Министерство науки и высшего образования РФ

Федеральное бюджетное государственное образовательное учреждение высшего профессионального образования

Пермский национальный исследовательский политехнический университет (ПНИПУ)

Факультет: Электротехнический (ЭТФ)

Направление: 09.03.01 «Информатика и вычислительная техника (Автоматизированные системы обработки информации и управления)»

Кафедра: «Информационные технологии и автоматизированных систем» (ИТАС)

Информатика

Лабораторная работа №2

Классы и объекты. Использование конструкторов

Студент: Балтаев Э. А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, подпись)

Группа: АСУ-20-1бзу

Работу проверил:

доцент кафедры ИТАС

Полякова О. А.

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Подпись:\_\_\_\_\_\_\_\_\_\_\_

Пермь 2021г.

Постановка задачи:

1. Определить пользовательский класс

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 (селекторы и модификаторы)

5. Написать демонстрационную программу, в которой продемонстрированы все три случая вызова конструктора-копирования, вызов конструктора с параметрами и конструктора без параметров.

Задание варианта:

Пользовательский класс КОМПЬЮТЕР

- Процессор – string

- Объем ОП – int

- Объем ЖД – int

Описание класса

#pragma once

#include <iostream>

#include <string>

using namespace std;

class computer

{

private:

string cpu;

int ram;

int rom;

public:

computer();

computer(string \_cpu, int \_ram, int \_rom);

computer(const computer& c);

~computer();

string get\_cpu();

int get\_ram();

int get\_rom();

void set\_cpu(string \_cpu);

void set\_ram(int \_ram);

void set\_rom(int \_rom);

void show();

};

Определение компонентных функций

#include "computer.h"

computer::computer()

{

cpu = "deafult";

ram = 0;

rom = 0;

}

computer::computer(string \_cpu, int \_ram, int \_rom)

{

cpu = \_cpu;

ram = \_ram;

rom = \_rom;

}

computer::computer(const computer& c)

{

cpu = c.cpu;

ram = c.ram;

rom = c.rom;

}

computer::~computer()

{

cout << "Деструктор" << endl;

}

string computer::get\_cpu()

{

return cpu;

}

int computer::get\_ram()

{

return ram;

}

int computer::get\_rom()

{

return rom;

}

void computer::set\_cpu(string \_cpu)

{

if (\_cpu == "")

cpu = "default";

else

cpu = \_cpu;

}

void computer::set\_ram(int \_ram)

{

if (\_ram < 0)

ram = 0;

else

ram = \_ram;

}

void computer::set\_rom(int \_rom)

{

if (\_rom < 0)

rom = 0;

else

rom = \_rom;

}

void computer::show()

{

cout << cpu << " [О: " << ram << ", Ж: " << rom << "]" << endl;

}

Определение главного файла

#include <iostream>

#include "computer.h"

computer make\_computer() {

string cpu;

int ram, rom;

cout << "Процессор? "; cin >> cpu;

cout << "Оперативная память? "; cin >> ram;

cout << "Жесткий диск? "; cin >> rom;

computer temp(cpu, ram, rom);

return temp;

}

int main()

{

system("chcp 1251");

cout << "\nКонструктор без параметров\n";

computer c1;

c1.show();

c1.set\_cpu("GTX 9000");

c1.set\_ram(10);

c1.set\_rom(500);

c1.show();

cout << "\nКонструктор с параметрами\n";

computer c2 = make\_computer();

c2.show();

cout << "\nКонструктор копирования\n";

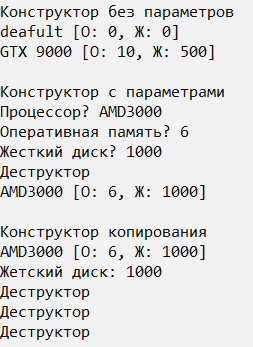
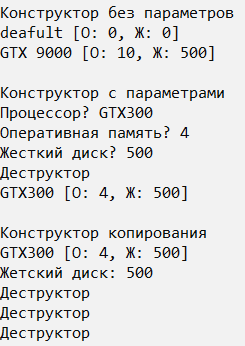
computer c3(c2);

c3.show();

cout << "Жетский диск: " << c3.get\_rom() << endl;

}

Результаты программы:

Ответы на контрольные вопросы

1) Для инициализации объекта

2) 3: с параметрами, без и копирования

3) Для выполнения действий перед удалением объекта. Если есть массивы или указатели

4) Создание по-умолчанию, с помощью аргументов и для копирования

5) При передаче в функцию или получение из функции

6) Не возвращает значение. Может быть несколько. Если не создано, будет создано автоматически. Любой тип в аргументе. Не наследуются. Без модификаторов.

7) Для освобождения ресурсов. Вызывается автоматически перед удалением. Создано автоматически, если забыли создать.

8) Всем

9) Указатель на текущий объект

10) Обращение внутри класса непосредственно сразу к полям, вне класса с помощью объекта

11) Не возвращает

12) Конструктор, деструктор

13) Не возвращает

14) конструктор по-умолчанию

15) конструктор по-умолчанию

16) конструктор с параметрами

17) конструктор с параметрами и копирования

18) конструктор с параметрами и оператор присвоения ссылки

19) копирования

20) p.set\_name(“newValue”);