**Документация**

**Проект №10 – Библиотека**

**Изготвил: Георги Патриков ФН: 72097**

**Линк към хранилище в *Github*:** [***https://github.com/patrikov01/OOP-Library***](https://github.com/patrikov01/OOP-Library)

1. Увод

Проектът цели да реализира информационна система, поддържаща библиотека. Поддържа се информация за налични книги и техните характеристики, както и за потребители със съответните операции, които могат да извършват. Програмата запазва и обработва информацията от файл.

Целта на проекта е да се изградят класове, които да обработват данните за книги и потребители, като позволяват на текущия потребител чрез команди да извършва операции в информационната система. Програмата трябва да позволява работа с файлове и достъпът до съответна команда да става посредством някаква проверка дали е влезнал потребител, дали той е администратор или дали е отворен файл, от който да четем съответната информация.

Структура на текущата документация:

1. Увод – описание, идея и цел
2. Реализация – основни характеристики на класовете и техните методи
3. Идеи за бъдещо развитие

1. Реализация

За целта са реализирани класовете *Book*, описващ характеристиките и методите за дадена книга, клас *User,* който описва поведението на един потребител, класовете *VectorBooks* и *VectorUsers*, поддържащи произволен брой съответно книги и потребители заедно с операциите нужни за нашия проект и най- важния клас *Library*, който реализира идеите на информационната система. Помощни класове са *Vector* и *String*, които да улеснят нашата задача за заделяне на памет, освобождаване на памет и други операции. Направени са и юнит тестове с помощта на вмъкнати *Doctests.*

**class Book**

private:

String author;

String title;

String genre;

String description;

int yearOfPublishing; член-данни на клас Книга

String keyWords;

double rating;

long int uniqueNumber;

public:

Book(); - дефолтен конструктор

String getAuthorName() const;

String getTitle() const;

String getGenre() const;

String getDescription() const;

int getYearOfPublishing() const;

String getKeyWords() const;

double getRating() const;

long int getUniqueNumber() const;

сетъри и гетъри

void setAuthorName(const String& \_authorName);

void setTitle(const String& \_title);

void setGenre(const String& \_genre);

void setDescription(const String& \_description);

void setYearOfPublishing(int \_yearOfPublishing);

void setKeyWords(const String& \_keyWords);

void setRating(double \_rating);

void setUniqueNumber(long int \_uniqueNumber);

Оператори за вход и изход:

friend std::istream& operator>>(std::istream& in, Book& book);

friend std::ostream& operator<<(std::ostream& out, const Book& book);

void printInfo() const; - принтиране на информацията

class User

private:

String username;

String password; член-данни на клас Потребител

bool isAdmin;

public:

User(); - дефолтен коструктор

User(String& , String& , bool ); - конструктор с параметри

String getUsername() const;

String getPassword() const;

bool getIsAdmin() const;

сетъри и гетъри

void setUsername(String& \_username);

void setPassword(String& \_password);

void setIsAdmin(bool \_isAdmin);

Оператори за вход и изход:

friend std::ostream& operator<<(std::ostream& out, const User& user);

friend std::istream& operator>>(std::istream& in, User& user);

class VectorUsers

private:

Vector<User> users; - вектор от потребители

size\_t indexOfCurrentUser; - индекс на текущ потребител

bool isLoggedUser; - булева стойност за това дали е влезнал потребител

public:

VectorUsers(); - дефолтен конструктор

void login(); - команда за влизане в системата

void logout(); - команда за излизане от системата

void usersAdd(); - команда за добавяне на потребител

void usersRemove(); - команда за премахване на потребител

void saveUsers(std::ostream& out); - запазване на потребителите във файл

void loadUsers(std::istream& in); - четене на потребители от файл

class VectorBooks

private:

Vector<Book> library; - вектор от книги

public:

void booksAll() const; - команда за извеждане на всички книги

void booksInfo() const; - команда за извеждане на книга по въведен номер

void booksFind(); - команда за намиране на книга по зададен критерий

void booksSort(); - команда за сортиране на книги по зададен критерий по “Метод на мехурчето”

void booksAdd(); - команда за добавяне на книга

void booksRemove(); - команда за премахване на книга

void saveUsers(std::ostream& out); - запазване на книгите във файл

void loadUsers(std::istream& in); - четене на книги от файл

class Library

private:

VectorBooks library; - книгите в системата

VectorUsers users; - потребителите в системата

public:

void open(); - команда за отваряне на файл

void close(); - команда за затваряне на текущия файл

void save(); - команда за запазване на текущата информация в текущия файл

void saveAs(); - команда за запазване на текущата информация в произволен файл

void help(); - команда за извеждане на наличните команди с тяхното действие

void exit(); - команда за излизане от системата

void start(); - метод реализиращ въведените команди чрез множество проверки

1. Идеи за бъдещо развитие

Нашата информационна система библиотека не разгръща пълния си потенциал и за в бъдеще са възможни още множество операции. Възможно е за удобство на потребителя да има опция за извеждане на книгите с най- висок рейтинг за дадена година. Подобрение би било всеки читател да може да има опция за добавяне на книга или автор в „Любими“ за по- пряк достъп до тях. При отварянето на дадена книга е възможно визуализиране на текста или откъс от текста на дадената книга. Редом с извеждането на характеристиките на книга, може да се зарежда снимков материал на корицата, някоя страница или снимка на автора, в това число и биография за самия автор. Извеждането на информация за това колко потребители има в системата и наредба на най- четените книги от тях, също е приложимо в проекта.