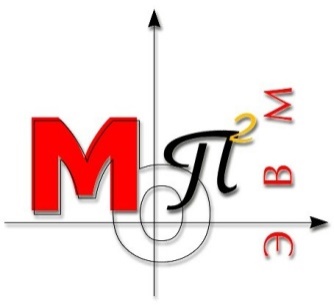
Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждения высшего образования   
«Южный федеральный университет»

Инженерно-технологическая академия

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ

**

Лабораторная работа №4

по дисциплине

"ООП"

на тему

"Исследование STL"

**Выполнил:**

студент группы КТбо2-7

Борисов Я. П.

**Проверил:**

Преподаватель  
Тарасов С.А.

**ВАРИАНТ №12**

**СПЕЦИФИКАЦИЯ КЛАССОВ**

**КЛАСС MAPCREATOR**

class MapCreator {

public:

MapCreator(const std::string filename);

bool readData();

std::vector<Book> findByString(const std::string value, const int type);

std::vector<Book> findByInt(const int value, const int type);

std::vector<Book> getAllBoks();

void deleteBook(const Book &book);

void addBook(const Book& book);

~MapCreator();

private:

std::string filename;

std::vector<Book> books;

std::map < std::string, std::set <int >> nameKeyMap;

std::map < std::string, std::set <int >> surnameKeyMap;

std::map < std::string, std::set <int >> midlenameKeyMap;

std::map < std::string, std::set <int >> titleKeyMap;

std::map<int, std::set<int>> yearKeyMap;

std::map<int, std::set<int>> countKeyMap;

void readFile();

void rewriteFile();

void updateData();

};

**СТРУКТУРА BOOK**

struct Book {

std::string name="", surname = "", midlename = "",

title = "";

unsigned int year = 0, count = 0;

};

class Pentagon : public Figure {

public:

Pentagon(const int count);

double getArea() override;

double getPerimeter() override;

void getWeight() override;

void onRotate(const double angle) override;

void onMove(const double x, const double y) override;

~Pentagon();

private:

static const int \_P = 5;

};

**КЛАСС ИТЕРФЕЙСА**

class Console {

public:

Console();

void onConsoleCall();

~Console();

private:

std::vector<Book> books;

MapCreator\* creator;

static const int \_COMMANDS\_LENGS = 6;

std::string strings[\_COMMANDS\_LENGS] = { "Get all by author or title", "Get all book from year or count", "Delete book", "Add Book", "Show all books", "Exit" };

bool onConversationStart();

void setColor(const int text);

void openFile();

bool getData();

std::vector<Book> getByString();

std::vector<Book> getByInteger();

void deleteBook();

void addBook();

void showAllBooks();

void printBook(const Book book);

void showBooks(const std::vector<Book> books);

void showBooks(const std::vector<Book> books, const bool numbering);

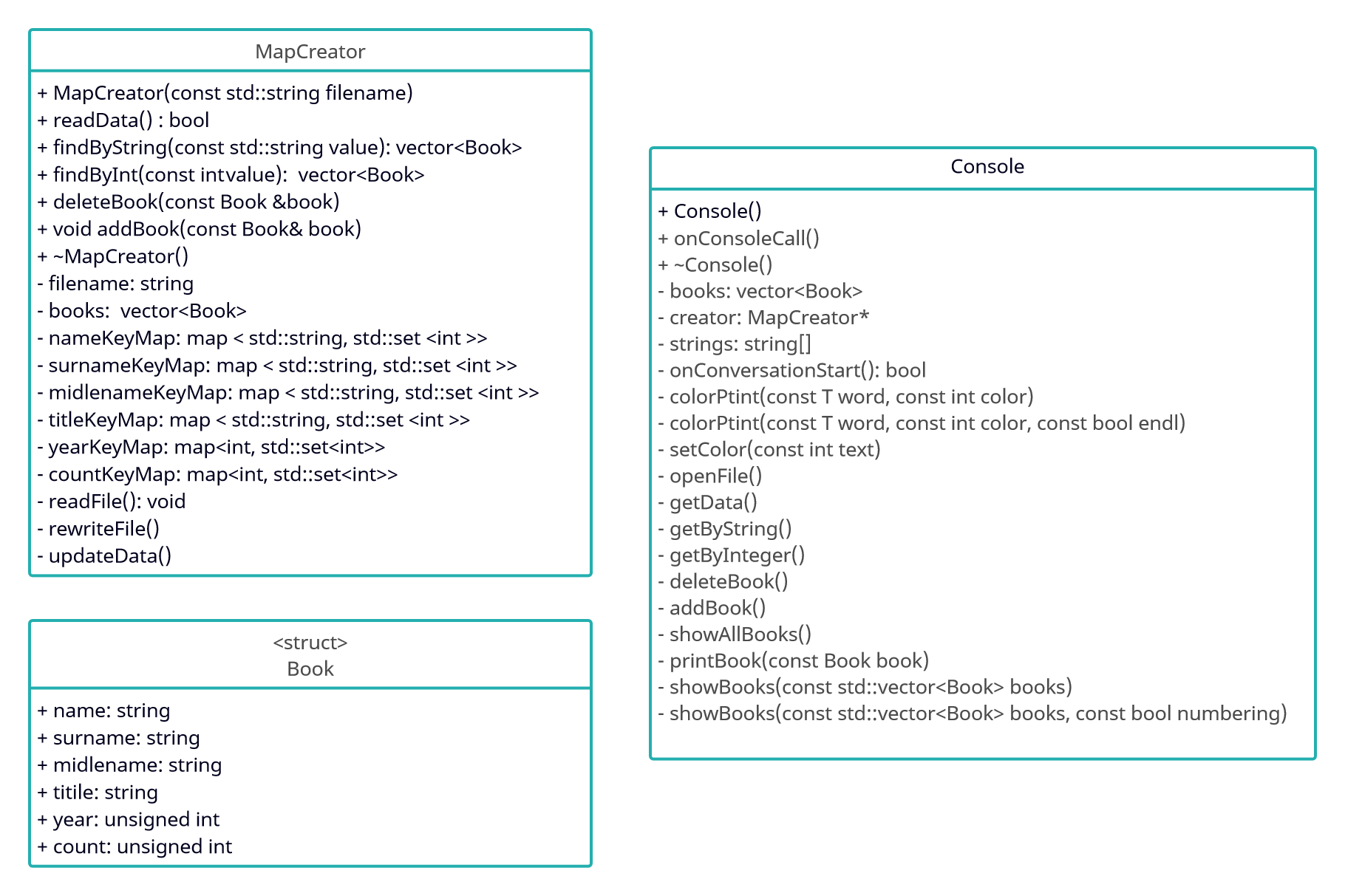
template <typename T>

void colorPrint(const T word, const int color);

template <typename T>

void colorPrint(const T word, const int color, const bool endl);

};

**Диаграмма классов**