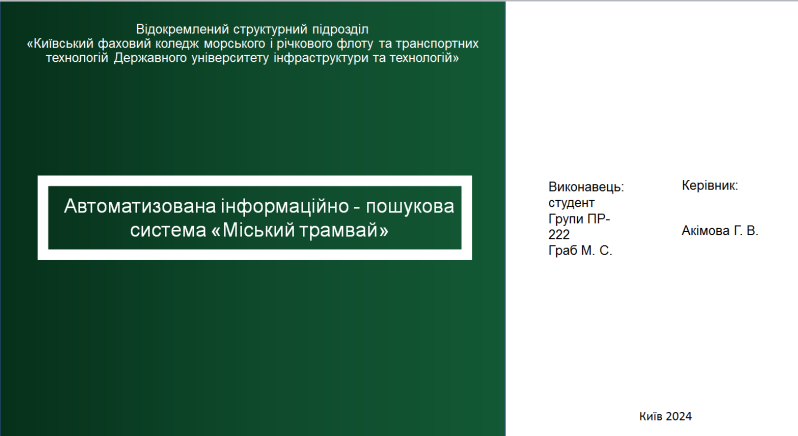
**Тезиси доповіді**

**Автоматизована інформаційно-пошукова система**

Автоматизована інформаційно-пошукова система «Міський трамвай».

Слайд 1.



Дозвольте представити на розгляд курсовий проект на тему «Автоматизована інформаційно - пошукова система «Міський трамвай».

Розробив студент групи ПР\_222 Граб М.С. Керівник проекту - Акімова А.Г.

У курсовому проекті мова піде про створення нескладної програмної системи. Розроблена система дозволить зберігати дані про трамваї та маршрути в зручному для використання вигляді, здійснювати доступ до необхідної інформації у будь-який момент часу, редагувати та видаляти дані, формувати звіти.

Майбутні фахівці з розробки програмного забезпечення повинні мати чітке уявлення про всі етапи створення та експлуатації програм, вміти здійснювати вибір сучасних засобів і методів їх проектування та розробки. Тому основні причини актуальності теми такі:

1. Закріплення практичних навичок програмування, отриманих на заняттях з дисципліни "Основи програмування та алгоритмічні мови";
2. Поглиблення теоретичних і практичних знань у галузі методології програмування та розробки програмних комплексів.

  Слайд 2. Мета проекту



Метою виконання курсового проекту є розробка комп'ютерної програми "Міський трамвай", яка дозволяє оптимальним способом зберігати та оброблювати текстову інформацію.

Ця мета досягається шляхом створення бази даних – текстового файлу на диску,  який містить інформацію про трамваї та маршрути. Для доступу до бази даних, розроблена  проста система управління базою даних, яка дозволяє  виконувати основні операції з даними: зчитувати дані з файлу в оперативну пам'ять, оброблювати їх, записувати дані з оперативної пам’яті у файл.

Практична цінність проекту полягає в тому, що розробка проекту сприяє поглибленню теоретичних і практичних знань в області методології програмування та розробки програм, забезпечує якісне й вчасне отримання інформації про маршрути та трамваї, завдяки автоматизації процесу представлення цих даних.

  Слайд 3. Опис предметної області

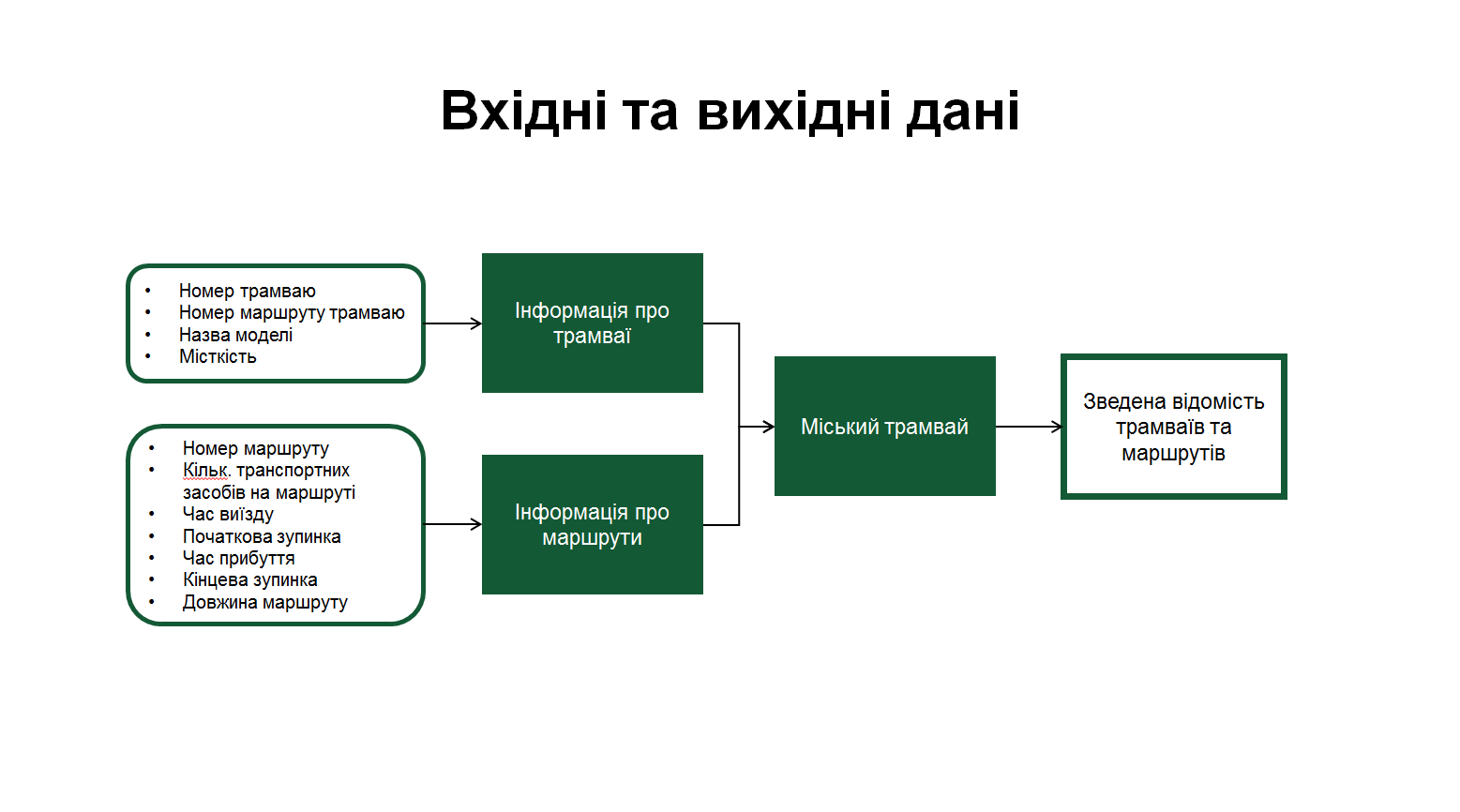


На етапі розроблення проекту,  необхідно зрозуміти, як працює трамвайне депо, тобто описати предметну область.

Трамвайне депо займається організацією руху міських трамваїв. Трамваї відправляються по різним маршрутам. Рух трамваїв регламентується графіком.

Розподіл наявного вагонного парку маршрутами залежить від пасажиропотоків: чим більше пасажирів, тим більше рухомого складу. Час обороту трамваю маршрутом залежить від довжини маршруту та частоти розташування зупинок.

  Слайд 4 . Вхідні та вихідні дані



При додаванні трамваю, працівник повинен ввести такі дані:

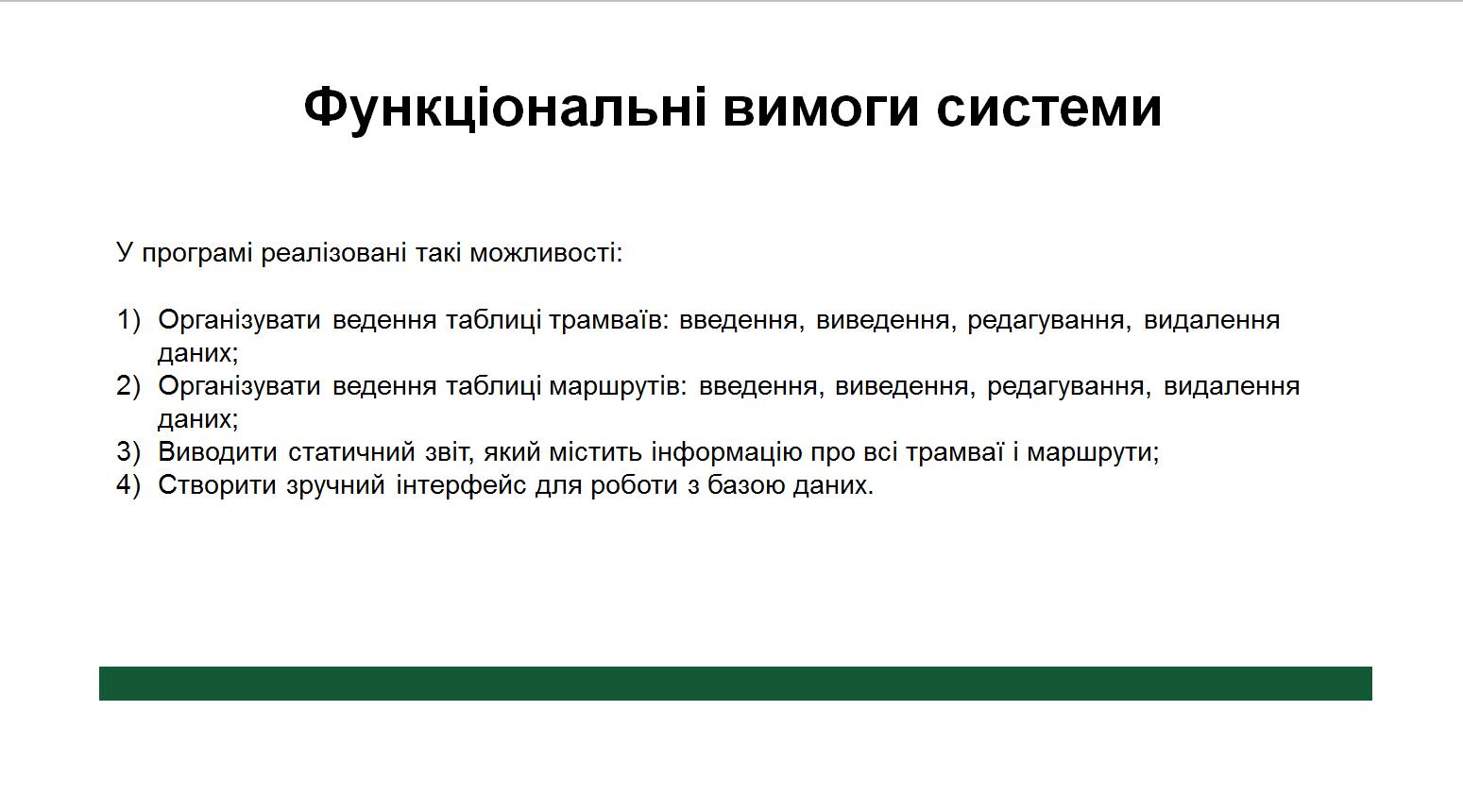
* Номер трамваю
* Номер маршруту трамваю
* Назва моделі
* Місткість

При створенні нового маршруту, працівник повинен ввести такі дані:

* Номер маршруту
* Кількість. транспортних засобів на маршруті
* Час виїзду
* Початкова зупинка
* Час прибуття
* Кінцева зупинка
* Довжина маршруту

Система, яка розробляється, повинна формувати наступні документи: звіт трамваїв, прив’язаних до маршрутів

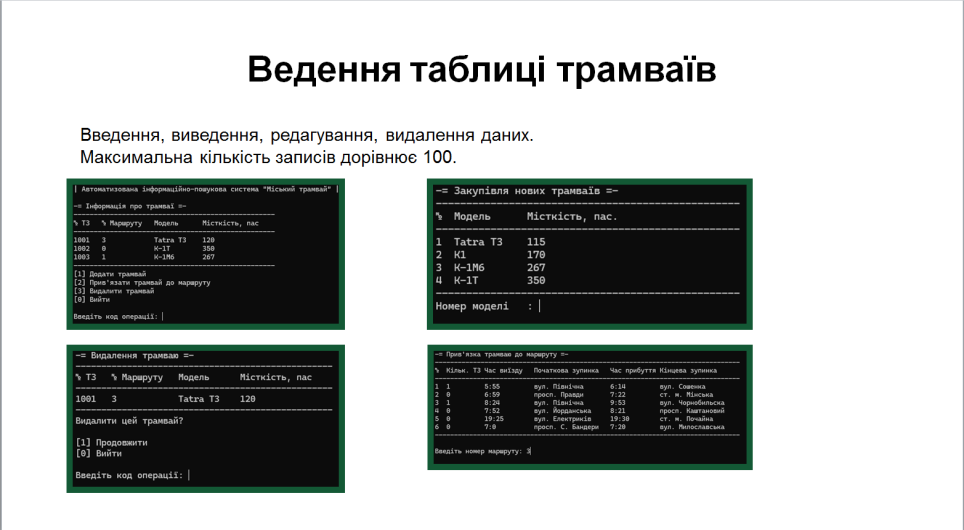
  Слайд 5. Функціональні характеристики системи



У програмі реалізовані такі можливості:

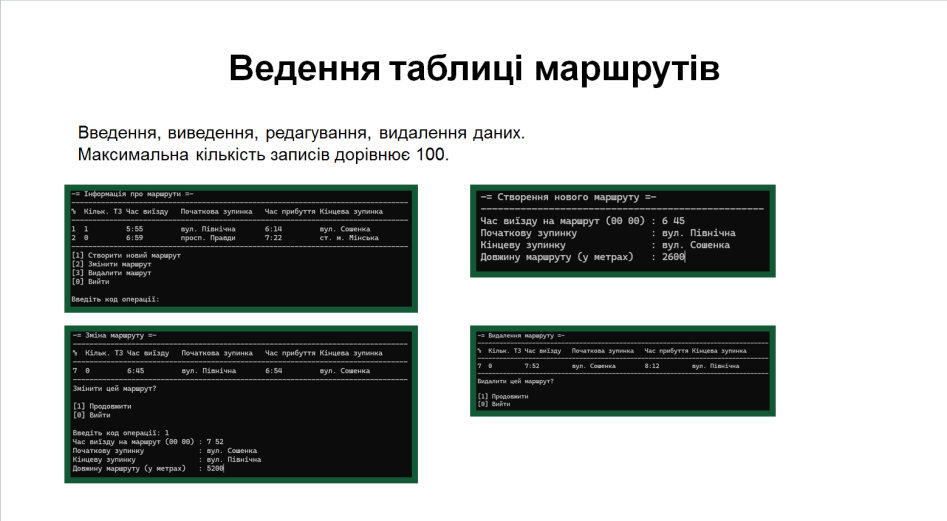
1. Організувати ведення таблиці трамваїв: введення, виведення, редагування, видалення даних;
2. Організувати ведення таблиці маршрутів: введення, виведення, редагування, видалення даних;
3. Виводити статичний звіт, який містить інформацію про всі трамваї і маршрути;
4. Створити зручний інтерфейс для роботи з базою даних.

  Слайд 6. Функціональні характеристики системи: ведення таблиці трамваїв



У базі даних мають знаходитися відомості про трамваї, що були додані. Максимальна кількість записів дорівнює 100.

  Слайд 7. Функціональні характеристики системи: ведення таблиці маршрутів



У базі даних мають знаходитися відомості про маршрути, що були створені. Максимальна кількість записів дорівнює 100.

  Слайд 8. Інструментальні засоби розробки програмного продукту



**Мова програмування**

Для розробки програми було використано мову програмування С++, яка вивчається в курсі "Основи програмування і алгоритмічні мови".

Ця мова має достатню функціональність для вирішення поставленого завдання. Вона поєднує декілька парадигм - об'єктно-орієнтоване програмування, узагальнене програмування і традиційно-процедурне програмування.

**Середовище розробки**

Середовище розробки  - це сукупність програмних засобів, які використовуються  програмістами, для розробки програмного забезпечення. Просте середовище розробки включає текстовий редактор, компілятор, засоби компонувальник та відладчик. Коли ці компоненти зібрані в єдиний програмний комплекс, говорять про інтегроване середовище розробки (Integrated development environment - IDE). Таке середовище представлене однією програмою, не виходячи з якої можна робити увесь цикл розробки.

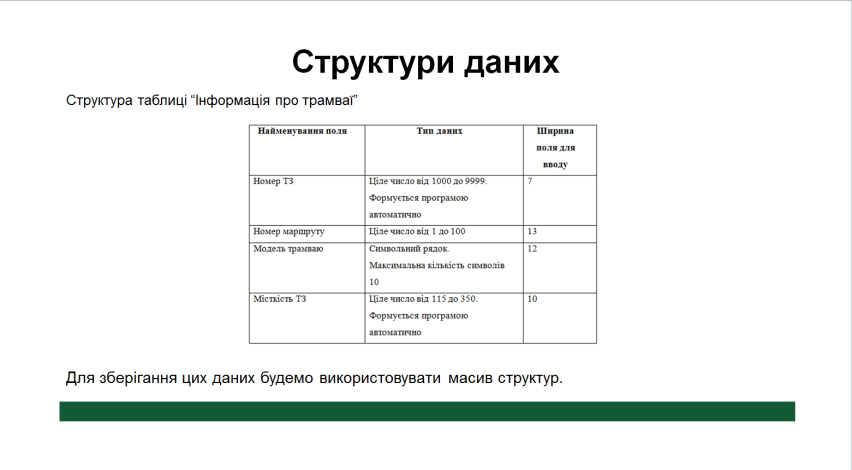
Для розробки проекту, було обрано середовище розробки MS Visual Studio. На сьогодні воно є переважним вибором багатьох розробників, працюючих на платформі Windows. Середовище дозволяє ефективно створювати складні застосування впродовж короткого періоду часу.

Для розробки пояснювальної записки було обрано текстовий процесор, що випускається фірмою Microsoft, що  входить до складу офісного пакету «Microsoft Office». Для оформлення  діаграм було обрано графічний редактор  Microsoft Visio, який входить в пакет «Microsoft Office».

**Тестування програми було зроблено в ручному режимі.**

Коли основна функціональність програмного забезпечення знаходиться в стадії розробки, розробники тестують роботу кожної частини програми вручну. Ручне тестування також використовується на останніх етапах розробки, коли програма має створений інтерфейс користувача. Тестування інтерфейсу передбачає перевірку того, як реальний користувач реагує на те, як розроблено меню і як працює система.

  Слайд 9. Структури даних



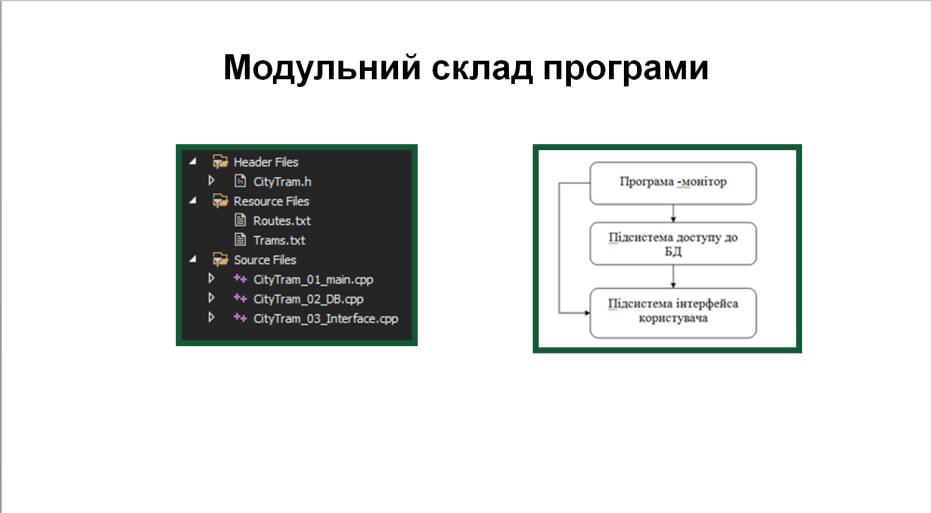
  Слайд 10. Структури даних



База даних  - це текстовий файл, який має свою структуру.  Кожен рядок даних файлу складається з декількох складових частин, кожна з яких має свій тип. Кількість рядків  у файлі обмежена виходячи з постановки завдання.

Дані з файлу зчитуються в оперативну пам'ять. Так як кожен рядок файлу містить різнорідні дані, які стосуються одного маршруту або трамваю, то для опису цих даних створимо структуру. Для зберігання даних про всі маршрути та трамваї будемо використовувати масив структур: одновимірний масив, кожен елемент якого є структурою.

Слайд 11. Модульний склад програми



Для полегшення аналізу програмного коду і внесення до нього змін, розіб’ємо програму на модулі.  Якщо зміни будуть внесені тільки до одного модуля, то потрібно буде перекомпілювати тільки його, і потім виконати повторну зборку усієї програми  компонувальником. Перекомпонування об'єктних модулів програми  виконується набагато швидше, ніж компіляція великого початкового файлу.

Розподіл функцій між модулями зробимо наступним чином:

* 1. Перший модуль складатиме головна функція. У модулі розміщується програма-монітор, яка управляє порядком виконання функцій програмної системи.

*Ім'я модуля*  CityTram\_01\_main.cpp.

* 1. Другий модуль складатимуть функції роботи з базою даних: додавання даних про новий трамвай, прив’язка трамваю до маршруту, видалення трамваю, запис даних про трамвай у файл бази даних, додавання даних про новий маршрут, зміна маршруту, видалення маршруту, запис даних про маршрут у файл бази даних

*Ім'я модуля*  CityTram\_02\_DB.cpp

* 1. Третій модуль складатимуть допоміжні функції: виведення  на екран шапки для таблиці «Інформація про трамваї», виведення на екран таблиці “Інформація про трамваї”, виведення  на екран шапки для таблиці «Інформація про маршрути», виведення на екран таблиці “Інформація про маршрути”, виведення на екран лінії заданого розміру

*Ім'я модуля*  CityTram\_03\_Interface.cpp

* 1. Четвертий модуль -  *заголовний файл*. В ньому визначимо прототипи функцій і деякі іменовані константи. Це визвано тим, що функції розташовані в різних модулях.  Для того, щоб можна було б викликати функцію до того, як вона буде визначена, необхідно задати для кожної з них прототипи. Можна сказати, що заголовний файл  містить *інтерфейс* для деякого набору функцій.

*Ім'я модуля*  CityTram.h

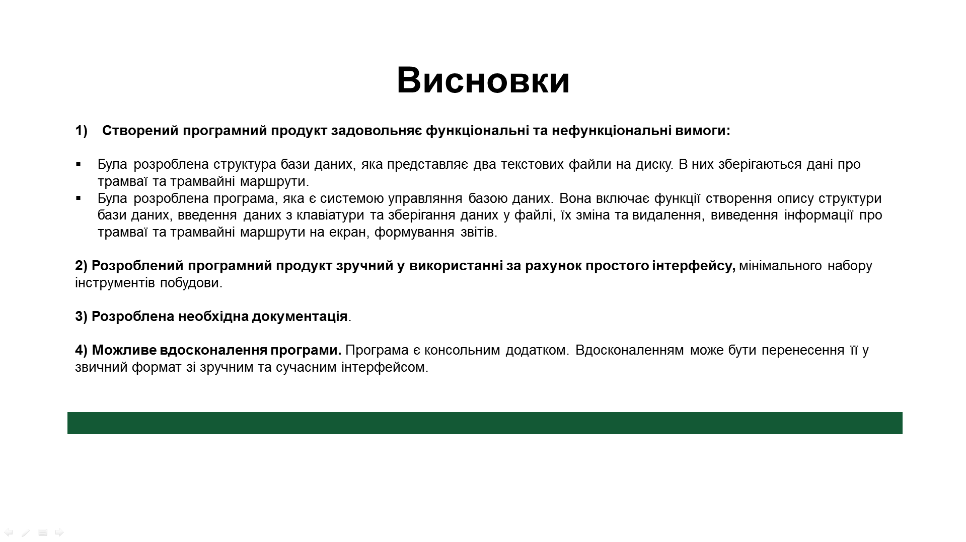
  Слайд 12. Узагальнений алгоритм логічної структури програми

Проект реалізує систему зберігання та обліку даних для таблиць “Інформація про маршрути” та “інформація про трамваї”.

Проект складається з 22 функцій.

| **№** | **Ім’я функції** | **Дії, які виконує функція** | **Ім’я модуля** |
| --- | --- | --- | --- |
|  | main | Головна функція | CityTram\_01\_main.cpp |
|  | addNewTram | Додавання нового трамваю | CityTram\_02\_DB.cpp |
|  | writeTram | Записує дані про трамваї в файл бази даних | CityTram\_02\_DB.cpp |
|  | readTram | Зчитує дані про трамвай з файлу баз даних | CityTram\_02\_DB.cpp |
|  | bindTramToRoute | Прив’язує трамвай до маршруту | CityTram\_02\_DB.cpp |
|  | deleteTram | Видаляє трамвай | CityTram\_02\_DB.cpp |
|  | addOrUpdRoute | Додавання або зміна маршруту | CityTram\_02\_DB.cpp |
|  | writeRoute | Записує дані про маршрути в файл бази даних | CityTram\_02\_DB.cpp |
|  | readRoute | Зчитує дані про маршрут з файлу баз даних | CityTram\_02\_DB.cpp |
|  | calcRouteTimeEnd | Розраховує час прибуття на кінцеву зупинку маршруту | CityTram\_02\_DB.cpp |
|  | breakRouteTime | Розбиває час на маршруті на елементи | CityTram\_02\_DB.cpp |
|  | deleteRoute | Видаляє маршрут | CityTram\_02\_DB.cpp |
|  | findTramIndex | Знаходить індекс трамваю за номером | CityTram\_02\_DB.cpp |
|  | findRouteIndex | Знаходить індекс маршруту за номером | CityTram\_02\_DB.cpp |
|  | tramInfoBanner | Виводить на екран шапку таблиці “Інформація про трамваї” | CityTram\_03\_interface.cpp |
|  | routeInfoBanner | Виводить на екран шапку таблиці “Інформація про маршрути” | CityTram\_03\_interface.cpp |
|  | PrintTramLine | Виводить на екран таблицю “Інформація про трамваї” | CityTram\_03\_interface.cpp |
|  | PrintRouteLine | Виводить на екран таблицю “Інформація про маршрути” | CityTram\_03\_interface.cpp |
|  | printNewTram | Виводить на екран таблицю можливих варіантів трамваїв для покупки | CityTram\_03\_interface.cpp |
|  | printLine | Виводить на екран лінію заданого розміру | CityTram\_03\_interface.cpp |
|  | printReportBanner | Виводить на екран шапку звітної таблиці | CityTram\_03\_interface.cpp |
|  | printReportLine | Виводить на екран звітну таблицю | CityTram\_03\_interface.cpp |

  Слайд 13. Висновки



Висновки

1. Створений програмний продукт задовольняє функціональні та нефункціональні вимоги:

* Була розроблена структура бази даних, яка представляє два текстових файли на диску. В них зберігаються дані про трамваї та трамвайні маршрути.
* Була розроблена програма, яка є системою управляння базою даних. Вона включає функції створення опису структури бази даних, введення даних з клавіатури та зберігання даних у файлі, їх зміна та видалення, виведення інформації про трамваї та трамвайні маршрути на екран, формування звітів.

1. Розроблений програмний продукт зручний у використанні за рахунок простого інтерфейсу, мінімального набору інструментів побудови.
2. Розроблена необхідна документація.
3. Можливе вдосконалення програми. Програма є консольним додатком. Вдосконаленням може бути перенесення її у звичний формат зі зручним та сучасним інтерфейсом.