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 ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

Дисциплина: «Основы алгоритмизации и программирования»

Тема : "Классы и объекты. Использование конструкторов."

Выполнила работу

Студентка группы РИС-22-1Б

Лихачев Д.А.

Проверила

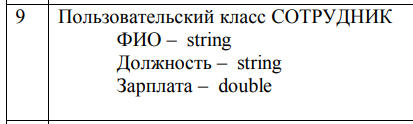
Доцент кафедры ИТАС

Полякова О.А.

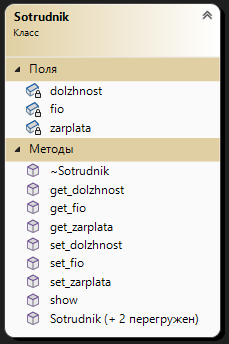
Г. Пермь-2023

**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.



**Диаграмма класса**



**Код программы**

Sotrudnik.h:

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Sotrudnik

{

private:

string fio;

string dolzhnost;

double zarplata;

public:

Sotrudnik();

Sotrudnik(string, string, double);

Sotrudnik(const Sotrudnik& p);

~Sotrudnik();

string get\_fio();

string get\_dolzhnost();

double get\_zarplata();

void set\_fio(string);

void set\_dolzhnost(string);

void set\_zarplata(double);

void show();

};

Sotrudnik.cpp:

#include "Sotrudnik.h"

#include <iostream>

#include <string>

using namespace std;

Sotrudnik::Sotrudnik() {

this->fio = "";

this->dolzhnost = "";

this->zarplata = 0;

}

Sotrudnik::Sotrudnik(string fio, string dolzhnost, double zarplata) {

this->fio = fio;

this->dolzhnost = dolzhnost;

this->zarplata = zarplata;

}

Sotrudnik::Sotrudnik(const Sotrudnik& p) {

fio = p.fio;

dolzhnost = p.dolzhnost;

zarplata = p.zarplata;

}

Sotrudnik::~Sotrudnik() {

cout << "Destructor for object: " << this << endl;

}

string Sotrudnik::get\_fio() {

return fio;

}

string Sotrudnik::get\_dolzhnost() {

return dolzhnost;

}

double Sotrudnik::get\_zarplata() {

return zarplata;

}

void Sotrudnik::set\_fio(string fio) {

this->fio = fio;

}

void Sotrudnik::set\_dolzhnost(string dolzhnost) {

this->dolzhnost = dolzhnost;

}

void Sotrudnik::set\_zarplata(double zarplata) {

this->zarplata = zarplata;

}

void Sotrudnik::show() {

cout << "FIO: " << fio << endl;

cout << "Dolzhnost: " << dolzhnost << endl;

cout << "Zarplata: " << zarplata << endl;

cout << endl;

}

Lab2\_main.cpp:

#include "Sotrudnik.h"

#include <iostream>

#include <string>

using namespace std;

Sotrudnik make\_sotrudnik() {

string fio, dolzhnost;

double zarplata;

cout << "Enter FIO: "; getline(cin,fio);

cout << "Enter dolzhnost: "; getline(cin,dolzhnost);

cout << "Enter double: "; cin >> zarplata;

Sotrudnik ex(fio, dolzhnost, zarplata);

return ex;

}

void print\_sotrudnik(Sotrudnik ex) {

ex.show();

}

int main()

{

Sotrudnik s1;

s1.show();

Sotrudnik s2("Bulochkin Vasiliy Olegovich","Lawyer", 30000);

s2.show();

Sotrudnik s3 = s2;

cout << "Dolzhnost s3: " << s3.get\_dolzhnost() << endl;

s3.set\_fio("Tapochkin Oleg Viktorovich");

s3.set\_dolzhnost("Driver");

s3.set\_zarplata(100000);

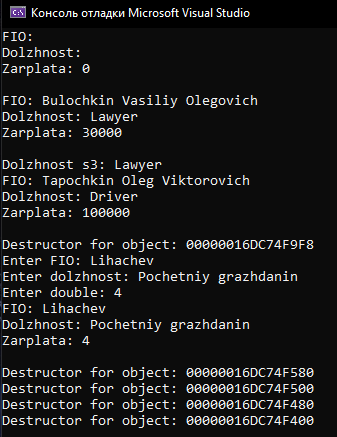
print\_sotrudnik(s3);

Sotrudnik s4 = make\_sotrudnik();

s4.show();

}

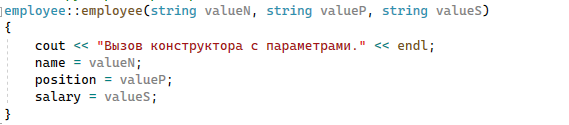
**Результаты работы программы**



**Ответы на контрольные вопросы**

1. Для чего нужен конструктор?

Конструктор необходим для инициализации полей класса.

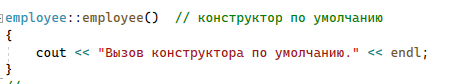


  
2. Сколько типов конструкторов существует в C++?

Существует 3 типа конструкторов: Без параметров, с параметрами, конструктор копирования.

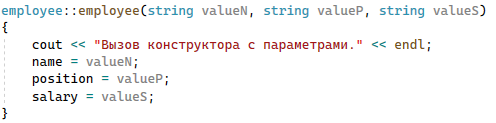
- конструктор без параметров:





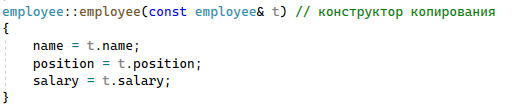
- конструктор с параметрами:



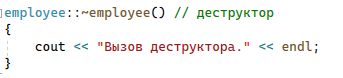


- конструктор копирования:



  
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?

Без параметров – для инициализации полей по умолчанию(заранее заданными значениями). С параметрами – для инициализации полей переданными в конструктор значениями. Копирования – для инициализации полей с помощью копирования их из другого обьекта.

5. В каких случаях вызывается конструктор копирования?

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

6. Перечислить свойства конструкторов.

Инициализация полей переданными параметрами, инициализация полей по умолчанию, инициализация полей с помощью копирования, перегрузка конструкторов, конструктор списка по умолчанию, конструктор списка инициализаторов.

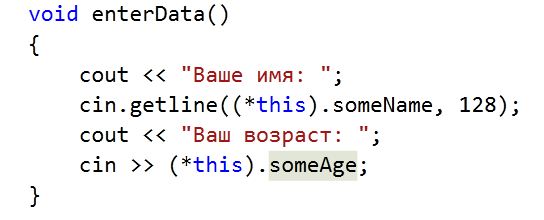
7. Перечислить свойства деструкторов.

Не могут иметь аргументов; Не возвращают значение (или void ); Не может быть объявлен как const , volatile или static . Однако они могут вызываться для уничтожения объектов, объявленных как const , volatile или static ; Может быть объявлен как virtual . Используя виртуальные деструкторы, можно уничтожать объекты, не зная их тип — правильный деструктор для объекта вызывается с помощью механизма виртуальных функций. Обратите внимание, что для абстрактных классов деструкторы также могут объявляться как чисто виртуальные функции; не описаны явно – автоматически вызовутся.

8. К каким атрибутам имеют доступ методы класса?

Private, Public, Protected;  
9. Что представляет собой указатель this?

Адрес текущего обьекта класса.



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

Внутри класса можно не передать параметры, вне – обязательно. Использование шаблонов ф-ий порой обязательно.

11. Какое значение возвращает конструктор?

Конструкторы не возвращают значений.

12. Какие методы создаются по умолчанию?

Конструктор и 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?  
15. Какой метод будет вызван при выполнении следующих операторов:  
student\*s;  
s=new student;

16. Какой метод будет вызван при выполнении следующих операторов: student s("Ivanov",20);  
17. Какие методы будут вызваны при выполнении следующих операторов: student s1("Ivanov",20);  
student s2=s1;  
18. Какие методы будут вызваны при выполнении следующих операторов: student s1("Ivanov",20);  
student s2;  
s2=s1;  
19. Какой конструктор будет использоваться при передаче параметра в функцию print():  
void print(student a)  
{a.show();}  
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?

* Аналогично, x
* Сеттер.
* Никакой.
* Конструктор с параметрами.
* Конструктор с параметрами для первого обьекта и конструктор копирования для второго.
* Аналогично предыдущему, только сначала второй обьект вызовет конструктор без параметров.
* С параметрами.
* p.set\_name(“name”);