



Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика

Специалност “Компютърни Науки” , Първи Курс , Трета Група
Първи проект - Обектно Ориентирано Програмиране
Тема 1: Дневник на пътешественика

Изготвила:

Гергана Свиленова Ангелова (факултетен номер: 7MI0800133)



Съдържание:

1. Основна идея..... страница 3
2. Начин на работа..... страница 4
3. Структура..... страница 5-7
4. Алгоритми..... страница 7
5. Използвани материали и допълнителна информация

Основна Идея

Проекта представлява конзолно приложение даващо право на любителите пътешественици да споделят впечатления, оценка и имена на снимкови файлове от своите пътувания. За направата на регистрация е необходимо да бъдат въведени потребителско име, съдържащо само главни или малки латински букви както и цифри, но без други специални символи. Нужни са още имейл и парола, за които няма ограничение в начина на изписване.

Така регистрирания потребител може да добави информация за своите пътувания. Въвеждат се последователно дестинацията, началната дата на пътуването, последвана от крайната дата, оценка от 1 до 5 в зависимост от преживяването, коментар, описващ накратко пътуването, брой изображения, които по късно се въвеждат като имена на снимкови файлове.

Всеки потребител, вписал се във вече съществуващ профил има възможност да провери отзивите за дадена дестинация като потърси информация с името на мястото.

При наличен запис в базата данни, съхраняваща всички регистрирани до момента пътувания, се извеждат впечатленията на останалите потребители за избраната дестинация. Приложението работи с текстови файлове, в които се записва информацията.

Начин на работа

При стартиране на приложението се изписва следна информация на конзолата:

Потребителят трябва да избере едната от следните три опции:

1. за да се регистрира;
2. за да влезе във вече съществуващ профил;
3. за да напусне приложението;

При избор на опцията за регистрация, потребителят следва да въведе валидно име, ако то не отговаря за изискванията, се извежда съответно съобщение, парола и имейл. Това създава лична база данни - файл със съответстващо име на потребителя и разширение .db, съдържащо информацията за пътуванията. Създава се още запис в обща база данни, където се съхраняват наличните потребители - отново текстов файл (database.txt), за да може след това да бъдат достъпени, при вписване в наличен профил (опция 2). Приложението работи и с допълнителен текстов файл, в който се записват всички дестинации (all_users_destinations.txt), така че да бъде използван, когато потребител потърси отзивите за дадено място, след това.

При избор на опцията за вписване във вече съществуващ профил, потребителят трябва да въведе своето име, като ако не бъде открито регистрирано име се извежда съответно съобщение. При намерен профил, следва сравнение на паролата предоставена от потребителя с тази записана в базата данни.

След влизане в профила си потребителя има възможност да потърси отзивите на останалите.

Структура

За проекта са използвани следните класове:

- String - представлява интерпретация на `std::string`;

Включва като private- данни : указател към тип `char`, променлива от тип `size_t`, пажеща размера; private-функции са за копиране и освобождаване на динамичната памет; Има конструктор по подразбиране, конструктор с параметри, копиращ конструктор, оператор равно, деструктор, селектори, предефинирани оператори `+=`, `+`, `==`, `<=`, `<` както и оператори `<<`, `>>`;

- DynamicArray - темплейтен клас, представлява интерпретация на `std::vector`;

Включва като private- данни: указател `T`, променливи от тип `size_t`, пажещи размера и капацитета; private-функции са за копиране, заделяне и освобождаване на динамичната памет; Има конструктор по подразбиране, конструктор с параметри, копиращ конструктор, оператор равно, деструктор, селектор, метод за добавяне на елемент, предефинирани оператори `[]`, както и оператори `<<`, `>>`;

- Constants- заглавен файл, съдържащ константи;
- Dates - клас, интерпретиращ дати

Включва като private- данни: променливи от тип `unsigned`, пажещи информация за ден, месец и година; private-функция за проверка за високосна година; Има конструктор по подразбиране, конструктор с параметри, селектори, предефинирани оператори `==`, `<`, както и оператори `<<`, `>>`;

- Journey - клас, интерпретиращ пътуване

Включва като private- данни: променливи от тип String за дестинацията, коментара и снимките, променливи от тип Dates за начална и крайна дата и променлива от тип int за оценка; Има конструктор по подразбиране, конструктор с параметри, селектори, мутатори и предефиниран оператор <<;

- User - клас, интерпретиращ потребител

Включва като private- данни: променливи от тип String за потребителско име, парола и имейл, динамичен масив от пътувания; private-функции за проверка на валидно потребителско - то трябва да съдържа само главни и малки латински букви както и цифри; Има конструктор по подразбиране, конструктор с параметри, селектори, мутатори, методи за добавяне на пътуване, за запазване на информацията за потребителя в текстов файл- регистрира потребителското име, паролата и имейла, като ги разделя със запетаи, както и метод, който запазва информацията за пътуванията на отделния потребител - отново разделяйки ги запетаи. Данните за дестинацията, начална, крайната дата, оценката, коментара и снимките се запазват в лична база данни съответстваща на името на потребителя, както и в общ текстов файл, съдържащ всички дестинации.

- Commands - помощен клас за избор на команда

Включва като private - данни: променлива от тип int; Има следните методи: принтиране за отварящата страница и тази за търсене на дестинация, както и селектор;

- Application - основния клас, управляващ приложението;
Включва като private- данни: променливи от тип User, Journey, Commands; private-функции за проверка за коректно изписани имена на снимките и правилен формат, две функции, записващи потребител във базата данни и пътуване, които се използват във функцията за регистрация; функция за вписване във вече съществуващ профил, както и функция за принтиране на информация за дадена дестинация; има още и метод, задействащ самото приложение;
- main - Използва метода за задействане на приложението на класа Application

Алгоритми

Използвани са алгоритми за четене и писане във файл и алгоритми за валидация на данни.

Използвани материали и допълнителна информация

- функция за изчистване на конзолата : **system("cls")** - <https://www.cplusplus.com/forum/beginner/46492/>
- Github: <https://github.com/GerganaAngelova02>