Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: «Классы и объекты. Использование конструкторов»

Выполнил работу

Студент группы РИС-22-1Б

Шуракова А.А.

Проверил

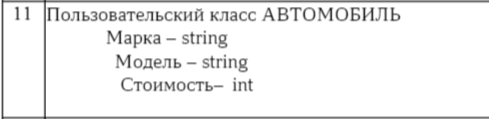
Доцент кафедры ИТАС

Полякова О.А.

Г. Пермь-2024

**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 конструктора-копирования, вызов конструктора с параметрами и конструктора без параметров.



**Код программы:**

Car.h:

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Car {

string brand;

string model;

int price;

public:

Car();//конструктор без параметров

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

Car(const Car&);//конструктор копирования

~Car();//деструктор

string getBrand();//селектор

void setBrand(string);//модификатор

string getModel();//селектор

void setModel(string); //модификатор

int getPrice();//селектор

void setPrice(int); //модификатор

void show();//просмотр атрибутов

};

Lab2\_main.cpp:

#include "Car.h"

#include <iostream>

#include <string>

using namespace std;

//конструктор без параметров

Car::Car()

{

brand = "";

model = "";

price = 0;

cout << "Constructor without parameters " << this << endl;

}

//конструктор с параметрами

Car::Car(string N, string K, int S)

{

brand = N;

model = K;

price = S;

cout << "Constructor with parameters " << this << endl;

}

//конструктор копирования

Car::Car(const Car& t)

{

brand = t.brand;

model = t.model;

price = t.price;

cout << "Copy constructor " << this << endl;

}

//деструктор

Car::~Car()

{

cout << "Destructor " << this << endl;

}

//селекторы

string Car::getBrand()

{

return brand;

}

string Car::getModel()

{

return model;

}

int Car::getPrice()

{

return price;

}

//модификаторы

void Car::setBrand(string N)

{

brand = N;

}

void Car::setModel(string K)

{

model = K;

}

void Car::setPrice(int S)

{

price = S;

}

//метод для просмотра атрибутов

void Car::show()

{

cout << "Brand :" << brand << endl;

cout << "Model :" << model << endl;

cout << "Price :" << price << endl;

}

Car.cpp:

#include "Car.h"

#include <iostream>

#include <string>

using namespace std;

//функция для возврата объекта как результата

Car makeCar()

{

string b;

string m;

int c;

cout << "Enter brand: ";

cin >> b;

cout << "Enter model: ";

cin >> m;

cout << "Enter price: ";

cin >> c;

Car s(b, m, c);

return s;

}

//функция для передачи объекта как параметра

void printCar(Car s)

{

s.show();

}

int main()

{

system("chcp 1251");

//конструктор без параметров

Car t1;

t1.show();

//коструктор с параметрами

Car t2("Lada ", "Vesta", 1500000);

t2.show();

//конструктор копирования

Car t3 = t2;

t3.setBrand("Mersedes");

t3.setModel("Benz");

t3.setPrice(5000000);

//конструктор копирования

printCar(t3);

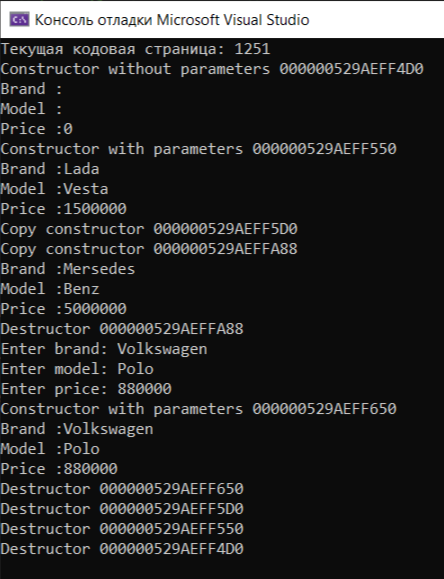
//конструктор копирования

Car t4 = makeCar();

t4.show();

}

**Результат работы программы:**

****

**Ответы на контрольные вопросы**

1. Для чего нужен конструктор?

Конструктор необходим для инициализации полей класса.

2. Сколько типов конструкторов существует в C++?

* Default Constructor (Конструктор без параметров) — конструктор, который не принимает параметров.
* Parameterized Constructor (Конструктор с параметрами) — конструктор, который может принимать один или несколько аргументов.
* Copy Constructor (Конструктор копирования) — конструктор, который используется для создания копии уже существующего объекта класса.
* Dynamic Constructor (Динамический конструктор) — конструктор, который позволяет динамически инициализировать объекты.

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

Деструктор используется для завершения работы с конструктором класса после выхода из области видимости функции. Если конструктор описан явно, то необходимо и деструктор описать явно.

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

* Конструктор без параметров используется для инициализации объекта с дефолтными значениями или значениями по умолчанию (заранее заданными значениями).
* Конструктор с параметрами используется для инициализации объекта с переданными значениями.
* Конструктор копирования используется для создания копии объекта существующего объекта того же типа.

5. В каких случаях вызывается конструктор копирования?

Конструктор копирования вызывается в случае, когда нужно заполнить n обьектов одинаковыми значениями полей.

6. Перечислить свойства конструкторов.

* Конструктор не возвращает значение, даже типа void. Нельзя получить указатель на конструктор.
* Класс может иметь несколько конструкторов с разными параметрами для разных видов инициализации (при этом используется механизм перегрузки).
* Конструктор, вызываемый без параметров, называется конструктором по умолчанию.
* Параметры конструктора могут иметь любой тип, кроме этого же класса. Можно задавать значения параметров по умолчанию. Их может содержать только один из конструкторов.
* Если программист не указал ни одного конструктора, компилятор создает его автоматически.
* Конструкторы не наследуются.

7. Перечислить свойства деструкторов.

* Класс может иметь только один деструктор.
* Деструктор имеет то же имя, что и класс, в котором он объявляется, с префиксом тильда (~).
* Деструктор не имеет параметров (аргументов).
* Деструктор не возвращает значения (даже типа void).
* Деструктор не наследуется в производных классах.
* Деструктор — это функция, и он может быть виртуальным (его можно объявить с ключевым словом virtual).
* В программе невозможно получить адрес деструктора (указатель на деструктор).
* Если деструктор не задан в программе, он будет автоматически сгенерирован компилятором для уничтожения соответствующих объектов.
* Деструктор можно вызвать так же, как обычную функцию, если использовать его полное имя.
* Деструктор вызывается автоматически только при разрушении (уничтожении) объекта.

8. К каким атрибутам имеют доступ методы класса?

Методы класса имеют доступ ко всем членам класса, включая его публичные, защищенные и приватные атрибуты. (Private, Public, Protected)

9. Что представляет собой указатель this?

Указатель this - это указатель, который хранит адрес конкретного объекта класса.

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

Методы, определенные внутри класса, считаются методами-членами класса и имеют прямой доступ ко всем членам класса, включая приватные. Методы, определенные вне класса, являются методами-прототипами и требуют явного указания класса при их определении.

11. Какое значение возвращает конструктор?

Конструкторы не возвращают значений.

12. Какие методы создаются по умолчанию?

* Конструктор по умолчанию (без параметров)
* Конструктор копирования
* Оператор присваивания (assignment operator)
* Деструктор

13. Какое значение возвращает деструктор?

Деструктор не возвращает значений.

14. Дано описание класса  
class Student  
{  
 string name;  
 int group;  
 public:  
 student(string, int);  
 student(const student&)  
 ~student();  
};  
Какой метод отсутствует в описании класса?

Отсутствует конструктор без параметров.

15. Какой метод будет вызван при выполнении следующих операторов:

student\*s;

s=new student;

Конструктор без параметров.

16. Какой метод будет вызван при выполнении следующих операторов: student s("Ivanov",20);

Конструктор с параметрами.

17. Какие методы будут вызваны при выполнении следующих операторов: student s1("Ivanov",20);

student s2=s1;

Конструктор с параметрами, конструктор копирования.

18. Какие методы будут вызваны при выполнении следующих операторов: student s1("Ivanov",20);

student s2;

s2=s1;

Конструктор с параметрами, конструктор без параметров, конструктор копирования.

19. Какой конструктор будет использоваться при передаче параметра в функцию print():

void print(student a)

{a.show();}

Конструктор копирования.

20. 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?

p.set\_name("Новое имя");