Министерство образования и науки Российской Федерации

Федеральное государственное образовательное автономное учреждение высшего образования

"Пермский национальный исследовательский политехнический университет"

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

Дисциплина: Основы алгоритмизации и программирования

Тема: Классы и объекты. Использование конструкторов.

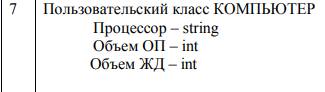
|  |  |
| --- | --- |
| Выполнил работу | |
| Студент группы РИС-22-1б | |
| Бреднев М.П. | |
|  | |
| Проверил работу | |
| Доцент кафедры ИТАС | |
| Полякова О.А. | |
|  | |

Пермь – 2023

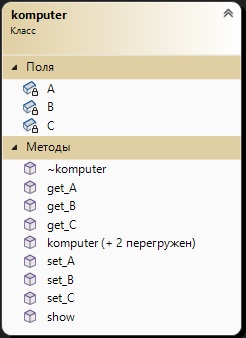
**Анализ предметной области**

**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 продемонстрировать все три случая вызова конструкторов.



**Диаграмма классов**



**Код программы**

Computer.h

#pragma once

#include <iostream>

#include <string>

using namespace std;

class komputer

{

string A;

int B;

int C;

public:

komputer(); // конструктор без параметров

komputer(string, int, int); // конструктор с параметрами

komputer(const komputer&); // консутруктор копирования

~komputer(); // деструктор

string get\_A(); // селектор

void set\_A(string); // модификатор

int get\_B();

void set\_B(int);

int get\_C();

void set\_C(int);

void show(); // просмотр атрибутов

};

Computer.cpp

#include "Computer.h"

#include <iostream>

#include <string>

using namespace std;

// конструктор без параметров

komputer::komputer()

{

string A = " ";

B = 0;

C = 0;

cout << "Constructor bez parametrov dlya objecta " << this << endl;

}

// конструктор с параметрами

komputer::komputer(string a, int b, int c)

{

A = a;

B = b;

C = c;

cout << "Constructor s parametrami dlya objecta " << this << endl;

}

// конструктор копирования

komputer::komputer(const komputer& t)

{

A = t.A;

B = t.B;

C = t.C;

cout << "Constructor copirovaniya dlya objecta " << this << endl;

}

// детструктор

komputer::~komputer()

{

cout << "Destructor dlya objecta " << this << endl;

}

// селекторы

string komputer::get\_A()

{

return A;

}

int komputer::get\_B()

{

return B;

}

int komputer::get\_C()

{

return C;

}

// модификаторы

void komputer::set\_A(string a)

{

A = a;

}

void komputer::set\_B(int b)

{

B = b;

}

void komputer::set\_C(int c)

{

C = C;

}

// просмотр атрибутов

void komputer::show()

{

cout << "CPU : " << A << endl;

cout << "RAM : " << B << endl;

cout << "HDD : " << C << endl;

cout << endl;

}

main.cpp

#include "Computer.h"

#include <iostream>

#include <string>

using namespace std;

komputer add\_komputer()

{

string a;

int b, c;

cout << "Enter CPU >> ";

cin >> a;

cout << "Enter size of RAM >> ";

cin >> b;

cout << "Enter size of HDD >> ";

cin >> c;

komputer t(a, b, c);

return t;

}

void print\_komputer(komputer t)

{

t.show();

}

void main()

{

// конструктор без параметров

komputer t1;

// конструктор с параметром

komputer t2("intel core i5", 8, 1000);

komputer t3 = t2;

t3.set\_A("ryzen 5600");

t3.set\_B(4);

t3.set\_C(500);

// вывод

print\_komputer(t1);

print\_komputer(t2);

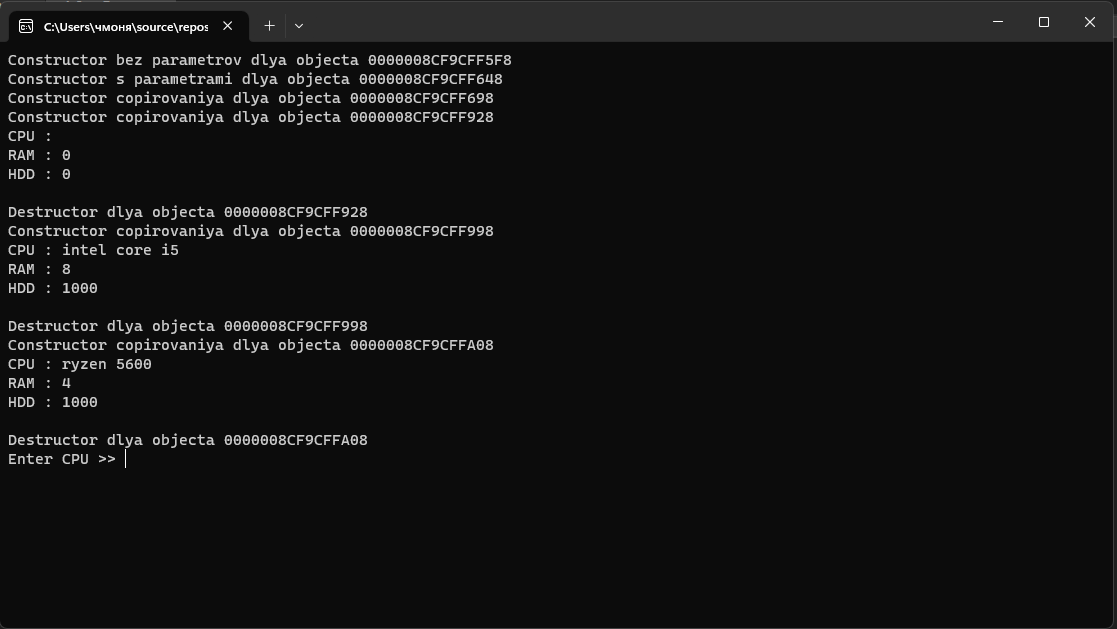
print\_komputer(t3);

t1 = add\_komputer();

t1.show();

}

**Тестирование программы**

****

**Ответы на контрольные вопросы**

1. Для чего нужен конструктор?

Конструктор в языке программирования С++ нужен для инициализации объектов класса.

1. Сколько типов конструкторов существует в С++?

В языке С++ существует три типа конструкторов: конструктор без параметров, конструктор с параметрами, конструктор копирования и конструктор переноса.

1. Для чего используется деструктор? В каких случаях деструктор описывается явно?

Деструктор в С++ используется для освобождения памяти, занятой объектом класса. Деструктор описывается явно в тех случаях, когда объект класса содержит динамические данные, которые должны быть освобождены при уничтожении объекта.

1. Для чего используется конструктор без параметров? Конструктор с параметрами? Конструктор копирования?

Конструктор без параметров используется для создания объекта класса с значениями по умолчанию. Конструктор с параметрами используется для создания объекта класса с заданными значениями атрибутов. Конструктор копирования используется для создания копии объекта класса.

1. В каких случаях вызывается конструктор копирования?

Конструктор копирования вызывается в тех случаях, когда объект класса передается в качестве аргумента в функцию по значению, возвращается из функции объект класса по значению или происходит инициализация одного объекта класса другим объектом класса.

1. Перечислить свойства конструкторов.

Свойства конструкторов:

* Имя конструктора совпадает с именем класса
* Не имеют возвращаемого значения
* Могут быть перегружены

1. Перечислить свойства деструкторов.

Свойства деструкторов:

* Имя деструктора совпадает с именем класса с добавлением символа ~ в начале
* Не имеют параметров и возвращаемого значения
* Могут быть перегружены
* Вызываются автоматически при уничтожении объекта

1. К каким атрибутам имеют доступ методы класса?

Методы класса имеют доступ к атрибутам объекта класса.

1. Что представляет собой указатель this?

Указатель this представляет адрес объекта, для которого был вызван метод.

1. Какая разница между методами, определенными внутри класса и вне класса?

Методы, определенные внутри класса, имеют доступ к приватным и защищенным атрибутам класса, а методы, определенные вне класса, могут иметь доступ только к публичным атрибутам класса.

1. Какое значение возвращает конструктор?

Конструктор не возвращает никакого значения.

1. Какие методы создаются по умолчанию?

Методы, создаваемые по умолчанию: конструктор без параметров, конструктор копирования, деструктор, оператор присваивания и оператор сравнения.

1. Какое значение возвращает деструктор?

Деструктор не возвращает никакого значения.

1. Дано описание класса

class Student {

string name;

int group;

public:

student(string, int);

student(const student&)

~student(); }; Какой метод отсутствует в описании класса?

В описании класса отсутствует определение метода конструктора по умолчанию.

1. Какой метод будет вызван при выполнении следующих операторов: student\*s; s=new student;

Будет вызван конструктор без параметров.

1. Какой метод будет вызван при выполнении следующих операторов: student s(“Ivanov”,20);

Будет вызван конструктор с параметрами, переданными в качестве аргументов.

1. Какие методы будут вызваны при выполнении следующих операторов: student s1(“Ivanov”,20); student s2=s1;

Будет вызван конструктор копирования.

1. Какие методы будут вызваны при выполнении следующих операторов: student s1(“Ivanov”,20); student s2; s2=s1;

Будет вызван конструктор с параметрами затем будет вызван конструктор копирования для создания объекта s2 на основе объекта s1, а затем будет вызван оператор присваивания для копирования значений полей объекта s1 в объект s2.

1. Какой конструктор будет использоваться при передаче параметра в функцию print(): void print(student a) {a.show();}

Будет вызван конструктор копирования, так как объект a передается в функцию print() по значению, а не по ссылке.

1. Класс описан следующим образом:

class Student {

string name;

int age;

public:

void set\_name(string);

void set\_age(int ); ….. };

Student p;

Каким образом можно присвоить новое значение атрибуту name объекта р? Для того чтобы присвоить новое значение атрибуту name объекта р, можно использовать метод set\_name(), например: p.set\_name("New name"), а также метод set\_age.