Министерство высшего образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет» (ПНИПУ)**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ

по лабораторной работе №2

Тема: «Классы и объекты. Инкапсуляция.»

Выполнил

Студент группы РИС-22-2б

Худеньких В.Д.

Проверил доц. Кафедры ИТАС

Полякова Ольга Андреевна

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.

ВАРИАНТ 15:

Пользовательский класс ЗАРПЛАТА

- ФИО – string

- Оклад – double

- Премия (% от оклада) – int

**Контрольные вопросы**

1. Для чего нужен конструктор?

*Метод для инициализации объекта класса, который позволяет избежать ошибки, связанные с использованием неинициализированных переменных*

1. Сколько типов конструкторов существует

*3 типа: конструктор с параметрами и без, конструктор копирования*

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

*Для удаления ресурсов, выделенных под объект (включая локальные атрибуты). Явно, если создаются динамические типы данных (т.е. указатели), которые нужно очищать.*

4-5. Для чего конструктор без параметров? С параметрами? Копирования?

*Конструктор копирования вызывается в тех случаях, когда новый объект создается путем копирования существующего:*

* *при описании нового объекта с инициализацией другим объектом;*
* *при передаче объекта в функцию по значению;*
* *при возврате объекта из функции.*

*Конструктор без параметров нужен для создания “пустого” экземпляра класса.*

*Конструктор с параметрами для инициализации объекта с заданными параметрами.*

1. Перечислите св-ва 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, но их может содержать только один из конструкторов.*
* *Если программист не указал ни одного конструктора, компилятор создает его автоматически. Такой конструктор вызывает конструкторы по умолчанию для полей класса. В случае, когда класс содержит константы или ссылки, при попытке создания объекта класса будет выдана ошибка, поскольку их необходимо инициализировать конкретными значениями, а конструктор по умолчанию этого делать не умеет.*
* *Конструкторы не наследуются.*
* *Конструкторы нельзя описывать с модификаторами const, virtual и static.*
* *Конструкторы глобальных объектов вызываются до вызова функции main. Локальные объекты создаются, как только становится активной область их действия. Конструктор запускается и при создании временного объекта (на пример, при передаче объекта из функции).*

1. Св-ва деструкторов
   * + *не имеет аргументов и возвращаемого значения;*
     + *не наследуется;*
     + *не может быть объявлен как const или static (далее);*
     + *может быть виртуальным (далее).*
2. К каким атрибутам имеют доступ методы класса

*Ко всем элементам класса вне зависимости от спецификатора доступа*

1. Что представляет собой запись: this

*Указатель на текущий объект*

1. Как разница между методами определенными внутри класса и вне класса

*Нет разницы.*

1. Какое значение возвращает конструктор?

*Конструктор ничего не возвращает*

1. Какие методы создаются по умолчанию?

*Конструктор без параметров, копирования, деструктор*

1. Какое значение возвращает деструктор?

*Деструктор ничего не возвращает*

1. *Конструктор без параметров*
2. *Конструктор без параметров*
3. *Конструктор с параметрами*
4. *Конкструктор с параметрами, копирования*
5. *Конструктор с параметрами и без, копирования*
6. *Конструктор копирования*
7. *С помощью сеттера set\_name(string);*

***Код программы***

*Заголовочный файл, который содержит описание класса (point.h)*

#pragma once

#include <string>;

class **Point**

{

public:

std::string FIO;

**Point**();

**Point**(std::string name, double salary, int award);

**Point**(const Point& obj);

~**Point**();

double **get\_salary**();

void **set\_salary**(double salary);

int **get\_award**();

void **set\_award**(int award);

void **show**();

private:

double salary;

int award;

};

*Файл, содержащий описание методов класса salary (point.cpp)*

#pragma once

#include "Point.h"

#include <iostream>

Point::**Point**()

{

FIO = "";

this->salary = 0;

this->award = 0;

}

Point::**Point**(std::string name, double salary, int award)

{

FIO = name;

this->salary = salary;

this->award = award;

}

Point::**Point**(const Point& obj)

{

this->FIO = obj.FIO;

this->salary = obj.salary;

this->award = obj.award;

}

Point::~**Point**()

{

std::cout << this << std::endl;

}

double Point::**get\_salary**()

{

return salary;

}

void Point::**set\_salary**(double salary)

{

this->salary = salary;

}

int Point::**get\_award**()

{

return award;

}

void Point::**set\_award**(int award)

{

this->award = award;

}

void Point::**show**()

{

std::cout << "FIO: " << FIO << std::endl << "salary: " << salary << std::endl << "award: " << award << std::endl;

}

*Файл, содержащий основную программу (main.cpp)*

#include "Point.h";

#include <iostream>

void **print**(Point& t)

{

t.show();

}

Point **set\_obj**(std::string FIO, double salary, int award) {

Point s(FIO, salary, award);

return s;

}

int **main**() {

setlocale(LC\_ALL, ".1251");

Point p1;

p1.show();

p1.FIO = "Danil";

p1.set\_award(15170.7);

p1.set\_salary(1000);

p1.show();

Point p2("Mihail", 9251, 500);

print(*p2*);

std::cout << p2.get\_salary() << " " << p2.get\_award() << std::endl;

Point p3 = set\_obj("Sasha", 1251.4, 200);

p3.show();

return 0;

}