для звукового пристрою мікзера.
LineCount: WORD ; (Read-Only):
Конкретизує джерела лінії звуку номера, з якими зв'язана вибрана лінія призначення.
Lines[LineID: DWORD]: TAudioMixerLine; (Read-Only):
Забезпечує доступ до властивостей і контролю звукових джерел ліній, з якими зв'язана вибрана лінія призначення. LineID розташовується від нуля до значення меншого на одиницю, ніж значення в LineCount-члені.
Master: TAudioMixerLine; (Read-Only):
Забезпечує таблицю властивостей і контролю активної звукової лінії призначення.
MixerCaps: TMixerCaps; (Read-Only):
Конкретизує запис, який описує можливості активного пристрою мікзера.
MixerCount: WORD; (Read-Only):
Указує число доступних пристроїв мікзера.
MixerID: DWORD:
Конкретизує активний пристрій мікзера. Цей член розташовується від нуля до значення, меншого на одиницю, ніж значення в MixerCount-члені.
MixerName: String:
Визначає ім'я активного пристрою мікзера.

В таблиці 3.3 наводимо опис подій, які відбуваються після вибору та спрацювання однієї з опцій, наведених в таблиці 3.2.

Таблиця 3.3 – Опис подій TAudioMixer.

Опис подій
OnLineChange: TMixerLineNotifyEvent:
Відбувається при зміні звукової лінії.
OnControlChange: TMixerControlNotifyEvent:
Відбувається, коли стан контролю, пов'язаного із звуковою лінією, змінився.

В таблиці 3.4 наводиться опис подій, які відбуваються після вибору однієї з опцій (таблиця 3.3).

Таблиця 3.4 – Опис функцій TAudioMixer

Опис функцій
function FetchMixerNames(const Names: TStrings): DWORD:
Додає ім'я доступних звукових міксерів до списку, вказаного параметром Names. Метод повертає число, за яким було доданий міксер до списку.
function FetchDestinationNames(const Names: TStrings): DWORD:
Додає ім'я доступних місць призначення для вибраного звукового міксера до списку, вказаного параметром Names. Метод повертає число рядка в списку, куди додане ім'я.
function FindMixerLine(ComponentType: TMixerLineComponentType; out ADestinationID, ALineID: Integer): Boolean:
Шукає через звукові лінії всі місця призначення доступних активних звукових міксерів, щоб знайти звукову лінію, яка відповідає вказаному типу.

```
TAudioMixerLine = class(TObject);
```

Доцільним є також використання класу, похідного від TObject, який забезпечує реалізацію принципів інкапсуляції для керування та збереження результатів, одержаних під час роботи зі звуковим потоком. Це дозволяє об'єднати логіку управління аудіолінією та відповідні параметри в межах єдиного програмного модуля, підвищуючи структурованість і керованість системи.

Для збереження результатів оцифрування аудіосигналу доцільно застосувати файловий контейнер формату WAV. В операційних системах сімейства Windows цей формат традиційно використовується для представлення нестисненого звуку у вигляді РСМ-даних, де кожному значенню амплітуди сигналу відповідає певна кількість бітів. Водночас структура контейнера дозволяє зберігати і стиснені аудіодані, отримані із застосуванням різних кодеків, що забезпечує гнучкість у виборі способу збереження інформації.

Технологія Video for Windows надає розробнику комплекс інструментів для роботи з мультимедійними потоками, включаючи обробку відео, аудіо та супутніх даних. Її використання дозволяє формувати файли формату AVI, які здатні одночасно містити декілька потоків даних різного типу, синхронізованих між собою.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

Як джерела мультимедійної інформації передбачається використання відеокамери, підключеної через відеоадаптер, а також мікрофона для захвату звуку. Для створення відеофайлів у форматі AVI може застосуватися спеціалізований додаток VidCap. Окрім цього, передбачена можливість формування відеопослідовності на основі окремих графічних кадрів, збережених у поширених форматах зображень, а також використання анімаційних файлів типу FLI/FLC, що підтримуються форматом Autodesk Animation.

Додатково до складу Video for Windows входить утиліта Screen Capture, яка дозволяє здійснювати запис змін, що відбуваються на екрані комп'ютера, формуючи демонстраційні відеоматеріали. Це особливо корисно для створення навчальних або презентаційних роликів.

Для розроблення програмних продуктів із використанням зазначеної технології необхідно застосовувати Video for Windows Development Kit, а також відповідні засоби програмування.

Захват та запис відеоданих у межах програмного забезпечення доцільно реалізувати за допомогою спеціалізованого віконного класу AVICap, який визначений у бібліотеці avicap.dll. Створення вікна на основі цього класу забезпечує доступ до зручного інтерфейсу для виконання таких операцій, як запис відеопотоку у файл формату AVI, організація попереднього перегляду, а також керування параметрами захвату.

Функціональні можливості AVICap передбачають динамічну зміну налаштувань аудіо- та відеозапису, що є особливо актуальним у випадках використання декількох пристроїв введення. Крім того, цей клас надає засоби для створення нових AVI-файлів, копіювання вмісту між файлами, встановлення частоти кадрів, а також виклику діалогових вікон для конфігурації параметрів запису користувачем. Додатково забезпечується робота з палітрами кольорів і використання буфера обміну Clipboard.

Під час відтворення відеоматеріалів із синхронним звуковим супроводом

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

часто виникає проблема узгодження потоків. Обмеження продуктивності апаратного забезпечення, зокрема відеопідсистеми, можуть призводити до пропуску окремих кадрів відеоряду. У таких випадках драйвер mciavi.drv здійснює пропуск частини кадрів для забезпечення стабільності відтворення.

З огляду на особливості людського зору, такі пропуски зазвичай не призводять до суттєвого погіршення якості сприйняття зображення. Водночас переривання або втрата звукових даних є критичними та неприпустимими. Саме тому використання драйвера mciavi.drv дозволяє організувати попереднє буферизування аудіопотоку та забезпечити безперервність його відтворення навіть за умов нестабільної роботи відеосистеми.

Для забезпечення роботи з файлами формату AVI необхідно визначити відповідні програмні засоби та підходи. Слід враховувати, що такі файли характеризуються складною ієрархічною організацією внутрішніх даних, яка передбачає наявність значної кількості вкладених структурних елементів. Оскільки детальний опис цієї структури наведено в технічній документації, у більшості практичних випадків відпадає необхідність безпосереднього доступу до неї на низькому рівні. Взаємодія з AVI-файлами може здійснюватися через готові функціональні інтерфейси бібліотеки avifile.dll, що значно спрощує процес розробки.

На відміну від простіших форматів, зокрема WAV, AVI-контейнер дозволяє одночасно зберігати декілька потоків даних, які обробляються синхронно. Інформація в ньому організована у вигляді послідовності блоків (кадрів), причому аудіо- та відеодані чергуються між собою з метою забезпечення безперервності відтворення. Зважаючи на значні обсяги мультимедійної інформації, для оптимізації розміру файлів застосовуються алгоритми стиснення, тому результати роботи системи доцільно зберігати у компресованому вигляді.

Застосування функцій бібліотеки avifile.dll дозволяє працювати не зі структурою файлу безпосередньо, а з абстракціями вищого рівня — потоками та кадрами. Це дає можливість отримувати інформацію про вміст файлів,

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

виконувати операції читання і запису, а також вибірково обробляти окремі кадри. При цьому операції стиснення та розпакування даних виконуються автоматично, що значно спрощує реалізацію відповідного функціоналу.

Для обробки відеоданих передбачається використання стандартних кодеків, що входять до складу технології Video for Windows. Вибір конкретного алгоритму стиснення залежить від характеристик зображення та вимог до якості й обсягу кінцевого файлу. Наприклад, у випадках, коли відеоряд містить значні ділянки з однорідним кольоровим заповненням, доцільним є застосування алгоритму Microsoft RLE, який забезпечує велику ефективність при роботі із записами екрану, особливо за умов незначних змін між сусідніми кадрами.

Іншим варіантом є використання кодека Cinerak, який забезпечує збалансоване співвідношення якості та бітрейту відео, що робить його зручним для підготовки відеоматеріалів, орієнтованих на обмежену пропускну здатність носіїв інформації. Водночас метод Microsoft Video 1 характеризується більш високою швидкістю обробки, хоча й поступається за якістю, надаючи можливість вибору глибини кольору.

Також заслуговує уваги алгоритм Intel Indeo Video, який підтримує представлення зображення у форматі True Color та дозволяє здійснювати стиснення відеоданих у режимі реального часу за умови використання відповідного апаратного забезпечення.

З метою забезпечення високої якості зображення, відеодані доречно зберігати у форматі True Color (24 біти на піксель). У разі необхідності відтворення на пристроях із нижчою кольоровою роздільною здатністю, система автоматично виконує перетворення кольорів та формування палітр. Водночас слід враховувати, що процес стиснення відео може бути ресурсомістким і тривалим навіть за використання сучасного обладнання.

У процесі створення AVI-файлів доцільно експериментально визначити найбільш ефективний метод компресії, який дозволить стиснути зображення/файл без втрати якості.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Програмний комплекс, що створюється, під час функціонування взаємодіє з менеджером ICM (Installable Compression Manager), який відповідає за виконання операцій стиснення та декомпресії відеоданих. За потреби цей менеджер також може здійснювати безпосереднє відображення обробленої інформації на екрані. Як правило, пряме звернення до ICM необхідне лише у випадках редагування відеоматеріалів, оскільки стандартні засоби, такі як AVISar або MCI, виконують ці операції автоматично у фоновому режимі.

Додатково технологія Video for Windows надає набір функцій для відображення графічних даних у форматі DIB. Зокрема, функції сімейства DrawDib забезпечують високоефективне виведення зображень на екран. Наприклад, функція DrawDibDraw перевершує за швидкістю аналогічні засоби, одночасно забезпечуючи автоматичне відновлення стиснених даних перед їх відображенням. У випадках невідповідності кольорових форматів застосовується механізм згладжування (dithering), що дозволяє коректно відтворювати зображення навіть за обмежених можливостей відеосистеми.

Запропонований підхід до організації обробки мультимедійних даних забезпечить повноцінну реалізацію функціональних вимог, визначених технічним завданням.

Розроблюване програмне забезпечення повинне відповідати вимогам компактності, зрозумілості та зручності використання. Інтерфейс користувача має бути інтуїтивно доступним, а також забезпечувати надання довідкової інформації в процесі роботи системи. Візуалізація результатів буде здійснюватися у форматах, звичних для кінцевого користувача.

Для забезпечення стабільної та ефективної роботи розроблюваного програмного забезпечення визначено актуальні вимоги до апаратного та програмного середовища:

– центральний процесор із тактовою частотою не нижче 2.0 ГГц (рекомендовано багатоядерні рішення);

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

- не менше 1–2 ГБ вільного місця на накопичувачі (з урахуванням збереження мультимедійних даних);
- сучасний веб-браузер (наприклад, Google Chrome, Mozilla Firefox, Microsoft Edge останніх версій);
- оперативна пам'ять обсягом від 8 ГБ для забезпечення стабільної обробки відео- та аудіопотоків;
- операційна система Windows 10 або Windows 11, 64-розрядна версія.

Таким чином, беручи до уваги визначені структурні елементи системи, їх функціональне призначення, сформованих вимог та встановлених обмежень, а також обраних підходів до реалізації і сучасного технічного забезпечення, створено достатнє підґрунтя для подальшого проектування структурної та функціональної організації програмного продукту.

3.2 Розробка структурної схеми

Розроблювана система реалізується за модульною архітектурою, що передбачає наявність центрального керуючого компонента та набору функціонально незалежних підсистем. Кожен із таких модулів відповідає за виконання окремого класу завдань, визначених у межах правил, за якими функціонує програма, і послідовності її дій для виконання завдань.

Фундаментальною складовою системи виступає головний модуль, який представлений у вигляді графічного інтерфейсу користувача. Саме через нього ініціюється виконання основних операцій: взаємодія з функціональними компонентами здійснюється шляхом активації відповідних елементів керування. Кожна окрема дія обробляється або спеціалізованим модулем, або процедурою, що входить до його складу, та викликається за потреби.

Однією з ключових переваг запропонованої архітектури є можливість спрощеного оновлення компонентів системи. Зокрема, окремий модуль може бути замінений на новішу або вдосконалену версію без необхідності втручання в інші частини програми.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

Такий підхід дозволяє розробнику оперативно усувати недоліки, розширювати функціональність або оптимізувати роботу окремих підсистем. Оновлення реалізується шляхом заміни відповідного файлу модуля, після чого система одразу готова до подальшого використання без потреби у перезапуску.

З урахуванням того, що програмний продукт орієнтований на активну взаємодію з мережевим середовищем, особлива увага приділяється простоті та інтуїтивній зрозумілості інтерфейсу.

Після запуску користувачу відображається головне вікно, яке містить набір елементів керування для доступу до основних функцій системи. До них належать: процедура автентифікації користувача, завантаження конфігураційних модулів (режимів функціонування), обробка помилок системного рівня, а також довідковий режим (HELP). Результати виконання відповідних операцій відображаються безпосередньо в інтерфейсі.

Програмне забезпечення розробляється з урахуванням сумісності з сучасними операційними системами сімейства Windows, що забезпечує його використання на більшості персональних комп'ютерів.

Для наочного відображення внутрішньої організації системи сформовано структурну схему, яка ілюструє взаємозв'язки між її основними компонентами (рисунк 3.1). Функціональне призначення кожного елемента цієї схеми детально розкрито у відповідному підрозділі пояснювальної записки.

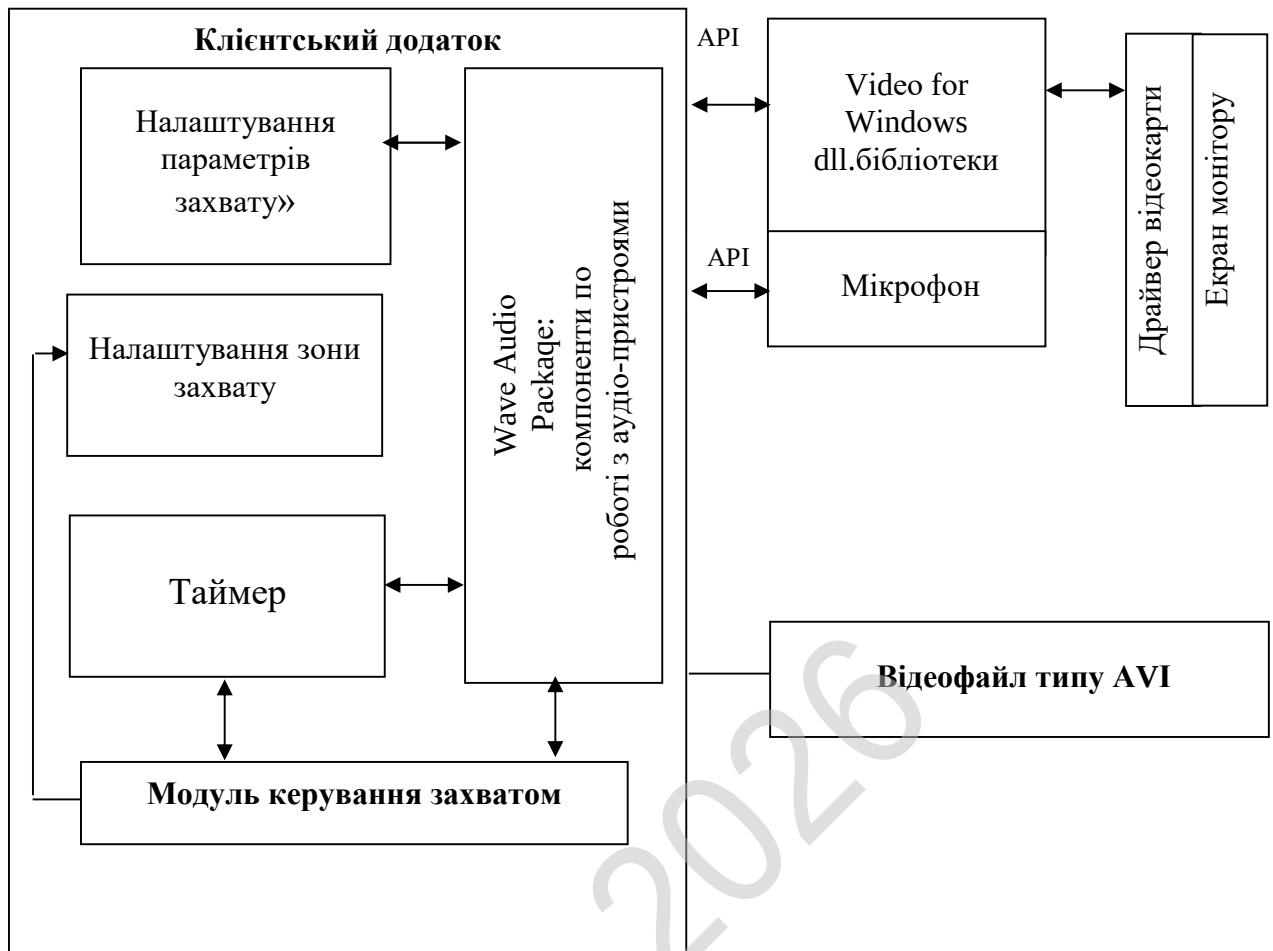


Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

Задля стабільності та архітектурної гнучкості майбутньої системи, її структуру доцільно розділити на логічні блоки (компоненти). Кожен блок відповідатиме за конкретний аспект функціонування ПЗ:

- центральне ядро (Main Core): точка входу, що об'єднує графічну оболонку (GUI), систему навігації та довідки, а також механізми перехвату та обробки виняткових ситуацій;

- блок конфігурації клієнта: керує загальними параметрами роботи програми;

- служба таймінгу: відповідає за часові інтервали та автоматизацію процесів;
- компонент керування робочою областю: дозволяє точно визначити межі екрана для зйомки;
- менеджер параметрів медіапотоку: регулює технічні властивості вихідного контенту.

Деталізація функціонального наповнення модулів:

1. Підсистема маніпулювання відеокадрами: даний функціональний блок реалізує безпосередню роботу з потоком даних:

- ініціалізація створення вихідного файлу та вибір шляху збереження;
- активація процесу фіксації зображення;
- коректне завершення сесії запису;
- постпроцесинг (компресія) відеоданих для оптимізації обсягу пам'яті.

2. Налаштування середовища користувача: тут зосереджені інструменти персоналізації роботи програми:

- встановлення пріоритетності процесу в ОС;
- коригування колірної гами (використання оригінальних кольорів або негатива);
- керування видимістю курсора миші на відео;
- візуалізація рамок активної зони захвату;
- управління режимом попереднього перегляду (Preview);
- автоматичне згортання інтерфейсу при старті роботи;
- налаштування поведінки системи після завершення фіксації кадру.

3. Робота з областю екрана та часом: спеціалізований модуль таймера дозволяє делегувати системі контроль за часом, встановлюючи момент автоматичного припинення запису. Своєю чергою, підсистема позиціонування забезпечує три сценарії вибору зони:

- ручне виділення довільного фрагмента екрана;

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

– використання динамічного вікна, що прив'язане до координат маніпулятора;

– повна фіксація конкретного активного вікна додатка.

4. Технічні специфікації запису: цей модуль є критичним для якості фінального продукту. Він дозволяє гнучко налаштовувати:

– алгоритми стиснення (відеокодеки) та рівень якості;

– активацію або деактивацію звукового супроводу;

– параметри аудіопотоку (PCM-формат, бітрейт, джерело сигналу);

– частоту дискретизації кадрів (від 1 до 50 FPS) та інтервал між ними (від 20 мс до 10 с).

Архітектурні особливості та інтерфейс. основним джерелом візуальних даних виступає монітор комп'ютера, а для аудіофіксації використовується мікрофон. Усі перелічені модулі інтегровані в єдину екосистему з централізованим керуванням. Такий підхід дозволяє користувачу запускати будь-яку функцію з головного меню, що робить взаємодію інтуїтивно зрозумілою.

Програмний продукт функціонує локально на ПК. Для мінімізації впливу на робочий простір, інтерфейс виконано у вигляді компактного вікна з меню, яке автоматично приховується під час активної фази запису.

Алгоритм роботи передбачає попереднє конфігурування: користувач обирає зону (весь робочий стіл або його частину) та технічні параметри (кодек, звук, частоту кадрів). Після активації процесу (у ручному або програмованому режимі) система миттєво розпочинає конвертацію візуального потоку у відеофайл формату AVI.

Результатом впровадження такої схеми є автоматизований інструмент, що забезпечує високу ефективність обробки помилок та надає вичерпну інформацію про стан поточних процесів, що відображено на функціональній схемі (рисунок 3.2).

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

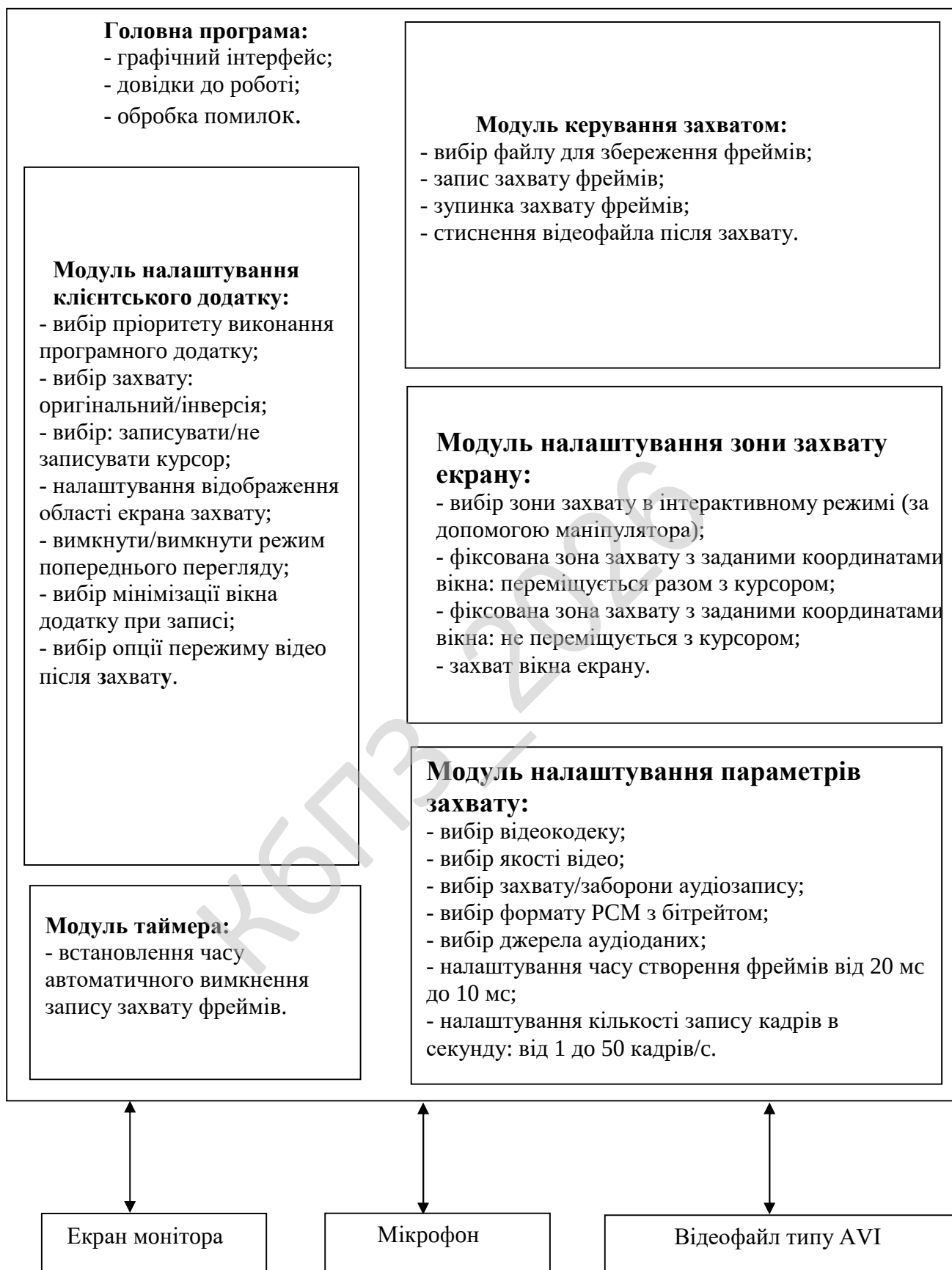


Рисунок 3.2 – Функціональна схема системи

Алгоритм функціонування та сценарії використання ПЗ.

Процес ініціалізації запису може відбуватися за двома сценаріями: в автоматичному режимі або за безпосередньої участі оператора. Після активації система негайно розпочинає трансляцію візуальних даних із визначеної ділянки екрана у файл контейнера AVI.

Аналогічно завершення сесії фіксації відеопотоку реалізовано двома способами:

- ручне припинення: здійснюється користувачем через інтерфейс програми;
- програмне припинення: активується інтегрованим модулем таймінгу за попередньо встановленим графіком.

Для кожного об'єкта візуалізації в системі передбачено розширений набір індивідуальних параметрів, а саме:

- визначення конкретного проміжку кадрів для фінального рендерингу;
- керування видимістю вказівника миші (активація/деактивація відображення);
- трансформація колірної палітри (збереження природних кольорів або застосування інверсійного фільтра);
- розподіл обчислювальних ресурсів ОС через налаштування пріоритетності процесу;
- використання алгоритмів стиснення для оптимізації обсягу готового медіафайлу.

Ключовою перевагою розробленого рішення є його ергономічність. Лаконічний інтерфейс спроектований таким чином, щоб взаємодія з програмою не потребувала від спеціаліста навичок програмування чи глибокої технічної підготовки.

Гнучкість системи дозволяє адаптувати її під специфічні завдання шляхом маніпуляції пріоритетами виконання та технічними характеристиками запису. Розглянемо типові моделі експлуатації:

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

1. Моніторинг активності (фоновий нагляд).

У випадках, коли необхідно проконтролювати використання робочого часу персоналом або активність дитини за комп'ютером, доцільно застосовувати «економний» режим:

- охоплення: повна площа робочого столу.
- якість: мінімальна (для зниження навантаження на апаратне забезпечення під час тривалої роботи).
- частота оновлення: рідка фіксація кадрів (інтервал від 10 секунд і більше), що дозволяє суттєво економити дисковий простір.

2. Створення навчального та медійного контенту.

Для підготовки відеоінструкцій, туторіалів або демонстраційних роликів, вимоги до якості суттєво зростають:

- охоплення: весь екран або фокусування на конкретному сегменті/вікні.
- якість: максимальна чіткість зображення (High Quality).
- динаміка: частота запису від 24 кадрів за секунду (FPS) і вище (залежно від продуктивності ЦП) для забезпечення плавності картинки.
- мультимедійність: обов'язкова синхронізація відеоряду з аудіосупроводом у реальному часі.

3.4 Розробка діаграми процесів

Для візуалізації внутрішньої архітектури програмного продукту, а також встановлення чіткої ієрархічної підпорядкованості окремих операцій, було спроектовано діаграму процесів (рисунок 3.3). Дана схема дозволяє простежити логіку взаємодії між модулями, визначити послідовність передачі керуючих сигналів та оцінити ступінь взаємного впливу компонентів системи під час їх паралельного виконання.

Зокрема, представлена модель відображає:

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

- вертикальні зв'язки: делегування повноважень від головного керуючого ядра до периферійних функціональних модулів;
- неформаційні потоки: обмін даними між підсистемою конфігурації та виконавчим механізмом захвату медіапотоку;
- контрольні точки: механізми зворотного зв'язку, що забезпечують стабільність роботи ПЗ при виникненні системних переривань.

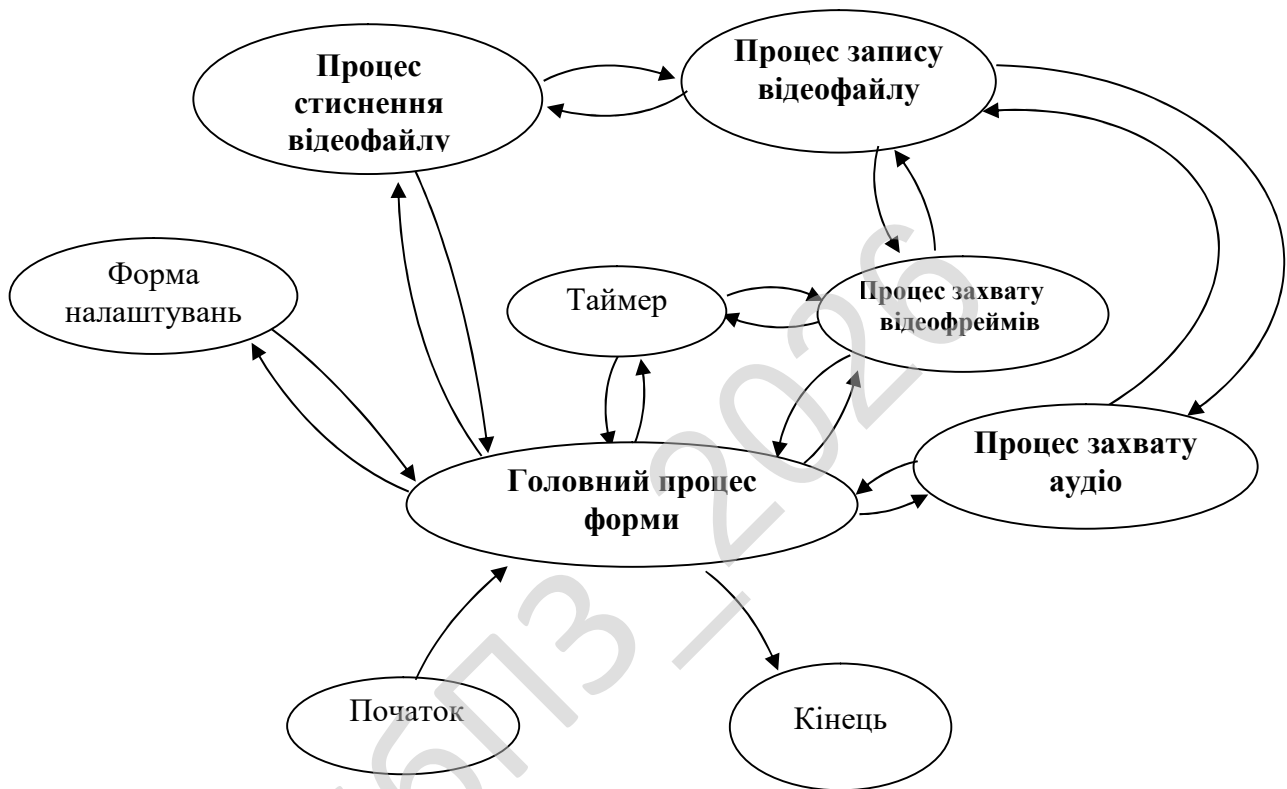


Рисунок 3.3 – Діаграма взаємодії процесів в системі

Підбиваючи підсумок, можна констатувати завершення етапу теоретичного моделювання архітектури майбутнього програмного комплексу. В ході проведеної роботи було досягнуто наступних результатів:

- аналітичний базис: сформовано перелік ключових вимог та технічних обмежень, які є критичними для стабільного функціонування системи в умовах реальної експлуатації;

– функціональне наповнення: деталізовано основне призначення системи, а для кожного робочого сценарію розроблено відповідний інструментарій налаштувань та опцій ї;

– технологічний стек: визначено оптимальні механізми та методи програмної реалізації окреслених інженерних рішень ї;

– модульна декомпозиція: на основі функціонального аналізу сформовано компонентний склад системи, де для кожного модуля чітко окреслено коло завдань та зону відповідальності ї;

– графічне моделювання: розроблено комплекс взаємопов'язаних схем, включаючи загальну структуру системи, її функціональну модель та ієрархічну діаграму процесів.

Отже, створений теоретичний фундамент та детальна проектна документація забезпечують повний обсяг вихідних даних, необхідних для переходу до стадії алгоритмізації та безпосереднього написання програмного коду.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

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

Програмне забезпечення системи автоматизованого запису дій користувача на ПК розроблене у відповідності з вимогами побудови систем аналогічного класу та спрямування та є багаторівневою системою реального часу, оскільки обробка даних ведеться за позначками реального часу.

4.1 Блок-схеми та опис алгоритмів функціонування системи

Згідно технічного завдання, в процесі виконання ВКРБ розробці підлягає ПЗ автоматизованого запису дій користувача на ПК. По суті – це інструмент для запису користувачем своїх дій за комп'ютером у відеофайл. Не на відеокамеру і навіть не на відеомагнітофон, а в один із стандартних і найбільш поширених відеоформатів, який, не сходячи з місця, можна буде продивитись чи програти, насолодившись результатом. Як звичайна відеокамера дозволяє записати окремі моменти для подальшого перегляду, так і система запише будь-які дії користувача на комп'ютері, будь то робота з офісними пакетами, створення відеоуроків, проходження улюбленої гри, читання пошти або тонке налаштування програми.

Найважливішою особливістю розробленої системи є те, що користувачу абсолютно не треба вдаватися до тонкощів роботи з комп'ютерним відео - досить натиснути кнопку "Старт", вибрати область захвату і весь екран чи його сектор будуть записуватись до відеофайлу.

Від інших програм подібного толку система буде вигідно відрізнятися також якістю створюваного ролика і можливістю накладати на відеоматеріал звук з мікрофону. А вбудований оптимізатор дозволить уникнути гальмування

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

системи і стрибків показчика миші при великих розмірах кадру. Цей недолік, властивий практично усім подібним програмам. В якості області запису користувач зможе навіть використати розгорнуте на увесь екран вікно, без побоювання дістати витік пам'яті і гальмування роботи системи.

Структура системи спроектована у відповідності з вимогами концепції побудови програмних системи (розділ 2 пояснювальної записки) і є системою реального часу, оскільки обробка даних здійснюється за позначками реального часу. Разом з тим, робота системи не мають перешкоджати основній діяльності інших програмних продуктів, не має виникати колізій і при роботі системи по безпосередньому призначенню з системним та програмним забезпеченням ПК. Тому при проектуванні алгоритмів системи використані засади системного підходу, в рамках якого окремі модулі системи розглядались та проектувались в сукупності з урахуванням їх можливого впливу на роботу інших програмних модулів, та додатків, що можуть знаходитись на ПК, де буде встановлена спроектована система. Таким чином, при проектуванні системи були враховані наступні аспекти:

- інтеграція неоднорідних модулів в єдину систему;
- урахування взаємного впливу окремих механізмів, одночасно працюючих з системою;
- ідентифікація та перевірка достовірності суб'єктів доступу при вході в систему по ідентифікатору/паролю/ключу.

Розроблене в процесі реалізації ВКРБ програмне забезпечення системи передбачає виконання наступних функцій, направлених на виконання основної задачі – організація процесу запису дій користувача на ПК:

- захват відеофреймів з повного екрана/визначеної геометричної області екрана;
- запис захвачених відеофреймів в обраний файл;
- стиснення відеофайла після захвату.

Необхідно визначитись з якісними вимогами до ПЗ. В процесі теоретичної побудови системи було вирішено за основу взяти наступні пріоритети цих вимог:

- коректність розробленого ПЗ: забезпечення одержання правильних результатів роботи кожної програми, модуля та системи в цілому;

- достовірність: одержання відповідної реакції системи на нестандартні ситуації та надання доцільної теоретичної інформації користувачу (в разі потреби).

Враховуючи зазначене вище, для побудови дійсно дієспроможної системи необхідно провести теоретичні розрахунки, розробити блок-схеми алгоритмів програм, алгоритми програм та описати їх програмним кодом.

Розпочнемо з головного. В якості механізму реалізації ПЗ було обрано компоненту Wave Audio Package та бібліотеку Microsoft Video for Windows. Обґрунтування цього вибору та перелік можливостей надаються в розділі 3 пояснювальної записки. Цей вибір дозволить розробити дійсно ефективне програмне забезпечення для виконання означених ТЗ функцій, використавши формат AVI (Audio Video Interleave), який є основним для зберігання відео в операційних системах родини Windows.

Основна перевага універсального формату AVI на відміну від потокових форматів типу MPEG, а тим більше - від таких спеціалізованих різновидів, як MP3 (MPEG Audio Layer 3), в тому, що «стандартних» AVI-файлів практично не існує : AVI - фактично не більше, ніж «контейнер», який містить загальний опис вмісту в стандартизованому різновиді.

Таким чином, AVI-файли тільки зовні виглядають однаково, але усередині вони можуть сильно розрізнятися, і тоді, як MPEG -, MP3 - і MJPEG - файли містять потоки тільки певного виду компресії, AVI може містити багато різних видів компресії у будь-яких поєднаннях. Проте працювати з AVI-файлом можна буде тільки при наявності кодека, доступного для кодування/декодування.

Відповідно до загальної структури RIFF-типа, AVI-файл має наступний вигляд:

RIFF 'AVI ' // чотирьохбуквений ідентифікатор файлу (у RIFF -форматі)

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

```
LIST 'hdrl' // список заголовків блоків, що визначають формати потоків
LIST 'movi' // блоки даних (потоків) AVI-файла
'idx1' // необов'язковий блок, що визначає розміщення блоків даних
усередині AVI-файла <AVI Index>
```

Тобто, в AVI-файлі має бути принаймні два обов'язкові блоки: заголовок і дані, які, у свою чергу, можуть містити підблоки. Перший блок міститиме загальну інформацію про відеоролик: дозвіл кадрів і їх частоту, формат аудіо, тощо.

Структура списку 'hdrl' має наступний вигляд:

```
LIST 'hdrl' // список заголовків блоків, що визначають формати потоків
'avih' // головний заголовок AVI-файла
LIST 'strl'
'strh' // заголовок потоку
'strf' // формат потоку
'strd' // додатковий заголовок даних
```

Список 'movi', у свою чергу, складається з підблоків:

```
LIST 'movi' // блоки даних (потоків) AVI-файла
SubChunk | LIST 'rec ' // підблок | список записів
'##wb(data) // звукові дані (блок)
'##dc(data) // відеодані (блок)
'##db(data) // відеодані (блок)
```

Отже, підблок даних організований у вигляді послідовності записів, кожен з яких складається з одного кадру відео і відповідного звукового супроводу. Спочатку ##dc-блок був призначений для зберігання стисненого зображення, а ##db-блок - для нестисненого DIB (Device Independent Bitmap). Але фактично вони обоє можуть містити стиснені дані.

Роботу програмного забезпечення системи означимо наступними кроками:

- Крок 1: ініціалізація системи.
- Крок 2: вибір пункту меню «Налаштування». Проведення налаштувань опцій системи: клієнтського додатка, зони захвату екрану, параметрів захвату, таймера в залежності від обраного спрямування роботи системи.
- Крок 3: в інтерактивному режимі в діалозі властивостей відео встановлення необхідних для роботи системи значень.
- Крок 4: вибір пункту меню «область запису» - «Записати окреме вікно».

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

- Крок 5: відкриття програми, дії з якою будуть записуватись.
- Крок 6: початок запису.
- Крок 7: зупинка запису натисненнями кнопки Num Lock.
- Крок 8: у діалозі, що з'явився, введення імені файлу для його збереження.

- Крок 9: кінець роботи системи.

Тепер можна перейти до визначення шляхів реалізації функціональних можливостей, які має забезпечити система. Почнемо з визначення типів моніторингу, які будуть реалізовані в системі. Всього їх шість, і кожен з них може бути включений і вимкнений незалежно від інших.

Перший - реєстрація натиснень користувача на клавіші. Причому програма зможе записувати і використання функціональних клавіш (F1 - F12, всілякі поєднання з кнопками Alt і Ctrl). Разом з тим, окрім безпосередньо самих натиснень, у балках зберігаються їх час, програма, якою вони були створені, і заголовок активного вікна. Усе це дозволяє дізнатися не лише, що саме вводив користувач, але і в якому ПЗ, в якому відкритому файлі він це робив.

Другий тип моніторингу – створення скриншотів екрана. Вони необхідні для того, щоб наглядно побачити, що робив користувач під час своєї роботи на ПК. Скриншоти програма зможе робити, як по таймеру (через визначену кількість секунд), так і при відкритті кожного нового вікна.

Наступний тип моніторингу, який може здійснювати система, полягає в контролі активності програмного забезпечення. При його використанні система зберігатиме у балці інформацію про все запущене і закрите ПЗ. Таким чином, власник ПК завжди знатиме, які програми використовувалися в його відсутність.

Четвертий тип моніторингу - відстежування того, яка інформація зберігалася у буфері обміну.

П'ятий вид інформації, який система буде зберігати у балках, - адреси відвіданих веб-сторінок. Це дуже важливо, тому що контроль над використанням Інтернету дуже актуальний. Відвідування деяких сайтів може не лише

відволікати співробітників від роботи і впливати на дитячу психіку, але і серйозно погрожувати безпеці комп'ютера. А тому контроль в цій області просто необхідний.

Ну і, нарешті, останній вид моніторингу дозволить відстежувати процеси включення, виключення і перенавантаження комп'ютера. В даному випадку моніторинг – це не програмно-реалізовані функції контролю, а вхідна інформація (відеоролики, набір скриншотів) для проведення моніторингу в інтерактивному режимі відповідальною особою.

Іншою цікавою можливістю системи буде виборочний моніторинг. По-перше, власник комп'ютера може для кожного користувача включити або відключити різні види контролю аж до повної відмови від стеження за діями деяких людей. Такий підхід дозволяє гнучко налаштувати систему контролю і мінімізувати розмір записуваних відеофайлів. Ця функція може виявитися дуже корисною, наприклад, в домашніх умовах.

В якості механізму програмної реалізації системи нами обрана система Microsoft Video for Windows версії 1.1, яка поставляється окремо в якості розширення для Microsoft Windows і Microsoft Windows for Workgroups. Video for Windows містить DLL - бібліотеки і драйвери, що забезпечують інтерфейс між додатками і мультимедіа-пристроями, призначеними для роботи з звуковими даними і відеоданими.

Програми, що програють мультимедіа-файли (у тому числі avi-файли), можуть використати драйвер mciavi.drv, що надає в розпорядження додатка високорівневий інтерфейс MCI. Але більш доцільне використання інтерфейсу, який забезпечується бібліотекою msvideo.dll . Ця бібліотека є надбудовою над інтерфейсом MCI, що значно спрощує його використання. Вона потрібна не лише для програвання, але і для запису захвачених відеоданих та аудіоданих.

Викликаючи функції інтерфейсу Video for Windows, система не буде мати справи з внутрішньою структурою avi-файлів, оскільки бібліотека avifile.dll забезпечує їм усю необхідну підтримку.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

Програми системи, призначені для запису звукових даних і відеоданих, використовують зручний високорівневий інтерфейс, що надається бібліотекою avicap.dll. Створюючи додатки для запису відео, користувачу не доведеться піклуватися про внутрішню структуру avi-файлів, про компресію (стискування) даних при записі, про інтерфейс з драйверами обладнання (чи пристроїв) запису. При необхідності, система використовує інтерфейс нижчого рівня, який забезпечується бібліотекою avifile.dll. Редактори файлів мультимедіа будуть використовувати бібліотеки avifile.dll і msvideo.dll для доступу до даних і засобів відображення, а також для компресії даних.

Бібліотека msvideo.dll містить визначення класа вікна MCI, на базі якого можна дуже легко створювати додатки, що управляють пристроями для відтворення відео, відтворення і запису звукових даних, відтворення MIDI, а також додатки, що управляють обладнаннями читання компакт-дисків.

Вікно MCI створити так же просто, як і звичайне вікно. Більше того, для його створення можна використовувати звичну функцію CreateWindow, але більш доцільно використати спеціальну функцію MCIWndCreate. Знаючи ідентифікатор вікна MCI, можна управляти вікном, використовуючи набір макрокоманд, таких, як MCIWndRecord (включення режиму запису) і MCIWndPlay (включення режиму відтворення). Є також засоби для передачі вікну звичайних команд або командних рядків MCI.

У більшості випадків для програвання файлів мультимедіа можливості вікна MCI більш ніж достатні. При розробці системи нами використане вікно класу AVICap. Але перш, ніж приступити до розробки алгоритмів системи необхідно визначитись з обмеженнями та кодом помилок MCI.

Наводимо основні коди помилок MCI, які можуть виникнути при роботі системи:

- MCIERR_BAD_TIME_FORMAT - неправильний формат часу;
- MCIERR_CANNOT_LOAD_DRIVER - неможливо завантажити драйвер;

- MCIERR_CANNOT_USE_ALL - для цієї команди не можна використовувати рядок all як ім'я пристрою;
- MCIERR_CREATEWINDOW - неможливо створити або використовувати вікно;
- MCIERR_DEVICE_LENGTH - занадто довге ім'я пристрою або драйвера (більше 79 символів);
- MCIERR_DEVICE_LOCKED - пристрій закритий, через декілька секунд можна спробувати повторити команду;
- MCIERR_DEVICE_NOT_INSTALLED - вказаний пристрій не встановлений в системі та інші.

Розглянемо програмну реалізацію основних алгоритмів системи. Почнемо з функції відтворення відео на поверхні вікна монітору. Для реалізації цієї функції до списку модулів, що підключаються, додаємо модуль VFW з бібліотекою avifile 32.dll, а в установках проекту – шлях до файлу ole2.dcu, якого потребує модуль, що підключається.

Наступні змінні пов'язані із специфікою задачі, яка в даний час вирішується:

```
var
  TmpBmp : TBitmap; // Допоміжне зображення
  AviStream : PAVISTREAM; // AVI-потік
  Frame : PGetFrame; // кадр відео
  pbmi : PBITMAPINFOHEADER; // покажчик на заголовок растру
  bits : Pointer; // покажчик на картинку растру
  CurrFrame DWORD - 0; // Лічильник кадрів
  AVIClock DWORD; // Емулятор таймера
  AVIDelay DWORD; // Величина паузи між кадрами
  AVIwidth DWORD; // Характеристики кадру
  AVIheight DWORD;
  AVILength DWORD; // Кількість кадрів в AVI
```

Перед ініціалізацією бібліотеки avifile 32.dll необхідно розпізнати характеристики відео, яке необхідно відобразити. Для цього використовуємо функції даного модуля, пов'язані з роботою AVI, як з файлом:

```
var
  AVIFile : PAVIFile; // Оброблюваний AVI, як файл
  AVIInfo : TAVIFileInfo; // Заголовок файлу, характеристики AVI
begin
  TmpBmp := TBitmap.Create; // Створюємо допоміжний растр
  AVIFileOpen(AVIFile, AVIName, OF_READ, nil); // Відкриття AVI
```

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

```
// читаємо заголовочну інформацію, заповнюємо поля AVIInfo
AVIFileInfo(AVIFile, AVIInfo, SizeOf (AVIInfo));
AVIWidth := AVIInfo.dwWidth; // Запам'ятовуємо розміри кадру
AVIHeight := AVIInfo.dwHeight;
AVILength := AVIInfo.dwLength; // Кількість кадрів
// Обчислюємо паузу між черговими кадрами
AVIDelay := 1000 div (AVIInfo.dwRate div AVIInfo.dwScale);
AVIFileRelease(AVIFile); // звільняємо AVI
```

Поверхня FDDSDImage, на якій відтворюватиметься відео, створюється з розмірами, рівними розмірам кадру відео:

```
ZeroMemory (@ddsd, SizeOf(ddsd));
with ddsd do begin
dwSize := SizeOf (ddsd);
dwFlags := DDS_DCAPS or DDS_HEIGHT or DDS_WIDTH;
ddsCaps.dwCaps := DDSCAPS_OFFSCREENPLAIN;
dwWidth := AVIWidth;
dwHeight := AVIHeight;
end;
hRet := FDD.CreateSurface(ddsd, FDDSDImage, nil);
if Failed(hRet) then ErrorOut(hRet, 'Create Second Surface');
```

Після ініціалізації DirectDraw наступним кроком готуємося до роботи з потоковим відео:

```
procedure TFRMDD.FirstFrame;
var
wrkDC : HDC;
begin
AVIFileInit; // Ініціалізація бібліотеки
// Відкриваємо AVI-файл для читання
AVIStreamOpenFromFile (AVIStream, AVIName, streamtypeVIDEO
0, OF_READ, nil);
// Завантажуємо потік
Frame:= AVIStreamGetFrameOpen(AVIStream, nil);
// Отримуємо перший кадр відео
pbmi:= AVIStreamGetFrame(Frame, CurrFrame);
// Отримуємо покажчик на картинку кадру
bits:= Pointer(Integer(pbmi)+ SizeOf(TBITMAPINFOHEADER));
// Отримуємо контекст для відтворення кадру на поверхню
if FDDSDImage.GetDC (wrkDC) = DD_OK then begin
// Відтворюємо кадр в допоміжний растр
TmpBmp.Handle := CreateDIBitmap(
// Допоміжним контекстом служить HDC поверхні
wrkDC
pbmi^, // Адреса розміру растру і формату даних
CBM_INIT, // Прапорець ініціалізації
bits, // Дані для ініціалізації
PBITMAPINFO (pbmi)^, // Дані про формат кольорів
DIB_RGB_COLORS); // Прапорець означення кольора растра
// Переносимо фрейм (кадр) допоміжного растра на поверхню
BitBlt (wrkDC, Про, Про, AVIWidth, AVIHeight
TmpBmp.Canvas.Handle, 0, 0, SRCCOPY);
FDDSDImage.ReleaseDC (wrkDC);
end;
AVIClock:= GetTickCount; // Ініціалізація допоміжного таймера
end;
```

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Дії з відтворення чергового кадру аналогічні, але витратити час на отримання адрес тепер не потрібно:

```
procedure Tfrmdd,nextframe;
var
wrkDC : HDC;
begin
// Настав час відтворити наступний кадр AVI
if GetTickCount - AVIClock > AVIDelay then begin
pbmi := AVIStreamGetFrame(Frame, CurrFrame);
if FDDSIImage.GetDC (wrkDC) = DD_OK then begin
TmpBmp.Handle := CreateDIBitmap(wrkDC, pbmi^, CBM_INIT
bits, PBITMAPINFO(pbmi л, DIB_RGB_COLORS) ;
BitBlt (wrkDC, 0, 0, AVIWidth, AVIHeight
TmpBmp.Canvas.Handle, 0, 0, SRCCOPY);
FDDSIImage.ReleaseDC (wrkDC);
end;
// Збільшуємо лічильник кадрів
CurrFrame:= (CurrFrame + 1) mod AVILength;
AVIClock:= GetTickCount;
end;
end;
```

В наведеному прикладі AVI-файл відновлюється нескінченим циклом, услід за останнім кадром все повторюється з початку. Кадр відтворюється на поверхні FDDSIimage, тобто в місці, визначеному користувачем при проведенні налаштувань системи.

Програма завершується наступними очікуваними діями:

```
AVIStreamRelease(AviStream); // Закриття потоку
AVIFileExit; // Завершення роботи з бібліотекою
TmpBmp.Free; // Видалення допоміжного растру
```

Відтворення якісного відео – задача складна, але необхідна для будь-якого користувача. Розробка в системі однієї програми на використання тільки модуля VFW з бібліотеки dvifill.dll не є правильним рішенням, оскільки не забезпечить якісного відтворення створеного відеофайлу на екрані ПК.

Розглянемо ще один механізм відтворення відео, базований на застосуванні модуля DirectShow. Ця бібліотека входить до складу DIRECTX, включає набір функцій для роботи із мультимедіа. Для цього застосуємо проект каталога Ex02, відтворюючого AVI-файл на поверхні екрана.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Якщо при відтворенні файлу за допомогою модуля VFW картинка виходить спотвореною, використаємо цей спосіб. Відмовитися від першого способу недоцільно, оскільки DirectShow теж придатна не для будь-якого файлу.

Модуль використовує COM-модель, тому перш за все означимо головний об'єкт і дочірні інтерфейси:

```
var
AMStream : IAMMultiMediaStream; // Головний об'єкт
PrimaryVidStream : IMediaStream; // Дочірній потік, пов'язаний з відео
Sample: IDirectDrawStreamSample; // Інтерфейс для виведення на поверхню
```

Першим кроком у процедуру ініціалізації потоку передається ім'я необхідного файлу:

```
procedure TFRMDD.PlayMedia(const FileName: WideString);
var
hRet: HRESULT;
begin
// Створення головного об'єкта ('filter graph1')
AMStream:=IAMMultiMediaStream(CreateComObject
(CLSID_AMMultiMediaStream;
// Ініціалізація потоку для читання
hRet:= AMStream Initialize(STREAMTYPE_READ, 0, nil);
if Failed (hRet) then ErrorOut (hRet, 'Stream Initialize');
// Додавання потоку відео до головного об'єкта
hRet:= AMStream.AddMediaStream(FDD MSPID_PrimaryVideo
0, IMediaStream(nil));
if Failed (hRet) then ErrorOut (hRet, 'Add Video Stream');
// Відкриття файла
hRet := AMStream.OpenFile(PWideChar (FileName), 0);
if Failed (hRet) then ErrorOut (hRet, 'Open AVI File');
// Наступні дії призначені для скріплення потоку і поверхні
// Отримання дочірнього потоку
hRet := (AMStream as IMultiMediaStream).
GetMediaStream (MSPID_PrimaryVideo, PrimaryVidStream);
if Failed (hRet) then ErrorOut (hRet, 'GetMediaStream');
// Перетворення інтерфейса в тип Isample
// і зв'язування його з поверхнею
hRet := (PrimaryVidStream as IDirectDrawMediaStream).
CreateSample (FDDS Image, TRect (nil/4), 0, Sample);
if Failed (hRet) then ErrorOut (hRet, 'CreateSample');
// Запуск потоку
hRet := (AMStream as IMultiMediaStream).SetState(STREAMSTATE_RUN);
if Failed (hRet) then ErrorOut (hRet, 'SetState');
end;
```

Обробка можливих помилок проводиться аналогічно першому варіанту, але коди помилок не розшифровуватимуться. Модуль DirectDraw не містить, пояснення цих помилок.

Перед блітингом поверхні FDDS image оновлює позицію в відео:

```
if Sample.Update(0, 0, nil, 0) <> S_OK
```

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

```
then (AMStream as IMultiMediaStream).Seek (0);
```

Невдале завершення вищезначеної процедури свідчить про те, що відео прокручене до кінця. В цьому випадку необхідно наново запустити його та виконати повторну спробу відтворення відеофайлу.

Після того, як захват фреймів екрану ПК вже здійснено, необхідно провести їх запис в відеофайл (AVI-файл). Розглянемо програмну реалізацію цієї процедури.

Пройде цей процес просто, оскільки скористаємося готовим модулем AviMaker. Він містить опис класу TAviMaker, що виконує всю виснажливу роботу. Залишається лише підготувати набір растрів, що становлять послідовність кадрів створюваного фільму, і викликати метод запису.

Наводимо набір властивостей і методів класу, які забезпечать програмну реалізацію функції запису фреймів в файл::

```
Bitmaps TList; // Список об'єктів класу Bitmap, кадри AVI
Height Integer; // Розміри кадрів AVI
Width Integer;
FrameTime Integer; // Величина паузи між кадрами
Stretch BOOL; // Ознака, чи треба масштабувати кадри
FileName String; // Ім'я файлу результату
PixelFormat : TPixelFormat; // Розрядність AVI
constructor Create;
destructor Destroy; override;
procedure Write; // Запис AVI
```

Використавши модуль VFW, в опціях проекту вказуємо шлях до файлу ole2.dcu. У проекті каталога Ex03 формуються кадри.

Для запису фільму чи дій користувача з екрана ПК, використаємо об'єкт AviMakerI класу TAviMaker:

```
AviMakerI := TAviMaker.Create;
with AviMakerI do begin
Width := 256;
Height := 256;
Stretch := True; // Кадри масштабуватимуться
PixelFormat := pf24bit; // 24-бітовий формат кадру
FrameTime := 100;
FileName := 'test.avi';
end;
```

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Переходимо безпосередньо до реалізації функції запису дій користувача з екрану ПК. В якості прикладу зазначимо, що його обсяг – 20 фреймів (кадрів), а тривалість – 2 сек.

```
function TFRMDD.UpdateFrame : HRESULT;
const
  step = 2 * Pi / 400;
var
  i : Integer;
  hRet : HRESULT;
  Bitmap : TBitmap; DC : HDC;
begin
  ThisTickCount:= GetTickCount;
  if ThisTickCount - LastTickCount > 30 then begin
    Angle := Angle + 0.25;
    if Angle > 2 * Pi then Angle := Angle - 2 * Pi;
    LastTickCount := GetTickCount;
    // Беремо 20 кадрів
    Inc (FrameCount);
    if FrameCount > 20 then begin
      FActive := False;
      Result := DD_OK;
      AviMakerl.Write; // Записуємо AVI
      Close;
      Exit;
    end;
    // Виводимо фон
    hRet := FDDSBck. BltFast (0, 0, E'DDSBackGround
    nil, DDBLTFast_WAIT);
    if hRet = DDERR_SURFACELOST thenbegin
      Result := hRet;
      if Failed (RestoreAll) then Exit;
    end;
    // Виводимо точки спіралі for i := 0 to 800 do
    if FDDSBck.BltFast (310 + trunc (cos (Angle -f i * step)* i / 4)
    230 + trunc (sin(Angle + i * step)*i / 4)
    FDDSIImage, nil
    DDBLTFast_WAIT) = DDERR_SURFACELOST then begin
      hRet := RestoreAll;
      if Failed (hRet) then begin
        Result := hRet; Expend;
      end;
      // Формуємо кадр фільму/ролика/контролю дій спостерігача/тощо
      Bitmap := TBitmap.Create;
      with Bitmap do begin Width := ScreenWidth;
      Height := ScreenHeight;
      PixelFormat := pf24bit; // Важливо, інакше встановлюється 8 біт
      FDDSBck.GetDC (DC);
      BitBlt(Canvas.Handle, 0, 0, // Копіюємо екран в растр
      ScreenWidth, ScreenHeight, DC, 0, 0, SRCCOPY);
      FDDSBck.ReleaseDC (DC) ;
      AviMakerl.Bitmaps.Add (Bitmap); // Додаємо кадр у фільм
    end;
  end;
  Result := DD_OK;
end;
```

В якості вхідних даних будемо розглядати фрейми та звукові потоки, які надають в автоматичному режимі джерела: монітор ПК, мікрофон. Система забезпечує їх введення у вигляді послідовності даних.

Вихідні дані забезпечуються часовою міткою, що фіксує момент їх передачі/прийняття. Дані обробляються в динамічному режимі на протязі всього безперервного циклічного процесу запису дій користувача з екрану ПК.

Мінімальна глибина зберігання даних відповідає нормативним документам. Система забезпечує їх архівацію у виділені відеофайли формату AVI.

При розробці ПЗ системи використане середовище Delphi та програмування під ОС Windows з застосуванням подій Windows та елементів ActiveX. Це дозволило підвищити зрозумілість/читабельність програмній моделі для сучасного користувача, який звик працювати в цій операційній системі.

Програмно реалізовано досить складну багатфункціональну систему. ПЗ системи має модульну структуру побудови. До його складу увійшли стандартні модулі і модулі, розроблені в процесі розробки ПЗ системи.

Керування модулями системи здійснюється основною програмою. Вона за своїм призначенням - диспетчер системи, забезпечуючи виконання наступних функцій: санкціонований доступ до роботи з системою користувачів; завантаження програм-модулів, допоміжних модулів та керування їх роботою: запуск, закриття; ведення режиму допомоги користувачу щодо налагодження режимів її роботи та роботи по прямому призначенню; обробка системних помилок, надання інформаційних повідомлень користувачу.

Головна програма використовує зовнішній графічний інтерфейс. Це зробить систему доступною для зовнішніх модулів, які звертаються до неї в процесі роботи, підключених до основної програми модулів, оголошених типів, класів, змінних, констант, функцій і процедур.

В своїй роботі головна програма використовує низькорівневі API-функції для звернення до ядра ОС Windows. Розглянемо блок-схему роботи її алгоритму, представлену на рисунку 4.1.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

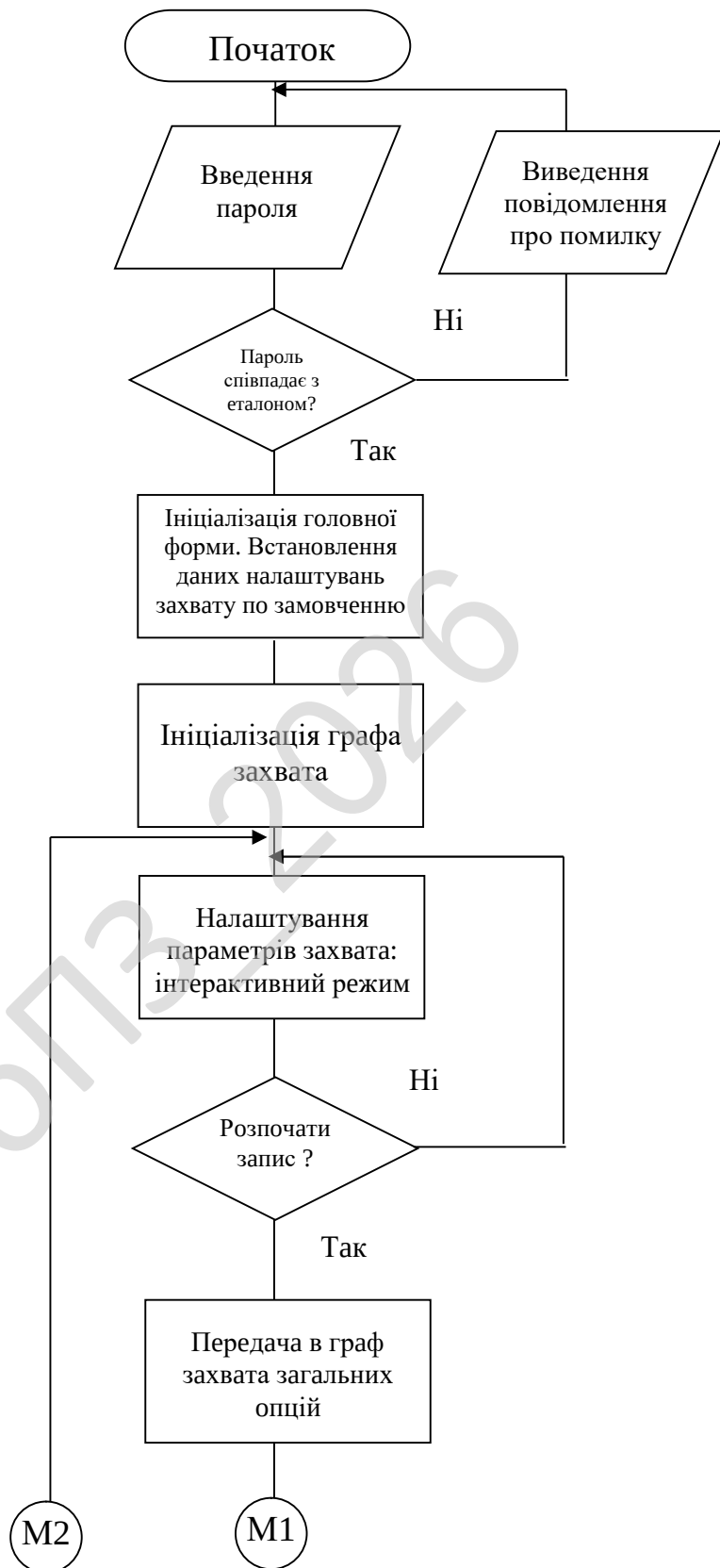


Рисунок 4.1 – Блок-схема алгоритму роботи головної програми

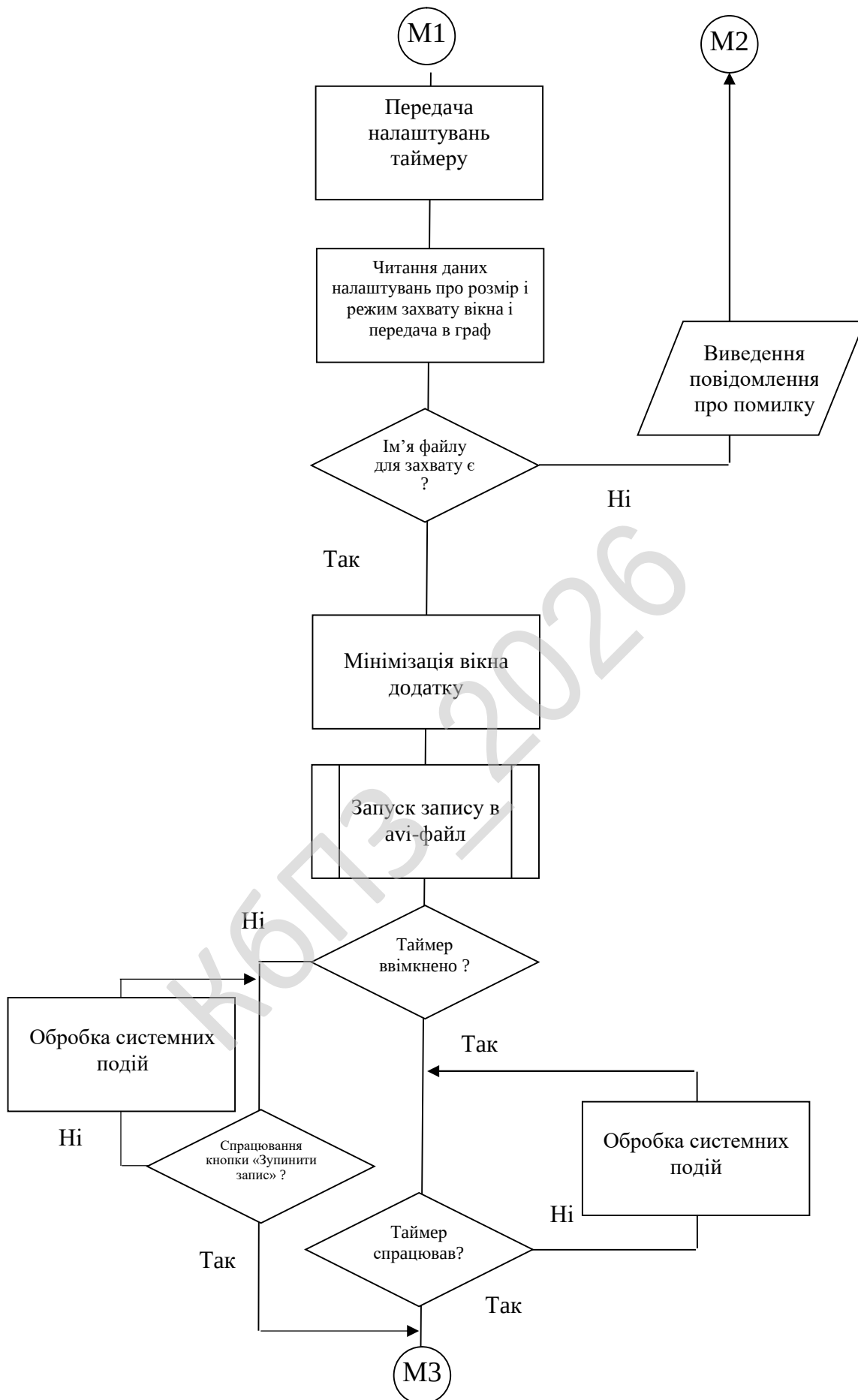


Рисунок 4.1, аркуш 2

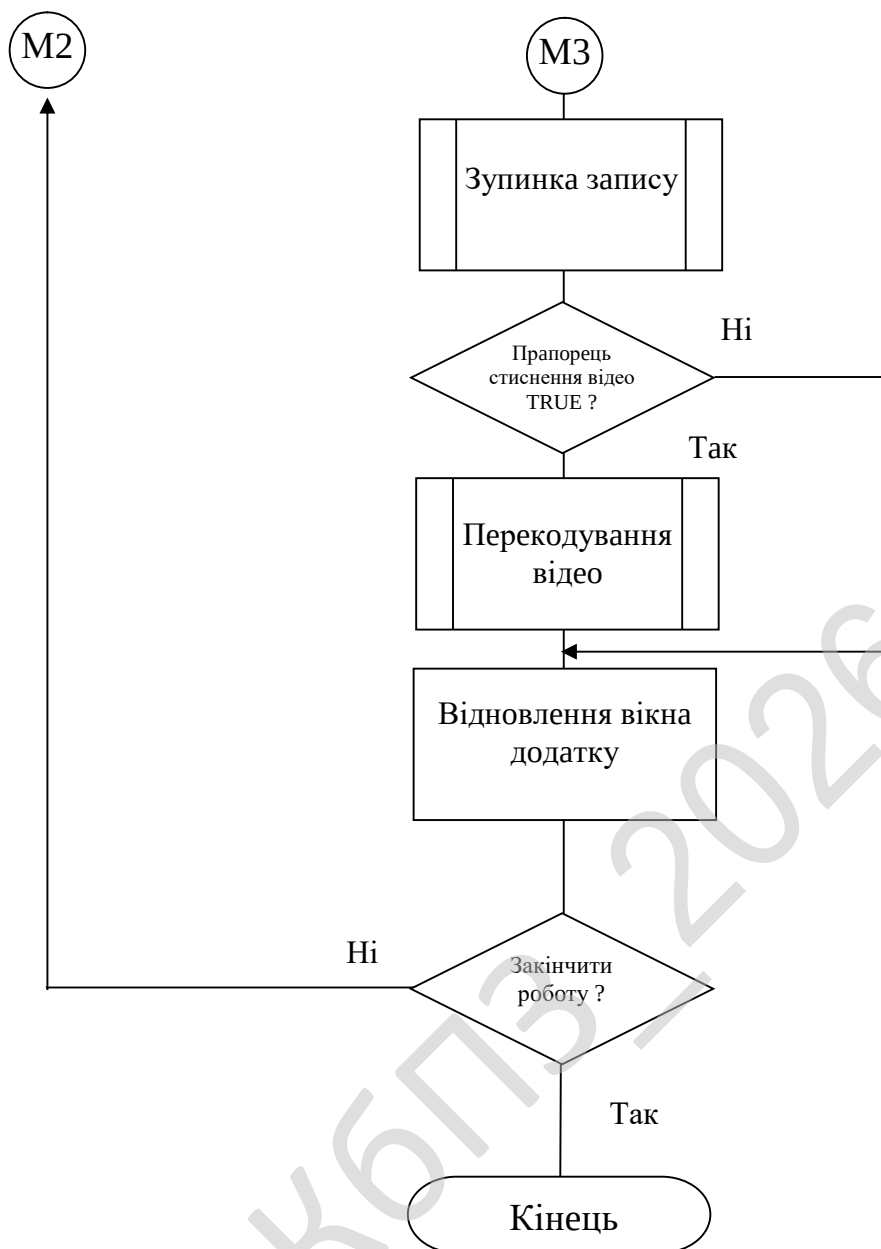


Рисунок 4.1, аркуш 3

В якості прикладу наводимо фрагмент коду зазначеного вище алгоритму:

```

procedure TfrmMain.UpdateForm(Sender: TObject);
//виведення інформації при оновленні форми
begin
  Label1.Caption := 'Інформація :' + #13#10 +
    Format(використовується кодек: %s',
[ScrCam.CurrentVideoCodecName]) + #13#10 +
    Format('Час запису: %s', [ScrCam.ElapsedTime]) + #13#10 +
    Format('Фрейм/S: %f', [ScrCam.Averagefps / 10]) + #13#10 +

```

```

#13#10 +
Format(записаних фреймів: %d', [ScrCam.FramesCaptured]) +
Format(пропущених фреймів: %d', [ScrCam.DroppedFrames]);
end;
procedure TfrmMain.Start(Sender: TObject);
//активація кнопки Стоп
begin
    Button2.Enabled := True;
end;

procedure TfrmMain.Stop(Sender: TObject);
//активація кнопки Старт
begin
    Button2.Enabled := False;
end;

procedure TfrmMain.Preview(Sender: TObject;
    PreviewBitmap: TBitmap; Active: Boolean; Recording: Boolean);
//обробка вікна попереднього перегляду
begin
    if Active then begin
        //якщо вибрано попередній перегляд
        CheckBox8.Checked := True;
        //якщо немає запису, то
        if not Recording then begin
            Label9.Caption := 'Предперегляд... Відмінити: Esc';
        end
        //інакше
        else begin
            Label9.Caption := почати запис... Відмінити: Esc --- Закінчити запис:
Shift+Esc';
        end;
    end
    else begin
        //якщо не пишемо
        if not Recording then
            Label9.Caption := 'Предперегляд вимкнено...'
        else
            Label9.Caption := почати запис... Відмінити: Esc --- Закінчити запис:
Shift+Esc';
        //вимкнути предперегляд
        CheckBox8.Checked := False;
    end;
    //предперегляд співставляємо з компонентою Image1
    Image1.Picture.Assign(PreviewBitmap);
end;

procedure TfrmMain.Saving(Sender: TObject; Percent: Integer; var Continue:
Boolean);
begin
    //обробка системних подій
    Application.ProcessMessages;

    //якщо стан прапорця FCancel False то
    if not FCancel then begin
        //якщо стан кнопки BCancel.Enabled дорівнює False
        if not BCancel.Enabled then begin
            //змінюємо стан кнопки
            BCancel.Enabled := True;
            //змінюємо стан статусу
            LStatus.Enabled := True;
            //виводимо підпис
            LStatus.Caption := 'Йде запис...';
        end;
    end;

```

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

```

Progress.Position := Percent;
//якщо значення стану прогресбара =100
if Percent = 100 then begin
// прапорець переводимо в стан False
FCancel := False;
//змінюємо стан статусу
LStatus.Enabled := False;
//змінюємо стан кнопки
BCancel.Enabled := False;
//виводимо надпис
LStatus.Caption := 'Немає статусу';
//обнуляємо прогресбар
Progress.Position := 0;
end;
//встановлюємо стан прапорця
Continue := True;
end
//якщо стан прапорця FCancel True, то
else begin
//прапорець переводимо в стан False
FCancel := False;
//змінюємо стан статусу
LStatus.Enabled := False;
//змінюємо стан кнопки
BCancel.Enabled := False;
//виводимо надпис
LStatus.Caption := 'Немає статусу';
//обнуляємо прогресбар
Progress.Position := 0;
//встановлюємо стан прапорця
Continue := False;
end;
end;

```

Розроблена система призначена для виконання наступних функцій:

- забезпечення зручної, якісної і наочної форми відеозапису дій користувача, які він здійснив під час роботи на ПК з повної поверхні екрану чи з визначеного ним сегменту, до відеофайлу;
- забезпечення декількох варіантів відеозапису: запис усього, що відбувається на екрані; запис тільки визначеної, заданої користувачем частини; запис відео із звуком з визначеною заданою прямокутною областю на екрані;
- відтворення дій користувача в довільних програмах. Інформація, одержана з відеофайлів, може використатись для: контролю дій власних співробітників, дітей, яким властиво витрачати робочий час на неробочу активність; контроль роботи працівників з конфіденційною інформацією; вирішення деяких задач керування підпорядкованим персоналом фірми чи установи, створення відеоуроків; запис фільмів, відеороликів, тощо.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

- контроль дій зовнішнього провайдера ІТ – послуг.

Перед початком роботи системи по безпосередньому призначенню, користувачу надається можливість провести в інтерактивному режимі налаштування параметрів захвату тих дій на екрані монітору, які він планує виконувати та синхронно записувати.

Можливість налаштування параметрів захвату екрану (screen recording) є обов'язковою функцією для програм для запису відео. Це дозволить оптимізувати якість відео та розмір файлу. Наводимо основні параметри, які можна налаштувати:

- Область захвату: запис всього екрана, окремого вікна або вибраної користувачем області.

- Частота кадрів (FPS): зазвичай від 24 до 60 кадрів в секунду. 60 FPS забезпечує плавне відео для ігор, тоді як 30 FPS достатньо для туторіалів.

- Роздільна здатність (Resolution): налаштування від 720p (HD) до 4K (Ultra HD), що впливає на чіткість зображення та розмір файлу.

- Якість відео та бітрейт: зміна бітрейту дозволяє балансувати між якістю картинки та вагою підсумкового файлу.

- Аудіоджерела: одночасний запис системних звуків, мікрофона або їх відключення.

Налаштування здійснюються через меню "Налаштування" у розробленій програмній системі перед початком запису в інтерактивному режимі. Проте користувачу надається можливість відновити всі попередньо виконанні налаштування в автоматичному режимі (по замовченню).

Для виконання цієї функції розроблено та програмно реалізовано ряд алгоритмів. На рисунку 4.2 наводимо блок-схему роботи алгоритму одного з них - модуля налаштувань параметрів захвату.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

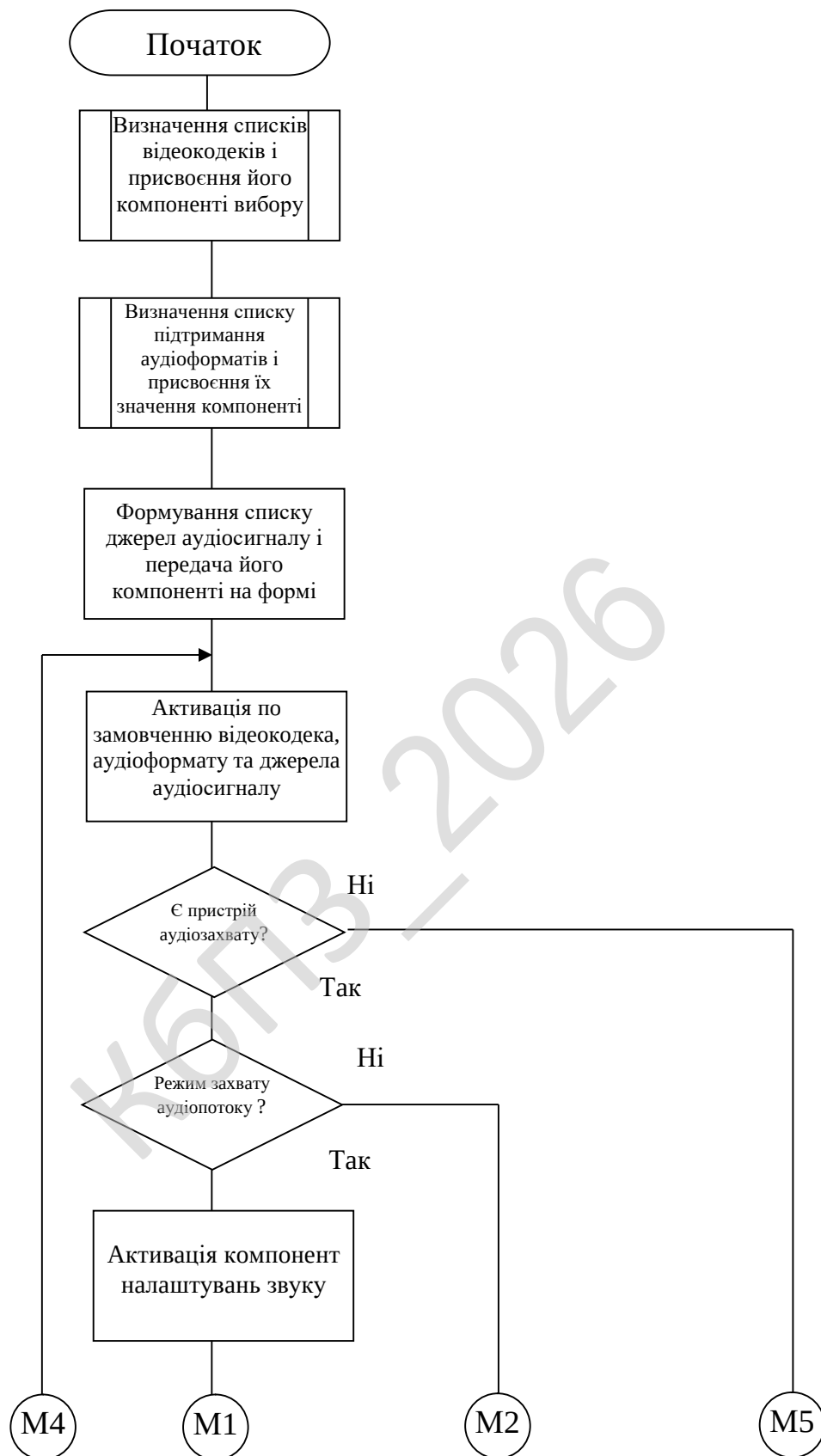


Рисунок 4.2 – Блок-схема алгоритму роботи модуля налаштувань параметрів захвату

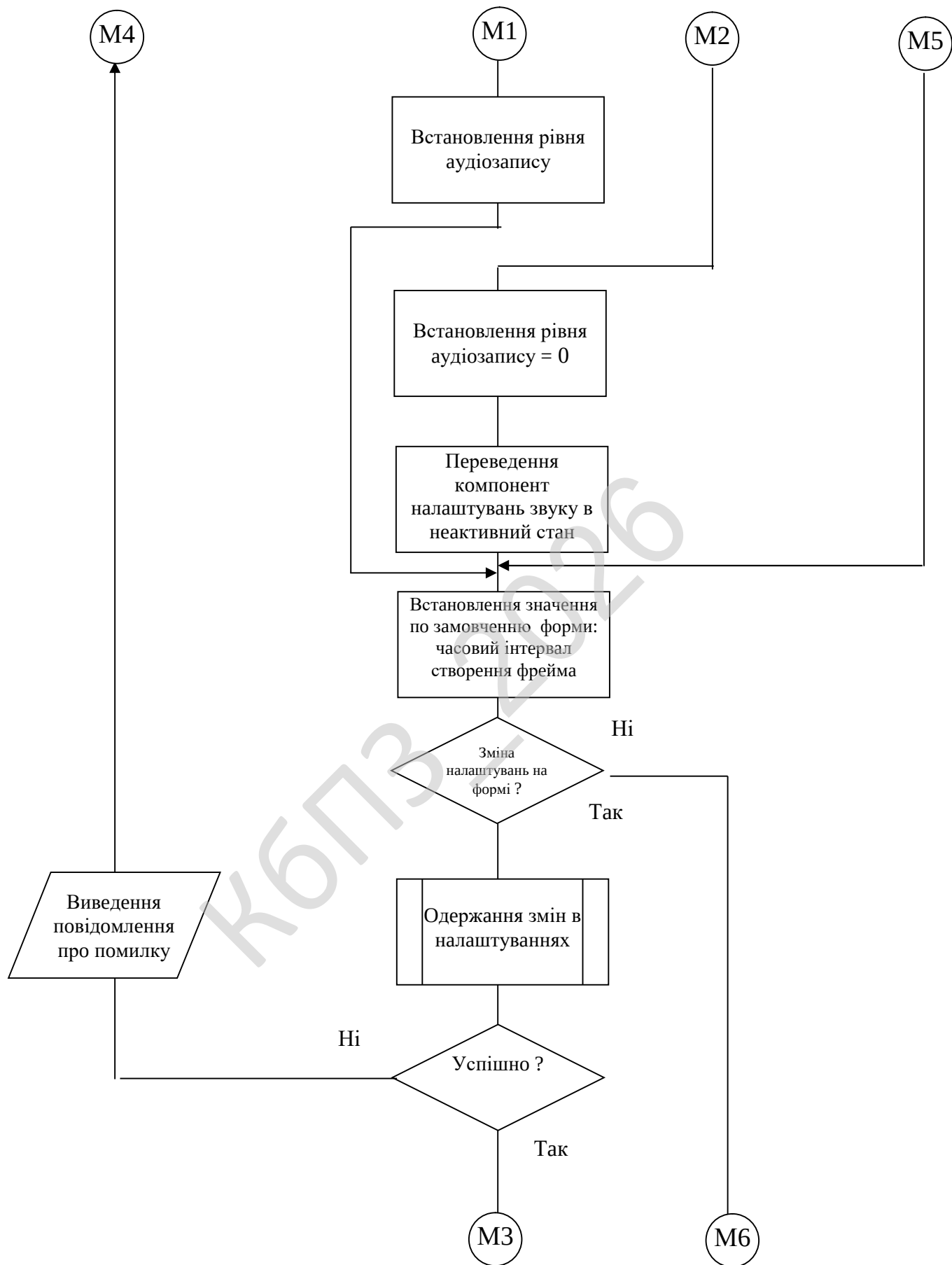


Рисунок 4.2, аркуш 2

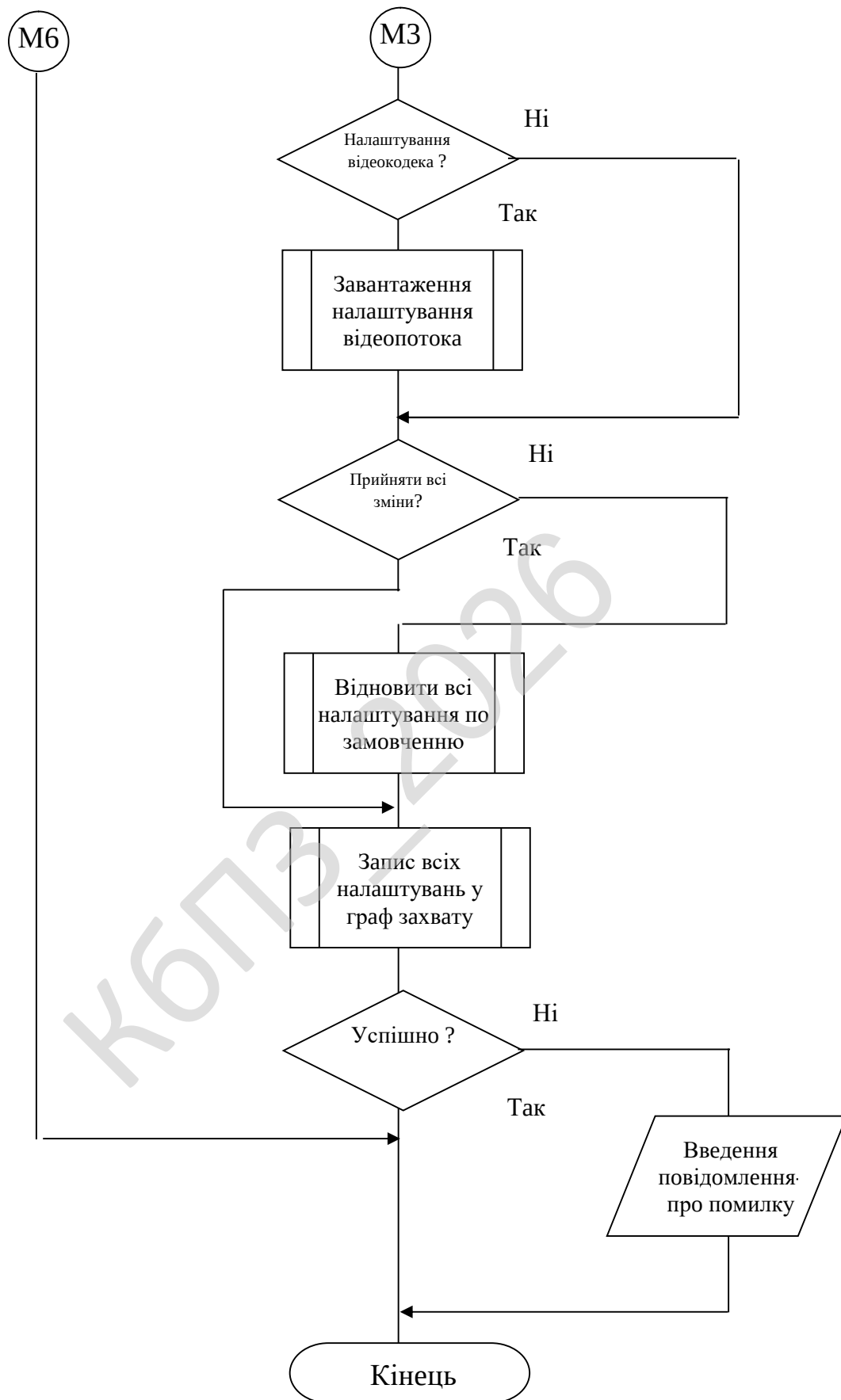


Рисунок 4.2, аркуш 3

Розроблена система працює в автоматичному режимі і може експлуатуватись безперервно чи під керуванням користувача (інтерактивний режим).

При записі екрану монітора (дій користувача) середовищем передачі даних є внутрішня шина комп'ютера, яка передає візуальну інформацію від відеокарти до центрального процесора або безпосередньо у відеопам'ять для подальшої обробки програмним забезпеченням. Це технічний процес захвату, кодування та стиснення зображень (фреймів) у реальному часі.

Технічні інтерфейси підключення монітора, такі як HDMI, DisplayPort або USB-C, не є середовищем передачі даних для самого запису, оскільки вони призначені для виведення зображення назовні до екрана. Проте, кабель USB-C може одночасно передавати відеосигнал на монітор і дані назад до ПК (наприклад, при використанні сенсорного екрана).

В наведених основних модулях програмного забезпечення підтримується принцип інформативності, а саме: надання користувачу інформації про виконання встановлених перевірок та результати роботи окремих програм та підпрограм, надання (в разі необхідності) теоретичних відомостей щодо принципів роботи системи, необхідних для роботи технічних параметрів, тощо. Інформаційні повідомлення в процесі роботи системи виводяться на екран монітора чи роздруковуються (по бажанню користувача).

Таким чином, завдання визначене ТЗ та постановкою задачі, виконане в повному обсязі. Основні етапи розробки ПЗ системи включають:

- розробку структурної і функціональної схем системи для визначення та обґрунтування вибору структурних частин програмної моделі ті їх функцій;
- розробку ПЗ системи відповідно принципам та методиці реалізації систем аналогічного класу та спрямування;
- теоретичне обґрунтування розробки основних рішень побудови програмної моделі та виконання необхідних розрахунків;
- розробка алгоритмів та ПЗ системи.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

Розроблене ПЗ має модульну структуру, що забезпечує виконання системою принципів мобільності та адаптивності і цілком дієздатне. Про це свідчать результати дослідної експлуатації. Розроблено досить потужну систему з додержанням усіх вимог до систем даного класу, яка в подальшому може вдосконалюватись шляхом введення нових функцій – сервісних та робочих, елементів експертних систем та штучного інтелекту.

Програмне забезпечення системи працює на платформі ОС Windows. При цьому вона пред'являє мінімальні вимоги до системних ресурсів, адже нею використовуються тільки ті процедури введення/виведення, які необхідні в роботі. Для зручності користувача використовуються звичні для нього панелі інструментів і меню та налаштування за допомогою ієрархічного дерева в стилі провідника ОС Windows.

Для більш повної ілюстрації роботи системи розроблено та виконано 6 листів графічного матеріалу формату А4, до складу якого ввійшли: структурна схема; функціональна схема; діаграма процесів; блок-схеми роботи алгоритмів основної програми та підпрограм, які найбільш ефективно характеризують та ілюструють розроблене ПЗ.

4.2 Захист розробленого програмного забезпечення

На сьогоднішній день реальний захист авторських прав на комп'ютерні програми можна забезпечити лише встановленням засобів програмного захисту. Задача розробника ПЗ полягає в тому, щоб користувач, встановивши піратську копію ПЗ на комп'ютер, не зміг би використовувати її внаслідок того, що спрацюють засоби програмного захисту від нелегального копіювання. Це – задача програмістів, що повинна бути реалізована на етапі розробки ПЗ або на початку промислової експлуатації системи на підприємстві (у організації).

При розробці програми захисту інформації було вирішено організувати санкціонований доступ до роботи з системою.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Перш, ніж отримати дозвіл на роботу з системою, користувач-абонент повинен на вимогу системи назвати свій пароль. Пароль встановлюється розробником системи або системним адміністратором організації-користувача. Еталон пароля розміщується в базі даних в спеціальному файлі. При введенні пароля власне користувачем, проводиться його порівняння з еталоном. В результаті цієї процедури можливі два випадки:

- пароль співпадає з еталоном;
- пароль не співпадає з еталоном.

Наслідок: система в обох випадках реагує однаково – повторно пропонує ввести пароль для підтвердження права доступу до системи.

У першому випадку – це ще один ступінь захисту системи, тобто користувач знає, що код доступу він повинен ввести п-у кількість разів (кількість введень встановлюється системним адміністратором) і якщо код доступу вірний – отримає право роботи з системою.

Якщо пароль не співпадає з еталоном, то після повторного неправильного його введення, версія ПЗ, що знаходиться на даному комп'ютері, буде знищена. Такі жорсткі дії були введені для захисту програмного забезпечення від “злому” встановленого захисту.

Цього ступеню захисту розробленого ПЗ на даному певному етапі цілком достатньо. Складніший захист може бути встановленим у разі виникнення попиту на розроблений програмний продукт.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Згідно з ТЗ в процесі роботи над випускною кваліфікаційною роботою за першим (бакалаврським) рівнем вищої освіти було розроблене програмне забезпечення системи відеозапису послідовності дій користувача на екрані ПК на основі використання API-функцій.

Впровадження системи відеозапису дій користувача на ПК з використанням API-функцій - це технічний процес, що базується на захваті зображення екрана (frame grabbing), стисненні та збереженні у відеофайл. Методика впровадження передбачає вибір відповідного API, налаштування середовища розробки, реалізацію логіки захвату та обробку дій користувача, які він виконує впродовж робочого часу на екрані ПК.

Впровадження системи відеозапису дій користувачів (Screen Monitoring/User Activity Monitoring - UAM) на ПК є складним процесом, який охоплює технічні, юридичні та організаційні аспекти. Така система дозволяє фіксувати екран, натискання клавіш, запуск програм, час роботи тощо.

Представляємо основні етапи впровадження системи:

1. Підготовчий етап та юридичне оформлення;

- Визначення мети: чітко окреслити навіщо впроваджується система (безпека даних, контроль продуктивності, розслідування інцидентів тощо).

- Юридична відповідність: внесення змін до посадових інструкцій або підписання згоди працівників на моніторинг робочого місця, щоб уникнути порушень трудового законодавства та права на приватність особистих даних.

- Розробка положення: створення внутрішнього наказу "Про використання засобів відеофіксації та моніторингу".

2. Технічна підготовка системи:

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

- Підготовка інфраструктури: встановлення відеореєстратора (сервера), забезпечення джерел живлення та місця для зберігання відеоархіву.

- Встановлення агентів: встановлення клієнтського ПЗ на комп'ютери користувачів, яке буде записувати екран у прихованому або відкритому режимі.

3. Введення в експлуатацію:

- Тестування: перевірка коректності запису та збереження даних.

- Тестування навантаження: перевірка впливу на продуктивність (завантаження CPU < 10% є критичним).

- Навчання персоналу: ознайомлення працівників з фактом моніторингу.

- Розгортання: впровадження через групові політики (GPO) у корпоративній мережі.

- Початок фіксації: запуск системи у робочому режимі. Записи дозволяють відстежувати час входу/виходу, активність на робочому столі та використані ресурси

- 4. Облік та обслуговування: систему відеоспостереження/моніторингу зазвичай обліковують як окремий об'єкт основних засобів, обліковуючи витрати на придбання та налагодження як незавершені капітальні інвестиції до моменту вводу в експлуатацію. Впровадження таких систем має відбуватися відповідно до законодавства України, зокрема з урахуванням захисту персональних даних.

Введення в експлуатацію любого побічного інструментарію, (тобто, розробленої системи), супроводжується певними складнощами. Вони пов'язані з тим, що необхідно розширити коло обов'язків супровідників ПЗ та служби технічної підтримки. Персоналу, що буде супроводжувати ПЗ, необхідно вивчити функції системи, представлені наявною технічною документацією на рівні, що буде достатнім для її безперебійної роботи та забезпечення виконання означених задач інтеграції в новому ПЗ.

Інтеграція системи в існуючу інфраструктуру передбачає сумісність із іншими платформами, наявність API-функцій для розширення функціональності, централізоване управління та масштабованість для адаптації до змін у мережі.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Відсутність належної інтеграції може призвести до прогалин у безпеці та конфліктів між компонентами.

Інтеграція також вимагає тісної взаємодії з системами управління ідентифікацією та доступом для забезпечення чіткої сегментації прав доступу. Це дозволяє мінімізувати ризики, пов'язані з людським фактором, і підвищити прозорість доступу до критичних активів.

В якості ілюстрації роботи системи по безпосередньому призначенню наводимо скриншоти ПЗ (рисунок 5.1, рисунок 5.2).

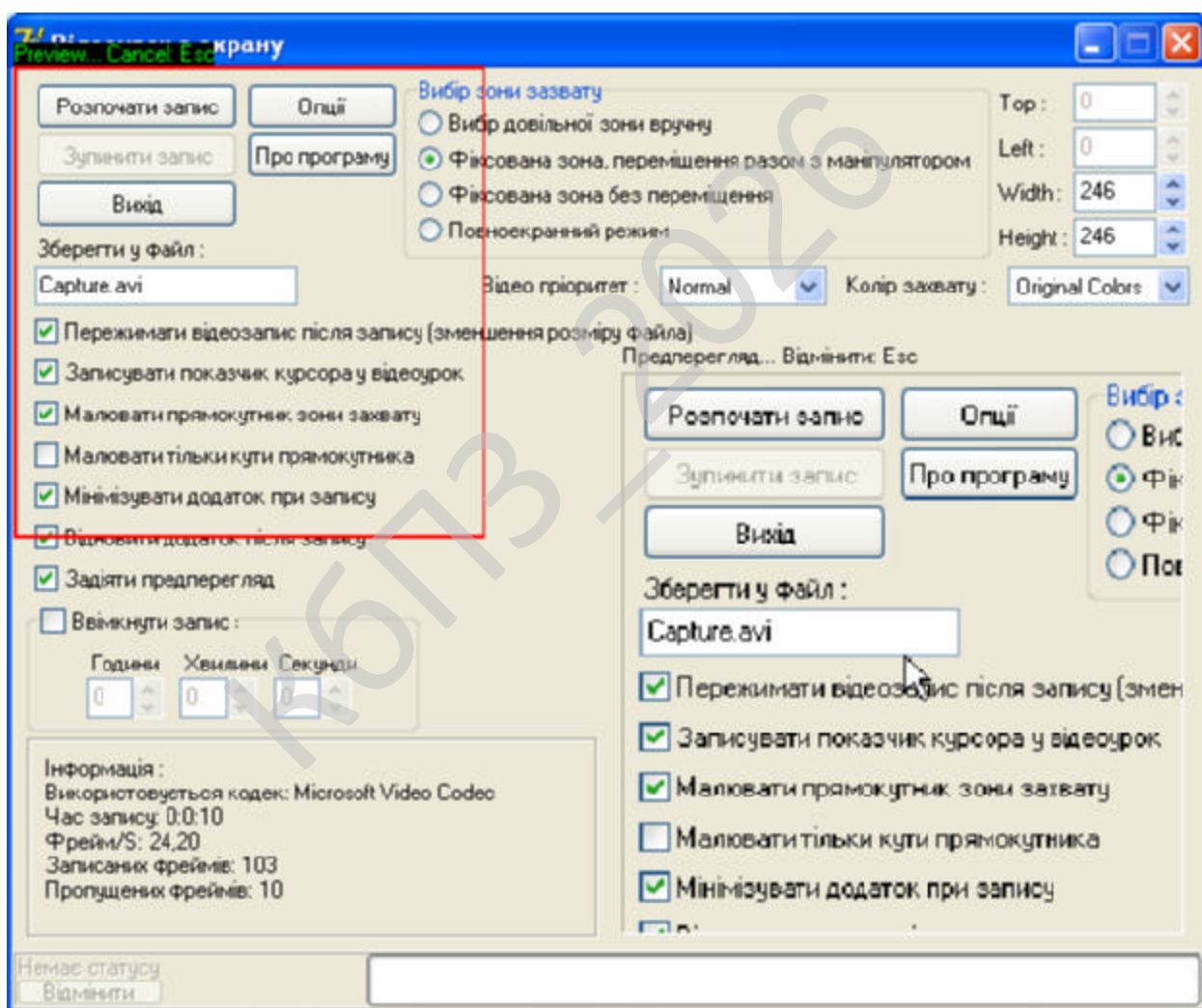


Рисунок 5.1 – Скриншот головного вікна інтерфейсу користувача

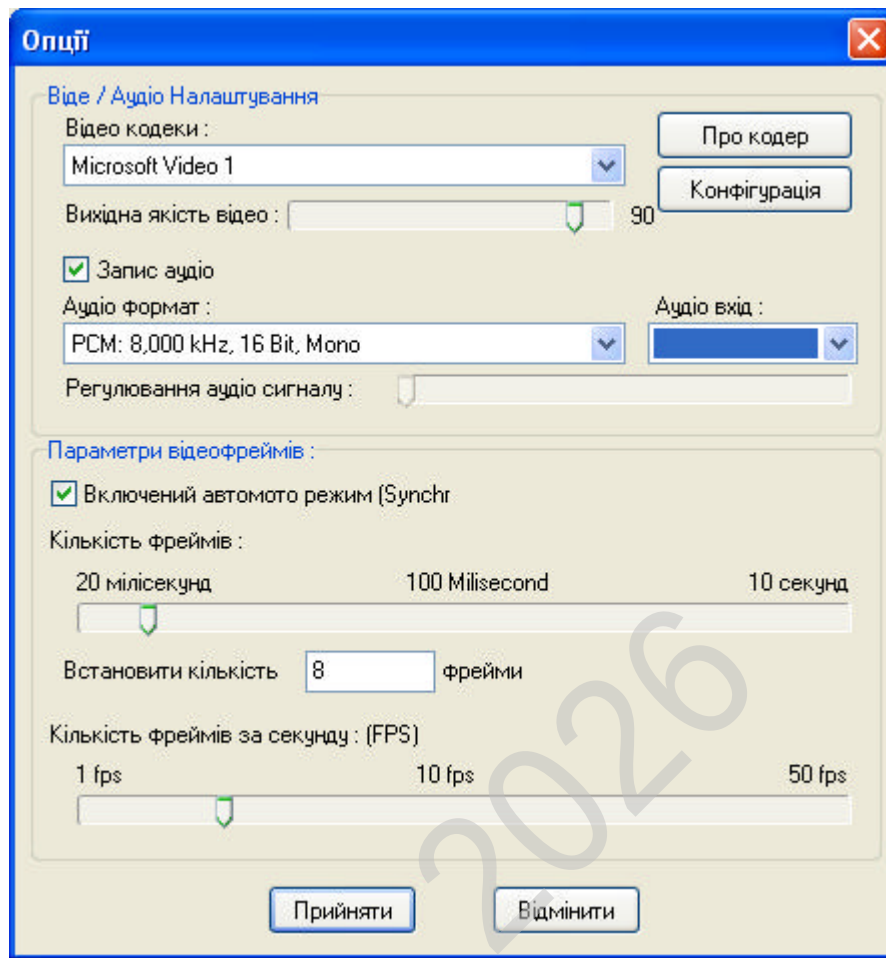


Рисунок 5.2 – Скриншот опцій налаштувань модуля налаштувань параметрів захвату фреймів

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначене для системи відеозапису послідовності дій користувача на екрані ПК на основі використання API-функцій.

Вітчизняні розробки програмних систем аналогічного класу та спрямування в межах України представлені не в достатній мірі, хоча на сьогодні це питання є надзвичайно актуальним.

Виконання завдання на розробку ПЗ системи полягало у вирішенні наступних задач:

- проведення огляду наявних на світовому ринку програмних продуктів систем – аналогів відеозапису послідовності дій користувача на екрані ПК на основі використання API-функцій;
- дослідження механізму відеозапису послідовності дій користувача на екрані ПК на основі використання API-функцій;
- виконання на основі отриманих результатів досліджень програмної реалізації системи відеозапису послідовності дій користувача на екрані ПК на основі використання API-функцій.

Створені у процесі виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми, дозволяють успішно вирішувати актуальні задачі не лише виконання обчислювальних процесів, а й детального аналізу взаємодії користувача з програмним середовищем.

Розроблена система:

- є апаратно невибагливою, що дає можливість працювати як на одному, окремо взятому ПК, так і на складній комбінації локальних мереж, що працюють на значній відстані одна від одної;

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

- дозволить зберегти раніше зроблені капіталовкладення при переході на більш продуктивний варіант апаратної платформи;
- забезпечує можливість роботи в режимі реального часу;
- забезпечує обмін даними з іншими системами КМ;
- надійно захищена від несанкціонованого доступу;
- має прийнятну вартість для мінімальної конфігурації комплексу технічних засобів і можливість використання обладнання, відповідного сучасним технічним досягненням.

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.

В якості середовища розробки ПЗ системи використане середовище швидкої розробки Delphi. Це одне з найбільш потужних візуально об'єктно-орієнтованих середовищ програмування, яке дозволяє на найсучаснішому рівні створювати розгалужені програмні комплекси, призначені для роботи в комп'ютерних мережах та в мережах Інтернет. Дана мова програмування дозволила ефективно обробити дані для системи та мінімізувати терміни розробки ПЗ. Розроблене ПЗ працює під керуванням ОС Windows.

Надаються необхідні рекомендації з розгортання та налаштування розробленого ПЗ, проведення дослідної експлуатації та подальшої промислової експлуатації.

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

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Axelrod. Complete Guidance, to Check Automation: Technique, Practices, and Standards for Building and Support of Effective Projects / of Software. Axelrod: Apress, – 2018. - 588 p.
2. Toledo T., Beinhaker R. Evaluation of the potential benefits of advanced traveler information systems// Journal of Intelligent Transportation Systems: Technology, Planning, and Operations. Vol. 10(4), 2006. – 173-183 p.
3. Babii, M., Tson, O., Kuchvara, I., & Chernii, V. Improving the efficiency of the road organization traffic at an unregulated crossroads. Transport Development, 2021, (1(8)), – 125-134 p.
4. . Rozhko N, Plekan U., Tson O., Matviishyn A. Digitalization of truck companies: current challenges and development prospects // Central Ukrainian Scientific Bulletin. Technical Sciences. – 2022. – Col.6(37). – 208-214 p.
5. Kumar A., Jaiswal J., Tiwari N. Blind Spot Monitoring System Using Ultrasonic Sensor. International Conference on Disruptive Technologies (ICDT). 11-12 May. 2023. Greater Noida, India. – 30-35 p.
6. Yang Z. Research on Causes of Rear-End Collision and Behavior of Avoiding Collision Based on Deep Data Analysis : Ph.D. Thesis, Northeast Forestry University. Harbin, China, 2021. – 118-127 p.
7. Cicchino J.B., Zuby D.S. Characteristics of rear-end crashes involving passenger vehicles with automatic emergency braking. Traffic Inj. Prev. 2019. Vol.20, No 1. – 112–118 p.
8. Fanca A., Puscasiu A., Folea S., Vălean H. Trauma accident detecting and reporting system. In Proceedings of the 2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR). 24–26 May 2018. Cluj-Napoca, Romania. 2018. – 1–5 p.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

9. Balasubramanian V., Otoum S., Aloqaily M., Y. Jararweh Y. Low-latency vehicular edge: A vehicular infrastructure model for 5G. Simul. Model. Pract. Theory, 2020. Vol. 98, No 7. – 119-132 p.
10. Gilly K., Filiposka S., Alcaraz S. Predictive migration performance in vehicular edge computing environments. APL Sci. 2021. Vol. 11, No 3. – 944 p.
11. Samani A., Sabyasachee M., Kakan D. Assessing the effect of long-automated driving operation, repeated take-over requests, and drivers' characteristics on commercial motor vehicle drivers' driving behavior and reaction time in highly automated vehicles. In Transportation Research Part F: Psychology and Behaviour. 2022. Vol. 84, No 8. – 239-261 p.
12. Batyrgareieva V. S., Shramko S. S., Samoilova O. M. Mortality and injury in Ukraine as a result of traffic accidents in measuring of public health: to the analysis of social–legal and criminological problem. Wiadomości Lekarskie. Vol. LXXIV. Is. 11. Part 2. November 2021. – 2870-2876 p.
13. Richards M. Fundamentals of Software Architecture: An Engineering Approach / M. Richards, N. Ford. – Sebastopol: O'Reilly Media, 2020. – 432 p.
14. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language / M. Fowler. – Boston: Addison-Wesley Professional, 2018. – 191 p.
15. Rumbaugh J. UML 2.0. Object-Oriented Modeling and Design with UML / J. Rumbaugh, M. Blaha. – London: Pearson, 2012. – 496 p.
16. Lamdakkar O., Ameer I., Mbarek Eleyatt M., Carlier F., Ait Ibourek L. Toward a modern secure network based on next-generation firewalls: recommendations and best practices. Procedia Computer Science. Vol. 238. 2024. – 1029–1035 p. DOI: <https://doi.org/10.1016/j.procs.2024.06.130>.
17. . Aho A.V., Hopcroft J.E., Ullman J.D. Data Structures and Algorithms. – Pearson, 2001. – 620 p.
18. Beller M., Gousios G., Panichella A., Proksch S., Amann S., & Zaidman A. (2017). Developer testing in the ide: Patterns, beliefs, and behavior. IEEE Transactions on Software Engineering, 45(3), – 261-284 p.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

19. Stamp M. Information Security: Principles and Practice, 2nd ed. Wiley-IEEE Press, 2011.- 121 p.
20. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.
21. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
22. Динамічна відеофіксація транспортних засобів для системи моніторингу дорожнього руху у мегаполісі / N. Kuchuk та ін. Системи управління, навігації та зв'язку. Збірник наукових праць. 2022. Т. 2. – 55–58 с
23. Моніторинг дорожньої обстановки: Навчальний посібник / В.Ф. Ступак, В.В. Задорожний, В.В. Яковчук – Львів: Видавництво Львівської політехніки, 2020 – 128 с.
24. Кір'янов О. Ф., Мороз М. М., Бойко Ю. О. Інформаційні технології на автомобільному транспорті : навч. посіб., Кременчуцький нац. ун-т ім. Михайла Остроградського. Харків : Дркарня Мадрид, 2015. 270 – с.
25. Краснодемський В. Безпека на дорозі – проблема національна. Голос України. Київ, 11.05.2011, – с. 5
26. Безвідний В.В. Ідентифікація транспортних засобів за допомогою відеокамер. Збірник наукових праць за матеріалами II Міжнародної науково-практичної конференції «Комп'ютерні технології і мехатроніка», Харків, ХНАДУ, 2020, – 198-201 с.
27. Симбірський Г.Д. Аналіз методів виявлення аномалій у відеоряді камер відеоспостереження на транспорті. Вісник ХНАДУ, зб. наук. праць, Харків, ХНАДУ, 2022, № 98, – 26-37 с.
28. Стрельнікова І.А., Артемова Ю.А. Супутникові системи навігації і моніторингу транспорту // Автомобіль і Електроніка. Сучасні Технології. Електронне наукове фахове видання. – Харків, 2013.-№;-с. 56-63.Алтуєв М. Перспективи розвитку цифрового CCTV. Думки фахівців // Алгоритм Безпеки. – 2004. – № 5. - 57с.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

29. Аулін В. В., Гриньків А. В. Методологічні основи проектування та функціонування інтелектуальних транспортних і виробничих систем: монографія. Кропивницький: Видавець Лисенко В.Ф., 2020. – 428 с.

30. Парасюк В. М., Демків Р. Я., Когут В. М. Безпека дорожнього руху : навчальний посібник. Львів : Львівський державний університет внутрішніх справ, 2022. – 340 с.

31. Захарченко А. М., Алфьоров С. М. Щодо обмеження руху транспортних засобів в умовах воєнного стану. Безпека дорожнього руху в умовах воєнного стану: матер. Всеукр. Наук.-практ. онлайн-конф. (в автор. ред.), (м. Кривий Ріг, 27 трав. 2022 р.). Кривий Ріг: Донецький державний університет внутрішніх справ, 2022. – 55-57 с.

32. Печук І. С. Кримінально-правова характеристика суб'єкта злочину в складі порушення правил безпеки дорожнього руху або експлуатації транспорту особами, які керують транспортними засобами в Україні. Право і суспільство. 2015. № 1. – 224–230 с.

33. Батиргареева В. С. Дорожній рух у прифронтовому регіоні: виклики воєнного стану. Безпека дорожнього руху в умовах воєнного стану: матеріали Всеукраїнської науково-практичної онлайн-конференції (в авторській редакції), (м. Кривий Ріг, 27 травня 2022 року). Кривий Ріг, 2022. 148 с. Моніторинг дорожньої обстановки: Навчальний посібник / В.Ф. Ступак, В.В. Задорожний, В.В. Яковчук – Львів: Видавництво Львівської політехніки, 2020 – 128 с.

34. Динамічна відеофіксація транспортних засобів для системи моніторингу дорожнього руху у мегаполісі / N. Kuchuk та ін. Системи управління, навігації та зв'язку. Збірник наукових праць. 2022. Т. 2. 55–58 с.

35. О.М. Поліщук, В.В. Гурін – К.: Національний технічний університет Моніторинг інтенсивності руху на дорогах: Навчальний посібник / В.М. Мельник, України "Київський політехнічний інститут імені Ігоря Сікорського", 2022 – 108 с.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

36. «Про Стратегія інформаційної безпеки України» Указ Президента України: Стратегія від 28.12.2021 № 685/202. URL: <https://zakon.rada.gov.ua/laws/show/685/2021#Text>.

37. «Про Стратегію національної безпеки України» Указ Президента України: Стратегія від 14.09.2020 № 392/2020. URL: <https://zakon.rada.gov.ua/laws/show/392/2020#Text>.

38. Новіков Ф.А. Навчально-методичний посібник з дисципліни Технологічні підходи до розробки програмного забезпечення/ Ф.А. Новіков. – СПб.: СПбГУ ІТМО, 2007. – 137 с.

39. Гагаріна Л. Технології розробки програмного забезпечення: навчальний посібник / Л. Гагаріна, Є. Кокорева, Б. Віснадул. – М.: ВБ «ФОРУМ»: ІНФРА-М, 2008. – 400 с.

40. Безменов М. І. Основи програмування у середовищі Delphi: навч. посіб. – Харків: НТУ «ХП», 2010. – 608 с.

41. Короткий курс програмування в середовищі Delphi (для студентів заочної форми навчання спеціальності 7.050201 “Менеджмент організацій” спеціалізації “Інформаційні системи в менеджменті”). Укл. Г.А. Мірошніченко. - Харків: ХНАМГ, 2004. - 81 с.

42. Сопілко І. М. Інформаційна безпека та кібербезпека: порівняльноправовий аспект// Юридичний вісник Повітряне і космічне право. – Київ: НАУ, 2021. – № 2(59). – 110-115 с.

43. Галіпчак В. Інформаційна війна як складова гібридної війни в умовах російської агресії. Вісник Прикарпатського університету, 2023. Вип. 15. – 26-32 с.

44. Чмир Я. Сучасні проблеми інформаційної безпеки України та перспективні напрями їх вирішення». Наукові праці Міжрегіональної Академії управління персоналом. Політичні науки та публічне управління, 2022. №2(62). – 149-154с.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

45. Коваленко О.В. Моделі та методи розроблення програмного забезпечення комп'ютерних систем для підвищення безпеки даних: монографія / О.В. Коваленко // К.: Вид. «КОД» – 2019. – 305 с.

46. Коваленко А.В. Методи якісного аналізу і кількісної оцінки ризиків розробки програмного забезпечення / А.В. Коваленко, А.А. Смирнов // Інформаційні технології в управлінні, освіті, науці і промисловості: монографія / Під редакцією професора В. С. Пономаренко. - Х.: Видавець Рожко С. Г., 2016. - 566 с.

47. Коваленко А.В. Розробка методу управління ризиками розробки програмного забезпечення / А.В. Коваленко, А.А. Смирнов // Інформаційні технології : проблеми та перспективи : монографія / За загальною редакцією В. С. Пономаренка. - Х.: Видавець Рожко С. Г., 2017. - 447 с.

48. Коваленко О.В. Методи якісного аналізу та кількісної оцінки ризиків розроблення програмного забезпечення/ О.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 3 (49). – Полтава: ПолтНТУ. – 2018. – С. 116-125.

49. Коваленко О.В. Методи та засоби управління безпекою додатків / О.В. Коваленко // Інформаційно-керуючі системи на залізничному транспорті. №4, 2018. – С. 41-44.

50. Коваленко О.В. Розробка інформаційної технології передтестової компіляції та розподілу доступу / О.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 4 (50). – Полтава: ПолтНТУ. – 2018. – С. 115-119.

51. Коваленко О.В. Аналіз та дослідження інформаційних технологій розробки програмного забезпечення/ О.В. Коваленко // Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 29 (68) № 5, 2018. – С. 131-137.

					ВКРБ-123.26.0001.00.00. ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83