# Лабораторная работа 4. Создание информационно-справочных систем на С++

## Задание

Разработать классы с использованием контейнера, разработать методы ввода данных с клавиатуры в контейнер, сохранение элементов в файле, поиск и сортировку элементов в массиве c использованием библиотечных функций и вывод на экран результатов.

Предусмотреть обработку не менее 2-х исключительных ситуаций, возникающих, например, при вводе данных с консоли и работе с файлом; обработчики данных должны указывать на место и причину возникновения ситуации и предлагать способ ее преодоления.

Во всех нижеследующих вариантах написать программу, в основе которой лежит следующий класс:

Класс “Студент” с полями: ФИО студента, номер группы, успеваемость(массив из 3-х элементов – количество пятерок, четверок, троек).

Программа должна осуществлять:

* ввод с клавиатуры и из файла в динамический массив не менее 10 записей с проверкой данных;
* сортировку записей по номеру группы;
* вывод в файл и на дисплей фамилии и группы студентов, средний балл которых больше 4.

## Используемые математические алгоритмы

Средняя балл студента вычислялся, как сумма произведений соответствующей оценки на их количество у студента, и эта сумма делилась на все количество оценок.

## Диаграмма классов

Student

MyBase

main

Progress

## Перечень сигнатур методов с комментариями согласно заданию

Progress.h

#pragma once

class Progress

{

public:

int five;

int four;

int three;

Progress(int five, int four, int three);

Progress(void);

~Progress(void);

};

Student.h

#include <string>

#include <iostream>

#include <cctype>

#include "Progress.h"

using namespace std;

#pragma once

class Student

{

private:

string \_patronymic;

string \_name;

string \_lastName;

string \_NumberOfGroup;

Progress \_progress;

void Check();//Проверка верности введенных данных

public:

double AverageMark();//Вычисление среднего балла студента

Student(void);

~Student(void);

bool operator<(Student student);

friend istream& operator>>(istream &cin, Student &student);//потоковый ввод

friend ostream& operator<<(ostream &cout, Student &student);//потоковый вывод

};

MyBase.h

#include <string>

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

#include "Student.h"

using namespace std;

#pragma once

class MyBase

{

private:

vector<Student> students;

public:

void Sort();//сортировка по номеру группы

void In(int n, istream& in);//ввод данных с указанного потока

void InFile(int n = -1, string path = "BaseIn.txt");//ввод данных с указанного файла

void Out();//Вывод на экран и в файл студентов, средний балл которых больше 4

void Out(ostream& out);//вывод списка студентов в указанный поток

void OutFile(string path = "BaseOut.txt");//вывод списка студентов в указанный файл

MyBase(void);

~MyBase(void);

};

## Листинг

Progress.cpp

#include "Progress.h"

Progress::Progress(void)

{

five = 0;

four = 0;

three = 0;

}

Progress::Progress(int five\_, int four\_, int three\_)

{

five = five\_;

four = four\_;

three = three\_;

}

Progress::~Progress(void)

{

}

Student.cpp

#include "Student.h"

Student::Student(void)

{

}

Student::~Student(void)

{

}

double Student::AverageMark()

{

return (\_progress.five \* 5 + \_progress.four \* 4 + \_progress.three \* 3) / (double)(\_progress.five + \_progress.four + \_progress.three);

}

bool Student::operator<(Student student)

{

return \_NumberOfGroup < student.\_NumberOfGroup;

}

istream& operator>>(istream &cin, Student &student)

{

cin >> student.\_lastName >> student.\_name >> student.\_patronymic >> student.\_NumberOfGroup >> student.\_progress.five >> student.\_progress.four >> student.\_progress.three;

try

{

student.Check();

}

catch(string error)

{

cout << "Input error with " << error << endl;

cout << "Chech your data and try do this operation again!" << endl;

}

return cin;

}

ostream& operator<<(ostream &cout, Student &student)

{

return cout << student.\_lastName << " " << student.\_NumberOfGroup;

}

void Student::Check()

{

for (int i = 0; i < \_lastName.length(); i++)

{

if(isdigit(\_lastName[i]))

throw (string)"LastName";

}

for (int i = 0; i < \_name.length(); i++)

{

if(isdigit(\_name[i]))

throw (string)"Name";

}

for (int i = 0; i < \_patronymic.length(); i++)

{

if(isdigit(\_patronymic[i]))

throw (string)"Patronymic";

}

if (\_progress.five < 0 || \_progress.four < 0 || \_progress.three < 0)

{

throw (string)"Progress";

}

}

MyBase.cpp

#include "MyBase.h"

MyBase::MyBase(void)

{

}

MyBase::~MyBase(void)

{

}

void MyBase::Out()

{

Out(cout);

OutFile();

}

void MyBase::Out(ostream &out)

{

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

{

if (students[i].AverageMark() > 4)

{

out << students[i] << endl;

}

}

}

void MyBase::OutFile(string path)

{

try

{

fstream out(path, ios::out | ios::trunc);

if (!out)

{

throw (string)"Error open and write";

}

Out(out);

}

catch(string error)

{

cout << error << endl;

cout << "Chech path output, attr file and try do this operation later!" << endl;

}

}

void MyBase::In(int n, istream &in)

{

Student tmp;

if (n == -1)

{

while (!in.eof())

{

in >> tmp;

students.push\_back(tmp);

}

}

else

{

int size = students.size();

students.resize(n + size);

for (int i = 0; i < n; i++)

{

if(!in.eof())

{

in >> tmp;

students[i] = Student(tmp);

}

else

{

throw (string)"End the file! Input cann't be!";

}

}

}

}

void MyBase::InFile(int n, string path)

{

fstream in;

try

{

in.open(path, ios::in);

if (!in)

{

throw (string)"Error open and read";

}

In(n, in);

}

catch(string error)

{

cout << error << endl;

cout << "Chech path input and try do this operation later!" << endl;

}

in.close();

}

void MyBase::Sort()

{

sort(students.begin(), students.end());

}

#include <iostream>

#include "MyBase.h"

using namespace std;

int main()

{

MyBase mb;

mb.In(2, cin);

mb.Out(cout);

mb.InFile();

mb.Out(cout);

mb.Sort();

mb.Out();

return 0;

}

Знание кода Жиденко Н.Н. Отметки

Контрольные вопросы Жиденко Н.Н. Отметки

Автор Преподаватель

Жиденко Н.Н. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ (подпись) Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись \_\_\_\_\_\_\_\_ Дата