Министерство образования Республики Беларусь

УО «Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

По дисциплине: “Языки программирования”

Тема: “Классы и объекты в С++”Вариант №6

Выполнил: студент 2 курса

группы ПО-7

Комиссаров Андрей

Проверила: Дряпко А. В.

Брест 2021

Постановка задачи:

1. Определить пользовательский класс в соответствии с вариантом задания
2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования
3. Определить в классе деструктор
4. Определить в классе компоненты-функции для просмотра и установки полей данных
5. Определить указатель на компоненту-функцию
6. Определить указатель на экземпляр класса
7. Написать демонстрационную программу, в которой создаются и разрушаются объекты пользовательского класса и каждый вызов конструктора и деструктора сопровождается выдачей соответствующего сообщения
8. Показать в программе использование указателя на объект и указателя на компоненту-функцию

Цель: получить практические навыки реализации классов на С++

Вариант задания:

Экзамен

Имя студента – char\*

Дата – int

Оценка – int

Код программы:

**Main.cpp**

#include "exam.h"

int main() {

setlocale(LC\_ALL, "russian");

cout << "Создание объектов класса разными способами:" << endl;

auto\* exam\_noParam = new Exam; //без параметров

auto\* exam\_param = new Exam("Andrey", 2, 3); //с параметрами

auto\* exam\_copy = new Exam(\*exam\_param); //копированиe

cout << "\n\nДоступ к объектам через методы изменения полей: " << endl;

cout << "\nсмена имени: ";

exam\_noParam->SetName("первый объект");

exam\_noParam->Print();

cout << "\nсмена даты: ";

exam\_param->SetDate(9);

exam\_param->Print();

cout << "\n\nИспользование указателя на объект(экземпляр) класса: " << endl;

Exam\* exam\_ptr; //создание указателя на объект

exam\_ptr = &\*exam\_copy; //присвоение адреса другого объектa

cout << "\nОбъект, до изменения его через указатель: ";

exam\_copy->Print();

cout << "\nИзменение объекта через указатель...";

exam\_ptr->SetName("Виктор");

cout << "\nВывод объекта через указатель: ";

exam\_ptr->Print();

cout << "\nВывод самого объекта: ";

exam\_copy->Print();

//Использование указателя на метод класса

cout << "\n\nИспользование указателя на метод класса: " << endl;

cout << "\nСоздание указателя на метод...";

void (Exam::\* PrintP)() const;

cout << "\nПрисвоение метода указателю...";

PrintP = &Exam::Print;

cout << "\nВывод объекта через указатель: ";

(exam\_copy->\*PrintP)();

cout << "\n\nУдаление объектов: \n";

delete exam\_noParam;

delete exam\_param;

delete exam\_copy;

}

**Exam.h**

#ifndef EXAM\_H

#define EXAM\_H

#include <iostream>

#include <string>

#include <string.h>

using namespace std;

class Exam {

private:

string name;

int date;

int score;

public:

Exam();

Exam(string name, int date, int score);

Exam(const Exam& exam);

~Exam();

void Print() const;

void SetName(string \_name);

string GetName() const;

void SetDate(int \_date);

int GetDate() const;

void SetScore(int \_score);

int GetScore() const;

};

#endif

**Exam.cpp**

#include "exam.h"

Exam::Exam() : name("NoName"), date(0), score(0)

{

cout << "\nСоздание без параметров - (" << name << ", " << date << ", " << score << ")";

}

Exam::Exam(string name, int date, int score) : name(name), date(date), score(score)

{

cout << "\nСоздание с параметрами - (" << name << ", " << date << ", " << score << ")";

}

Exam::Exam(const Exam& exam) : name(exam.name), date(exam.date), score(exam.score)

{

cout << "\nСоздание с помощью копирования (" <<

name << ", " << date << ", " << score << ")";

}

Exam::~Exam() {

cout << "\nУдаление (" << name << ", " << date << ", " << score << ")";

}

void Exam::Print() const {

cout << '(' << name << ", " << date << ", " << score << ")";

}

void Exam::SetName(string \_name) {

name = \_name;

}

string Exam::GetName() const {

return name;

}

void Exam::SetDate(int \_date) {

date = \_date;

}

int Exam::GetDate() const {

return date;

}

void Exam::SetScore(int \_score) {

score = \_score;

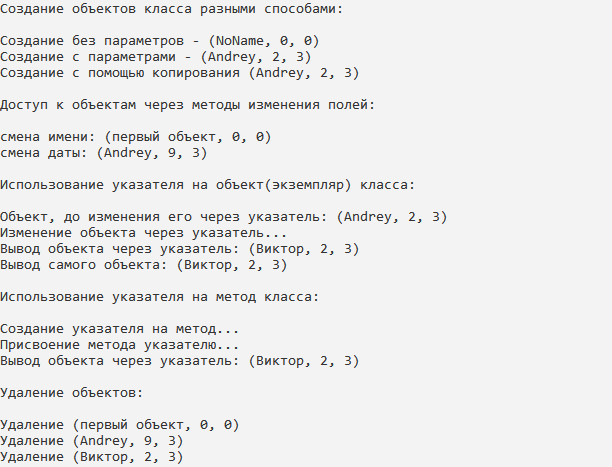
}

int Exam::GetScore() const {

return score;

}

Результат работы программы:



Вывод: Я получил практические навыки реализации классов на С++