Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

за 1 семестр

По дисциплине: «Языки программирования»

Тема: « Классы и объекты в С++ »

Выполнил:

Студент 2 курса

Группы ПО-4 (2)

Синяк Д.А

Проверил:

Хацкевич М. В.

2020

Лабораторная работа №1

Классы и объекты в С++

**Цель работы:** *получить практические навыки реализации классов на С++. Написать программу, в которой создаются и разрушаются объекты, определенного пользователем класса. Выполнить исследование вызовов конструкторов и деструкторов.*

Вариант 1

Задание:

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 (какой объект какой конструктор или деструктор вызвал).

8. Показать в программе использование указателя на объект и указателя на компоненту-функцию.

СТУДЕНТ

имя – char\*

дата – int

курс – int

**Ход работы:**

1) Определение пользовательского класса с комментариями:

class Student {

char\* student\_name;

int student\_kyrs;

int student\_date;

public:

Student(); //конструктор без параметров

Student(char\* name, int date, int kyrs);

Student(char\* name, int kyrs, char\* date); //конструктор с параметрами

Student(const Student& cop); //конструктор копирования

~Student(); //деструктор

void set\_name(char\* name); //функция для установки значений поля имя

void set\_date(int data); //функция для установки значений поля дата

void set\_kyrs(int kyrs); //функция для установки значений поля курса

void set(char\* name, int data, int kyrs); //функция для установки значений полей имя, дата, оценка

void show(); //функция для просмотра данных

void view(Student a); //функция для просмотра данных (нужна, чтобы вызвать констурктор копирования, когда объект передается функции по значению

};

2) Реализация конструкторов и деструкторов:

Student::Student() {

cout << "// Конструктор без параметров " << this << endl;

}

Student::Student(char\* name, int date, int kyrs)

{

student\_name = name;

student\_date = date;

student\_kyrs = kyrs;

cout << "// Конструктор с параметром" << this << endl;

}

Student::Student(const Student& cop)

{

student\_name = cop.student\_name;

student\_date = cop.student\_date;

student\_kyrs = cop.student\_kyrs;

cout << "//// Конструктор копирования " << this << endl;

}

Student::~Student()

{

cout << "// Деструктор" << this << endl;

}

3) Фрагмент программы, показывающий использование указателя на объект (а) и указателя на функцию(б) с объяснением:

а) arr1[1].set(name, 1402, 5);

Student\* p = &arr1[1];//указатель на экземпляр класса

p->show();

//использование динамической памяти

б) Student\* object1 = new Student (name, 2007, 5);

void(Student:: \* pfunction)() = & Student::show; //указатель на компоненту-функцию

(object1 ->\*pfunction)(); //вывод объекта1

4) Код основной программы:

Модуль файла заголовка Class.h:

#pragma once

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

using namespace std;

class Student {

char\* student\_name;

int student\_kyrs;

int student\_date;

public:

Student(); //конструктор без параметров

Student(char\* name, int date, int kyrs);

Student(char\* name, int kyrs, char\* date); //конструктор с параметрами

Student(const Student& cop); //конструктор копирования

~Student(); //деструктор

void set\_name(char\* name); //функция для установки значений поля имя

void set\_date(int data); //функция для установки значений поля дата

void set\_kyrs(int kyrs); //функция для установки значений поля курса

void set(char\* name, int data, int kyrs); //функция для установки значений полей имя, дата, оценка

void show(); //функция для просмотра данных

void view(Student a); //функция для просмотра данных (нужна, чтобы вызвать констурктор копирования, когда объект передается функции по значению

};

Модуль файла ресурсов demo.cpp

#include "Class.h"

Student function(Student& cop) { //функция, необходимая для вызова конструктора копирования при построении временного объекта как возвращаемого значения функции

Student temp(cop);

char\* name = new char[100];

cin.getline(name, 100);

temp.set\_name(name);

return temp;

delete[]name;

}

int main() {

setlocale(LC\_ALL, "rus");

system("color f0");

char\* name = new char[100];

//использование статической памяти

Student arr1[2];

cin.getline(name, 100);

arr1[0].set(name, 1502, 1);

arr1[0].show();

cin.getline(name, 100);

arr1[1].set(name, 1402, 5);

Student\* p = &arr1[1];//указатель на экземпляр класса

p->show();

//использование динамической памяти

Student\* arr2 = new Student[2];

cin.getline(name, 100);

arr2[0].set(name, 1502, 5);

arr2[0].show();

cin.getline(name, 100);

(arr1 + 1)->set(name, 1602, 3);

(arr1 + 1)->view(arr1[1]); //вызов конструктора копирования, когда объект передается фукнции по значению

delete[] arr2;

cin.getline(name, 100);

Student\* object1 = new Student (name, 2007, 5);

void(Student:: \* pfunction)() = & Student::show; //указатель на компоненту-функцию

(object1 ->\*pfunction)(); //вывод объекта1

Student object2 = Student(\*object1); //вызов конструктора копирования при использовании объекта для инициализации другого объекта

object2.set\_date(5);

object2.show();

Student object3 = function(object2); //вызов конструктора копирования при построении временного объекта как возвращаемого значения функции

object3.show();

delete[] name;

delete object1;

system("pause");

return 0;

}

Модуль файлов ресурсов realiz.cpp

#include "class.h"

Student::Student() {

cout << "// Конструктор без параметров " << this << endl;

}

Student::Student(char\* name, int date, int kyrs)

{

student\_name = name;

student\_date = date;

student\_kyrs = kyrs;

cout << "// Конструктор с параметром" << this << endl;

}

Student::Student(const Student& cop)

{

student\_name = cop.student\_name;

student\_date = cop.student\_date;

student\_kyrs = cop.student\_kyrs;

cout << "//// Конструктор копирования " << this << endl;

}

Student::~Student()

{

cout << "// Деструктор" << this << endl;

}

void Student::set\_name(char\* name)

{

student\_name = name;

}

void Student::set\_kyrs(int kyrs)

{

student\_kyrs = kyrs;

}

void Student::set\_date(int date)

{

student\_date = date;

}

void Student::set(char\* name, int date, int kyrs) {

student\_name = name;

student\_date = date;

student\_kyrs = kyrs;

}

void Student::show()

{

cout << "Имя: " << student\_name << endl;

cout << "Курс: " << student\_kyrs << endl;

cout << "Дата: " << student\_date << endl;

}

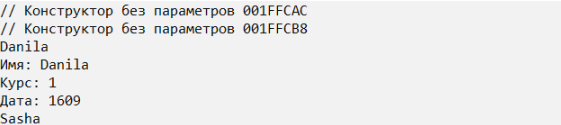
void Student::view(Student a)

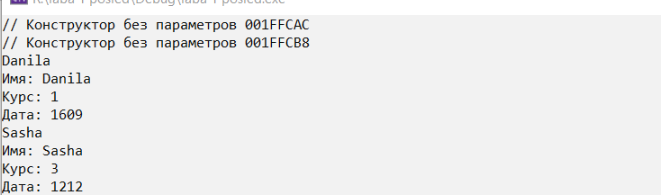
{

a.show();

}

Результат программы:





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