

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ ЗАКЛАД
„ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА”

Навчально-науковий інститут фізики, математики та інформаційних
технологій

Кафедра інформаційних технологій та систем

Бородін Микита Сергійович

**СИСТЕМА АВТОМАТИЗАЦІЇ ВИБОРУ
МУЛЬТИМЕДІЙНОГО КОНТЕНТУ ДЛЯ ПУТІВНИКА**

Бакалаврська робота
за напрямом підготовки 121 Інженерія програмного забезпечення

Особистий підпис – _____ Бородін М.С.

Науковий керівник – _____
(підпис)

доцент кафедри ІТС
Г.А. Могильний
(посада, науковий ступінь,
наукове звання, ініціали, прізвище)

Зав. кафедри – _____
(підпис)

зав. кафедри ІТС, кандидат
педагогічних наук, доцент,
М.А. Семенов
(посада, науковий ступінь,
наукове звання, ініціали, прізвище)

Полтава – 2023

Міністерство освіти і науки України
Державний заклад „Луганський національний університет
імені Тараса Шевченка”

Факультет (інститут)

Навчально-науковий інститут фізики,
математики та інформаційних технологій

Кафедра, циклова комісія

Інформаційних технологій та систем

Освітній ступень

Бакалавр

Напрямок підготовки (спеціальність)

121 «Інженерія програмного забезпечення»
(код, назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТС

М.А. Семенов

(підпис)

(ініціали, прізвище)

“ ”

2022 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Бородіна Микити Сергійовича

(прізвище, ім'я, по батькові)

**1. Тема проекту (роботи) Система автоматизації вибору мультимедійного
контенту для путівника**

Керівник кваліфікаційної роботи

Могильний Г.А.

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

затверджена наказом по університету

Від“ ” 2022 року №

2. Строк подання студентом проекту (роботи)

3. Вихідні дані до роботи (проекту)

У результаті виконання роботи

проекту) повинна бути розроблена програмна система, яка автоматизує вибір
, перейменування, копіювання файлів з фотоапарату та створення HTML коду.
мапи для додатку додатка «Віртуальний путівник»

(визначаються кількісні або (та) якісні показники, яким повинен відповідати об'єкт розробки)

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

МОВА HTML, ПРОГРАМУВАННЯ МАПИ МАЛЮНКА У HTML,

ДОДАТОК JAVA FX, ОПИС ДОДАТКУ, МУЛЬТИМЕДІЙНІ

МОЖЛИВОСТІ JAVA

(визначаються назви розділів або (та) перелік питань, які повинні увійти до тексту ПЗ)

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

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання „_____” _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
	Вибір теми роботи, вивчення наукової літератури, затвердження теми та керівника.	До 15 жовтня	
	Аналіз літературних джерел за темою роботи. Розробка та апробація методики дослідно-експериментальної роботи. Подання структури теоретичної частини роботи та плану експериментальних досліджень.	Другий тиждень листопада (10 листопада)	
	Робота над теоретичною частиною. Подання теоретичної частини роботи для першого читання науковим керівником.	До 15 грудня	
	Усунення зауважень, урахування рекомендацій наукового керівника. Подання теоретичної частини роботи на друге читання.	До 28 січня	
	Проведення експериментальної роботи. Поетапний аналіз та обговорення її результатів. Перевірка стану виконання роботи.	Перший тиждень березня	
	Урахування рекомендацій наукового керівника, усунення недоліків, підготовка варіанта роботи до передзахисту. Розробка презентації.	До 31 березня	
	Попередній захист роботи на кафедрі	квітень	
	Доопрацювання роботи з урахуванням рекомендацій після передзахисту. Подання роботи науковому керівникові та рецензентові на підготовку відгуку та рецензії	За 10 днів до державної атестації	
	Подання на кафедру остаточного варіанта роботи, переплетеного та підписаного автором, науковим керівником і рецензентом.	За 5 днів до державної атестації	

Студент

підпис

М.С Бородін.

(ініціали, прізвище)

Керівник проекту (роботи)

підпис

Г.А. Могильний

АНОТАЦІЯ

Бородін Микита Сергійович

Тема: Система автоматизації вибору мультимедійного контенту для путівника.

Спеціальність: 121 "Програмна інженерія"

Установа: ЛНУ імені Тараса Шевченка, 2023р.

Бакалаврська робота містить: загальна кількість сторінок 90, з них 68 – Пояснювальна записка, 1– відомості проекту, 8 –Технічне завдання, 12 – Методика тестування, 26 рис., 1 додат., 17 джерел. Робота містить анотації та завдання на кваліфікаційну роботу

Об’єкт дослідження –додатки автоматизації створення тегу «мапа» у мові HTML.

Предмет дослідження –додатки та утиліти обробки мультимедійних файлів.

Мета роботи – розробка спеціалізованого додатку для автоматизації процесу попереднього перегляду, вибору мультимедійних файлів з можливістю підготовки мультимедійного контенту програми «Путівник».

Результати роботи. Проведено аналіз особливостей використання програми «Путівник», спеціалізованого тегу «мапа» мови HTML та технологій JavaFX. Розроблені класи для роботи перегляду, вибору та копіювання/перейменування мультимедійних файлів, а також класи для автоматичного створення малюнків та тега «мапа» програми «Путівник». Розроблено технічне завдання на програмний комплекс, наведено опис процесу його моделювання та тестування.

Висновок. В результаті розробки було отримано додатковий сервісний програмний комплекс для створення мультимедійного контенту додатка «Путівник».

Ключові слова. АВТОМАТИЗАЦІЯ, УТИЛІТИ, КЛАС, МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, мова HTML, JAVA FX, ТЕГ “MAP”, ТЕГ AREA

ABSTRACT

Borodin M.S.

Theme: Automation system for selecting multimedia content for the guide.

Speciality: 121 "Software Engineering"

Institution: Luhansk Taras Shevchenko National University (LTSNU), 2023.

Diploma work contains: total number of pages 90, of which 68 Explanatory note, 1 – project information, 8 – Technical task, 12 – Testing methodology, 26 fig., 2 appendices, 17 sources. The work contains annotations and tasks for qualification work.

A research object is applications to automate the creation of the tag "map" in HTML.

The article of research is process design automation plates, including modeling and analysis of stress-strain state of these structures.

An aim of work is applications and utilities for processing media files.

Job performances. An analysis of the peculiarities of using the program "Guide", a specialized tag "map" of HTML and JavaFX technologies. Developed classes for viewing, selecting and copying / renaming media files, as well as classes for automatic image creation and the "map" tag of the "Guide" program. The technical task for the software package is developed, the description of process of its modeling and testing is given.

Conclusions. As a result of development the additional service software complex for creation of multimedia content of the Guide application was received.

Keywords. AUTOMATION, UTILITIES, CLASS, SOFTWARE SIMULATION, HTML language, JAVA FX, MAP TAG, AREA TAG.

Відомість проекту. Система автоматизації вибору мультимедійного контенту для путівника

[illegible]

Міністерство освіти і науки України
Державний заклад «Луганський національний університет
імені Тараса Шевченка»
Факультет (інститут) Навчально-науковий інститут фізики, математики та
інформаційних технологій
(повна назва)
Кафедра Інформаційних технологій та систем
(повна назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТС

М.А. Семенов
(підпис) (ініціали, прізвище)

“ ” 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на виконання програмної розробки (ПР):
" СИСТЕМА АВТОМАТИЗАЦІЇ ВИБОРУ МУЛЬТИМЕДІЙНОГО КОНТЕНТУ
ДЛЯ ПУТІВНИКА "

ІТС.ІП04.0123-02-ТЗ

ПОГОДЖЕНО
Керівник кваліфікаційної роботи

Могильний Г.А.

“ ” 2023р

ВИКОНАВЕЦЬ
Студент групи 4ПЗ

Бородін М.С.

“ ” 2023р

Полтава 2023

ЗМІСТ

ВСТУП	3
1. ХАРАКТЕРИСТИКА ОБ'ЄКТА	3
2. ПРИЗНАЧЕННЯ ТОВАРІВ	4
3. ОСНОВНІ ВИМОГИ ДО ПРОГРАМНОГО КОМПЛЕКСУ	4
4. ТЕХНІКО - ЕКОНОМІЧНІ ВИМОГ ДО КІНЦЕВОГО ПРОДУКТУ	6
5. ВИМОГИ ДО МАТЕРІАЛІВ І КОМПЛЕКТУЮЧИХ	6
6. ЕТАПИ ВИКОНАННЯ ПР	6
7. ПРИЙОМ	7
8. ПОРЯДОК ВНЕСЕННЯ ЗМІН ДО ТЕХНІЧНЕ ЗАВДАННЯ, ЩО ЗАТВЕРДЖЕНО	8

ВСТУП

1.1 Найменування: Система автоматизації вибору мультимедійного контенту для путівника.

1.2 Шифр ПР: ІТС.П04.0120

1.3 Підстава для виконання ПР: Підставою для виконання даної розробки є необхідність розробки додаткового сервісного додатку, який автоматизує вибір, перейменування, необхідне розташування файлів відповідно до вимог програмного додатку «Путівник».

1.4 Терміни розробки:

1.4.1 Початок 30 жовтня 2022 р.

1.4.2 Закінчення 30 квітня 2023р.

1.5 Фінансується за рахунок коштів замовника. Умови фінансування - за договором № 12 / а і протоколу узгодження ціни № 12 / б.

1. ХАРАКТЕРИСТИКА ОБ'ЄКТА

1.1. Розроблюваний програмний комплекс повинен автоматизувати підготовку мультимедійного контенту для програми «Путівник». До складу об'єкта, який створюється має входити:

1.1.1. Розроблюваний програмний комплекс,

1.2. До вхідної інформації відносяться вимоги програмного додатку «Путівник» щодо визначення правил розташування, йменування мультимедійних файлів.

1.3. До вихідної інформації належить структура розташування файлів:

Чотири файли – фото на кожну точку мапи

Один файл – кругове відео;

Один файл – корегуює з мапою пупівника та розташовує точки маршруту

Один файл – фрагмент коду HTML, який описує подію «onClick» на точці мапи.

2. ПРИЗНАЧЕННЯ ТОВАРІВ

2.1. Призначення: Розробка та налаштування програмного комплексу, який виконує додаткові сервісні функції для програми «Путівник».

2.2. Основні критерії ефективності.

2.2.1. Зручний інтерфейс.

2.2.1.1. Оператор повинен мати можливість обрати шлях до каталогу з відеофайлами;

2.2.1.2. Оператор повинен мати можливість обрати файл мапи, на якому створюється маршрут;

2.2.1.3. Оператор повинен мати можливість попереднього перегляду мультимедійних файлів;

2.2.1.4. Оператор повинен мати можливість проводити зручний перегляд всієї інформації.

2.2.2. Програмний комплекс повинен:

2.2.2.1. Давати можливість малювання вузлів, задавання коментаря на вузлі мапи, виділити помилкову точку мапи;

2.2.2.2. Автоматично створювати фрагмент текст HTML;

2.2.2.3. Надавати можливість графічного відображення результатів створених точок на мапі.

2.3. Основні функції оператора:

2.3.1. Запустити програмний комплекс.

2.3.2. Вибрати файл з мапою.

2.3.3. Вибрати каталог з мультимедійною інформацією.

2.3.4. Створити вузли – точки на мапі.

2.3.5. Обрати чотири фото та один відеофайл.

2.4. Основною функцією програмного комплексу є автоматизація послідовності дій оператора.

3. ОСНОВНІ ВИМОГИ ДО ПРОГРАМНОГО КОМПЛЕКСУ

3.1. Загальні вимоги

- 3.1.1. Програмний комплекс працює під операційною системою WINDOWS 10;
- 3.1.2. Вимоги до апаратного забезпечення персонального комп'ютера - не передбачені і можуть встановлюватися розробником програмного комплексу;
- 3.1.3. Програмний комплекс повинен мати зручний інтерфейс;
- 3.1.4. У складі програмного комплексу передбачити можливість створення циліндричних точок на маршруті мапи;
- 3.1.5. До складу програмного комплексу входить бібліотека Java;
- 3.1.6. До складу програмного комплексу як файл з програмою розробкою;
- 3.1.7. Програмний комплекс повинен проводити зручне відображення процесу підготовки мультимедійної інформації;
- 3.1.8. Програмний комплекс повинен забезпечувати збереження в файл створеного додаткового малюнку мати з круговими точками на ньому.
- 3.2. Додаткові вимоги
 - 3.2.1. Мова програмування Java/JavaFX.
 - 3.2.2. Вимоги до ліцензійного ПЗ не передбачаються і вирішуються замовником.
- 3.3. Вимоги до складу і архітектури
 - 3.3.1. Розробник самостійно вибирає склад і виконує розробку архітектури ПР
 - 3.3.2. Особливих умов до складу і архітектури ПР не передбачено.
- 3.4. Вимоги до якості і надійності
 - 3.4.1. Програмний комплекс повинен надійно працювати/
 - 3.4.2. Розробник вибирає технічні характеристики персонального комп'ютера, налаштовує системне програмне забезпечення.
 - 3.4.3. Розробник гарантує роботу програмного комплексу без збоїв і перенастроювань.
 - 3.4.4. Виконавець гарантує придбання додаткового обладнання (кондиціонер, UPS) за власні кошти.
- 3.5. Вимоги до експлуатації

3.5.1. Розробник використовує персональний комп'ютер, на якому програмний комплекс повинен надійно працювати.

3.5.2. Персональний комп'ютер буде задіяно в розрахунках і буде встановлений в приміщенні обчислювального центру.

4. ТЕХНІКО - ЕКОНОМІЧНІ ВИМОГ ДО КІНЦЕВОГО ПРОДУКТУ

Вартість робіт по розробці даної ПР визначається згідно з договором на розробку. Вартість пропонованих аналогів повинна забезпечити економічну доцільність їх застосування.

5. ВИМОГИ ДО МАТЕРІАЛІВ І КОМПЛЕКТУЮЧИХ

В процесі розробки програмного комплексу можливе використання програмних засобів Java.

5.1. Вимоги до екологічної безпеки при експлуатації.

Не пред'являються.

5.2. Спеціальні вимоги до кінцевого продукту.

Не пред'являються.

5.3. Вимоги до безпеки для населення при експлуатації продукції.

Не пред'являються.

6. ЕТАПИ ВИКОНАННЯ ПР

Етапи виконання ПР можуть уточнюватися згідно календарного плану робіт за погодженням між замовником і виконавцем

№	Етапи виконання роботи	Термін виконання і обсяг робіт	Звітні матеріали
1	Аналіз розробки програмного комплексу та розробка першої версії. Аналіз вимог. Розробка структури. Попереднє тестування.		Фрагмент програмного комплексу на ЕОМ замовника, який виконує всі основні функції і звітна документація п.8.2
2	Коригування структури. Розробка допоміжних функцій. Розробка остаточної версії програмного комплексу і його обробки.		Готовий програмний комплекс на ЕОМ замовника і звітна

№	Етапи виконання роботи	Термін виконання і обсяг робіт	Звітні матеріали
	Тестування.		документація п.8.2
3	Доопрацювання окремих модулів і навчання користувачів. Розробка звітних матеріалів по п.8 цього ТЗ		Звітні матеріали згідно з пунктом 8.

7. ПРИЙОМ

7.1. Необхідні вимоги для впровадження ПР і завершення робіт.

Оцінка результатів розробки і доцільність її продовження здійснюється замовником за поданням наступних матеріалів:

- встановлено програмний комплекс на ЕОМ замовника;
- перелік файлів на резервному носії;
- короткий опис роботи ПР і опис всіх файлів, які необхідні для роботи ПР.
- перелік документів
 - Технічне завдання
 - Пояснювальна записка
 - Програма і методика тестування
 - Керівництво користувача

7.2. Перелік звітних документів, необхідних для прийняття етапів роботи:

- короткий опис результатів етапу у вигляді анотованого звіту (для 1 та 2 етапів);
- частковий програмний комплекс на ЕОМ замовника згідно календарного плану робіт;
- акт приймання продукції.

Звітні матеріали подаються у вигляді звітів на папері по ДСТУ

7.3. Загальний перелік до прийому звітних документів, макетів, експериментальних зразків.

До прийому пред'являються: акт здачі-приймання продукції, акт впровадження ПР.

7.4.Тестування ПР

Тестування виконується в "Програма і методика тестування", яка розробляється виконавцем і затверджується замовником

8. ПОРЯДОК ВНЕСЕННЯ ЗМІН ДО ТЕХНІЧНЕ ЗАВДАННЯ, ЩО ЗАТВЕРДЖЕНО.

Дане технічне завдання може уточнюватися в процесі розробки ПР при узгодженні сторін з оформленням доповнень до ТЗ.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЗ «ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА»

Навчально-науковий інститут фізики, математики та
інформаційних технологій

(назва факультету, інституту)

Інформаційних технологій та систем

(назва кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

БАКАЛАВРА

(освітньо-кваліфікаційний рівень)

на тему: " СИСТЕМА АВТОМАТИЗАЦІЇ ВИБОРУ МУЛЬТИМЕДІЙНОГО
КОНТЕНТУ ДЛЯ ПУТІВНИКА "

ІТС.ІП04.0323-05-МТ

Виконав: студент 4 курсу
напряму підготовки (спеціальності)
121 «Інженерія програмного
забезпечення»

(шифр і назва напряму підготовки, спеціальності)

Бородін М.С..

(прізвище та ініціали)

Керівник Могильний Г.А.

(прізвище та ініціали)

Рецензент Козуб Ю.Г.

(прізвище та ініціали)

Полтава – 2023

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	3
ВСТУП.....	4
Розділ 1. Мультимедійні можливості Java	6
1.1 API платформи Java.....	6
1.2 Огляд sampled пакету API Java Sound	8
РОЗДІЛ II. АНАЛІЗ ВИМОГ ПРОГРАМИ «ПУТІВНИК»	18
2.1. Карти зображень.....	18
2.2. Додаткові вимоги програми «ПУТІВНИК».....	23
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ ПІДГОТОВКИ МУЛЬТИМЕДІЙНОГО КНТЕНТУ	26
3.1. Обґрунтування вибору середовища розробки додатку.....	26
3.2. Загальна структура і функціональна схема додатку	27
ВИСНОВКИ	37
ДОДАТОК.....	41
Лістинг коду	41

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПК	Программный комплекс
ОС	Операционная система
JVM	Java Viertual Michine (Виртуальная Java машина)
IDE	Integrated Development Environment (Интегрированная среда разботки)
JIT	Just-in-time («На лету»)
ARM	Advanced RISC Machines
RMI	Remote Method Invocation (удаленный вызов методово)
HTTP	HyperText Transfer Protocol (Протокол передачи гипертекста)
JDBC	Java DataBase Connectivity (Соединение с базами данных)
RIA	Rich Internet Application (Насыщенное интернет приложение)
CSS	Cascading Style Sheets (Касткадные таблицы стилей)
MSN	Microsoft Network
IPTV	Internet Protocol Television (Телевиденье по протоколу интернета)
IGMP	Internet Group Management Protocol (Протокол упправления группами интернета)
IP	Internet Protocol

ВСТУП

У сучасних умовах високої автоматизації виробничих процесів значну роль грають все можливі допоміжні утиліти. До таких програм належать різноманітні додатки, які автоматизують процес створення різноманітної інформації. На кафедрі інформаційних технологій та систем було створено програму «Путівник». Основною метою її є візуальна демонстрація різноманітних варіантів шляху до навчально наукового інституту фізики, математики та інформаційних технологій.

Однією з особливостей цієї програми є використання великою кількістю відео та фото інформації. Три цьому, ця інформація повинна бути розташована у певних папках та мати особливі імена файлів. Тільки в цьому випадку програма буде вірно відображати інформацію на веб сторінках. Тому процес підготовки мультимедійного контенту для програми «Путівник» є окремою та важливою задачею.

Об’єкт дослідження –додатки автоматизації створення тегу «мапа» у мові HTML.

Предмет дослідження –додатки та утиліти обробки мультимедійних файлів.

Мета роботи – розробка спеціалізованого додатку для автоматизації процесу попереднього перегляду, вибору мультимедійних файлів з можливістю підготовки мультимедійного контенту програми «Путівник».

Досягнення зазначеної мети передбачає вирішення таких основних завдань:

- Розглянути особливості програми «Путівник»;
- Обрати шляхи та вимоги вирішення завдання – автоматизації підготовки мультимедійного контенту;
- Розробити та реалізувати програмний додаток.

У першому розділі розглянуто особливості тега мапа у мові HTML.
Наведено опис сучасної технології Java/JavaFX

У другому розділі проведено аналіз програми «Путівник».

Третій розділ присвячений розробці програмного модуля для автоматизації процесу підготовки мультимедійного контенту.

РОЗДІЛ 1. МУЛЬТИМЕДІЙНІ МОЖЛИВОСТІ JAVA

1.1 API платформи Java

API Java Sound - це низькорівневий API для здійснення та керування входом і виведенням звукових носіїв, включаючи як аудіо, так і музичний інструмент цифрового інтерфейсу (MIDI). API Java Sound забезпечує явний контроль над можливостями, які зазвичай потрібні для введення та виведення звуку, в рамках, що сприяє розширенню та гнучкості.

Оскільки звук настільки фундаментальний, API Java Sound задовольняє потреби широкого кола розробників програм. Потенційні сфери застосування включають:

- Фреймворки комунікації, такі як конференц-зв'язок та телефонія
- Системи доставки контенту кінцевого користувача, такі як мультимедійні плеєри та музика за допомогою потокового вмісту
- Інтерактивні прикладні програми, такі як ігри та веб-сайти, які використовують динамічний вміст
- Створення та редагування вмісту
- Інструменти, інструментарій та утиліти як аудіо, так і мультимедійний цифровий інтерфейс (MIDI).

API Java Sound забезпечує явний контроль над можливостями, які зазвичай потрібні для введення та виведення звуку, в рамках, що сприяє розширенню та гнучкості.

API Java Sound забезпечує найнижчий рівень підтримки звуку на платформі Java. Це забезпечує прикладні програми з великою кількістю контролю над звуковими операціями, і це розширюване. Наприклад, Java Sound API поставляє механізми для встановлення, доступу та керування системними ресурсами, такими як аудіо мікшери, MIDI-синтезатори, інші аудіо- та MIDI-пристрої, читачі файлів та письменники, а також перетворювачі форматів звуку. API Java Sound не включає складних

редакторів звуку або графічних інструментів, але надає можливості для створення таких програм. Це підкреслює низький рівень контролю за межами того, що зазвичай очікує кінцевий користувач.

Є інші API платформи Java, які мають елементи, пов'язані із звуком. **Java Media Framework (JMF)** - це API вищого рівня, який наразі доступний як стандартний розширення для платформи Java. JMF визначає уніфіковану архітектуру, протокол обміну повідомленнями та інтерфейс програмування для захоплення та відтворення медіа на основі часу. JMF забезпечує просте рішення для базових прикладних програм медіапрограваців та дозволяє синхронізувати різні типи носіїв, наприклад, аудіо та відео.

З іншого боку, програми, що фокусуються на звучанні, можуть скористатися API Java Sound, особливо якщо вони потребують більш просунутих функцій, таких як можливість ретельно контролювати відтворення буферизованого звуку або безпосередньо управляти MIDI-синтезатором. Інші API Java із звуковими аспектами включають Java 3D та API для телефонії та мови. Реалізація будь-якого з цих API може використовувати внутрішню програму Java Sound API, але не вимагає цього.

Пакети

API Java Sound включає підтримку як цифрового аудіо, так і MIDI-даних. Ці два основних модуля функціональності надаються в окремих пакунках:

- `javax.sound.sampled` Цей пакет вказує інтерфейси для захоплення, змішування та відтворення цифрового (вибіркового) звуку.
- `javax.sound.midi` Цей пакет забезпечує інтерфейси для синтезу MIDI, послідовності та транспортування подій.

Два інших пакети дозволяють постачальникам послуг (на відміну від розробників додатків) створювати власні програмні компоненти, які розширюють можливості реалізації API Java Sound [1]:

- `javax.sound.sampled.spi`
- `javax.sound.midi.spi`

1.2 Огляд **sampled** пакету **API Java Sound**

Пакет `javax.sound.sampled` пов'язаний із аудіо транспортом - іншими словами, **Java Sound API** фокусується на відтворенні та захопленні аудіо. Центральне завдання, яке стосується адрес **Java-звуку API**, полягає в тому, як перемістити байт форматованих аудіо-даних у систему та за його межами. Це завдання передбачає відкриття аудіовхідних та вихідних пристроїв та керування буферами, які наповнюються звуковими даними у реальному часі. Це також може включати в себе змішування декількох потоків аудіо в один потік (будь то для введення або виведення). Передача звуку в систему або її виймання повинна правильно оброблятися, коли користувач запитує, щоб потік звуку був запущений, призупинений, відновлений або зупинений.

Щоб підтримати цю основну увагу на базовому вхідному та вихідному звукових даних, **API Java Sound** пропонує методи конвертації між різними форматами звукових даних, а також для читання та запису звичайних типів звукових файлів. Однак він не намагається бути всеосяжним інструментарієм звукових файлів. Певна реалізація **API Java Sound** не повинна підтримувати великий набір типів файлів або перетворення формату даних. Сторонні постачальники послуг можуть надавати модулі, які "підключаються" до існуючої реалізації для підтримки додаткових типів файлів і конверсій.

Буферизоване та небуферне керування звуком

API Java Sound може обробляти аудіо транспорту як в потоковому режимі, так і в буфері, і в пам'яті, без буферу. "Потокове передавання" тут використовується в загальному сенсі для позначення обробки аудіо-байтів у режимі реального часу; це не стосується конкретного, добре відомого випадку надсилання аудіо через Інтернет в певному форматі. Іншими словами, потік аудіо - це просто безліч аудіо-байтів, які прибувають більш-менш з тією ж швидкістю, до якої вони обробляються (відтворюються, записуються тощо). Операції над байтами починаються до того, як всі дані надійдуть. У потоковому режимі, особливо у випадку аудіо введення, а не аудіо виходу, користувачу не обов'язково знати заздалегідь, якої довжини

звук і коли він закінчиться. Ви просто обробляєте один буфер аудіоданих одночасно, доки операція не буде зупинена. У випадку виведення звуку (відтворення), також потрібно буде буферизація даних, якщо звук, який ви хочете відтворити, занадто великий, щоб він міг уміститися повністю в пам'яті. Іншими словами, ви передаєте аудіо-байти до звукового движка кусочками, і він дбає про відтворення кожного зразка в потрібний час. Існують механізми, які дозволяють легко зрозуміти, скільки даних потрібно доставляти в кожному фрагменті.

API Java Sound також дозволяє небуферизоване транспортування лише у випадку відтворення, якщо ви вже маєте всі аудіодані, які знаходяться під рукою, і не є надто великими, щоб вписати їх у пам'ять. У цій ситуації додаткова програма не потребує буферизації звуку, хоча буферний підхід до реального часу все ще доступний за бажанням. Замість цього весь звук може бути попередньо завантажений в пам'ять для подальшого відтворення. Оскільки всі звукові дані завантажуються заздалегідь, відтворення може розпочатися негайно, наприклад, як тільки користувач натискає кнопку «Пуск». Це може бути перевагою в порівнянні з буферною моделлю, де відтворення повинно очікувати першого заповнення буфера. Крім того, вбудована модель, що зберігається в пам'яті, дозволяє легко звучати з циклом (циклічно) або встановлювати довільні позиції в даних.

Щоб відтворити або зафіксувати звук за допомогою API Java Sound, потрібно як мінімум три речі: відформатовані аудіодані, змішувач та рядок. Кожен з них пояснюється нижче.

Формати даних

Формат даних розповідає вам про те, як інтерпретувати серію байтів "сирих" вибіркового аудіоданих, таких як зразки, які вже були прочитані із звукового файлу, або зразки, зняті з входу мікрофона. Можливо, вам знадобиться знати, наприклад, скільки бітів складають один зразок (представлення найкоротшого моменту звуку), і, подібно, вам може знадобитися знати частоту дискретизації звуку (наскільки швидко зразки

повинні слідувати один одному). Під час налаштування для відтворення або зйомки ви задаєте формат даних звуку, який ви захоплюєте чи відтворюєте.

У API Java Sound формат даних представлений об'єктом `AudioFormat`, який включає в себе наступні атрибути:

- Технологія кодування, як правило, модуляція імпульсного коду (PCM)
- Кількість каналів (1 для моно, 2 для стерео та ін.)
- Коефіцієнт вибірки (кількість зразків в секунду на канал)
- Кількість бітів на вибірку (на канал)
- Частота кадрів
- Розмір кадру в байтах
- Порядок байтів (big-endian або small-endian)

PCM - це один з видів кодування звукової форми сигналу. API Java Sound включає в себе два кодування PCM, які використовують лінійне квантування амплітуди, а також цілі значення, що підписані або не підписані. Лінійне квантування означає, що кількість, що зберігається в кожному зразку, прямо пропорційна (за винятком будь-яких спотворень) до вихідного звукового тиску в той момент і аналогічно пропорційно зміщенню гучномовця або барабанної перетинки, яка вібрує звуком в той момент. Нелінійне кодування відображає амплітуду вихідного звуку до збереженого значення, використовуючи нелінійну функцію, яка може бути розроблена для отримання більшої амплітудної роздільної здатності для тихого звучання, ніж для гучних звуків.

Змішувач

Багато програмних інтерфейсів програмного забезпечення (API) для звуку використовують поняття звукового пристрою. Пристрій часто є програмним інтерфейсом для фізичного пристрою введення / виведення. Наприклад, пристрій звукового введення може представляти можливості

входу звукової карти, включаючи мікрофонний вхід, аналоговий вхід на рівні лінії та, можливо, цифровий аудіовхід.

У Java Sound API пристрої представлені об'єктами Mixer. Метою змішувача є обробка одного або декількох потоків аудіовходу та одного або декількох потоків аудіовиходу. У типовому випадку він фактично об'єднує декілька вхідних потоків у один вихідний потік. Об'єкт Mixer може представляти можливості звукового змішування на фізичному пристрої, наприклад звуковій картці, яка може потребувати змішувати звук, який надходить до комп'ютера з різних входів, або звучання, що надходить із програм, і відбувається на виході.

Іншими словами, об'єкт Mixer може представляти можливості звукового змішування, які повністю реалізовані в програмному забезпеченні без будь-якого внутрішнього інтерфейсу для фізичних пристроїв.

У Java Sound API є такий компонент, як вхід мікрофона на звуковій картці, сам не вважається пристроєм, тобто змішувачем, а не портом в змішувач або з нього. Порт, як правило, забезпечує один потік аудіо в міксері або з нього (хоча потік може бути багатоканальним, наприклад, стерео). Змішувач може мати кілька таких портів. Наприклад, змішувач, який представляє вихідні можливості звукової карти, може об'єднати декілька потоків аудіо разом, а потім відправити змішаний сигнал на будь-який або всі різні вихідні порти, підключені до змішувача. Ці вихідні порти можуть бути, наприклад, роз'ємом для навушників, вбудованим динаміком або виходом на рівні лінії.

Лінія

Лінія це метафора фізичної консолі змішування також корисна для розуміння концепції лінії Java Sound API.

Лінія є елементом цифрового аудіо "трубопроводу", тобто шлях для переміщення аудіо в систему або виходу з неї. Зазвичай ця лінія являє собою шлях до міксера або з неї (хоча в технічному сенсі сам мікшер теж є своєрідною лінією).

Вхідні і вихідні аудіосигнали - це лінії. Вони аналогічні мікрофонам і динамікам, підключеним до фізичної консолі змішування. Інший вид лінії - це шлях до даних, через який програма може отримати вхідний аудіосигнал або відправити вихідний аудіосигнал у змішувач. Ці шляхи даних аналогічні до трас мультитрейкового записуючого пристрою, підключеного до фізичної консолі змішування.

Одна різниця між лініями в API Java Sound і функціями фізичного змішувача полягає в тому, що аудіодані, що проходять через лінію в API Java Sound, можуть бути моно- або багатоканальними (наприклад, стерео). На відміну від цього, кожен з входів і виходів фізичного мікшера, як правило, є єдиним каналом звуку. Щоб отримати два або більше каналів виведення з фізичного змішувача, зазвичай використовуються два або більше фізичних виходів (принаймні, у випадку аналогового звуку, цифровий вихідний роз'єм часто багатоканальний). У Java Sound API кількість каналів у рядку визначається за допомогою AudioFormat даних, які поточно проходять через рядок. [2]

AudioSystem Class

Клас AudioSystem виступає в якості клірингової установи для аудіокомпонентів, включаючи вбудовані служби та окремо встановлені сервіси від сторонніх постачальників. AudioSystem слугує точкою входу додатка для доступу до цих встановлених зразкових аудіо ресурсів. Ви можете задати запит AudioSystem, щоб дізнатись, які різновиди ресурсів було встановлено, а потім ви можете отримати доступ до них. Наприклад, програма може починатися з запрошення у класі AudioSystem, чи існує міксер, який має певну конфігурацію, наприклад, один з конфігурацій вводу чи виводу, ілюстрований раніше під час обговорення ліній. З мікшера програма потім отримає лінії даних і так далі.

Ось деякі з ресурсів, які прикладна програма може отримати від AudioSystem:

- Змішувачі У системі, як правило, встановлено кілька змішувачів. Зазвичай принаймні один - для вхідного аудіо та один - для виведення звуку. Також можуть бути змішувачі, які не мають портів вводу-виводу, але замість цього приймають аудіо з прикладної програми та передають змішане звучання назад у програму. Клас `AudioSystem` надає список всіх встановлених мікшери.
- Лінії Незважаючи на те, що кожна лінія пов'язана з мікшером, програма може отримати лінію безпосередньо з `AudioSystem`, не маючи чіткого зв'язку з змішувачами.
- Форматування конверсій Програма може використовувати перетворення формату для перекладу аудіоданих з одного формату в інший.

Файли та потоки Клас `AudioSystem` забезпечує методи перекладу між аудіофайлами та аудіопотоками. Він також може повідомляти про формат файлу звукового файлу і може записувати файли у різних форматах.

Кілька класів у `API Java Sound` надають корисну інформацію про пов'язані інтерфейси. Наприклад, `Mixer.Info` містить відомості про встановлений мікшер, наприклад постачальника мікшери, назву, опис та версію. `Line.Info` отримує клас певної лінії. Підкласи `Line.Info` включають `Port.Info` та `DataLine.Info`, які отримують інформацію, що стосується конкретного порту та лінії передачі даних, відповідно. Кожен з цих класів описаний далі в відповідному розділі нижче. Важливо не плутати об'єкт `Info` із змішувачем чи об'єктом лінії, який він описує.

Отримання змішувача

Зазвичай, одна з перших речей, яку потрібно виконати для програми, яка використовує `Java Sound API`, - отримати мікшер або принаймні один рядок змішувача, щоб ви могли отримувати звук на комп'ютері або виходити з нього. Для вашої програми може знадобитися конкретний змішувач, або ви, можливо, захочете відобразити список всіх доступних змішувачів, щоб

користувач міг вибрати його. У будь-якому випадку ви повинні дізнатись, які типи змішувачів встановлені. `AudioSystem` забезпечує наступний метод:

```
static Mixer.Info [] getMixerInfo ()
```

Кожний об'єкт `Mixer.Info`, який повертається за допомогою цього методу, визначає один тип встановленого змішувача. (Зазвичай у системі є не більше одного змішувача даного типу. Якщо трапляється, що це більше, ніж один з даного типу, то повернутому масиву до цих пір є тільки один `Mixer.Info` для цього типу). Програма може робити ітерацію над об'єктами `Mixer .Inf`, щоб знайти відповідний, відповідно до його потреб. `Mixer.Info` містить наступні рядки, щоб визначити вид змішувача:

- Ім'я
- Версія
- Постачальник
- Опис

Це довільні рядки, тому прикладна програма, яка потребує певного змішувача, повинна знати, чого очікувати і як порівнювати рядки з. Компанія, яка надає змішувач, повинна включити цю інформацію в свою документацію. Як альтернатива, і, можливо, більш типово, прикладна програма відобразить всі рядки об'єктів `Mixer.Info` користувача та дозволить користувачеві вибрати відповідний мікшер.

Коли знайдено відповідний мікшер, прикладна програма викличе наступний метод `AudioSystem` для отримання бажаного міксера:

```
static Mixer getMixer (Mixer.Info info)
```

Отримання лінії бажаного типу

Існує два способи отримати лінію:

- Безпосередньо з об'єкта `AudioSystem`
- З мікшера, який ви вже отримали від об'єкта `AudioSystem`.

В першому випадку, безпосередньо з `AudioSystem`, якщо змішувач не був отримано, і програма є простою, який дійсно потребує певної лінії, можна використовувати метод `AudioSystem`:

```
static Line getLine(Line.Info info),
```

що аналогічно методу `getMixer`, описаному вище.

На відміну від `Mixer.Info`, `Line.Info`, використаний як аргумент, не зберігає текстову інформацію, щоб вказати потрібну лінію. Замість цього зберігається інформація про клас потрібної лінії.

`Line.Info` є абстрактним класом, тому треба використовувати один з його підкласів (`Port.Info` або `DataLine.Info`), щоб отримати лінію. Наступний витяг коду використовує підклас `DataLine.Info`, щоб отримати та відкрити цільову рядок даних:

```
TargetDataLine line;
DataLine.Info info = new DataLine.Info(TargetDataLine.class, format);
// format is an AudioFormat object
if (!AudioSystem.isLineSupported(info)) {
    // Handle the error.
}
// Obtain and open the line.
try {
    line = (TargetDataLine) AudioSystem.getLine(info);
    line.open(format);
} catch (LineUnavailableException ex) {
    // Handle the error.
    //...
}
```

Цей код отримує об'єкт `TargetDataLine`, не вказуючи жодних атрибутів крім його класу та його аудіоформату. Ви можете використовувати аналогічний код для отримання інших видів ліній. Для `SourceDataLine` або `Clip` просто замініть цей клас для `TargetDataLine` як класу змінної лінії, а також у першому аргументі конструктора `DataLine.Info`.

Для порту можна використовувати статичні екземпляри `Port.Info` у коді, подібному до наступного:

```
if (AudioSystem.isLineSupported(Port.Info.MICROPHONE)) {
    try {
        line = (Port) AudioSystem.getLine(
            Port.Info.MICROPHONE);
    }
}
```

Другий варіант отримання лінії з міксера полягає у наступному.

Інтерфейс Mixer включає варіації методів доступу AudioSystem для вихідних та цільових ліній, описаних вище. До таких методів міксерів відносяться ті, що приймають аргументи Line.Info, як це роблять методи AudioSystem. Тим не менш, міксер також включає ці варіанти, які не приймають жодних аргументів:

```
Line.Info[] getSourceLineInfo()  
Line.Info[] getTargetLineInfo()
```

Ці методи повертають масиви всіх об'єктів Line.Info для конкретного міксера. Після того, як ви отримаєте масиви, ви можете виконати перебір над ними, викликаючи метод getLine Mixer для отримання кожного рядка, а потім відкритим методом Line для резервування використання кожного рядка для вашої програми.

Дозвіл на використання аудіо ресурсів

API Java Sound включає в себе клас AudioPermission, який вказує, які види доступу до аплету (або програми, запущеної менеджером безпеки) можуть мати у системі sampled-audio. Дозвіл на запис звуку регулюється окремо. Ці дозволи слід надавати обережно, щоб запобігти небезпеці безпеки, такому як несанкціоноване підслуховування. За замовчуванням аплети та додатки отримують дозвіл наступним чином:

- Апет, що працює з менеджером безпеки аpletів, може відтворювати, але не записувати звук.
- Програма, що працює без менеджера безпеки, може як грати, так і записувати звук.
- Програма, запущена за допомогою менеджера безпеки за замовчуванням, може відтворювати, але не записувати аудіо.

Загалом, аплети запускаються під контролем менеджера безпеки і не дозволяється запис звуку. З іншого боку, програми не автоматично встановлюють менеджера безпеки і можуть записувати звук. (Однак, якщо

менеджер безпеки за замовчуванням викликається явно для програми, програмі не дозволяється записувати звук).

Обидва апплети та програми можуть записувати звук навіть під час роботи з диспетчером безпеки, якщо їм надано явне дозвіл на це. [3]

РОЗДІЛ II. АНАЛІЗ ВИМОГ ПРОГРАМИ «ПУТІВНИК»

Програма «Путівник» в використовує HTML сторінки. Основою її є карта шляху до якогось підприємства, підрозділу, кімнати, наприклад, до ІФМІТ. Для більш швидкої автоматизації структура додатку розроблена таким чином, що ім'я малюнку карати повинно співпадати з ім'ям HTML. Сама програма не автоматизує окремі точки карти зображень. Тому першим завданням – є автоматизація створення мапи карти на засадах створення додаткового HTML коду, який потрібно вставити в кождну окрему сторінку програми «Путівник»

Розглянемо особливості створення карти.

2.1. Карти зображень

Карти зображень

Карта зображень - це зображення, яке було розділене на регіони, або "гарячі точки"; при натисканні на точку доступу відбувається дія, наприклад, відкривається новий файл. Зображення, знайдене внизу, є прикладом карти зображень. Усі посилання є в одному зображенні, які пов'язані на різних сторінках(рис. 2.1).

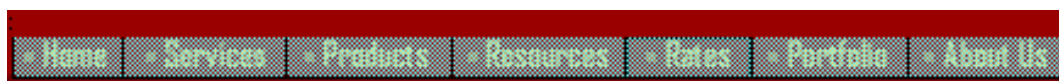


Рис. 2.1. Прямокутні регіони карти зображень

Можливо створити кругові та багатокутні ланки, як показано нижче (рис. 2.2).

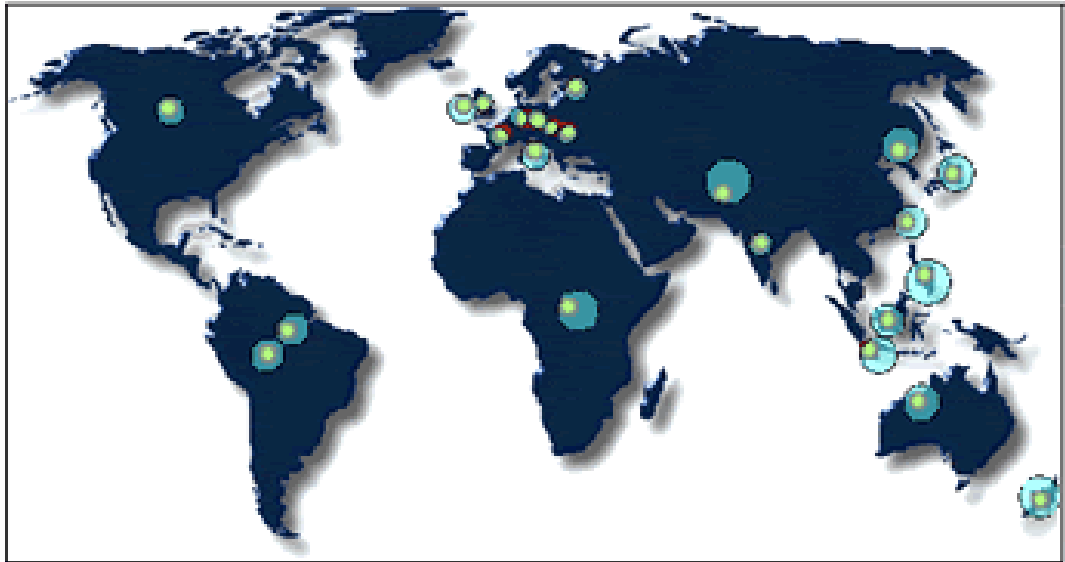


Рис. 2.2. Приклад кругових та багатокутних ланок

В історії веб-дизайну був момент, коли багато сайтів використовували функцію, відому як "карти зображень". Це список координат, приєднаних до певного зображення на сторінці. Ці координати створюють ділянки гіперпосилання на цьому зображенні, істотно додаючи графіку "гарячих точок", кожен з яких можна кодувати для посилання на різні місця. Це набагато інакше, ніж просто додати тег посилання до зображення, що призведе до того, що вся графіка стане однією великою ланкою до одного пункту призначення.

Карти зображень, звичайно, були, але вони не вподобали сьогоденню Мережу. Це, принаймні частково, тому що карти зображень потребують конкретних координат для роботи. Сьогодні сайти створені таким чином, щоб реагувати на зображення та масштабувати зображення залежно від розміру екрана чи пристрою. Це означає, що заздалегідь задані координати, як це працює карти зображень, розпадаються, коли сайт масштабує і зображення змінюють розмір. Ось чому карти зображень сьогодні рідко використовуються на сайтах з виробництва, але вони все ще мають переваги для демонстрацій чи випадків, коли ви назначаете чіткий розмір сторінки.

Хочете знати, як створити карту зображень, зокрема, як це зробити за допомогою Dreamweaver? Процес не особливо складний, але теж непростий, тому перед початком роботи ви повинні мати певний досвід.

Перший крок, який потрібно зробити, - це додати зображення на свою веб-сторінку. Потім натисніть на зображення, щоб виділити його. Звідти вам потрібно перейти до меню властивостей (і натиснути один з трьох інструментів малювання точкової точки: прямокутник, коло або багатокутник. Не забудьте назвати своє зображення, яке ви можете зробити в панелі властивостей. Ви можете назвати що завгодно. Використовуйте " карту " як приклад.

Тепер намалюйте потрібну форму на своєму зображенні за допомогою одного з цих інструментів. Якщо вам потрібні прямокутні плями, використовуйте прямокутник. Те саме для кола. Якщо ви хочете більш складні форми гарячої точки, використовуйте багатокутник. Це те, що ви, ймовірно, використовуєте на прикладі карти США, оскільки багатокутник дозволить вам опускати точки і створювати дуже складні і неправильні фігури на зображенні

У вікні властивостей точки доступу введіть або перейдіть на сторінку, на яку хотспот повинен посилатися. Це те, що створює цю пов'язану область. Продовжуйте додавати гарячі точки до тих пір, поки ваша карта не буде заповнена і не будуть додані всі посилання, які ви хочете додати.

Після закінчення перегляньте свою картинку в браузері, щоб переконатися, що вона працює правильно. Клацніть кожне посилання, щоб переконатися, що воно переходить до належного ресурсу чи веб-сторінки.

Недоліки зображень Карти

Карти зображень мають кілька мінусів, навіть за межами вищезазначеної відсутності підтримки веб-сайтами.

По-перше, невеликі деталі можуть бути затемнені на карті зображення. Наприклад, географічні карти зображень можуть допомогти визначити, з якого континенту є користувач, але ці карти можуть бути

недостатньо деталізованими, щоб точно визначити країну походження користувача. Це означає, що карта зображень може допомогти визначити, чи є користувач з Азії, а не з Камбоджі, зокрема.

По-друге, карти зображень також можуть повільно завантажуватися. Їх не слід використовувати кілька разів на веб-сайті, оскільки вони займають занадто багато місця для використання на кожній сторінці веб-сайту. Занадто багато мап зображень на одній сторінці створило б серйозне вузьке місце та масовавший вплив на продуктивність сайту.

Нарешті, карти зображень може бути непростим для користувачів, які мають проблеми із зором. Якщо ви використовували карти зображень, ви також повинні створити іншу систему навігації для цих користувачів як альтернативу.

Таким чином, час від часу використовуйте карти зображень, намагаючись скласти швидку демонстрацію дизайну та способів його роботи. Наприклад, якщо ви знушаєтесь над дизайном для мобільного додатка і хочете використовувати карти зображень для створення точкових точок для імітації інтерактивності програми. Це зробити набагато простіше, ніж було б кодувати додаток або навіть будувати підроблені веб-сторінки, побудовані відповідно до сучасних стандартів за допомогою HTML та CSS. Але розміщувати їх на виробничому сайті чи додатку дуже складно.

Приклад створення карти зображень у Dreamweaver

Необхідно обрати зображення, на якому треба зробити кілька посилань. Клацніть на прямокутному інструменті гарячої точки, знайденому в Інспекторі властивостей. Виберіть інструмент прямокутника та перетягніть вказівник на зображення, щоб створити прямокутну точку доступу. Також можливо вибрати інструмент гарячої точки овальної або багатокутної форми, щоб зробити вибір овалу чи полігона.

У полі "Посилання" інспектора властивостей точки доступу натисніть на значок папки, щоб перейти до потрібного файлу, який ви відкрили після натискання точки доступу. Крім того, введіть ім'я файлу.

Повторіть описані вище кроки, щоб визначити додаткові точки доступу на карті зображень.

Код буде наступний вид.

```
<map name = "tabsMap">  
< area shape = "rect" coords = "- 1,0,46,16" href = "/ default.asp" alt =  
    "веб-розробка">  
<area shape = "rect" coords = "46,0,108,16" alt = "services" href = "/"  
    services.asp">  
<area shape = "rect" coords = "108,0,172,16" alt = "шаблони" href = "/"  
    website_templates /">  
<area shape = "rect" coords = "172,0,243,16" alt = "portfolio" href = "/"  
    portfolio.asp">  
<area shape = "rect" coords = "243,0,295,16" alt = "testimonials" href = "/"  
    testimonials.asp">  
<area shape = "rect" coords = "296,0,363,16" href = "/ contactus.asp"  
    target = "_ blank" alt = "зв'яжіться з нами">  
</map>  
.
```

Типи областей

Для карт зображень існує 3 можливих типи областей - прямокутник, коло і багатокутник.

Прямокутна область (рис. 2.3).

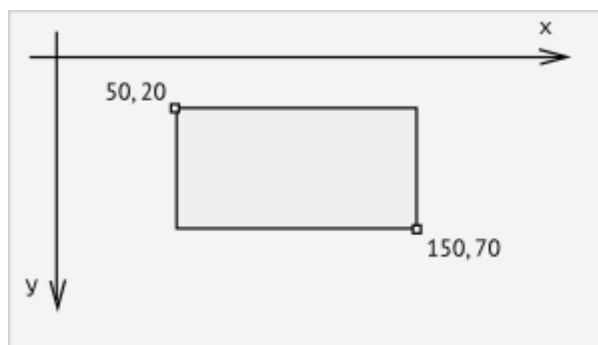


Рис 2.3. Параметри прямокутника

У мові html буде:

```
<area shape="rect" coords="50, 20, 150, 70" />
```

Кругова область (рис. 2.4)

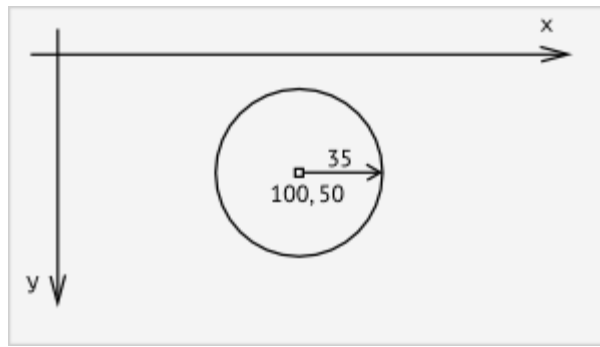


Рис. 2.4 Приклад кругової області

У мові html буде

```
<area shape="circle" coords="100, 50, 35" />
```

2.2. Додаткові вимоги програми «ПУТІВНИК»

В процесі аналізу програми «Путівник» було встановлено наступне:

1. Ім'я мапи співпадає с ім'ям файлу HTML та має цифру (1,2,3 і таке інше).
2. Для кожної мапи необхідно автоматично створити підкаталог с ім'ям мапи.
3. Для кожної мапи необхідно створити додатковий html файл та по згоді з розробником ім'я цього файлу повинно бути цифра_avto_generic.html.
4. Для кожної мапи необхідно створити додатковий малюнок мапи з намальованими точками маршруту на мапі. Ім'я цього файлу повинно бути "цифра_fon.png".
5. Для кожної точки мапи у своєму підкаталозі (1, 2,3 і т.і) скопіювати чотири фото з ім'ям цифра_1.jpg, цифра_2.jpg, цифра_3.jpg, цифра_4.jpg та один відео фвіл цифра.mp4.
6. Для зручного вибору файлів програма повинна мати попередній перегляд обраних файлів.

Таким чином в результаті повинно бути, підкаталог для кожної мапи, у підкаталозі – для кожної точки 4 фото та один відео файл. Для кожної мапи необхідно – один фоновий малюнок та один додатковий HTML файл.

Всі ці вимоги знайшли своє відображення у технічному завданні до розробки.

Основні критерії ефективності.

1.Зручний інтерфейс.

- Оператор повинен мати можливість обрати шлях до каталогу з відеофайлами;
- Оператор повинен мати можливість обрати файл мапи, на якому створюється маршрут;
- Оператор повинен мати можливість попереднього перегляду мультимедійних файлів;
- Оператор повинен мати можливість проводити зручний перегляд всієї інформації.

2.Програмний комплекс повинен:

- Давати можливість малювання вузлів, задавання коментаря на вузлі мапи, виділити помилкову точку мапи;
- Автоматично створювати фрагмент текст HTML;
- Надавати можливість графічного відображення результатів створених точок на мапі.

3.Основні функції оператора:

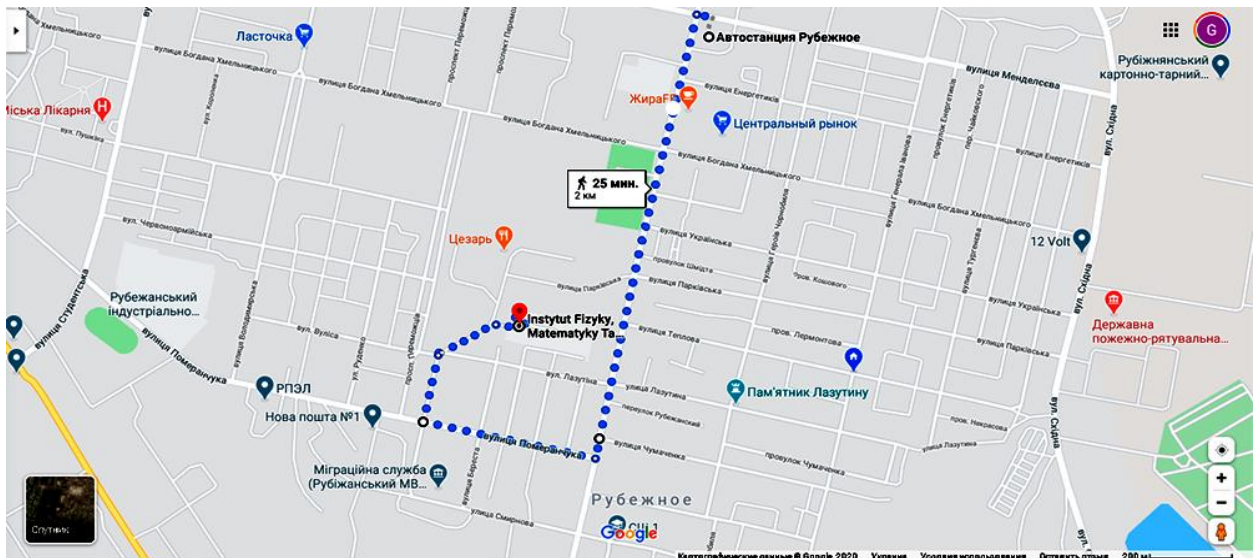
- Запустити програмний комплекс.
- Вибрати файл з мапою.
- Вибрати каталог з мультимедійною інформацією.
- Створити вузли – точки на мапі.
- Обрати чотири фото та один відеофайл.

4. Основною функцією програмного комплексу є автоматизація послідовності дій оператора.

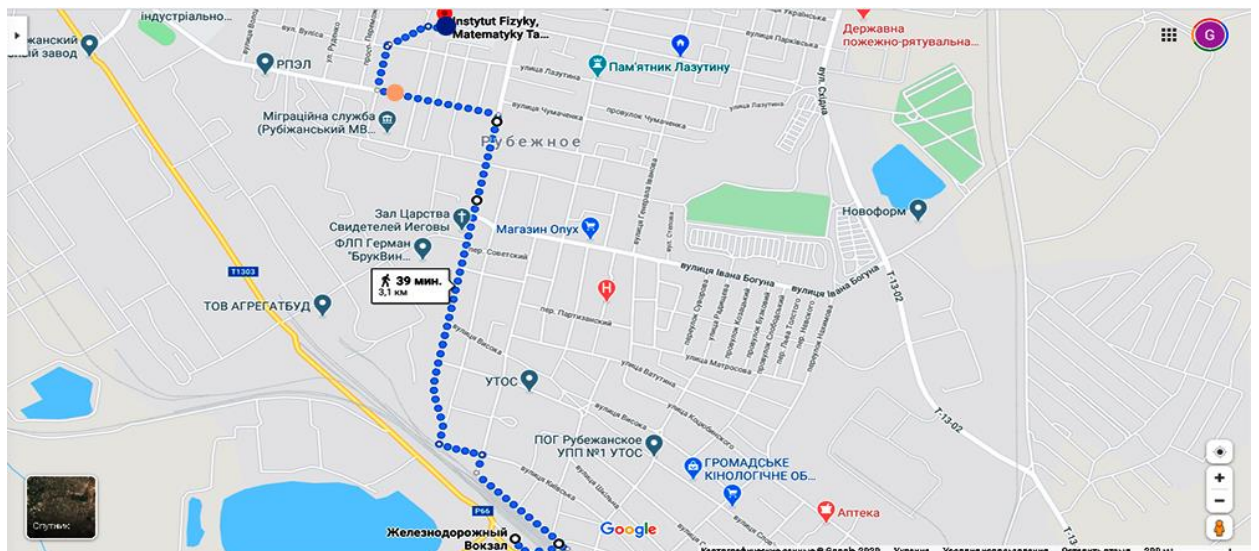
Приклад додаткового файлу

```
map name="map1_auto_insert">
<area shape=" circle" href="#" coords="353 , 16 , 8 " onClick="loadris(1);"
      title="оглянути 1 " alt="оглянути 1" >
<area shape=" circle" href="#" coords="312 , 72 , 7 " onClick="loadris(2);"
      title="hsgdhsqd 2 " alt="hsgdhsqd 2" >
</map>
```

Приклад додаткового малюнку мапи



А) біла точка



Б) синя та розова точки

Рис. 2.5 Приклад мапи з встановленими точками

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ ПІДГОТОВКИ МУЛЬТИМЕДІЙНОГО КНТЕНТУ

3.1. Обґрунтування вибору середовища розробки додатку

Мова програмування Java зародилася в 1991 р. в лабораторіях компанії Sun Microsystems. Розробку проекту започаткував Джеймс Гослінг, сам проект мав назву «Green» (Зелений). Створення першої робочої версії, яка мала назву «Oak» (дуб), зайняло 18 місяців. Оскільки виявилось, що ім'я Oak уже використовувалось іншою фірмою, то в результаті тривалих суперечок навколо назви нової мови з-поміж ряду запропонованих було вибрано назву Java[5], у 1995 р. мову було офіційно перейменовано[6].

Головним мотивом створення Java була потреба в мові програмування, яка б не залежала від платформи (тобто від архітектури) і яку можна було б використовувати для створення програмного забезпечення, що вбудовується в різноманітні побутові електронні прилади, такі як мобільні засоби зв'язку, пристрої дистанційного керування тощо.

13 листопада 2006 року Sun випустили більшу частину Java як вільне та відкрите програмне забезпечення згідно з умовами GNU General Public License (GPL). 8 травня 2007 корпорація закінчила процес, в результаті якого всі початкові коди Java були випущені під GPL, за винятком невеликої частини коду, на який Sun не мала авторського права.

В якості основної технології програмування обираємо java/javafx за основне середовище обираємо NetBeans IDE 8.2 (рис. 3.1)

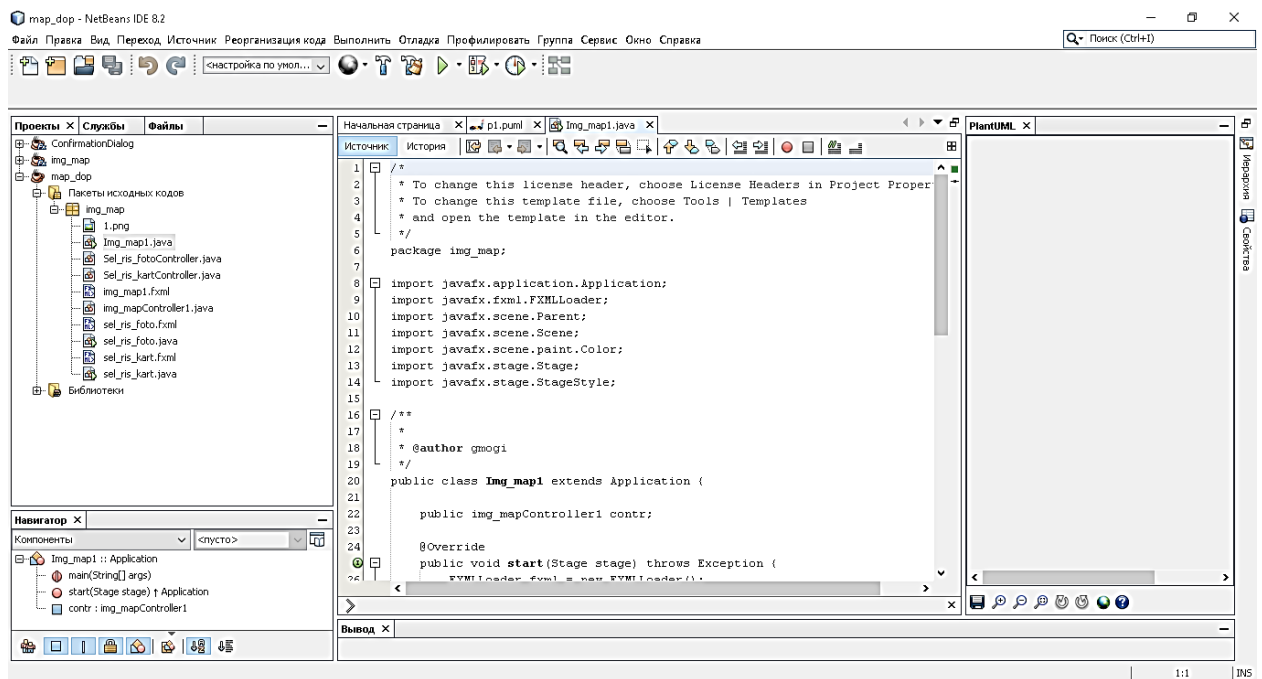


Рис. 3.1 Середовище NetBeans IDE 8.2

3.2. Загальна структура і функціональна схема додатку

Розгляд поставленого завдання передбачає наступні етапи розробки.

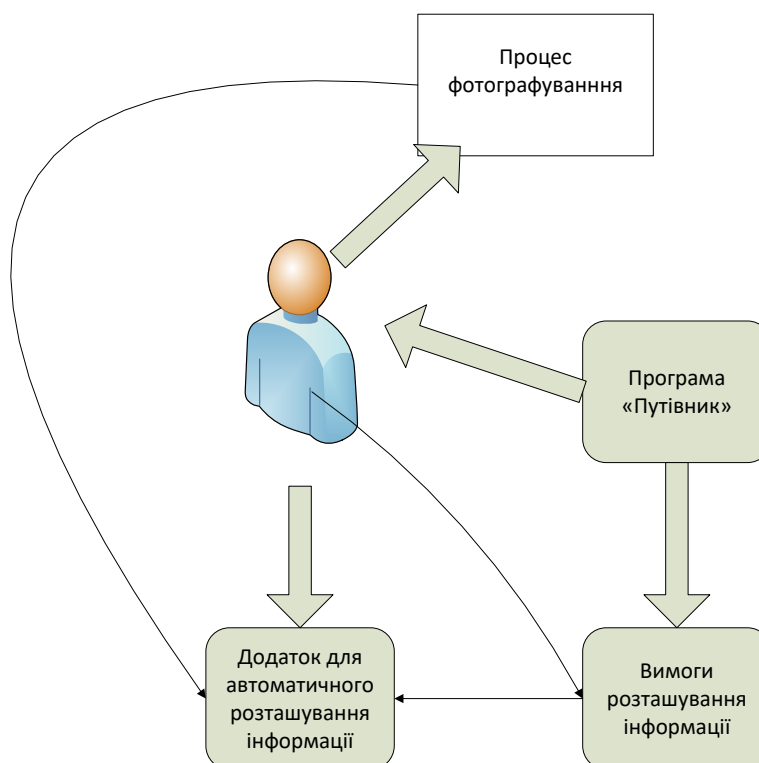


Рис. 3.2 Загальна схема процесу розробки

В цілому процес роботи програми може бути зображено на рисунку 3.3.
В першу чергу завантажується основний модуль (рис. 3.4).

Основний модуль активує меню та відстежує клік на формі (рис.3.5)

Активується вибір мапи програми (рис.3.6), а після його завершення повертаємся у основне вікно додатку (рис. 3.7).

Загальний перелік методів модулю sel_ris_kart наведено на рисунку 3.8.

Основна програма чекає на створення точок мапи. Та відстежує клік мишкою. При цьому є можливість зміни кольору. Після введення точки з'являється вікно з підтвердженням

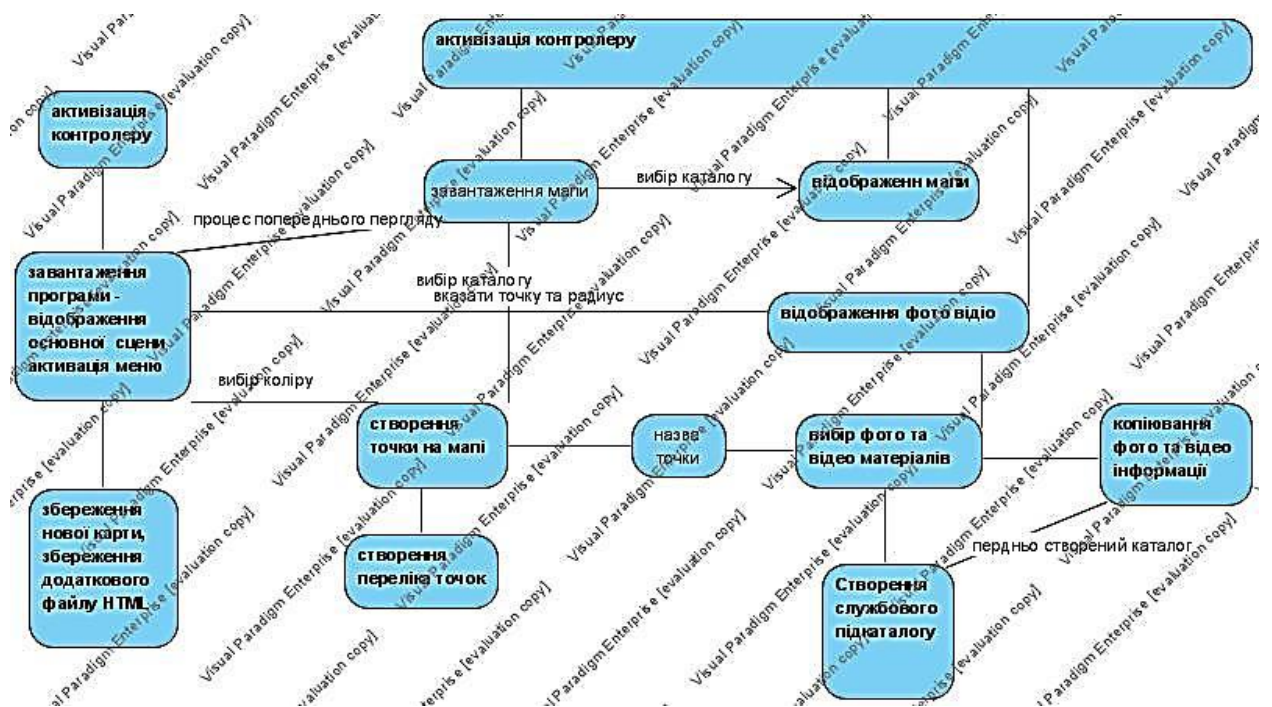
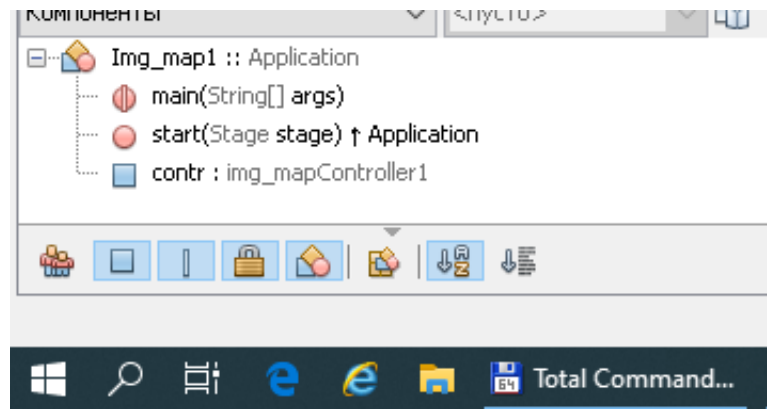
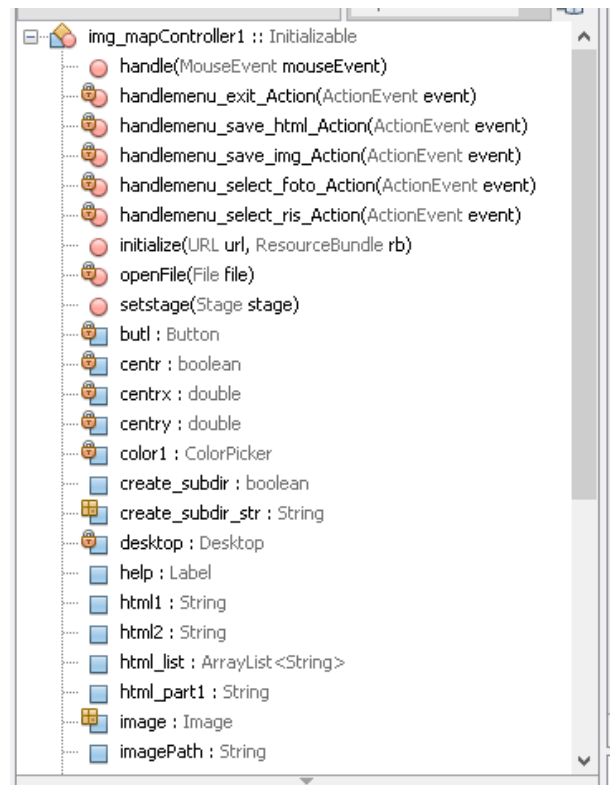


Рис. 3.3 Схема загальних процесів



А) стартовый модуль



Б) контролер стартового модулю

Рис.3.4 Основні методи стартового класу

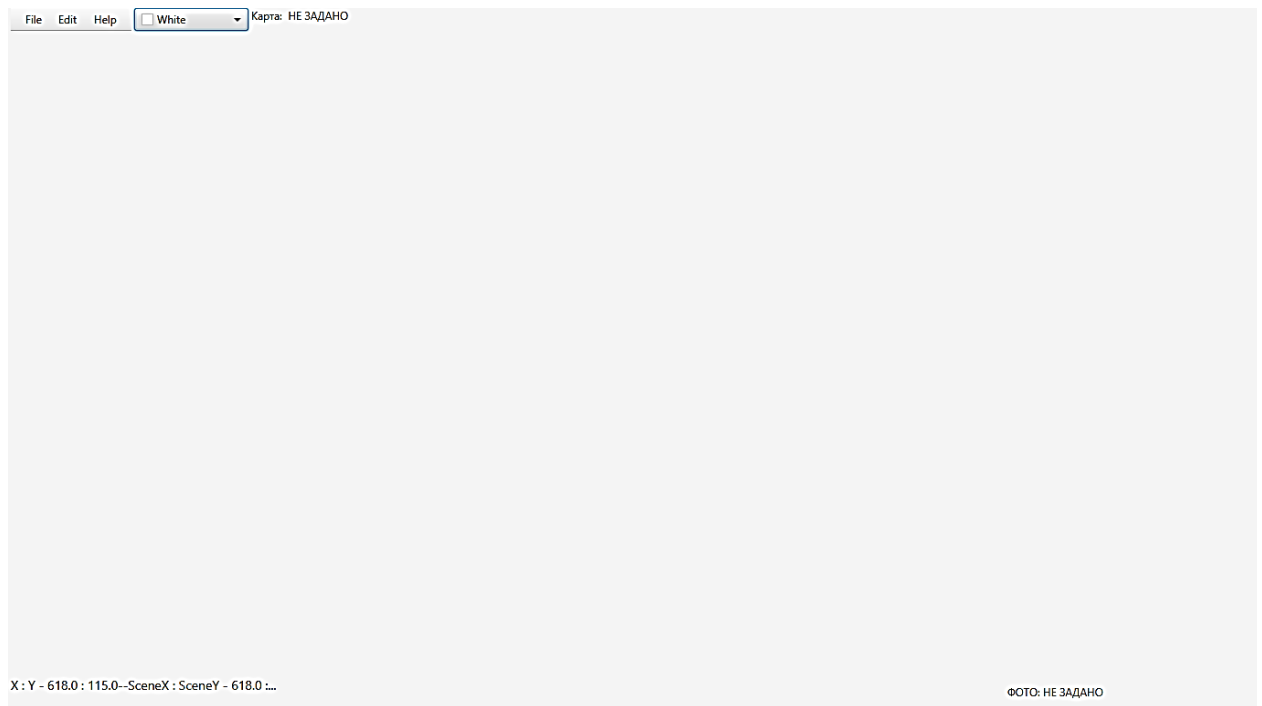


Рис.3.5 Основне вікно додатку

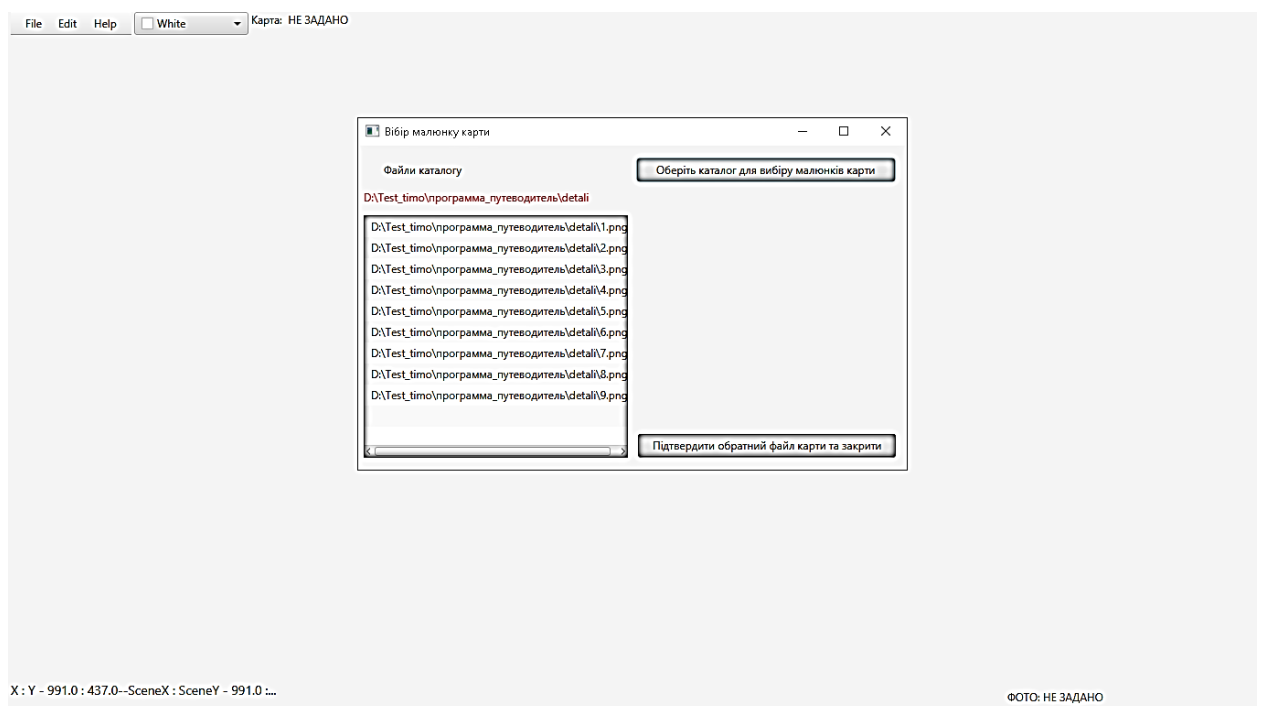
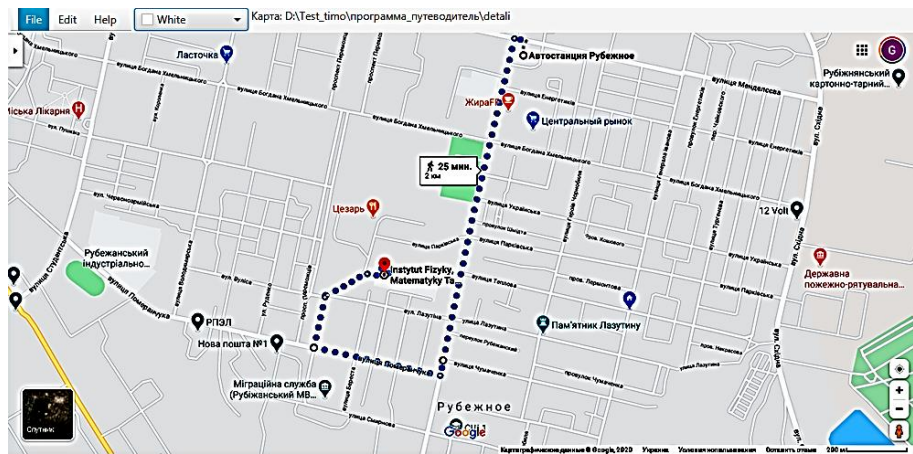


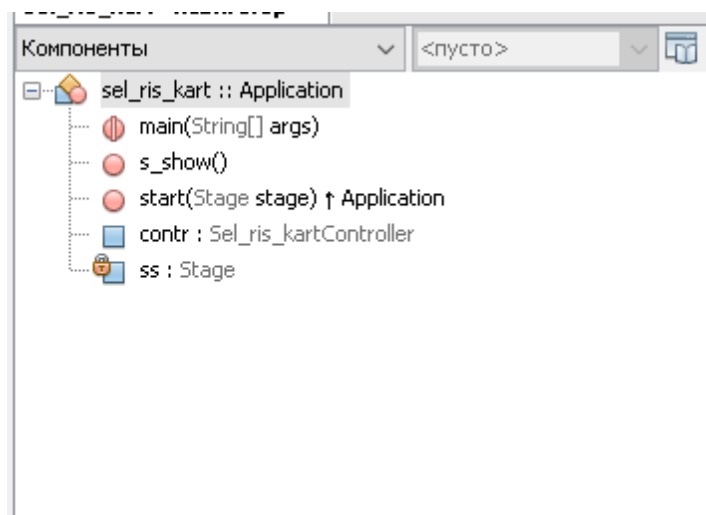
Рис. 3.6 Вибір мапи



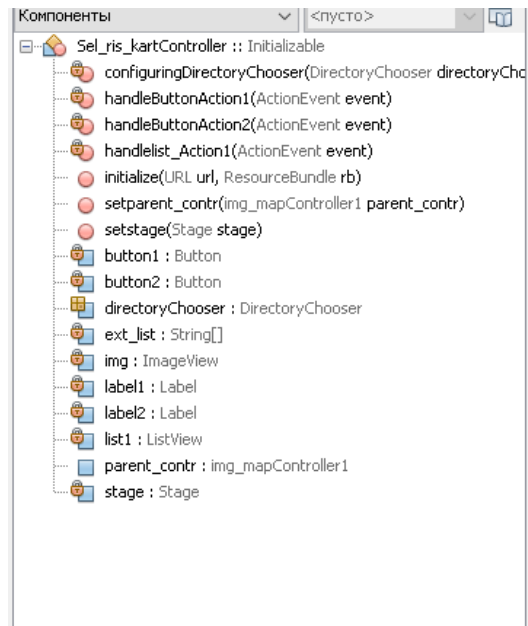
X : Y - 1019.0 : 634.0--SceneX : SceneY - 1019...

ФОТО: НЕ ЗАДАНО

Рис. 3.7 завершения выбора мапи

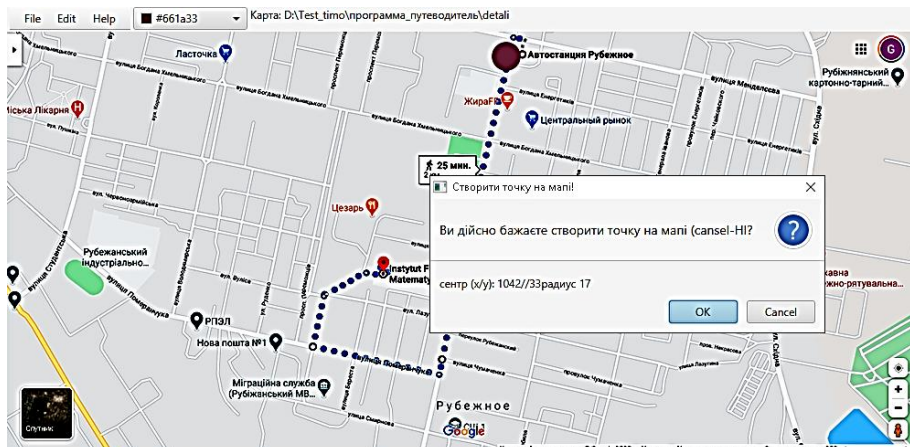


А) основной модуль выбора мапи



Б) контролер модулю вибору мапи

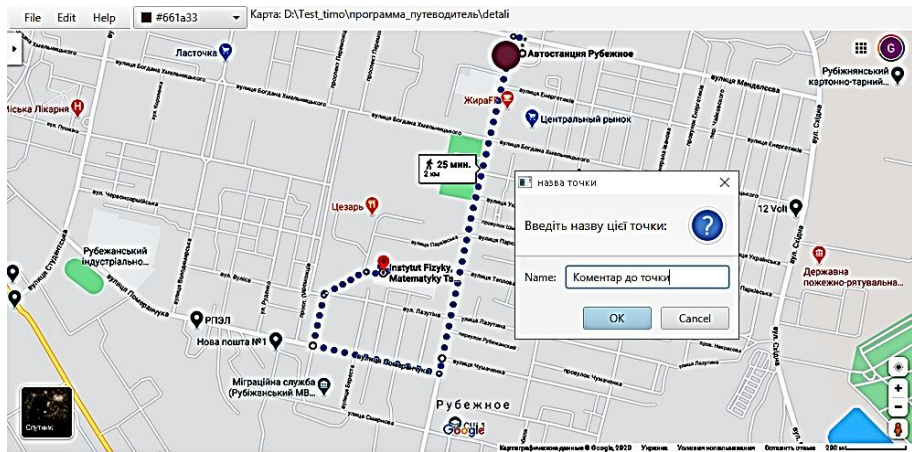
Рис. 3.8 Вибір мапи



X : Y - 555.0 : 40.0--SceneX : SceneY - 555.0 : ...

ФОТО: НЕ ЗАДАНО

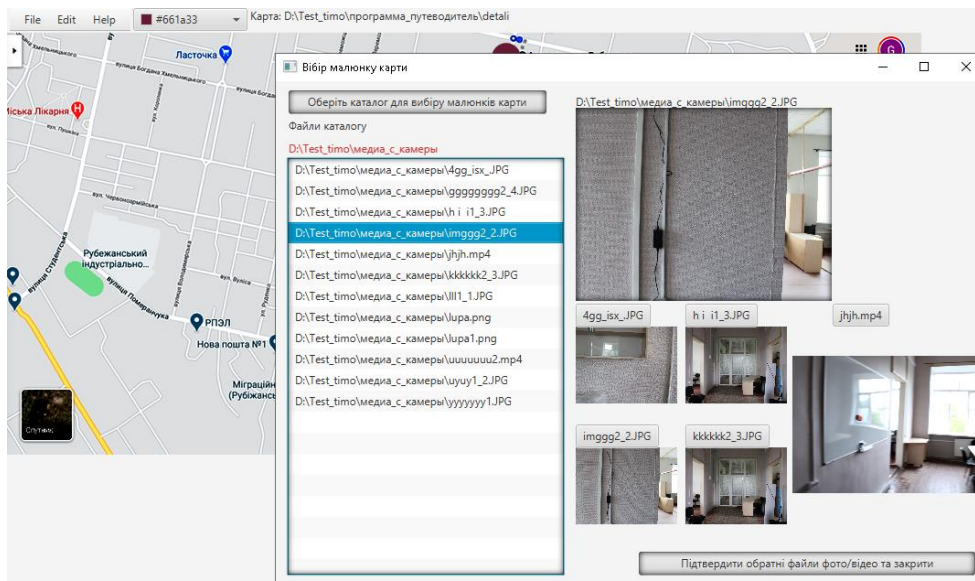
А) перше повідомлення вибору точки



X : Y - 555.0 : 40.0--SceneX : SceneY - 555.0 : ...

ФОТО: НЕ ЗАДАНО

Б) створення коментарю до точки

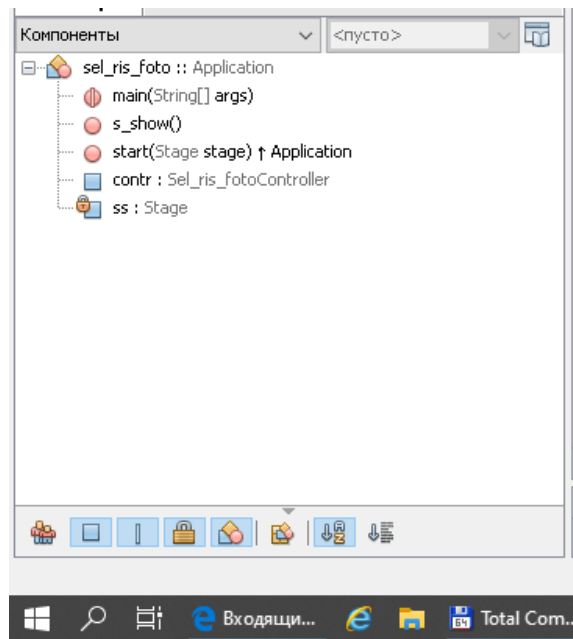


X : Y - 555.0 : 40.0--SceneX : SceneY - 555.0 : ...

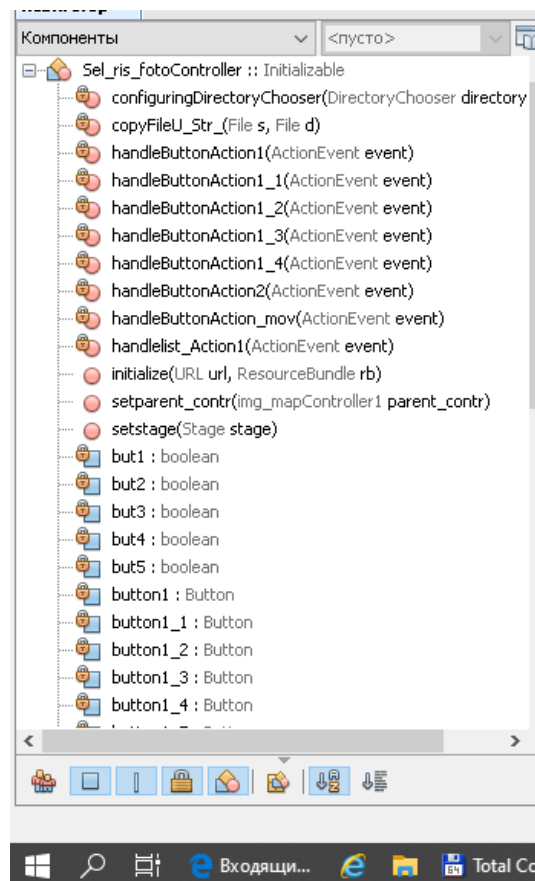
ФОТО: НЕ ЗАДАНО

В) вибір фотоматеріалів

Рис 3.9 Вибір точки на мапі



А) основний модуль вибору фото



Б) контролер вибору фото

Рис. 3.10 Модулю створення та обробки фотоматеріалів
Загальна діаграма класів показана на рисунку 3.11


```
map name="map1_auto_insert">  
<area shape="circle" href="#" coords="546 , 27 , 15 " onClick="loadris(1);"  
      title="Коментар до точки " alt="Коментар до точки" >  
</map>
```

Рис. 3.13 Додатковий файл

ВИСНОВКИ

У дипломній роботі було розглянуто розробку програмного забезпечення для автоматизації підготовки мультимедійного контенту.

В результаті комплексного аналізу сучасних технологій Java було запропоновано в якості мови розробки використовувати технологію JavaFX, в об'єктах та класах якої є необхідні модулі по роботі з мультимедійною інформацією.

В роботі проаналізовано використання тега «мапа» у HTML сторінках, та обрано в якості області вузлів (точек) кругову область. Її спосіб створення потребує всього трьох параметрів: координати центру та радіус.

Аналіз вимог програми «Путівник» дозволив створити технічне завдання до роботи. В результаті встановлено наступне:

1. Ім'я мапи співпадає з ім'ям файлу HTML та має цифру (1,2,3 і таке інше).
2. Для кожної мапи необхідно автоматично створити підкаталог з ім'ям мапи та додатковий html файл та по згоді з розробником ім'я цього файлу повинно бути *_avto_generic.html.
3. Для кожної мапи необхідно створити додатковий малюнок мапи з намальованими точками маршруту на мапі. Ім'я цього файлу повинно бути “*_fon.png”, а для кожної точки мапи у своєму підкаталозі (1, 2,3 і т.і) скопіювати чотири фото з ім'ям *_1.jpg, *_2.jpg, *_3.jpg, *_4.jpg та один відео файл *.mp4.
4. Для зручного вибору файлів програма повинна мати попередній перегляд обраних файлів.

Таким чином розроблено додаток, який автоматизує підготовку мультимедійної інформації для програми «Путівник».

Серед недоліків слід відзначити:

1. Не має можливості зберігти інформацію по мапі та продовжити її редагування у наступному сеансі. Тоб-то все необхідно робити за один запуск.
2. Мається можливість видалити тільки останні точки мапи. Не має можливості видалити будь яку попередньо створену точку мапи. Ці недоліки будуть враховані у наступному.

СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Oracle Helper Center [Електроний ресурс]. – Режим доступу: https://docs.oracle.com/javase/8/docs/technotes/guides/sound/programmer_guide/chapter1.html - Chapter 1: Introduction to the Java Sound API
2. Oracle Helper Center [Електроний ресурс]. – Режим доступу: https://docs.oracle.com/javase/8/docs/technotes/guides/sound/programmer_guide/chapter2.html - Chapter 2: Overview of the Sampled Package
3. Oracle Helper Center [Електроний ресурс]. – Режим доступу: https://docs.oracle.com/javase/8/docs/technotes/guides/sound/programmer_guide/chapter3.html - Chapter 3: Accessing Audio System Resources
4. Wikipedia - the free encyclopedia [Електроний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Java_Media_Framework - Java Media Framework
5. TechRepublic [Електроний ресурс]. – Режим доступу: <https://www.techrepublic.com/blog/software-engineer/process-multimedia-with-the-java-media-framework-api/> - Process multimedia with the Java Media Framework API
6. Wikipedia - the free encyclopedia [Електроний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Real-time_Transport_Protocol - Real-time Transport Protocol
7. .What-when-how [Електроний ресурс]. - Режим доступу: <http://what-when-how.com/javafx-2/working-with-audio-clips-using-the-media-classes-javafx-2/> - Working with Audio Clips (Using the Media Classes) (JavaFX 2)

8. Iteranet IT журнал [Электроний ресурс]. – Режим доступу: <http://iteranet.ru/it-novosti/2014/03/18/vidy-setevyx-tekhnologij-lokalnyx-setej> - Виды сетевых технологий локальных сетей
9. Обучения в Интернет [Электроний ресурс]. – Режим доступу: http://www.lessons-tva.info/edu/telecom-loc/m1t5_3loc.html - 1.5. Базовые технологии или сетевые технологии локальных сетей
10. Информационные технологии [Электроний ресурс]. – Режим доступу: http://kunegin.com/ref1/net_prot/tcpprot.htm - Функции протокола TCP
11. Wikipedia - the free encyclopedia [Электроний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/UDP> - UDP
12. Wikipedia - the free encyclopedia [Электроний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Application_layer - Application layer
13. Wikipedia - the free encyclopedia [Электроний ресурс]. – Режим доступу: [https://en.wikipedia.org/wiki/Broadcasting_\(networking\)](https://en.wikipedia.org/wiki/Broadcasting_(networking)) - Broadcasting (networking)
14. Baeldung [Электроний ресурс]. – Режим доступу: <http://www.baeldung.com/java-broadcast-multicast> – Broadcasting and Multicasting in Java
15. Wikipedia - the free encyclopedia [Электроний ресурс]. – Режим доступу: <https://en.wikipedia.org/wiki/Multicast> - Multicast
16. Wikipedia - the free encyclopedia [Электроний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Client%E2%80%93server_model - Client–server model
17. Шеховцов В. А. Операційні системи / В. А. Шеховцов. – К.: Видавнича група BHV, 2005. – 576 с.

ДОДАТОК

Лістинг коду

```
/*
 * To change this license header, choose License Headers in Project
 * Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package img_map;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
import javafx.stage.StageStyle;

/**
 *
 * @author Borodin
 */
public class Img_map1 extends Application {

    public img_mapController1 contr;

    @Override
    public void start(Stage stage) throws Exception {
        FXMLLoader fxml = new FXMLLoader();
        fxml.setLocation(getClass().getResource("img_map1.fxml"));
        Parent root = (Parent) fxml.load();
        contr = fxml.getController();
        contr.setstage(stage);

        Scene scene = new Scene(root);
        stage.initStyle(StageStyle.TRANSPARENT);
        scene.setFill(Color.TRANSPARENT);
        stage.setScene(scene);
        stage.setTitle("Програма створення додаткових посилань");
        stage.setMaximized(true);
        stage.setMaximized(true);
        // stage.setFullScreen(true);
        stage.show();
        stage.setMaximized(true);
        // stage.setFullScreen(true);
    }

    /**
     * @param args the command line arguments
     */
}
```

```

        public static void main(String[] args) {
            launch(args);
        }
    }
}

```

Контролер основного модуля

```

/*
 * To change this license header, choose License Headers in Project
 * Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package img_map;

import java.net.URL;
import java.io.*;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.*;
import java.awt.Desktop;
import java.io.File;
import java.io.IOException;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.Pane;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.VBox;
import javafx.stage.FileChooser;
import javafx.stage.Stage;
import javafx.scene.image.*;
import javafx.scene.Scene;
import javafx.scene.control.Menu;
import javafx.scene.control.MenuBar;
import javafx.scene.control.MenuItem;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.BorderPane;
import javafx.scene.paint.Color;

```



```

import javafx.scene.shape.Circle;
import javafx.stage.Stage;

import java.awt.image.BufferedImage;
import java.awt.image.RenderedImage;
import java.util.ArrayList;
import java.util.Optional;
import java.util.function.Consumer;
import javafx.embed.swing.SwingFXUtils;
import javafx.scene.SnapshotParameters;
import javafx.scene.control.Alert.AlertType;
import javax.imageio.ImageIO;

/**
 *
 * @author Borodin
 */
public class img_mapController1 implements Initializable {

    @FXML
    private MenuItem men_exit;
    @FXML
    private MenuItem men_select_ris;
    @FXML
    private Button but1;
    @FXML
    public Label help;
    @FXML
    public Label kat_kart;
    @FXML
    public Label kat_foto;
    @FXML
    private ColorPicker color1;
    @FXML
    private AnchorPane pan1;
    private double centrx, centry;
    private boolean centr = false;
    @FXML
    public ImageView img;
    public String imagePath;
    Image image;
    sel_ris_kart sel = null;
    sel_ris_foto sel_f = null;
    public boolean sel_init = false, sel_f_init = false, create_subdir =
        false, sel_init_f_load = false, sel_init_load = false;
    String create_subdir_str = "";

    private Stage stage;
    private Desktop desktop = Desktop.getDesktop();
    public String html1 = "<map name=\"map1_auto_insert\">\n";
    public String html2 = " </map>\n";
    public String html_part1 = "<area shape=\"circle\" href=\"\"#\"
        coords=\"\"";

```

```

private String name_alt_titl = "";

public ArrayList<String> html_list = new ArrayList<String>();

public void setstage(Stage stage) {
    this.stage = stage;
}

@FXML
private void handlemenu_exit_Action(ActionEvent event) {
    System.exit(0);
}

@FXML
private void handlemenu_save_html_Action(ActionEvent event) {
    String html_file = imagePath + "_avto_generic.html";
}

@FXML
private void handlemenu_save_img_Action(ActionEvent event) {
    // image = new Image(imagePath);
    html_list.add(html2 + "\n");
    String html_file = imagePath + "_avto_generic.html";
    StringBuffer sb = new StringBuffer();
    int size = html_list.size();
    for (int i = 0; i < size; i++) {
        sb.append(html_list.get(i));
    }
    String html_in = sb.toString();
    File f_htm = new File(html_file);
    FileOutputStream fol;
    try {
        fol = new FileOutputStream(f_htm);
        fol.write(html_in.getBytes());
        fol.flush();
        fol.close();
    } catch (FileNotFoundException ex) {

        Logger.getLogger(img_mapController1.class.getName()).log(Level.
            SEVERE, null, ex);
    } catch (IOException ex) {

        Logger.getLogger(img_mapController1.class.getName()).log(Level.
            SEVERE, null, ex);
    }

    int w = (int) img.getImage().getWidth();
    int h = (int) img.getImage().getHeight();
    File file = new File(imagePath + "_fon.png");
    WritableImage writableImage = new WritableImage(w, h);
    javafx.scene.snapshot.Parameters sp = new SnapshotParameters();

```

```

pan1.snapshot(null, writableImage);
RenderedImage renderedImage =
    SwingFXUtils.fromFXImage(writableImage, null);
try {
    ImageIO.write(renderedImage, "png", file);
} catch (IOException ex) {

    Logger.getLogger(img_mapController1.class.getName()).log(Level.
        SEVERE, null, ex);
}

}

@FXML
private void handlemenu_select_ris_Action(ActionEvent event) {
    Stage st = new Stage();

    try {
        if (!sel_init_load) {
            sel = new sel_ris_kart();
            sel.start(st);
            sel_init_load = true;
            sel.contr.setparent_contr(this);
        } else {
            sel.s_show();
        }
    } catch (Exception ex) {

        Logger.getLogger(img_mapController1.class.getName()).log(Level.
            SEVERE, null, ex);
    }

}

@FXML
private void handlemenu_select_foto_Action(ActionEvent event) {
    Stage st = new Stage();

    try {
        if (!sel_init_f_load) {
            sel_f = new sel_ris_foto();
            sel_init_f_load = true;
            sel_f.start(st);

            sel_f.contr.setparent_contr(this);
        } else {
            sel_f.s_show();
        }
    } catch (Exception ex) {

        Logger.getLogger(img_mapController1.class.getName()).log(Level.
            SEVERE, null, ex);
    }
}

```

```

    }

}

private void openFile(File file) {
    try {
        desktop.open(file);

    } catch (IOException ex) {

    }
}

@Override
public void initialize(URL url, ResourceBundle rb) {
    sel = null;
    html_list.add(html1);
    // img.setOnMouseMoved(mouseHandler);
    // TODO
}

@FXML
public void handle(MouseEvent mouseEvent) {
    help.setText("X : Y - " + mouseEvent.getX() + " : " +
        mouseEvent.getY() + "--"
        + "SceneX : SceneY - " + mouseEvent.getSceneX() + " : " +
        mouseEvent.getSceneY() + "--"
        + "ScreenX : ScreenY - " + mouseEvent.getScreenX() + " : "
        + mouseEvent.getScreenY());

    if (mouseEvent.isControlDown() &&
        mouseEvent.getEventType().equals(MouseEvent.MOUSE_RELEASED)) {
        int last = pan1.getChildren().size() - 1;
        if (last > 0) {
            pan1.getChildren().remove(last);
            int html_list_leng = html_list.size() - 1;
            html_list.remove(html_list_leng);
            return;
        } else {
            return;
        }
    }
    if (mouseEvent.getEventType().equals(MouseEvent.MOUSE_PRESSED)) {

        if (sel_init == false) {
            handlemenu_select_ris_Action(new ActionEvent());
            return;
        }
        centr = true;
        centry = mouseEvent.getY();
        centrx = mouseEvent.getX();
    }
}

```

```

if (mouseEvent.getEventType().equals(MouseEvent.MOUSE_RELEASED) &&
    centr == true) {
    centr = false;
    double dcentry = Math.abs(mouseEvent.getY() - centry);
    double dcentrx = Math.abs(mouseEvent.getX() - centrx);
    double rad = Math.sqrt(dcentrx * dcentrx + dcentry * dcentry);
    // Circle circle = new Circle(50,Color.BLUE);
    Circle cir1 = new Circle(centrx, centry, rad,
        color1.getValue());
    pan1.getChildren().add(cir1);
    Alert alert = new Alert(AlertType.CONFIRMATION);
    alert.setTitle("Створити точку на мапі!");
    alert.setHeaderText("Ви дійсно бажаєте створити точку на мапі (cansel-HI?)");
    alert.setContentText("центр (x/y): " +
        Integer.toOctalString((int) centrx) + "/" +
        Integer.toOctalString((int) centry) + "радіус " +
        Integer.toOctalString((int) rad));

    // option != null.
    Optional<ButtonType> option = alert.showAndWait();

    if (option.get() == null || option.get() == ButtonType.CANCEL)
    {
        int last = pan1.getChildren().size() - 1;
        if (last > 0) {
            pan1.getChildren().remove(last);
            return;
        } else {
            return;
        }
    }

    // запросить имя точки
    int last = pan1.getChildren().size() - 1;
    TextInputDialog dialog = new TextInputDialog("оглянути " +
        Integer.toString(last));

    dialog.setTitle("назва точки");
    dialog.setHeaderText("Введіть назву цієї точки:");
    dialog.setContentText("Name:");

    Optional<String> result = dialog.showAndWait();

    result.ifPresent(new Consumer<String>() {
        @Override
        public void accept(String name) {
            // this.but1.setText(name);
            name_alt_titl = name;
        }
    });
});

```

```

        String temp1 = Integer.toString((int) centrx) + " , " +
        Integer.toString((int) centry)
            + " , " + Integer.toString((int) rad) + " \"
onClick=\"loadris(" + Integer.toString(last)
            + ");\" title=\"\" + name_alt_titl + " \" \" + "alt=\"\" +
name_alt_titl + "\" >";
        temp1 = html_part1 + temp1 + "\n";
        html_list.add(temp1);

        handlemenu_select_foto_Action(new(ActionEvent()));

    }

}

/*
EventHandler<MouseEvent> mouseHandler = new EventHandler<MouseEvent>()
{

    @Override
    public void handle(MouseEvent mouseEvent) {
        help.setText(mouseEvent.getEventType() + "--"
            + "X : Y - " + mouseEvent.getX() + " : " +
mouseEvent.getY() + "--"
            + "SceneX : SceneY - " + mouseEvent.getSceneX() + " : "
+ mouseEvent.getSceneY() + "--"
            + "ScreenX : ScreenY - " + mouseEvent.getScreenX() + "
: " + mouseEvent.getScreenY());

    }

};
*/
}

```

Модуль вибору карти

```
/*
 * To change this license header, choose License Headers in Project
 * Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package img_map;

import java.awt.event.WindowEvent;
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;

/**
 *
 * @author Borodin
 */
public class sel_ris_kart extends Application {

    public Sel_ris_kartController contr;
    private Stage ss;

    public void s_show() {
        this.ss.requestFocus();
        this.ss.show();
    }

    @Override
    public void start(Stage stage) throws Exception {
        this.ss = stage;
        FXMLLoader fxml = new FXMLLoader();
        fxml.setLocation(getClass().getResource("sel_ris_kart.fxml"));
        Parent root = (Parent) fxml.load();
        contr = fxml.getController();
        contr.setstage(stage);
        Scene scene = new Scene(root);

        stage.setScene(scene);
        stage.setTitle("Вибір малюнку карти ");

        stage.show();
    }
}
```

```

        stage.setOnCloseRequest(new
            EventHandler<javafx.stage.WindowEvent>() {
                @Override
                public void handle(javafx.stage.WindowEvent event) {

                    Alert alert = new Alert(Alert.AlertType.WARNING);
                    alert.setHeaderText("Можливо тільки скрити");
                    alert.showAndWait();
                    //    stage.hide();
                    event.consume();
                }
            });
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        launch(args);
    }
}

```

Контролер вибору мапи

```

/*
 * To change this license header, choose License Headers in Project
 * Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package img_map;

import javafx.scene.image.*;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileFilter;
import java.io.FileNotFoundException;
import java.io.FilenameFilter;
import java.net.URL;
import java.util.ResourceBundle;
import java.util.logging.Level;
import java.util.logging.Logger;
import javafx.beans.InvalidationListener;
import javafx.beans.Observable;
import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;

```



```

import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.*;
import static javafx.scene.input.KeyCode.T;
import javafx.scene.paint.Color;
import javafx.scene.shape.Rectangle;

import javafx.stage.DirectoryChooser;
import javafx.stage.Stage;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;

/**
 * FXML Controller class
 *
 * @author Borodin
 */
public class Sel_ris_kartController implements Initializable {

    @FXML
    private Button button1;
    @FXML
    private Button button2;
    @FXML
    private Label label1;
    @FXML
    private Label label2;
    @FXML
    private ListView list1;
    @FXML
    private ImageView img;
    final DirectoryChooser directoryChooser = new DirectoryChooser();
    private Stage stage;
    public img_mapController1 parent_contr;
    private String ext_list[] = {"PNG", "GIF", "JPG", "JPEG"};

    /**
     * Initializes the controller class.
     */
    @Override
    public void initialize(URL url, ResourceBundle rb) {
        configuringDirectoryChooser(directoryChooser);
        // TODO
    }

    public void setstage(Stage stage) {
        this.stage = stage;
    }

    public void setparent_contr(img_mapController1 parent_contr) {
        this.parent_contr = parent_contr;
    }
}

```

```

@FXML
private void handleList_Action1(ActionEvent event) {
    ObservableList sel_list1 =
        list1.getSelectionModel().getSelectedIndices();

    for (Object o : sel_list1) {
        System.out.println("o = " + o + " (" + o.getClass() + ")");
    }
}

@FXML
private void handleButtonAction2(ActionEvent event) {
    File file = new File(label2.getText());
    if (file.isFile()) {
        String katalog = file.getParent();
        parent_contr.kat_kart.setText("Katra: " + katalog);
        String name_kat = file.getName().substring(0,
            file.getName().lastIndexOf("."));
        File f_new = new File(katalog + File.separator + name_kat);
        f_new.mkdir();
        parent_contr.create_subdir = true;
        parent_contr.create_subdir_str = f_new.getAbsolutePath();
        FileInputStream input;
        try {

            input = new FileInputStream(file);
            Image image = new Image(input);

            parent_contr.img.setImage(image);
            parent_contr.imagePath = file.getParent() + File.separator
+ name_kat ;
        } catch (FileNotFoundException ex) {

        }
        parent_contr.sel_init = true;
        stage.hide();
    }
}

@FXML
private void handleButtonAction1(ActionEvent event) {
    File dir = directoryChooser.showDialog(new Stage());
    if (dir != null) {
        list1.getItems().clear();
        label1.setText(dir.getAbsolutePath());
        File[] ff = dir.listFiles(new FilenameFilter() {
            @Override
            public boolean accept(File dir, String name) {
                int y = ext_list.length;

                for (int i = 0; i < y; i++) {
                    if (name.toUpperCase().endsWith(ext_list[i])) {

```

```

        return true;
    }
}
return false;
}
});
String[] ff_string = dir.list(new FilenameFilter() {
    @Override
    public boolean accept(File dir, String name) {
        int y = ext_list.length;

        for (int i = 0; i < y; i++) {
            if (name.toUpperCase().endsWith(ext_list[i])) {
                return true;
            }
        }
        return false;
    }
});
ObservableList<File> data =
FXCollections.observableArrayList(ff);
list1.setItems(data);
list1.getSelectionModel().getSelectedItem().addListener(new
InvalidationListener() {
    @Override
    public void invalidated(Observable observable) {

        File file = (File)
list1.getSelectionModel().getSelectedItem();

        if (file.isFile()) {

            FileInputStream input;
            try {

                input = new FileInputStream(file);
                Image image = new Image(input);
                img.setImage(image);
                label2.setText(file.toString());
                parent_contr.img.setImage(image);
            } catch (FileNotFoundException ex) {

            }

System.out.println(list1.getSelectionModel().getSelectedItem()
);

System.out.println(list1.getSelectionModel().getSelectedIndex()
);

System.out.println(list1.getSelectionModel().getSelectedItem()
;

```

```

        System.out.println(list1.getSelectionModel().getSelectedIndices
        ());
    }
}
});
/*

list1.getSelectionModel().selectionModelProperty().addListener(new InvalidationListener() {
    @Override
    public void invalidated(Observable observable) {
        String ss= (String)
list1.getSelectionModel().getSelectedItem();
        System.out.println(".invalidated()");
        System.out.println(ss);
        //throw new UnsupportedOperationException("Not supported
yet."); //To change body of generated methods, choose Tools |
Templates.
    }
});
*/
/* list1.getSelectionModel().selectedItemProperty().addListener(
    (ObservableValue<? extends String> ov, String old_val,
        String new_val) -> {
        label1.setText(new_val);
        label1.setTextFill(Color.web(new_val));
    }
    );
*/

// list1.getItems().addAll(ff);
// MultipleSelectionModel<String> select_m =
list1.getSelectionModel();
} else {
    label1.setText("Не обрано каталог");
}
}

private void configuringDirectoryChooser(DirectoryChooser
directoryChooser) {
    // Set title for DirectoryChooser
    directoryChooser.setTitle("Select Some Directories");

    // Set Initial Directory
    directoryChooser.setInitialDirectory(new
        File(System.getProperty("user.home")));
}
}
}

```

Модуль вибору фото

```
/*
 * To change this license header, choose License Headers in Project
 * Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package img_map;

import java.awt.event.WindowEvent;
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;

/**
 *
 * @author Borodin
 */
public class sel_ris_foto extends Application {

    public Sel_ris_fotoController contr;
    private Stage ss;

    public void s_show() {
        this.ss.requestFocus();
        this.ss.show();
    }

    @Override
    public void start(Stage stage) throws Exception {
        this.ss = stage;
        FXMLLoader fxml = new FXMLLoader();
        fxml.setLocation(getClass().getResource("sel_ris_foto.fxml"));
        Parent root = (Parent) fxml.load();
        contr = fxml.getController();
        contr.setstage(stage);
        Scene scene = new Scene(root);

        stage.setScene(scene);
        stage.setTitle("Вибір малюнку карти ");

        stage.show();
    }
}
```

```

        stage.setOnCloseRequest(new
            EventHandler<javafx.stage.WindowEvent>() {
                @Override
                public void handle(javafx.stage.WindowEvent event) {

                    Alert alert = new Alert(Alert.AlertType.WARNING);
                    alert.setHeaderText("МОЖЛИВО ТІЛЬКИ СКРИТИ");
                    alert.showAndWait();
                    // stage.hide();
                    event.consume();
                }
            });
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        launch(args);
    }
}

```

Контролер вибору фото

```

/*
 * To change this license header, choose License Headers in Project
 * Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package img_map;

import javafx.scene.image.*;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FilteredReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ResourceBundle;
import java.util.logging.Level;
import java.util.logging.Logger;
import javafx.beans.InvalidationListener;
import javafx.beans.Observable;
import javafx.beans.value.ObservableValue;

```

```

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.*;
import static javafx.scene.input.KeyCode.T;
import javafx.scene.media.MediaView;
import javafx.scene.paint.Color;
import javafx.scene.shape.Rectangle;

import javafx.stage.DirectoryChooser;
import javafx.stage.Stage;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;
import javafx.scene.media.Media;
import javafx.scene.media.MediaPlayer;

/**
 * FXML Controller class
 *
 * @author Borodin
 */
public class Sel_ris_fotoController implements Initializable {

    @FXML
    private Button button1;

    @FXML
    private Button button2;

    @FXML
    private Button button1_1;

    @FXML
    private Button button1_2;

    @FXML
    private Button button1_3;

    @FXML
    private Button button1_4;
    @FXML
    private Button button1_5;
    @FXML
    private Label label1;
    @FXML
    private Label label2;
    @FXML
    private ListView list1;
    @FXML
    private ImageView img;

```

```

@FXML
private ImageView img1;
@FXML
private ImageView img2;
@FXML
private ImageView img3;
@FXML
private ImageView img4;
@FXML
private MediaView mov;
private Media media_ff;
private MediaPlayer m_p;

final DirectoryChooser directoryChooser = new DirectoryChooser();
private Stage stage;
private boolean but1 = false, but2 = false, but3 = false, but4 = false,
        but5 = false;
private String ext_list[] = {"PNG", "GIF", "JPG", "JPEG", "MP4",
        "MOV"};
public img_mapController1 parent_contr;
String f1 = "";
String f2 = "";
String f3 = "";
String f4 = "";
String fm = "";

/**
 * Initializes the controller class.
 */
@Override
public void initialize(URL url, ResourceBundle rb) {
    configuringDirectoryChooser(directoryChooser);
    // TODO
}

public void setstage(Stage stage) {
    this.stage = stage;
}

public void setparent_contr(img_mapController1 parent_contr) {
    this.parent_contr = parent_contr;
}

@FXML
private void handlelist_Action1(ActionEvent event) {
    ObservableList sel_list1 =
        list1.getSelectionModel().getSelectedIndices();

    for (Object o : sel_list1) {
        System.out.println("o = " + o + " (" + o.getClass() + ")");
    }
}

```



```

@FXML
private void handleButtonAction1_1(ActionEvent event) {
    File file = (File) list1.getSelectionModel().getSelectedItem();

    if (file.isFile()) {

        FileInputStream input;
        try {

            input = new FileInputStream(file);
            Image image = new Image(input);
            img1.setImage(image);
            button1_1.setText(file.getName());
            but1 = true;
            f1 = file.getAbsolutePath();

            // parent_contr.img.setImage(image);
        } catch (FileNotFoundException ex) {

        }

    }

}

@FXML
private void handleButtonAction1_2(ActionEvent event) {
    File file = (File) list1.getSelectionModel().getSelectedItem();

    if (file.isFile()) {

        FileInputStream input;
        try {

            input = new FileInputStream(file);
            Image image = new Image(input);
            img2.setImage(image);
            button1_2.setText(file.getName());
            but2 = true;
            f2 = file.getAbsolutePath();
            // parent_contr.img.setImage(image);
        } catch (FileNotFoundException ex) {

        }

    }

}

@FXML
private void handleButtonAction1_3(ActionEvent event) {
    File file = (File) list1.getSelectionModel().getSelectedItem();

```

```

        if (file.isFile()) {

            FileInputStream input;
            try {

                input = new FileInputStream(file);
                Image image = new Image(input);
                img3.setImage(image);
                button1_3.setText(file.getName());
                but3 = true;
                f3 = file.getAbsolutePath();
                // parent_contr.img.setImage(image);
            } catch (FileNotFoundException ex) {

            }

        }

    }

@FXML
private void handleButtonAction1_4(ActionEvent event) {
    File file = (File) list1.getSelectionModel().getSelectedItem();

    if (file.isFile()) {

        FileInputStream input;
        try {

            input = new FileInputStream(file);
            Image image = new Image(input);
            img4.setImage(image);
            button1_4.setText(file.getName());
            but4 = true;
            f4 = file.getAbsolutePath();
            // parent_contr.img.setImage(image);
        } catch (FileNotFoundException ex) {

        }

    }

}

@FXML
private void handleButtonAction_mov(ActionEvent event) {
    File file = (File) list1.getSelectionModel().getSelectedItem();

    if (file.isFile()) {

        //      media_ff=new Media(file.toURI().toURL().toString());
        media_ff = new Media(file.toURI().toString());
        m_p = new MediaPlayer(media_ff);
    }
}

```

```

        m_p.setAutoPlay(true);
        mov.setMediaPlayer(m_p);
        button1_5.setText(file.getName());
        but5 = true;
        fm = file.getAbsolutePath();
        // parent_contr.img.setImage(image);

    }

}

@FXML
private void handleButtonAction2(ActionEvent event) {
    File file = new File(parent_contr.create_subdir_str);
    if (file.isDirectory()) {

        if (but1 && but2 && but3 && but4 && but5) {
            //начинаем копировать
            int n_point = parent_contr.html_list.size() - 1;
            String name1 = Integer.toString(n_point) + "_1.jpg";
            File f1_res = new File(file, name1);

            String name2 = Integer.toString(n_point) + "_2.jpg";
            File f2_res = new File(file, name2);

            String name3 = Integer.toString(n_point) + "_3.jpg";
            File f3_res = new File(file, name3);

            String name4 = Integer.toString(n_point) + "_4.jpg";
            File f4_res = new File(file, name4);

            String name5 = Integer.toString(n_point) + ".mp4";
            File f5_res = new File(file, name5);

            File isx_1 = new File(f1);
            File isx_2 = new File(f2);
            File isx_3 = new File(f3);
            File isx_4 = new File(f4);
            File isx_5 = new File(fm);
            try {
                this.copyFileU_Str_(isx_1, f1_res);
                this.copyFileU_Str_(isx_2, f2_res);
                this.copyFileU_Str_(isx_3, f3_res);
                this.copyFileU_Str_(isx_4, f4_res);
                this.copyFileU_Str_(isx_5, f5_res);
            } catch (IOException ex) {

                Logger.getLogger(Sel_ris_fotoController.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }
}

```

```

////////////////////////////////////
///
    /*
    File f_new = new File(katalog + File.separator + name_kat);
    f_new.mkdir();
    FileInputStream input;
    try {

        input = new FileInputStream(file);
        Image image = new Image(input);
        parent_contr.img.setImage(image);
        parent_contr.imagePath = file.getParent() + File.separator
+ name_kat + "_fon.png";
    } catch (FileNotFoundException ex) {

    }

    */

////////////////////////////////////
////////////////////////////////////
        stage.hide();

    }

}

private void copyFileU_Str_(File s, File d) throws IOException {
    InputStream is = null;
    OutputStream os = null;
    try {
        is = new FileInputStream(s);
        os = new FileOutputStream(d);
        byte[] buffer = new byte[1024];
        int length;
        while ((length = is.read(buffer)) > 0) {
            os.write(buffer, 0, length);
        }
    } finally {
        is.close();
        os.close();
    }
}
}

```

@FXML

```

private void handleButtonAction1(ActionEvent event) {
    File dir = directoryChooser.showDialog(new Stage());
    if (dir != null) {
        list1.getItems().clear();
        labell1.setText(dir.getAbsolutePath());
        File[] ff = dir.listFiles(new FilenameFilter() {

```

```

@Override
public boolean accept(File dir, String name) {
    int y = ext_list.length;

    for (int i = 0; i < y; i++) {
        if (name.toUpperCase().endsWith(ext_list[i])) {
            return true;
        }
    }
    return false;
}

});
String[] ff_string = dir.list(new FilenameFilter() {
    @Override
    public boolean accept(File dir, String name) {
        int y = ext_list.length;

        for (int i = 0; i < y; i++) {
            if (name.toUpperCase().endsWith(ext_list[i])) {
                return true;
            }
        }
        return false;
    }
});
ObservableList<File> data =
FXCollections.observableArrayList(ff);
list1.setItems(data);
list1.getSelectionModel().getSelectedItem().addListener(new
InvalidationListener() {
    @Override
    public void invalidated(Observable observable) {

        File file = (File)
list1.getSelectionModel().getSelectedItem();

        if (file.isFile()) {

            FileInputStream input;
            try {

                input = new FileInputStream(file);
                Image image = new Image(input);
                img.setImage(image);
                label2.setText(file.toString());
                // parent_contr.img.setImage(image);
            } catch (FileNotFoundException ex) {

            }

        }
    }
});

```

```

        /*

list1.getSelectionModel().selectionModelProperty().addListener(new InvalidationListener() {
    @Override
    public void invalidated(Observable observable) {
        String ss= (String)
list1.getSelectionModel().getSelectedItem();
        System.out.println(".invalidated()");
        System.out.println(ss);
        //throw new UnsupportedOperationException("Not supported
yet."); //To change body of generated methods, choose Tools |
Templates.
    }
});
*/
/* list1.getSelectionModel().selectedItemProperty().addListener(
    (ObservableValue<? extends String> ov, String old_val,
        String new_val) -> {
        label1.setText(new_val);
        label1.setTextFill(Color.web(new_val));
    });
*/

    // list1.getItems().addAll(ff);
    // MultipleSelectionModel<String> select_m =
list1.getSelectionModel();
} else {
    label1.setText("Не обрано каталог");
}
}

private void configuringDirectoryChooser(DirectoryChooser
directoryChooser) {
    // Set title for DirectoryChooser
directoryChooser.setTitle("Select Some Directories");

    // Set Initial Directory
directoryChooser.setInitialDirectory(new
    File(System.getProperty("user.home")));
}
}

```

Міністерство освіти і науки, молоді та спорту України

**Державний заклад «Луганський національний університет
імені Тараса Шевченка»**

Факультет (інститут) Навчально-науковий інститут фізики, математики та
інформаційних технологій

(повна назва)

Кафедра

Інформаційних технологій та систем

(повна назва)

Програма та методика тестування

на виконання програмної розробки (ПР):

**" СИСТЕМА АВТОМАТИЗАЦІЇ ВИБОРУ МУЛЬТИМЕДІЙНОГО
КОНТЕНТУ ДЛЯ ПУТІВНИКА "**

ІТС.ІП04.0323-04-МТ

ПОГОДЖЕНО

Керівник кваліфікаційної роботи

Могильний Г.А..

“ _____ ” 2023р

ВИКОНАВЕЦЬ

Студент групи 4ІПЗ

Бородін М.С..

“ _____ ” 2023р

Потава 2023

ЗМІСТ

1. ОБ'ЄКТ ВИПРОБУВАНЬ.....	3
2. МЕТА ТЕСТУВАННЯ.....	3
3. МЕТОДИ ТЕСТУВАННЯ.....	3
Тест перший. Відображення мапи.	3
Тест другий Створення двох вузлів на мапі червоного та зеленого кольору.	8

1. ОБ'ЄКТ ВИПРОБУВАНЬ

Розроблюваний програмний комплекс повинен автоматизувати підготовки мультимедійної інформації для програми «Путівники».

Для вимірювання якості розроблюваного програмного забезпечення виконано тестування програми.

Програмний комплекс повинен:

- Давати можливість малювання вузлів на мапі маршруту, видаляти вузли (точки) на мапі маршруту;
- Автоматично виводити запит на введення коментаря до вузла мапи маршруту;
- Автоматично створювати новий додатковий малюнок мапи маршруту з вузлами (точками) на цій мапі;
- Автоматично створювати додатковий HTML код для використання у html файлі;
- Надавати можливість вибору фотографій та відео файлу для цього вузла.

2. МЕТА ТЕСТУВАННЯ

Метою тестування є процес перевірки про якість продукту щодо контексту, в якому він повинен використовуватися. До цього процесу входить виконання програми з метою виявлення помилок.

3. МЕТОДИ ТЕСТУВАННЯ

Тестова задача. Вважаються заданими наступні параметри:

1. Рисунок мапи маршруту. Каталог програми «Путівник» (рис.1)
2. Каталог з фото та відео файлами, який завантажено з фотоапарату (рис.2.)

Тест перший. Відображення мапи.

Умови виконання.

1. Відображений файл мапи
2. Створено каталог для медіа файлів програми «Путівник»

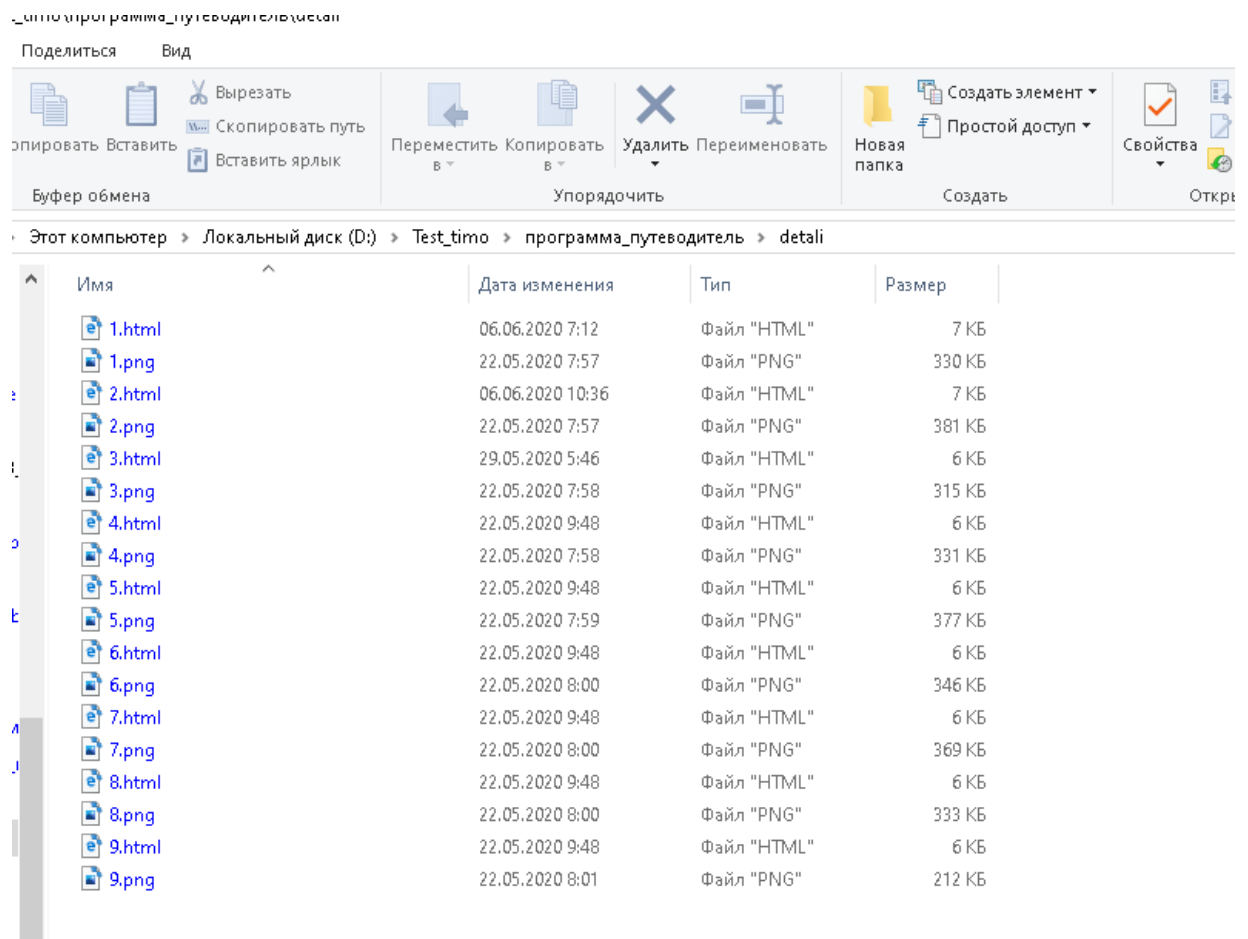


Рис. 1 Каталог програми «Путівник» з мапами .

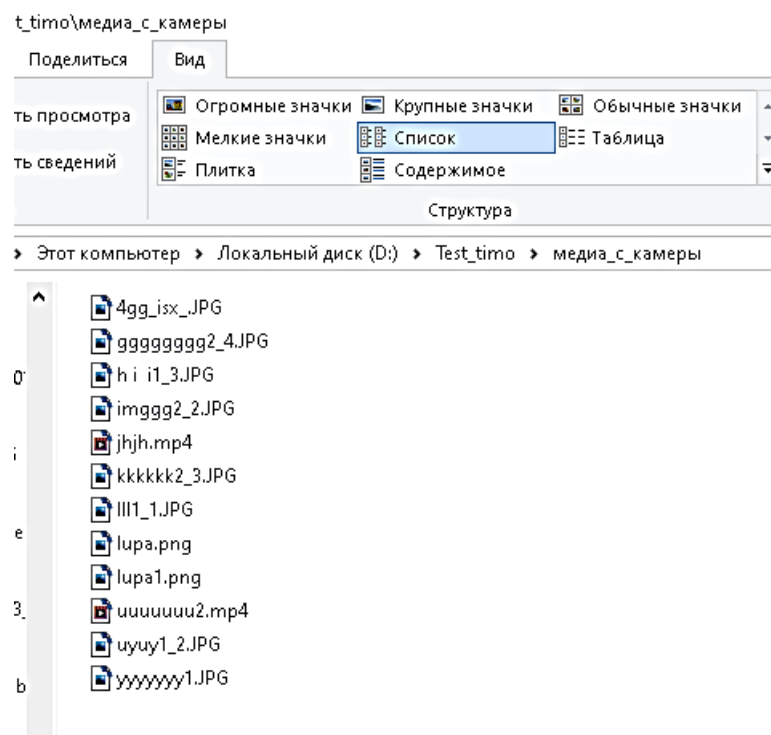


Рис. 2. Каталог файлів з фотоапарату

Результат

1. Запуск програми (рис.3-8)

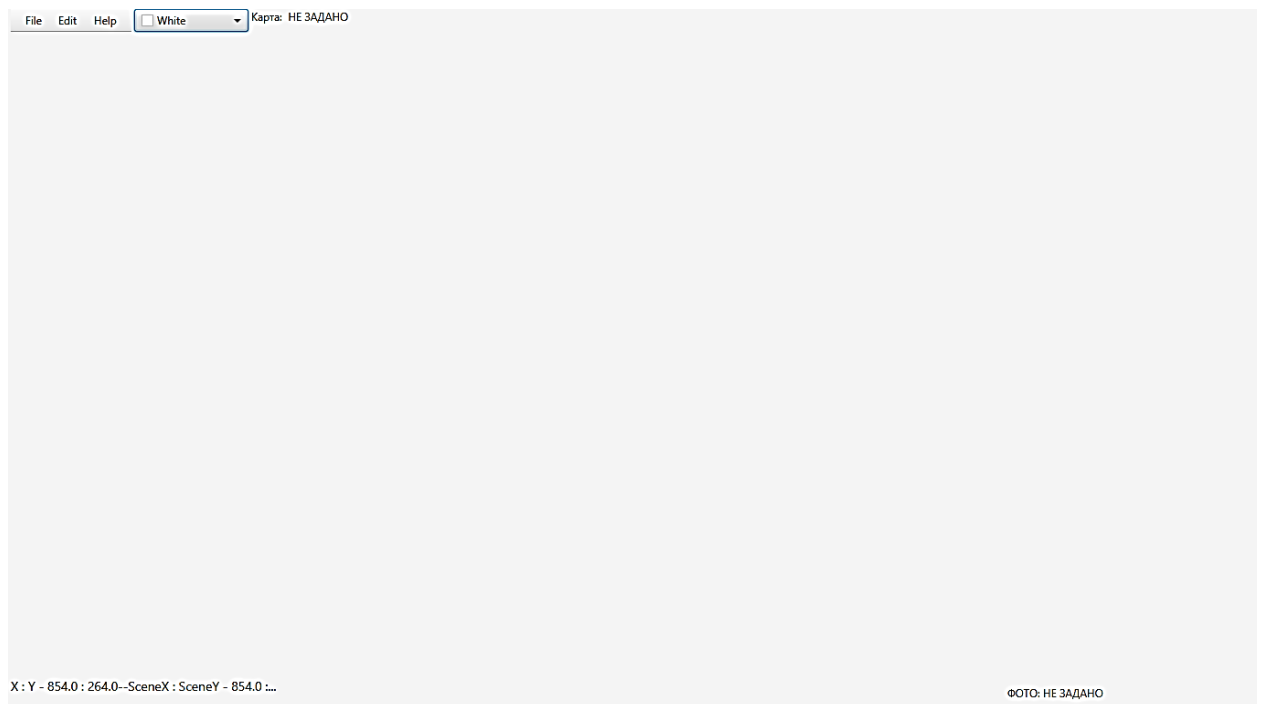


Рис. 3 Вікно старту

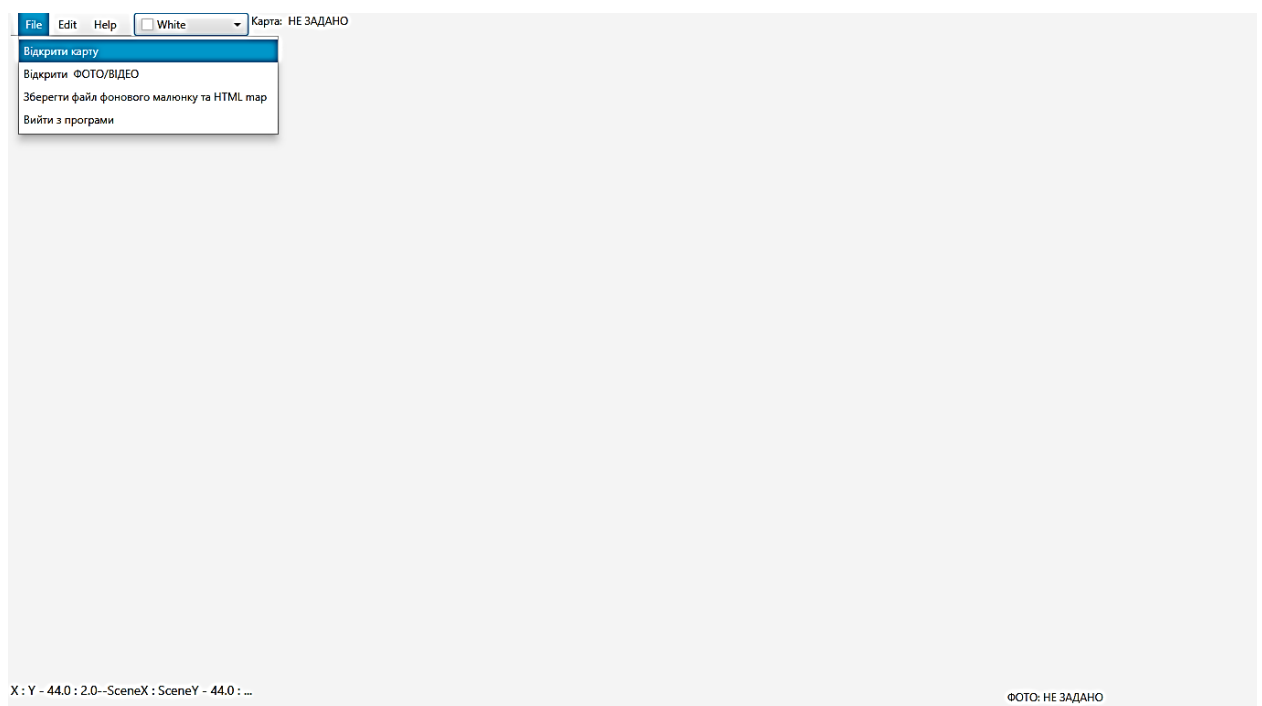


Рис. 4 Старт

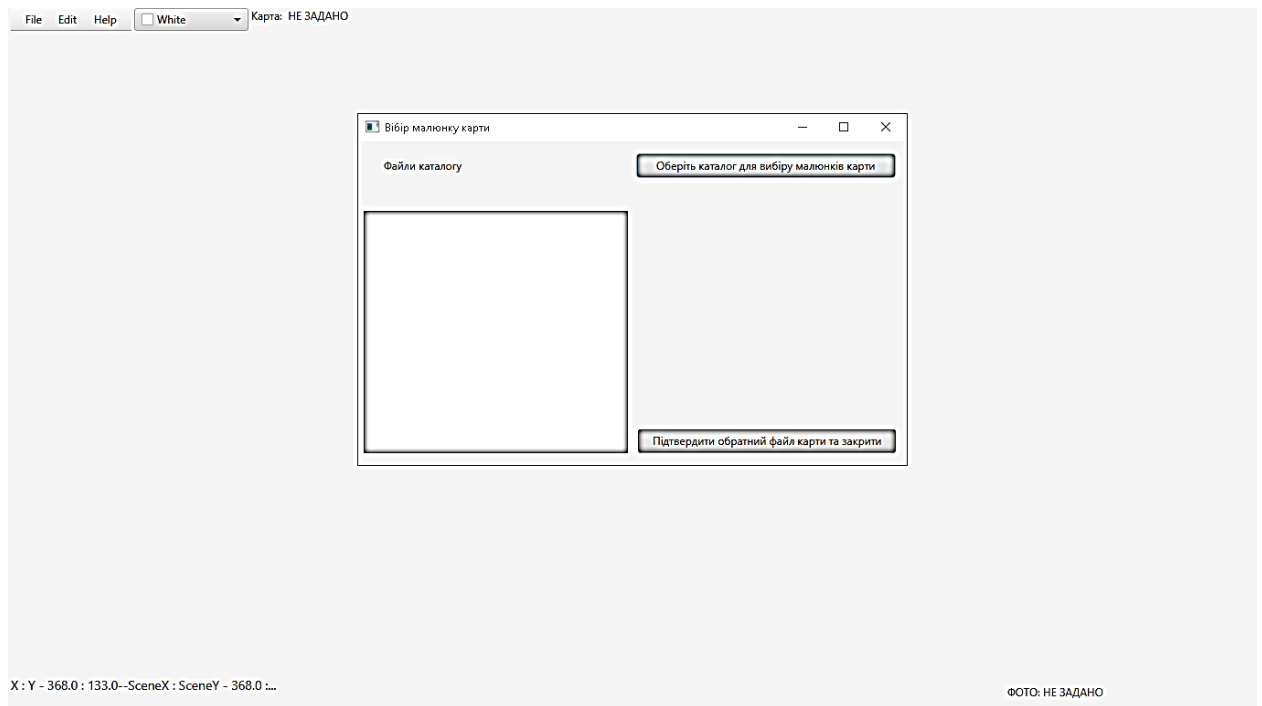


Рис. 5 Вікно вибору каталогу мап – початок

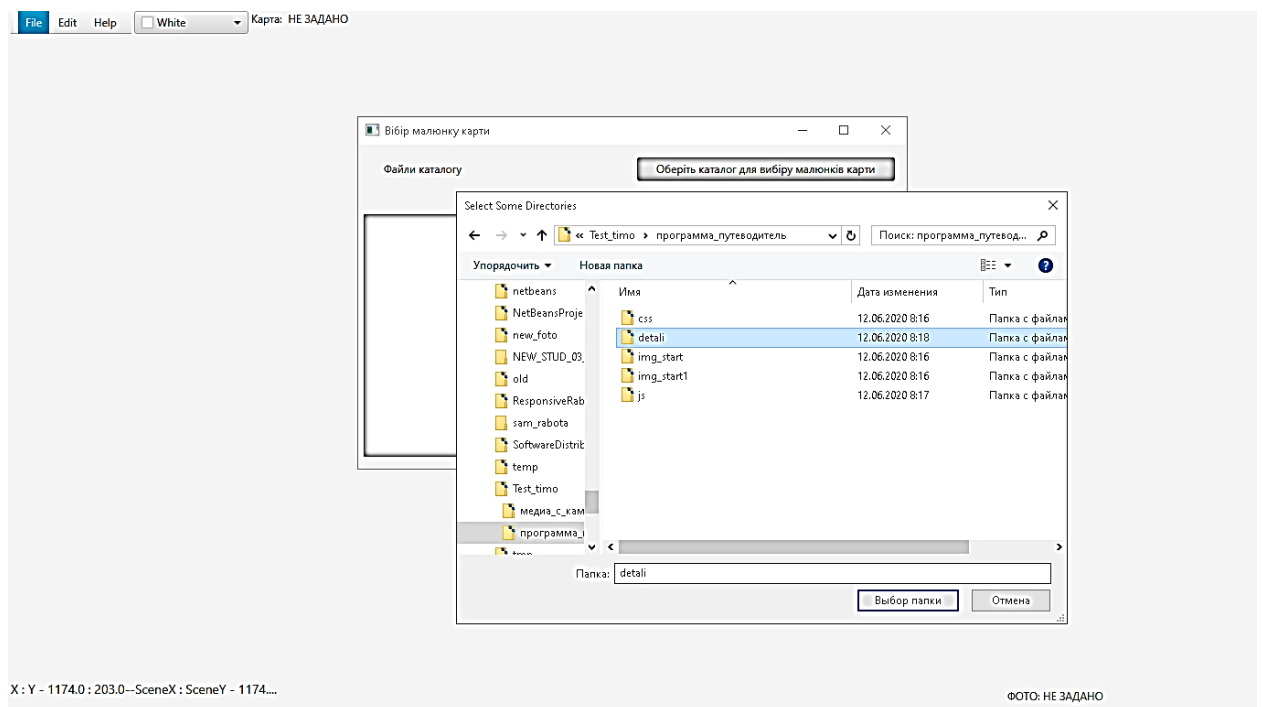


Рис. 6. Вибір каталогу з малюнками мати.

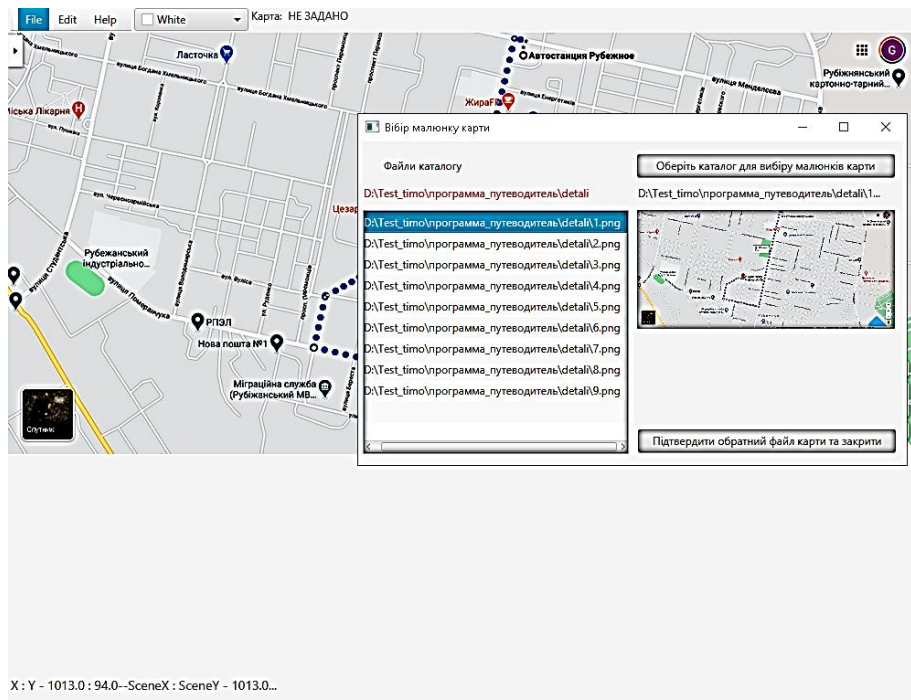


Рис. 7. Папереднє відображення мапи

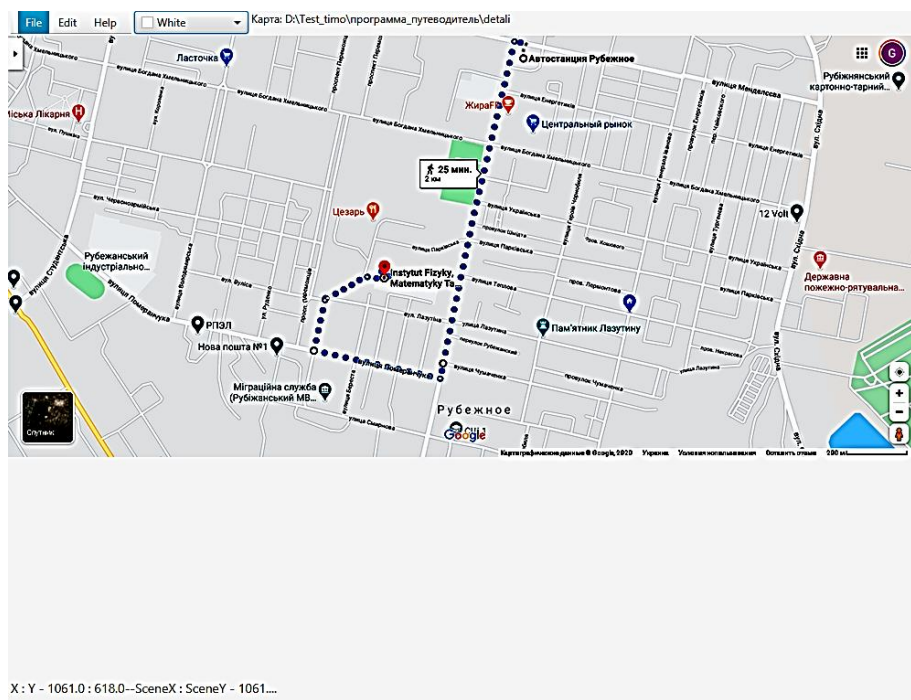


Рис. 8. Обрана мапа

2. Створено підкаталог -результати (рис.9)

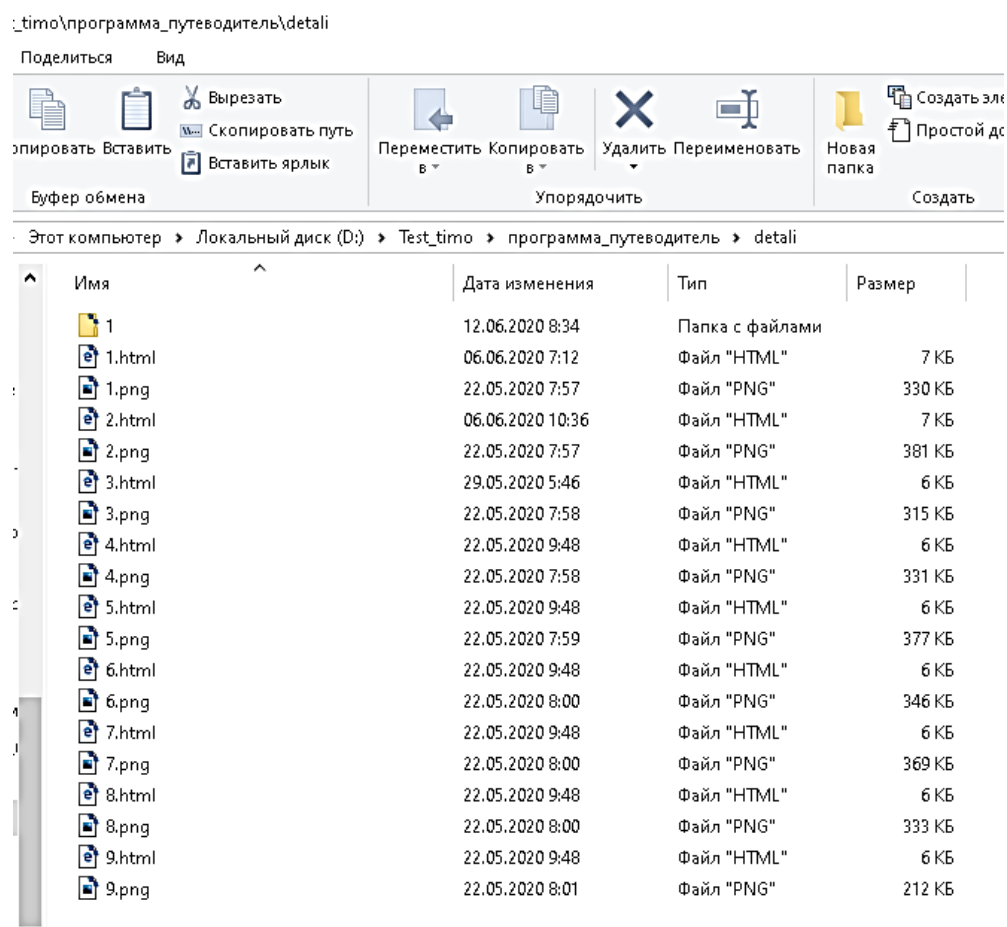
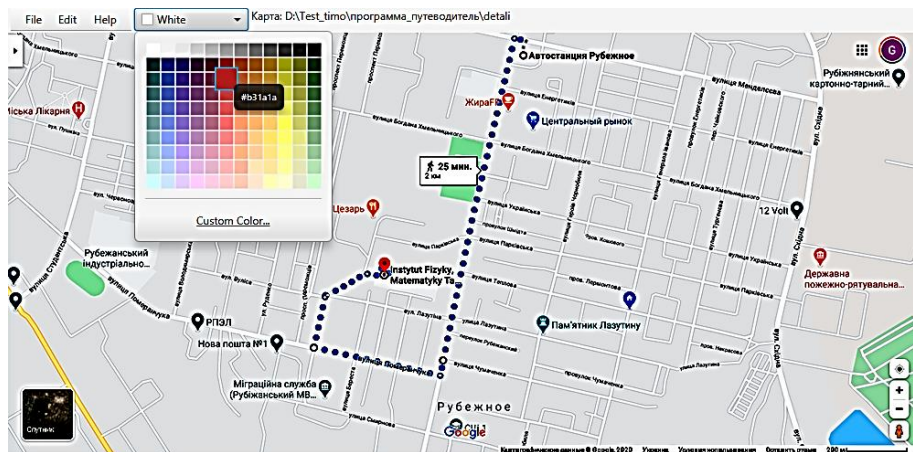


Рис.9 Автоматично створено каталог розташування мультимедійного контенту

Тест другий Створення двох вузлів на мапі червоного та зеленого кольору.
Умови виконання.

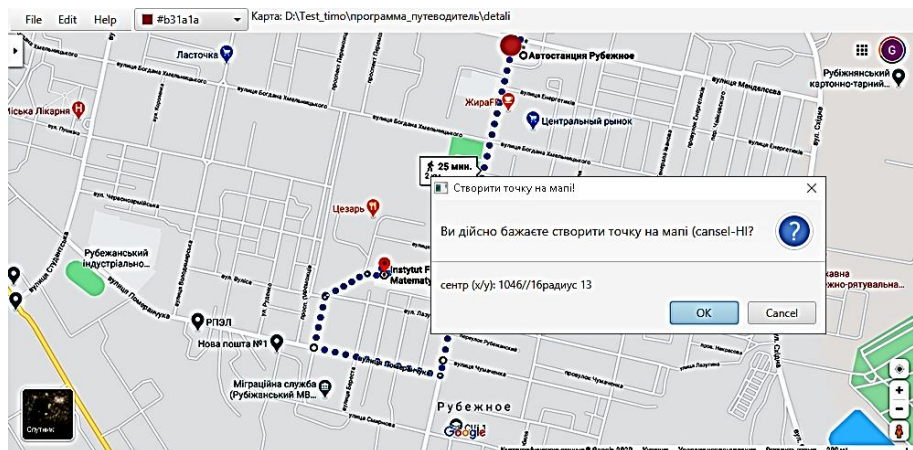
1. Мапа завантажена.
2. Скопійовано медіфайл для медіа файлів програми «Путівник»



X : Y - 252.0 : 2.0--SceneX : SceneY - 252.0 : ...

ФОТО: НЕ ЗАДАНО

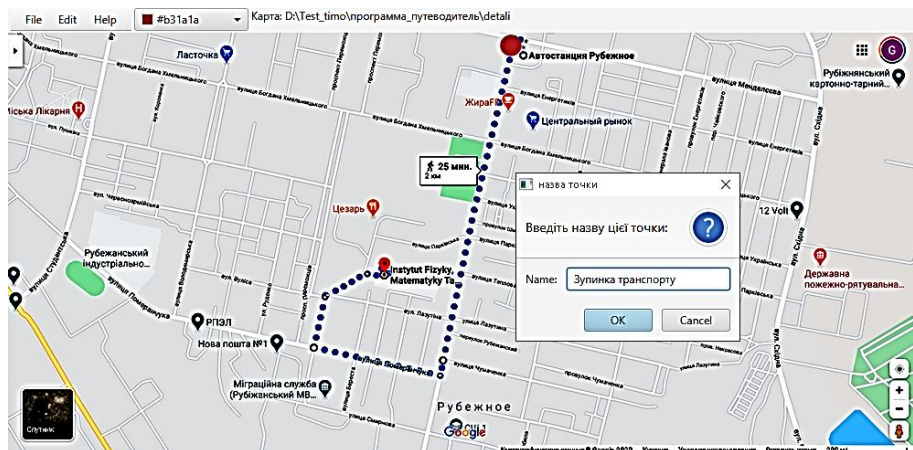
Рис.10 Обрано колір



X : Y - 554.0 : 25.0--SceneX : SceneY - 554.0 : ...

ФОТО: НЕ ЗАДАНО

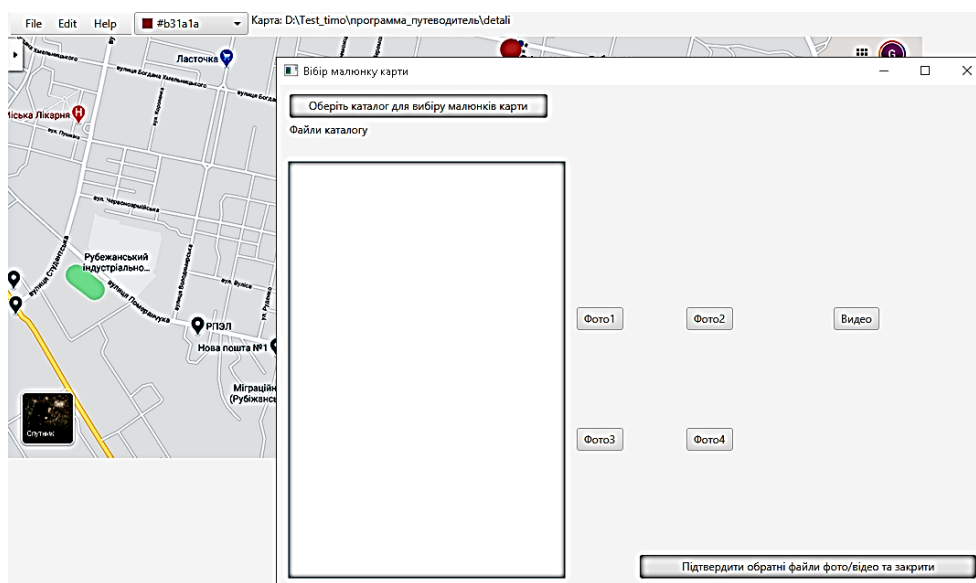
Рис.11 Підтвердження для створення



X : Y - 554.0 : 25.0--SceneX : SceneY - 554.0 : ...

ФОТО: НЕ ЗАДАНО

Рис.12 Введення коментаря



X : Y - 554.0 : 25.0--SceneX : SceneY - 554.0 : ...

ФОТО: НЕ ЗАДАНО

Рис.13 Завантажено вікно вибору медіа файлів

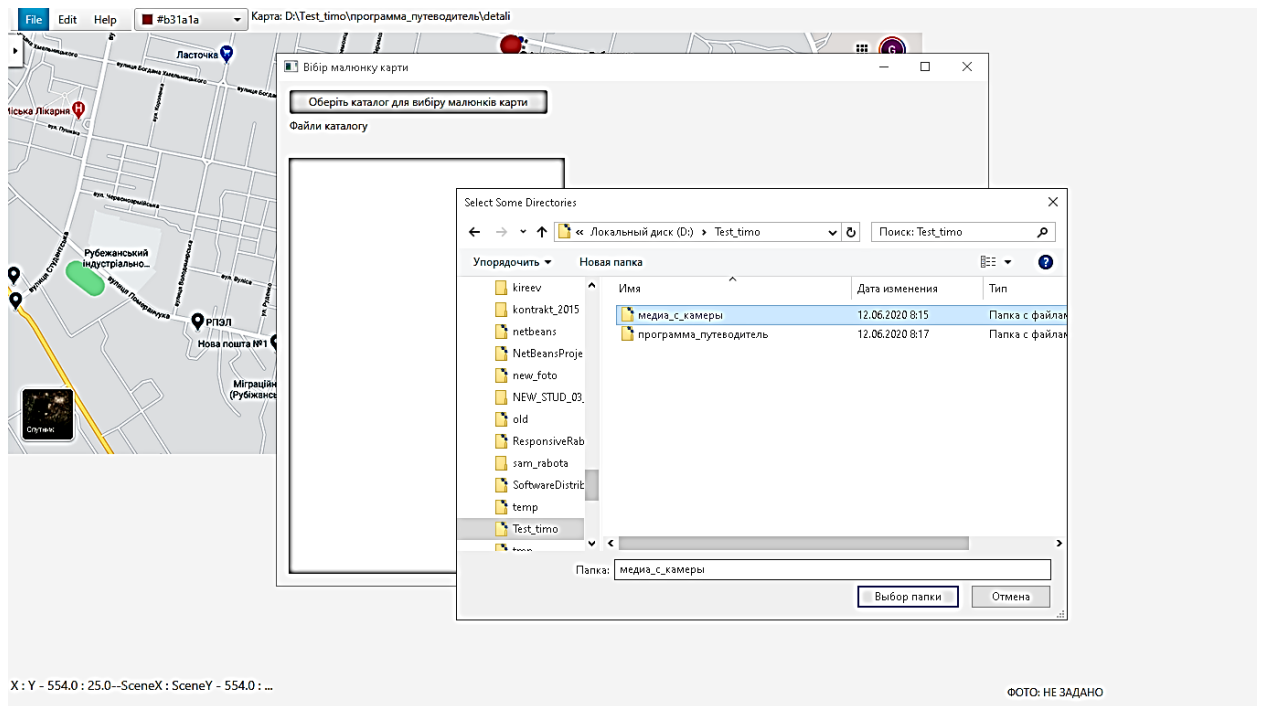


Рис.14 Завантажено вікно вибору медіа файлів

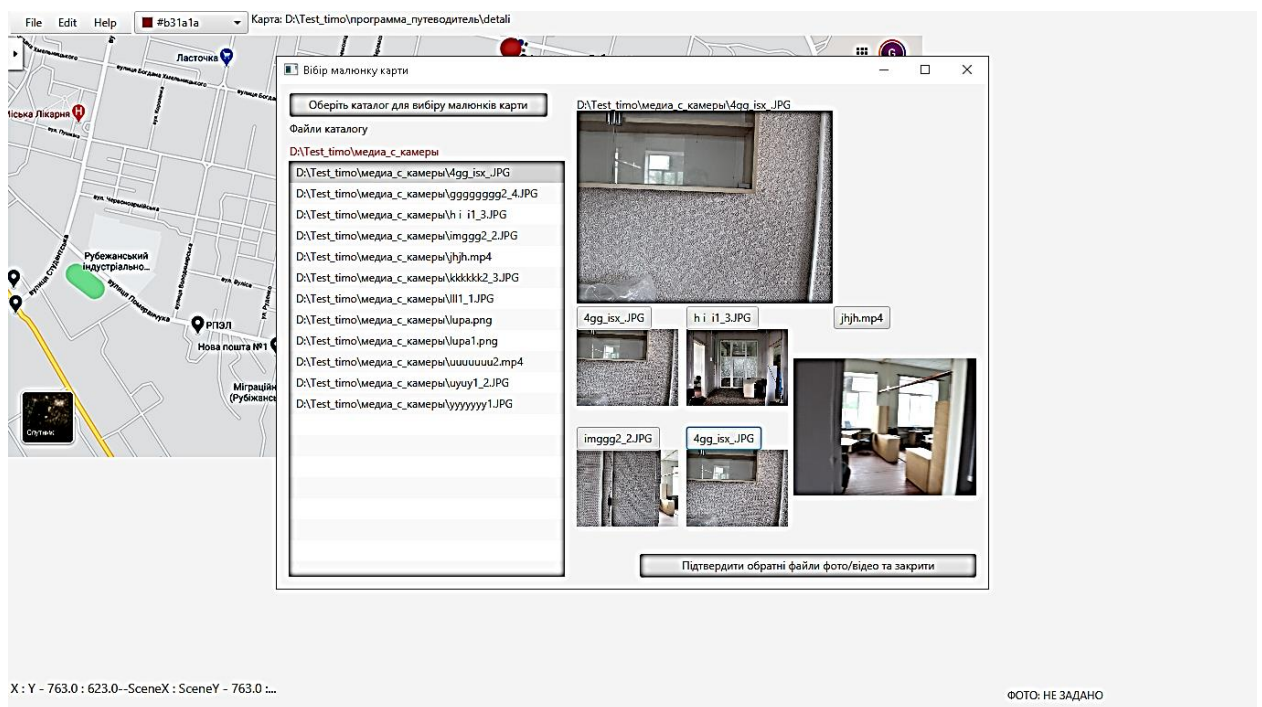


Рис.15 Обрано 4 фото та відеофайл

Результат виконання – створено 5 файлів з іменами 1*** для першої точки мапи (рис.16)

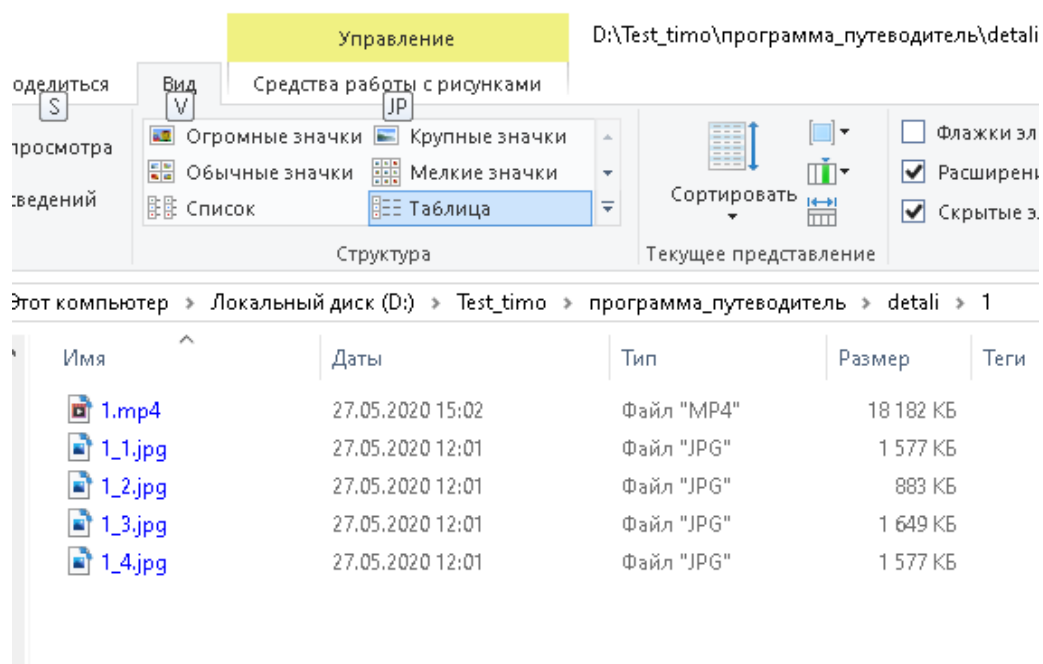


Рис. 15 Створено медіа контент для програми «Путівник»