Выполнил

студент КТбо1-4 \_\_\_\_\_\_\_\_\_\_\_\_\_ А. А. Воронов

Принял

ст. пр. каф. САПР, к.т.н. \_\_\_\_\_\_\_\_\_\_\_\_\_ В. И. Данильченко

Таганрог 2024

ЛАБОРАТОРНАЯ РАБОТА №9

по дисциплине «Алгоритмизация и программирование»

**«Классы, наследование»**

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

(ФГАОУ ВО «ЮФУ»)

1. **Цель лабораторной работы:** изучить работу классов, ознакомиться с .NET.
2. **Задача:** Написать программу, в которой реализован класс:  
   1) Перечислен список дисциплин и книг

2) Нужно получить набор книг, относящихся к заданной дисциплине, список дисциплин, использующих данную книгу.

1. **Ход работы:** я создал проект-консольное приложение в Visual Studio 2022. Я создал класс Discipline.Добавил приватные атрибуты, описывающие название вуза, подразделения, названия дисциплины, начало обучения (в семестрах), длительность (в семестрах), добавил флаг наличия экзамена и контейнер vector для хранения списка книг. В публичные методы добавил конструктор со значениями по умолчанию, а также методы addBook, deleteBook, printBooks, printInfo, getLen, getBook. В основном коде реализовал функцию генерации 5 дисциплин create\_data, функцию filter для отбора дисциплин, использующих заданную книгу. Весь проект я разбил на 3 файла: заголовочный файл класса Archive.h, файл с функциями класса Archive.cpp и основной код программы S2\_L4\_cut.cpp

**Программный код Archive.h:**

#pragma once

#include <iostream>

#include <string>

#include <vector>

class Discipline

{

private:

std::string p\_university, p\_department, p\_discipline\_name;

int p\_start\_semester, p\_semester\_amount;

bool p\_is\_exam;

std::vector<std::string> books;

public:

Discipline(std::string university = "SFedU", std::string department = "ICTIS", std::string discipline\_name = "None", int start\_semester = 1, int semester\_amount = 1, bool is\_exam = 1);

void addBook(std::string name);

void deleteBook(std::string name);

void printBooks();

void printInfo();

std::string getBook(int index);

int getLen();

};

**Программный код Archive.cpp:**

#include "Archive.h"

Discipline::Discipline(std::string university, std::string department, std::string discipline\_name, int start\_semester, int semester\_amount, bool is\_exam)

{

p\_university = university;

p\_department = department;

p\_discipline\_name = discipline\_name;

p\_start\_semester = start\_semester;

p\_semester\_amount = semester\_amount;

p\_is\_exam = is\_exam;

}

void Discipline::printInfo()

{

std::cout

<< "Discipline Name: " << p\_discipline\_name << "\n"

<< "University: " << p\_university << "\n"

<< "Department: " << p\_department << "\n"

<< "Start semester: " << p\_start\_semester << "\n"

<< "Amount of Semester: " << p\_semester\_amount << "\n"

<< "Books contained: " << books.size() << "\n"

<< "Exam: " << p\_is\_exam << "\n\n";

}

void::Discipline::addBook(std::string name)

{

books.push\_back(name);

}

void Discipline::deleteBook(std::string name)

{

for (int i = 0; i < books.size(); i++)

{

if (books[i] == name) books.erase(books.begin() + i);

}

}

void Discipline::printBooks()

{

std::cout << "Books connected with " << this->p\_discipline\_name << ":\n\n";

for (int i = 0; i < books.size(); i++) std::cout << i + 1 << ". " << books[i] << "\n";

std::cout << "\n";

}

std::string Discipline::getBook(int index)

{

return books[index];

}

int Discipline::getLen()

{

return books.size();

}

**Программный код S2\_L4\_cut:**

#include <iostream>

#include "Archive.h"

void create\_data(std::vector<Discipline>& container)

{

Discipline

/\*std::string university, std::string department, std::string discipline\_name, int start\_semester, int semester\_amount, bool is\_exam\*/

d1("SFedU", "ICTIS", "Descrete maths", 1, 1, 1),

d2("DSTU", "Wine faculty", "Wine tasting", 1, 4, 1),

d3("SFedU", "ICTIS", "History", 1, 2, 0),

d4("SFedU", "IRTCS", "Plane construction", 2, 3, 1),

d5("MEPhI", "LaPlaz", "Thermorectal cryptoanalyzis", 1, 4, 1);

container.push\_back(d1);

container.push\_back(d2);

container.push\_back(d3);

container.push\_back(d4);

container.push\_back(d5);

}

void filter(std::string book\_name, std::vector<Discipline>& container)

{

std::cout << "List of Disciplines the book <<" << book\_name << ">> is contained in: \n\n";

bool flag = false;

for (int i = 0; i < container.size(); i++)

{

for (int j = 0; j < container[i].getLen(); j++)

{

if (container[i].getBook(j) == book\_name)

{

container[i].printInfo();

flag = true;

}

}

}

if (flag == false) std::cout << "No Disciplines found \n";

}

int main()

{

std::vector<Discipline> archive;

create\_data(archive);

archive[0].addBook("Some random stuff1");

archive[0].addBook("Some random stuff2");

archive[0].addBook("Some random stuff3");

archive[1].addBook("Some random stuff1");

archive[0].printInfo();

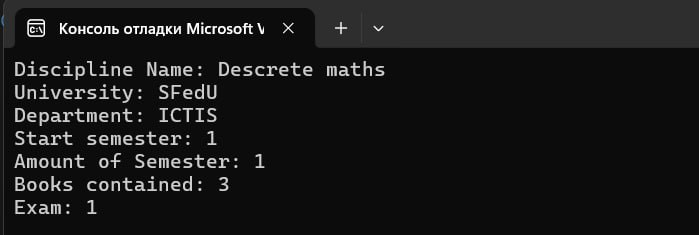
archive[0].printBooks();

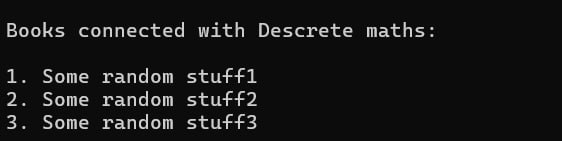
filter("Some random stuff1", archive);

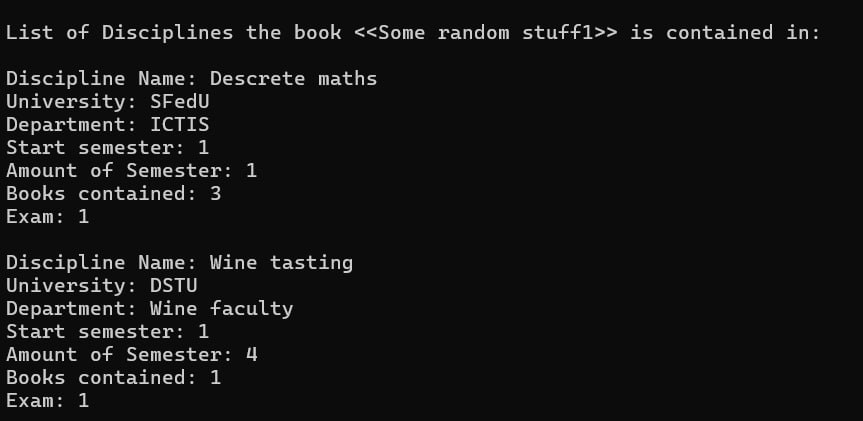
filter("Some random stuff4", archive);

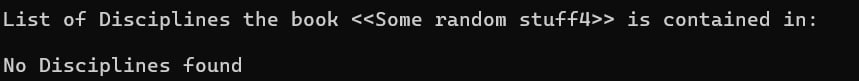
}

**Пример работы программы:**

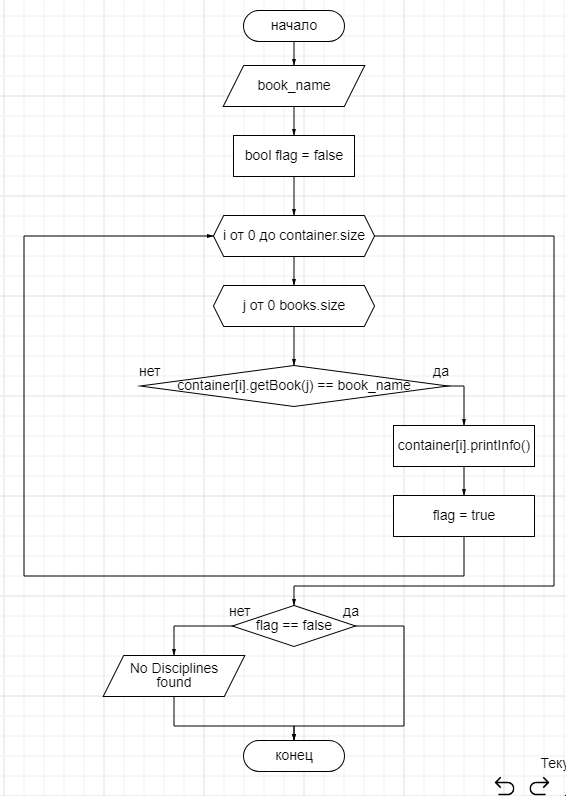








**4.Блок-схема фильтра**

****

**5. Вывод:** в ходе работы были выполнены все цели и задачи, было выяснено, как работают классы, как использовать заголовочные файлы классов и настраивать конструкторы. В проекте были сгенерированы входные данные, которые затем были обработаны по требованиям в задании.

**6. Источники:**

<http://cppstudio.com/post/439/>

<https://metanit.com/cpp/tutorial/5.17.php>

<https://metanit.com/cpp/tutorial/5.2.php>