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

Баяндин К. С.

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 (salary.h)*

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Salary

{

//атрибуты

string name;

double salary;

int prize;

public:

Salary(); //конструктор без параметров

Salary(string, double, int); //конструктор с параметрами

Salary(const Salary&); //конструктор копирования

~Salary(); //деструктор

string get\_name(); //селектор

void set\_name(string); //модификатор

double get\_salary(); //селектор

void set\_salary(double); //модификатор

int get\_prize(); //селектор

void set\_prize(int); //модификатор

void show(); //просмотр атрибутов

};

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

#include "salary.h"

#include <iostream>

#include <string>

using namespace std;

//конструктор без параметров

Salary::Salary()

{

name = "";

salary = 0;

prize = 0;

cout << "Конструктор без параметров для объекта " << this << endl;

}

//конструктор с параметрами

Salary::Salary(string N, double S, int P)

{

name = N;

salary = S;

prize = P;

cout << "Конструктор с параметрами для объекта " << this << endl;

}

//конструктор копирования

Salary::Salary(const Salary& s)

{

name = s.name;

salary = s.salary;

prize = s.prize;

cout << "Конструктор копирования для объекта " << this << endl;

}

//деструктор

Salary::~Salary()

{

cout << "Деструктор для объекта " << this << endl;

}

//селекторы

string Salary::get\_name()

{

return name;

}

double Salary::get\_salary()

{

return salary;

}

int Salary::get\_prize()

{

return prize;

}

//модификаторы

void Salary::set\_name(string N)

{

name = N;

}

void Salary::set\_salary(double S)

{

salary = S;

}

void Salary::set\_prize(int P)

{

prize = P;

}

//метод для просмотра атрибутов

void Salary::show()

{

cout << "ФИО: " << name << endl;

cout << "Оклад: " << salary << endl;

cout << "Премия (% от оклада): " << prize << endl;

}

*Файл, содержащий основную программу (Lab2\_main.cpp)*

#include "salary.h"

#include <iostream>

#include <string>

#include <Windows.h>

using namespace std;

//функция для возврата объекта как результата

Salary make\_salary()

{

string n;

cout << "Введите ФИО: ";

getline(cin, n);

cout << endl;

double s;

cout << "Введите оклад: ";

cin >> s;

cout << endl;

int p;

cout << "Введите премию(% от оклада): ";

cin >> p;

cout << endl;

Salary t(n, s, p);

return t;

}

//функция для передачи объекта как параметра

void print\_salary(Salary t)

{

t.show();

}

void main()

{

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//конструктор без параметров

Salary s1;

s1.show();

//конструктор с параметрами

Salary s2("Баяндин Кирилл Семенович", 100000, 10000);

s2.show();

//конструктор копирования

Salary s3 = s2;

s3.set\_name("Бражкин Егор Викторович");

s3.set\_salary(120000);

s3.set\_prize(12000);

//конструктор копирования

print\_salary(s3);

//конструктор копирования

s1 = make\_salary();

s1.show();

}