Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

**ОТЧЕТ**

**Тема:** Лабораторная работа №2 по ООП

Семестр: 2

Выполнил студент ИВТ-22-2б:

Мельников Глеб Владимирович

(дата, подпись)

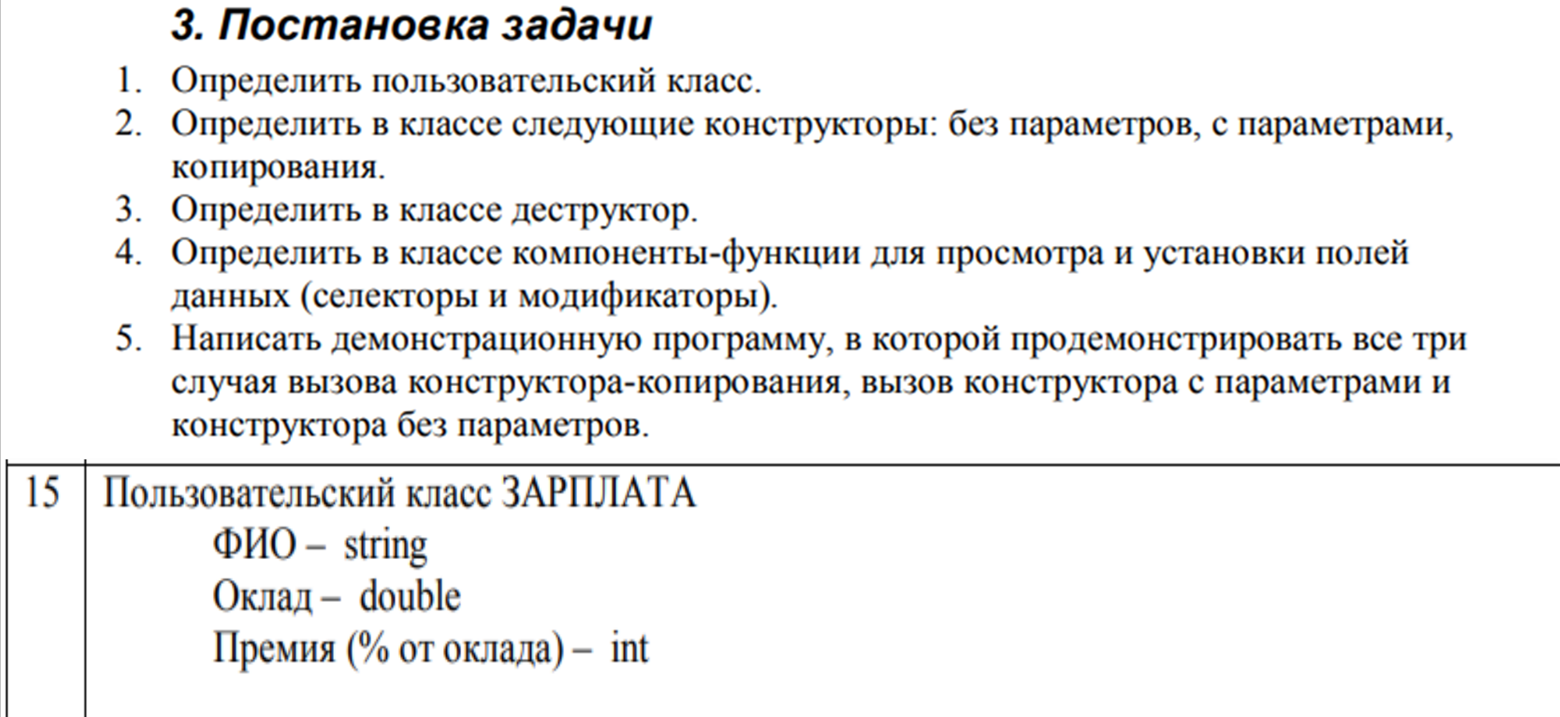
Проверила:

Полякова Ольга Андреевна

(дата, подпись)

Пермь 2023

**Задание**



**Анализ задачи**

1. Метод SetName – установление имени
2. Метод SetReward – установление зарплаты
3. Метод SetPrcnt – установление оклада
4. Метод GetPrcnt – вывод оклада
5. Метод Show – вывод данных

**UML-Диаграмма**

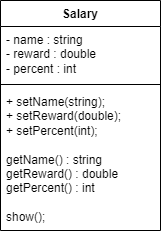
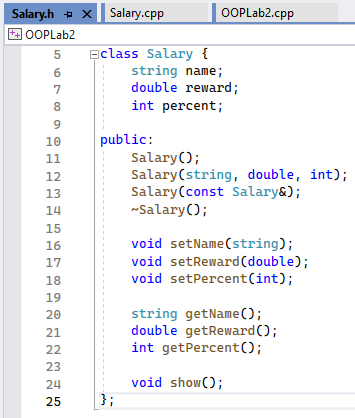


Рисунок 1 – UML-диаграмма класса «Salary»

**Программное решение**





#include <iostream>

#include <string>

#include "Salary.h"

using namespace std;

Salary::Salary() {

cout << "Вызов конструктора без параметров" << '\n';

name = "";

reward = 0;

percent = 0;

}

Salary::Salary(string s, double d, int i) {

cout << "Вызов конструктора с параметрами" << '\n';

name = s; reward = d; percent = i;

}

Salary::Salary(const Salary& t) {

cout << "Вызов конструктора копирования" << '\n';

name = t.name;

reward = t.reward;

percent = t.percent;

}

Salary::~Salary() {

cout << "Вызов деструктора" << '\n';

}

void Salary::setName(string name) {

this->name = name;

}

void Salary::setReward(double reward) {

this->reward = reward;

}

void Salary::setPercent(int percent) {

percent = percent;

}

string Salary::getName() {

return name;

}

double Salary::getReward() {

return reward;

}

int Salary::getPercent() {

return percent;

}

void Salary::show() {

cout << "Name: " << name << '\n';

cout << "Reward: " << reward << '\n';

cout << "Percent: " << percent << '\n';

}



#include <iostream>

#include <string>

#include "Salary.h"

using namespace std;

Salary makeStaff();

void print(Salary);

int main() {

system("chcp 1251 > NULL");

Salary s1; // конструктор без параметров

s1.show();

Salary s2("Антонов Дмитрий Сергеевич", 200054, 45); // Конструктор с параметрами

s2.show();

Salary s3 = s2; // Конструктор копирования

print(s3); // конструктор копирования

// Вызов деструктора (так как в параметрах функции не ссылка)

Salary s4 = makeStaff();

s4.show();

// Деструктор вызывается для каждого объекта

return 0;

}

Salary makeStaff() {

string name;

double r;

int i;

cout << "Введите ФИО сотрудника:" << '\n';

getline(cin, name);

cout << "Зарплата:" << '\n';

cin >> r;

cout << "Премия%:" << '\n';

cin >> i;

Salary t(name, r, i); // конструктор с параметрами

return t; // конструктор копирования, затем деструктор, так как заканчивается время жизни

}

void print(Salary t) {

t.show();

}

