**Проект „Билети“**

**Глава 1. Увод**

* 1. Описание и идея на проекта

Проектът ми „Билети“ симулира система, като тези в кината и театрите, която разработвам, за да улесня процеса за продаване на билети за събития.

Идеята на проектът е крайният потребител да въведе броя на залите, редовете и местата, които има на всеки ред, и с помощта на меню, той да избере каква операция иска да извърши. Има възможност да добави събитие, да купи или резервира билет, както и да отмени резервацията си, да провери свободните места или да получи списък за направените резервации в дадена зала за определено събитие, или да провери броя на запазените места за него.

* 1. Цел и задачи на разработката

Целта ми за проекта „Билети“ е да го разделя на малки подзадачи, които водят до решаването на целият проблем или казано по друг начин до разработване на системата.

Първите две малки подзадачи, които си поставям са да напиша клас Vector и клас String. Следващите подзадачи са да направя клас за дата, събитие, зала и билет, които да обединя в един общ клас, съдържащ всички събития и всички зали, както и операциите, които трябва да се извършат над тях.

Класовете Vector и String използвам, за да мога по-лесно да управлявам паметта на различните променливи. За да ги реализирам взех заглавните файлове от упражнения, но ги имплементирах сама. Класът за датата използвам, за да мога да проверя дали въведена от крайният потребител е валидна, както и за да мога лесно да определям хронологията на събитията.

2**. Обща архитектура – ООП дизайн**

Реализацията ми е постигната чрез 7 класа – Vector, String, Data, Ticket, Event, Auditorium и Cinema.

* Класът Vector го създадох, за да избегна проблеми при управлението на паметта.

Използвам го като контейнер, който да събира всички множества от данни.

За заглавния му файл, използвах като източник упражненията на 8-ма група, но го реализирах сама. Поддържа част от функциите на std::vector.

* Класът String го създадох, за да избегна проблеми при управлението на паметта, както и писане на голямата четворка за всеки клас. За заглавния му файл използвах отново като източник упражненията на 8-ма група, но реализацията си е моя.
* Класът за Date използвам, за да мога лесно да си въвеждам дата, както и да я проверявам за коректност. Чрез предефинираните оператори „<, >, >=, <= , == “ в него, подреждам хронологично събитията и намирам точната дата, която ми трябва.
* Класът Auditorium използвам за шаблон на залата, където „ Vector<size\_t> seats“ представлява масив от положителни числа, които показват броя на местата на всеки ред, а променливата „ size\_t numberOfAudit“ използвам за идентификатор на залата.
* Класът Ticket е самият билет, който съдържа: „String eventName“ - името на събитието, „size\_t rowNumber“ – номерът на реда на билета, „size\_t seatNumber“ – номерът на мястото на билета, „String password“ – парола, необходима за резервацията, както и за купуване на резервиран билет, „String notes“ – допълнителни бележки и „State state“ – състоянието на билета. Използвам функциите “setPassword(const String &passwordParam)” и „setNotes(const String& notesParam)“ за промяна на низът записан на билета(при резервация). Чрез булевите функции „isFree()“, „isReserved()“ и „isBought()“ проверявам състоянието на билета. А пък функциите „reserve()“, „buy(const String& passwordParam, const String& notesParam)“ и „cancelReservation(const String& passwordParam, const String& notesParam)“ използвам за резервирането, купуването и отказването на резервацията на текущия билет.
* Класът Event съдържа: „Date date“ – датата на събието, „String eventName“ – името на събитието, „size\_t auditoriumNumber“ – номерът на залата, в която ще се проведе, „Vector<Vector<Ticket>> tickets“ – билетите, направени по залата - шаблон, те са контейнер от редове с различна дължина.
* Класът Cinema съдържа два контейнера – единият от събития, другият от зали. В него са развити всички операции, които може да извършва системата. Функцията “ addEvent()” – служи за добавяне на събитие, „ buyTicket()“ – за закупуването на билет, „reserveTicket()“ – за резервирането на билет, „ cancelTicket()“ – отказ от вече направена резервация, „reportFreeSeats()“ – извежда справка за свободните места, „saveToFileReportForReservations()“ – записва във файл доклад за направените резервации, „reportForBoughtTickets()“ – извежда справка за закупените билети през определен период от време.

3. **Заключение**

3.1. Обобщение на изпълнението на началните цели

Успях да реализирам системата, като не успях да я направя да работи коректно за символен низ с разделители. За всички променливи от тип String, ако потребителят въведе символен низ с разделители, програмата няма да работи коректно.

5.2. Насоки за бъдещо развитие и усъвършенстване

Усъвършенстването на моята система, ще включва добавяне на час за различните събития или повтарящите се такива през деня, както и време за почистване, настаняване и напускане на залата. Ще подобря разработката на реализирания от мен стринг, така че да работи коректно и за имена с разделители. Ще добавя и още проверки за валидност на данните. Също така ще добавя и операция за проследяване на броя на закупените билети от един потребител и на база на този брой, ще правя намаление върху цената на билетите.