Міністерство освіти та науки України

Запорізький національний технічний університет

кафедра програмних засобів

з/к № //ХХХХХХХХХХХ//

**Пояснювальна записка**

з дисципліни «Об’єктно-орієнтоване програмування»

до курсової роботи на тему

//«Створення візуального інтерфейсу»//

Виконав

ст. гр. КНТ-415 Д. А. Рязанов

ст. гр. КНТ-415 О. В. Сергеев

ст. гр. КНТ-415 О. А. Бірюков

Прийняв

професор Г.В. Табунщик

доцент Т.І. Каплієнко

ст.. викл. Н.О. Миронова

Запоріжжя

2016

# ЗМІСТ

////////////////////////////////////////

# Перелік умовних позначень, символів, одиниць, скорочень і термінів

МП – мова програмування;

ПП – програмний продукт;

ПЗ – програмне забезпечення;

Qt – кроссплатформена бібліотека розробки GUI на C++;

MinGW – Minimalist GNU for Windows;

STL (Standard Template Library) – бібліотека для C++, що містить набір узгоджених узагальнених алгоритмів, контейнерів, засобів доступу до їхнього вмісту і різних допоміжних функцій;

JSON (JavaScript Object Notation) – текстовий формат обміну даними, заснований на JavaScript.

SQLite (Structured Query Language Lite) компактная [встраиваемая](https://ru.wikipedia.org/wiki/%D0%92%D1%81%D1%82%D1%80%D0%B0%D0%B8%D0%B2%D0%B0%D0%B5%D0%BC%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94" \o "Встраиваемая СУБД) [реляционная база данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85" \o "Реляционная база данных)

XML (Extensible Markup Language) - запропонований консорціумом World Wide Web (W3C) стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет.

# Вступ

Програма типу Ticketland – веб-сервісу, що являє собою он-лайн касу для придбання квитків. Призначена для зручного придбання квитків з любого місця де є інтернет.

Метою роботи є здобуття навичок створення програми за допомогою засобів ООП.

Задачею є розробка програми, аналогу Ticketland. Під час розробки необхідно реалізувати ергономічний інтерфейс, ефективні алгоритми, для роботи з програмою та реалізувати роботу з даними в форматах SQLite і JSON.

В процесі реалізації даної задачі використати основні властивості об’єктно-орієнтованого програмування. В якості інструментів для розробки було обрано мову програмування C++ в середовищі Qt. В процесі розробки необхідно виконати наступні завдання:

- провести аналіз предметної області;

- розробити відповідні структури даних;

- створити візуальний інтерфейс;

- розробити програму;

- провести тестування;

- оформити пояснювальну записку.

# 1 Опис предметної області

## 1.1 Огляд існуючих методів вирішення завдання

Предметна область даного проекту – веб-сервіс Ticketland.

Ticketland являє собою он-лайн касу для користувачів, які планують зручно та швидко купити квиток до бажаною вистави. Ticketland має вікно в якій користувач може обрати виставу яка йому необхідна, за вибором часу та жанру. Далі він потрапить на вікно з вибором місця. Також це вікно містить інформацію о вартості конкретного місця.

## 1.2 Огляд існуючих програмних засобів, що вирішують аналогічні завдання

Ticketland (ticketland.ru), розробник - ТОВ «МДТЗК». Переваги: робить персоналізовані рекомендації користувачам, орієнтуючись на їхні інтереси, надаються спеціальні бонуси і подарунки. Недоліки: не дуже зручний інтерфейс, відсутність сортування по типу місць. Вікно виконання цієї програми представлене на рисунку 1.1.

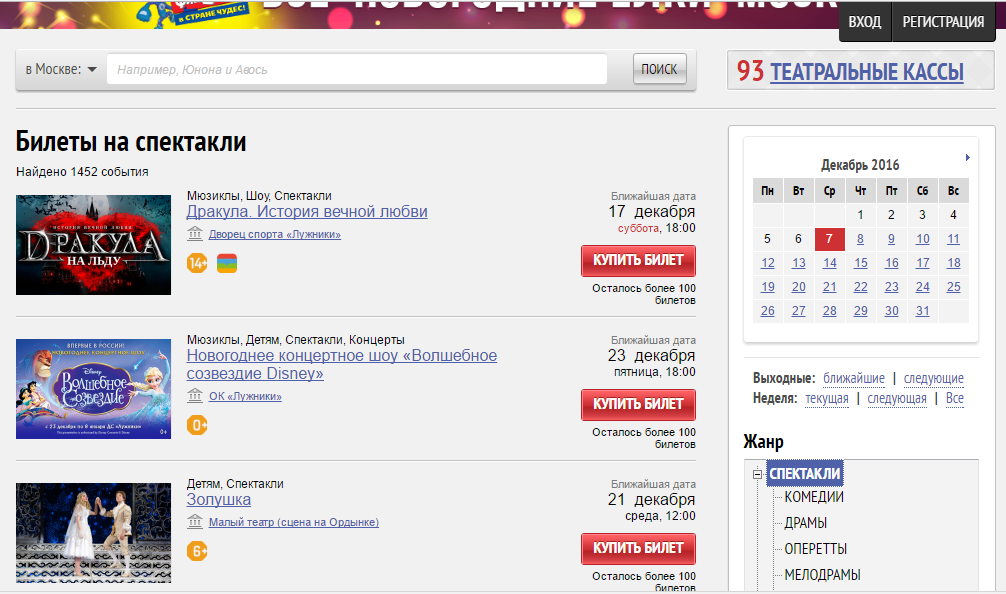


Рисунок 1.1 – Вікно виконання програми Ticketland

Wunderlist, розробники - 6Wunderkinder . Переваги: легкість, функціональність, мультіплатформеність. Недоліки: немає розподілу задач на категорії. Вікно виконання цієї програми представлене на рисунку 1.2.

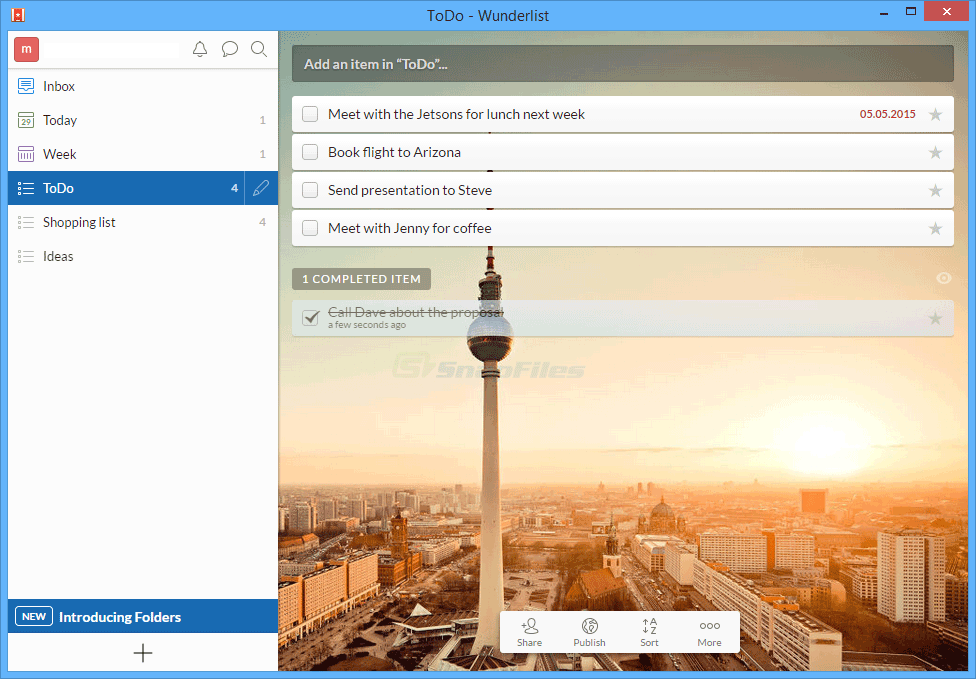


Рисунок 1.2 – Вікно виконання програми Wunderlist

Any.do, розробники - Any.do, Inc. Переваги: наявність підзадач, мультиплатформеність. Недоліки: порівнянна важкість в опануванні. Вікно виконання цієї програми представлене на рисунку 1.3.

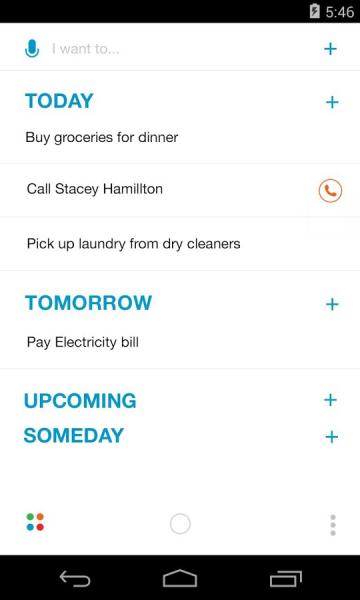


Рисунок 1.3 – Вікно виконання програми Any.do

LeaderTask, розробники - ТОВ «Органайзер ЛідерТаск». Переваги: богато додаткових функцій. Недоліки: нагромаджений інтерфейс. Вікно виконання цієї програми представлене на рисунку 1.4.

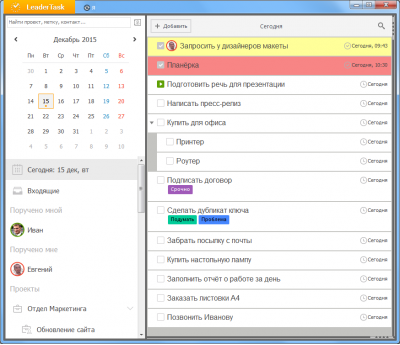


Рисунок 1.4 – Вікно виконання програми LeaderTask

ToDoList, розробники - AbstractSpoon. Переваги: можливість широкого індивідуального налаштування програми. Недоліки: порівнянна важкість в опануванні. Вікно виконання цієї програми представлене на рисунку 1.5.

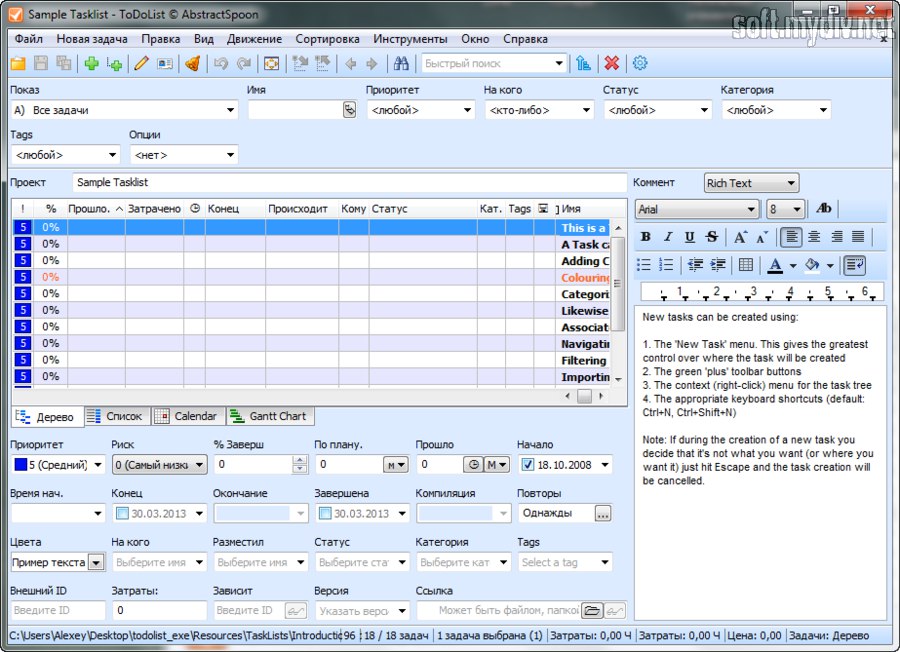


Рисунок 1.5 – Вікно виконання програми ToDoList

## 1.3 Постановка завдання роботи

Створити програмних продукт основною задачею якого є створення простого списку справ або задач. Він дозволить користувачу ефективно організовувати свій час.

Система повинна забезпечувати:

1. зручний доступ до календарю;

2. зручний запис подій;

3. пошук записів;

4. сортування записів за часом;

5. наглядне подання інформації.

# 2 Аналіз програмних засобів

## 2.1 Огляд особливостей мови програмування

Для написання програми було обрано мову програмування С++. Мова С++ - це мова програмування загального призначення, мета якого – зробити роботу програміста більш зручною. Вона надає програмісту інструмент для опису, що підлягають виконанню дій і набір концепцій, якими оперую програміст, обдумуючи, що можна зробити.

С++ підтримує такі парадигми програмування, як процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування, забезпечує модульність, роздільну компіляцію, обробку винятків, абстракцію даних, оголошення типів (класів) об'єктів, віртуальні функції. Стандартна бібліотека включає, в тому числі, загальновживані контейнери і алгоритми.

C ++ поєднує властивості як високорівневих, так і низькорівневих мов. Найбільшу увагу приділено підтримці об'єктно-орієнтованого і узагальненого програмування.

Для розробки програми було обрано середовище Qt. Qt - кросплатформовий інструментарій розробки ПЗ мовою програмування C++. Дозволяє запускати написане за його допомогою ПЗ на більшості сучасних операційних систем шляхом простої компіляції тексту програми для кожної ОС без зміни початкового коду. Включає всі основні класи, які можуть бути потрібні при розробці прикладного програмного забезпечення, починаючи з елементів графічного інтерфейсу і закінчуючи класами для роботи з мережею, базами даних, OpenGL, SVG і XML. Бібліотека дозволяє керувати потоками, працювати з мережею, і забезпечує кросплатформовий доступ до файлів.

## 2.2 Огляд особливостей обраного компілятора

У ході розробки використовувався компілятор MinGW версії 4.1.9, що є безкоштовним та представляє собою прямий програмний порт GNU Compiler Collection (GCC) під ОС Windows. Даний компілятор дозволяє розробляти рідні (native) додатки Windows. Увесь процес компіляції такий: препроцессінг, трансляція, лінкування. В результаті отримуємо готовий бінарний файл. Більшість мов, які підтримує GCC, також підтримуються і у MinGW. Це C, C++, Objective-C, Fortran, Ada.

MinGW за замовчуванням виконує компонування з бібліотекою компонентів MSVCRT, яка є бібліотекою мови C, створеної у 1998 році. Проект підтримує і поширює кілька різних ключових компонентів і додаткових пакетів, включаючи різні порти GNU toolchain, такі як GCC і Binutils, переведені в еквіваленті пакети. Ці утиліти можуть бути використані з командного рядка вікна або інтегровані в IDE. В доповнення, компонент MinGW, відомий як MSYS (Minimal SYStem) надає win32-порти легкої Unix-подібної shell оболонки, що включає rxvt і набір інструментів POSIX, достатній для запуску скриптів Autoconf. Реалізації заголовних файлів Win32 і бібліотек імпорту Win32 для зв'язування під час виконання програми від початку до її завершення мають пермісивну ліцензію, а порти GNU доступні під GNU General Public License. Бінарні збірки повного пакету MSYS та окремих MinGW GNU утиліт доступні для завантаження на сайті MinGW.

MinGW використовує виклики Win32 API. Внаслідок цього, він не може скомпілювати деякі програми Unix, що вимагають підтримку POSIX API. Зокрема, це відноситься до додатків, які вимагають такої функціональності POSIX, як fork(), mmap() або ioctl() і допускають запуск в середовищі POSIX.

У зв'язку з тим, що в рамках початкового проекту MinGW не обіцяв, що в його кодову базу будуть вноситися оновлення пов'язані з додаванням деяких нових ключових елементів Win32 API, а також найбільш необхідної підтримки 64-бітної архітектури, був створений проект MinGW-w64 (включений до останніх версій проекту MinGW). Він є новою чистою реалізацією порту GNU Compiler Collection (GCC) під Microsoft Windows здійсненої спочатку компанією OneVision та переданої у 2008 році в громадське користування (Public Domain).

MinGW-w64 забезпечує більш повну реалізацію Win32 API та включає:

* кращу підтримку стандарту C99;
* кращу підтримку Pthreads (включаючи можливість залучити функціонал стандарту C++11 бібліотеки libstdc++ компілятора GCC);
* GCC multilib;
* точки входу в програму з підтримкою Unicode (wmain / wWinMain);
* DDK (з проекту ReactOS);
* DirectX (з проекту WINE);
* підтримка великих файлів;
* підтримка 64-розрядних ОС Windows.

## 2.3 Огляд можливостей бібліотеки STL

Стандартна бібліотека шаблонів STL – частина стандартної бібліотеки C++ SCL, що забезпечує загальноцільові стандартні класи й функції, які реалізують найбільш популярні й широко використовувані алгоритми й структури даних. Ядро бібліотеки утворюють три складові: контейнери, алгоритми й ітератори.

Контейнери (containers) – це об'єкти, призначені для зберігання інших елементів. Тут представлені різні варіанти часто використовуваних структур даних – вектори, списки, стеки, черги й т.п. Точніше, навіть не самі структури, а заготівки для них. Конкретну структуру для свого типу даних програміст утворює сам за допомогою стандартних контейнерів.

Ітератори (iterators) – це об'єкти, які стосовно контейнера відіграють роль вказівників. Вони дозволяють одержати доступ до вмісту контейнера приблизно так само, як вказівники використовуються для доступу до елементів масиву. Ітератори застосуються як зв’язок між структурами даних (не обов'язково контейнерами) і алгоритмами або іншим кодом, який ці структури використовує.

Алгоритми (algorithms) виконують операції над даними контейнера. Існують алгоритми для ініціалізації, сортування, пошуку, заміни вмісту контейнерів. Багато алгоритмів призначені для роботи з послідовністю (sequence), що являє собою лінійний список елементів усередині контейнера. Велика частина алгоритмів STL побудована по єдиному принципу. Алгоритм отримує на вхід пару ітераторів(інтервал) і для елементів з цього інтервалу виконує деяку задачу, наприклад, сортування.

## 2.4 Огляд шаблонного класу бібліотеки STL, що використовується в роботі

В програмі використано шаблоний клас QVector. Клас QVector - це шаблоний клас в середовищі Qt, що представляє динамічний масив.

QVector є одним із загальних контейнерних класів Qt. Він зберігає свої елементи в суміжних комірках пам’яті та забезпечує швидкий доступ за індексом. QVector рекомендовано до використання, коли елементи є більшими, за розміром, ніж вказівник, для того щоб уникнути накладних витрат на індивідуальне виділення для них місця в "купі", під час вставки нового елемента.

QVector надає основні функції для додавання, переміщення та видалення елементів: insert(), replace(), remove(), prepend(), append(). Звернення за індексом відбувається за допомогою функцій at(), value() та перевантеженого оператора [].

В програмі, що розробляється, необхідний швидкий доступ до елементу контейнера за індексом. В контейнер поміщується обьект створенного классу, який суттєво більший за розміром, ніж вказівник. Саме через це, в якості контейнера було обрано шаблоний клас QVector.

## 2.5 Класи візуального інтерфейсу

Особливість Qt — використання Meta Object Compiller (MOC) — попередньої системи обробки початкового коду. MOC дозволяє в багато разів збільшити потужність бібліотек, вводячи такі поняття, як слоти (slots) і сигнали (signals). Qt комплектується графічним середовищем розробки графічного інтерфейсу QTDesigner, що дозволяює створювати діалоги і форми. Ідеологія створення форм у Qt базується на використанні менеджерів розташування, котрі надають «гумовий» дизайн, при якому розмір і розташування елементів форм визначаються автоматично, що значно прискорює розробку графічного інтерфейсу. В поставці Qt є «Qt Linguist» — могутня графічна утиліта, що дозволяє спростити локалізацію й переклад програми багатьма мовами, та «Qt Assistant» — довідкова система Qt, що спрощує роботу з документацією для бібліотек і дозволяє створювати крос-платформову довідку для ПЗ, розробленого на основі Qt.

Графічна бібліотека Qt розділена на декілька модулів:

QtDesigner — класи створення розширень QtDesigner'а для своїх власних віджетів;

QtGui — компоненти графічного інтерфейсу;

QtWidgets — містить класи для класичних застосувань на основі віджетів, модуль виділений з QtGui в Qt 5.

Ці модулі були використанні для створення інтерфейсу програми.

Інтерфейс розроблено за макетом «Центральна сцена». Тобто візуально в центрі робочого вікна програми будуть розміщені основні функціональні елементи інтерфейсу користувача.

Інтерфейс побудовано на основі Головного вікна програми з використанням додаткових діалогових вікон за необхідності. На Головному вікні розміщені: Панель меню, Рядок стану, Панель інструментів, різні варіанти списків, поля вводу, кнопки, таблиці. Всі елементі доповнені відповідними підписами. Є підтримка комбінацій гарячих клавіш та табуляції.

Додаткові діалогові вікна можуть бути функціональними або вікнами-повідомленнями з відповідними їм графічними елементами.

## 2.6 Висновки з розділу 2

Мовою програмування було обрано C++ з Qt Framework за зручність створення GUI, потужність та гнучкість мови, широкий спектр можливостей.

В якості компілятора було обрано MinGW 4.8.1 як доступний, зручний, налаштований під ОС Windows.

Інтерфейс було розроблено засобами Qt Designer за макетом «Центральна сцена» з Головним вікном як основним з викликом, за необхідності, додаткових вікон.

**3 Основні рішення з реалізації компонентів системи**

**3.1 Основні рішення щодо уявлення даних системи**

**3.2 Основні розроблені алгоритми**

**3.3 Основні рішення щодо модульного уявлення системи**

**3.4 Особливості реалізації системи**

**3.5 Результати тестування системи**

**3.6 Висновки з розділу 3**

**5 Інструкція користувача**

**Висновки**

**Список використаної літератури**

1. Методичні вказівки до виконання курсової роботи з дисципліни “Об’єктно-орієнтоване програмування” для студентів напряму підготовки 6.050103 «Програмна інженерія» всіх форм навчання. Частина 1. Організаційна частина / Г.В. Табунщик, Г.В. Неласа, Н.О. Миронова – Запоріжжя: ЗНТУ, 2010. – 22 с.

2. Лафоре Р. Объектно-ориентированное программирование в C++ / Роберт Лафоре. – СПБ: Питер, 2004. – 924 с.

3. Шлее М. Qt 5.3. Профессиональное программирование на C++ / Макс Шлее. – СПБ: БХВ-Петербург, 2015. – 928 с.:ил.

4. Qt Documentation [Електронний ресурс] – Режим доступу до ресурсу: <http://doc.qt.io/>.