Билети

Берай Сюлейман - 72084

Github Link: <https://github.com/DDooms/project---oop---tickets>

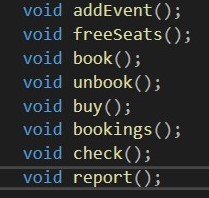
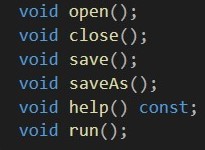
1. **Увод**
   1. Описание и идея на проекта

Проектът „Билети“ реализира програма за създаване на представления. Програмата реализира извършва определн брой действия спрямо резервирането на едно представление. Информация за книгите и потребителите може да се извлича от файлове.

* 1. Функционалности

За да функционира програмата, тя може да:

* + 1. зарежда информация от файл и да извършва редица команди, докато потребителя не реши да излезе от системата.



1. **Проектиране**

За осъществяване на проекта са използвание следните класе:

* Собствено написани помощни класове String и Vector

Class String

char\* string;

size\_t size; |

size\_t capacity; | член данни

void copy(const String& other); | копи конструктор

void erase(); | изтрива член данни

void resize(); | променя обема на стринга

public:

String(); | default – ен конструктор

String& operator=(const String& \_String); | предефиниране на оператор =

String& operator=(const char\*); |

friend std::ostream& operator<<(std::ostream& out, const String& current); | предефиниране на << и >>

friend std::istream& operator>>(std::istream& in, String& current); |

String(const char\* other); | нормални конструктори

String(const String& \_String); |

~String(); | деструктор

const char\* getString() const; |

size\_t getSize() const; | извеждат съответните член данни

size\_t getCapacity() const; |

void removeLast(); | маха последния елемент

String operator+(String&); |

String& operator+=(const char); |

bool operator==(const String& \_string) const; | предефинирани оператори +, +=, ==, [ ]

bool operator==(const char\*) const; |

const char operator[](size\_t index) const; |

void tolower(); | пренаписва стринг с малки букви

* Базови класове

Class Seat

size\_t rowNumber;

size\_t seatNumber;

bool booked;

String note;

size\_t ticketNum;

Date buyingDate;

size\_t random() const; | създава рандом число

public:

Seat();

Seat(size\_t \_rowNumber, size\_t \_seatNumber);

String getNote() const;

void setNote(const String& \_note); | прилага на член-данна съответна стойност

bool isBooked() const; | проверява дали е резервирано

void book(); | резервира

void unbook(); | отменя резервация

bool isBought() const; | проверява дали е купен билет

void buy(const Date& date); | купува билет

size\_t getTicketNum() const;

bool isBoughtInDateInterval(const Date& dateFrom, const Date& dateTo) const; | проверява в какъв интервал от време е купен билет

friend std::ostream& operator<<(std::ostream& out, const Seat& current);

friend std::istream& operator>>(std::istream& in, Seat& current);

Class Row

size\_t rowNumber;

Vector<Seat> seats;

public:

Row();

Row(size\_t \_rowNumber, size\_t numberOfSeats);

size\_t getSize() const;

Seat& operator[](size\_t index) const;

friend std::ostream& operator<<(std::ostream& out, const Row& current);

friend std::istream& operator>>(std::istream& in, Row& current);

Class Hall

size\_t rows;

size\_t seatsPerRow;

Vector<Date> bookedDates;

public:

Hall();

Hall(size\_t \_rows, size\_t \_seatsPerRow);

void book(const Date& date);

size\_t getRows() const;

size\_t getSeats() const;

bool isBooked(const Date& date) const; | проверява дали залата е резервирана

bool isFree(const Date& date) const; | проверява дали залата е свободна

friend std::ostream& operator<<(std::ostream& out, const Hall& current);

friend std::istream& operator>>(std::istream& in, Hall& current);

Class Event

String name;

Date date;

Vector<Row> rows;

size\_t hallNum;

public:

Event();

Event(const String& \_name, size\_t rowNumbers, size\_t colNumbers, const Date& \_date, size\_t \_hallNum);

String getName() const;

size\_t getHallNum() const;

size\_t getColNumber() const;

size\_t getAllSeats() const;

size\_t getBookedSeats() const;

size\_t getSeatNum(size\_t number);

size\_t getFreeSeats() const;

Date getDate() const;

bool isValid(size\_t rowNumber, size\_t seatNumber) const;

bool isBooked(size\_t rowNumber, size\_t seatNumber) const;

void bookSeat(size\_t rowNumber, size\_t seatNumber, const String& note); | резервира място

void unBookSeat(size\_t rowNumber, size\_t seatNumber); | отменя резервацията

void buyTicket(size\_t rowNumber, size\_t seatNumber, const Date& date); | купува билет

size\_t getBookedButNotBoughtSeats() const; | връща резервирани, но незакупени места

size\_t getBoughtSeatsOnDateInterval(const Date& dateFrom, const Date& dateTo) const; | връща закупени места във времеви интервал

void saveEvent(std::ostream& file); | записва информация от файл

void loadEvent(std::istream& file); | чете от файл

friend std::ostream& operator<<(std::ostream& out, const Event& current);

friend std::istream& operator>>(std::istream& in, Event& current);

Class Ticket

Vector<Hall> halls;

Vector<Event> events;

Event\* findEvent(const String& name, const Date& date);

public:

void addEvent(); | добавя представление в базата от данни

void freeSeats(); | извежда свободните места

void book(); | резервира място

void unbook(); | отменя резервацията

void buy(); | купува място

void bookings(); | извежда резервациите

void check(); | проверява дали има резервации

void report(); | извежда информация

Class Command

bool isOpened;

bool isSaved;

bool isChanged;

String FileName;

public:

Command();

void open(); | отваря файл

void close(); | затваря файл

void save(); | запаметява файл

void saveAs(); | запаметява файл на желаната от вас директория

void help() const; | извежда информацията с налични команди

void run(); | стартира програмата

Идея за развитие на проекта

* Разделянето на командите с файлове в различни методи
* Подробна информация за представление (продължителност, различни представления в един ден)
* Списък с гостите, купили билет
* Добавяне на актьори