

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

**Пермский национальный исследовательский  
политехнический университет**

Факультет электротехнический

Кафедра ИТАС

**ОТЧЁТ**

**о лабораторной работе №2 (классы)**

Выполнил:  
Студент группы ИВТ-23-1Б  
Пискунов Д. А.

---

Проверил:  
Доцент кафедры ИТАС  
Яруллин Д.В.

---

---

**Пермь 2024**

## Вариант 15

Пользовательский класс ЗАРПЛАТА

ФИО – string

Оклад – double

Премия (% от оклада) – int

## Текст программы

### Файл Payday.h

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  class Payday {
7
8      string ln_fn_p;
9      double salary;
10     int reward;
11
12 public:
13     Payday(string, double, int);
14
15     Payday();
16
17     Payday(const Payday&);
18
19     ~Payday();
20
21     string get_fio();
22     double get_salary();
23     int get_reward();
24
25     void set_fio(string);
26     void set_salary(double);
27     void set_reward(int prem);
28
29     void show();
30 };
```

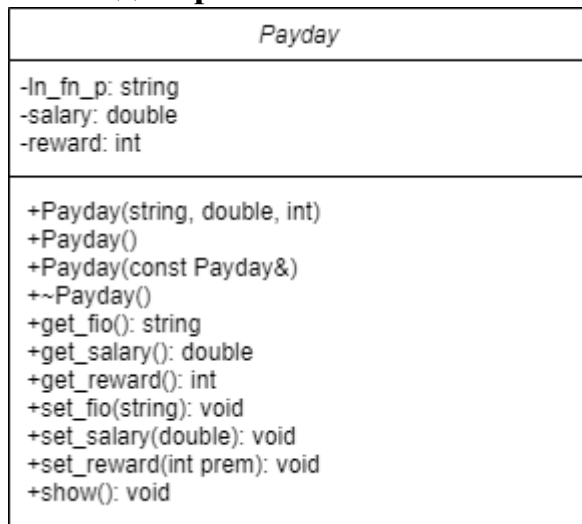
## Файл payday.cpp

```
1  #include<iostream>
2  #include"Payday.h"
3
4  Payday::Payday(string fio, double pay, int prem) {
5      ln_fn_p = fio;
6      salary = pay;
7      reward = prem;
8  }
9
10 Payday::Payday() {
11     ln_fn_p = "";
12     salary = 0;
13     reward = 0;
14 }
15
16 Payday::Payday(const Payday& p) {
17     ln_fn_p = p.ln_fn_p;
18     salary = p.salary;
19     reward = p.reward;
20 }
21
22 Payday::~Payday() {}
23
24 string Payday::get_fio() {
25     return ln_fn_p;
26 }
27 double Payday::get_salary() {
28     return salary;
29 }
30 int Payday::get_reward() {
31     return reward;
32 }
33
34 void Payday::set_fio(string fio) {
35     ln_fn_p = fio;
36 }
37 void Payday::set_salary(double zp) {
38     salary = zp;
39 }
40 void Payday::set_reward(int prem) {
41     reward = prem;
42 }
43
44 void Payday::show() {
45     cout << "\nФИО работника: " << ln_fn_p;
46     cout << "\nЗП работника: " << salary;
47     cout << "\nПремия работника: " << reward;
48 }
```

## Файл labs.cpp

```
1  #include "Payday.h"
2  #include <iostream>
3  #include <math.h>
4
5  using namespace std;
6
7  Payday make_payday() {
8      string fio;
9      double zp;
10     int premia;
11
12     cout << "Введите ФИО работника\n";
13     getline(cin, fio);
14
15     cout << "Введите зарплату работника\n";
16     cin >> zp;
17
18     cout << "Введите премию работника в процентах от зарплаты\n";
19     cin >> premia;
20
21     Payday p(fio, zp, premia);
22     return p;
23 }
24
25 void print_payday(Payday p) {
26     p.show();
27 }
28
29
30 int main() {
31     system("chcp 1251>null");
32
33     Payday p1;
34     p1.show();
35     cout << endl;
36
37     Payday p2("Зубенко Михаил Петрович", 800000, 30);
38     p2.show();
39     cout << endl;
40
41     Payday p3(p2);
42     p3.show();
43     cout << endl;
44     p3.set_fio("Бородач Александр Родионович");
45     p3.set_salary(90000);
46     p3.set_reward(0);
47     print_payday(p3);
48     cout << endl;
49     cout << endl;
50
51     p1 = make_payday();
52     p1.show();
53     cout << endl;
54
55
56
57     return 0;
58 }
```

## UML- диаграмма:



## Тест

```
ФИО работника:
ЗП работника: 0
Премия работника: 0

ФИО работника: Зубенко Михаил Петрович
ЗП работника: 800000
Премия работника: 30

ФИО работника: Зубенко Михаил Петрович
ЗП работника: 800000
Премия работника: 30

ФИО работника: Бородач Александр Родионович
ЗП работника: 9000
Премия работника: 0

Введите ФИО работника
Камышов Кирилл Кириллович
Введите зарплату работника
2
Введите премию работника в процентах от зарплаты
1

ФИО работника: Камышов Кирилл Кириллович
ЗП работника: 2
Премия работника: 1

C:\Users\MOkASiH\Desktop\labs\x64\Debug\labs.exe (процесс 17252) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...
```

## Ответы на вопросы:

1. Для чего нужен конструктор?

Ответ: чтобы избежать ошибки, связанные с использованием неинициализированных переменных.

2. Сколько типов конструкторов существует?

Ответ: 3: конструктор с параметрами, без параметров и конструктор копирования

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

Ответ: для освобождения памяти, выделенной под объект. Явно используется при выделении динамической памяти.

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

Ответ: конструктор копирования вызывается в тех случаях, когда новый объект создается путем копирования существующего

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

Конструктор с параметрами для инициализации объекта с заданными параметрами.

6. Перечислите свойства конструкторов

Ответ:

- Конструктор должен иметь то же имя, что и класс
- Класс может иметь несколько конструкторов с разными параметрами (конструктор можно перегружать)
- Конструктор не возвращает значение
- Если конструктор не был указан явно, то компилятор создаст пустой конструктор автоматически

7. Перечислите свойства деструкторов

Ответ:

- Должен иметь то же имя, что и класс, но со знаком тильда (~) перед именем
- Не принимает аргументы



- Не возвращает значение
- Вызывается автоматически, когда объект выходит из области видимости
- Если деструктор не был указан явно, то компилятор создаст пустой деструктор автоматически

8. К каким атрибутам имеют доступ методы класса?

Ответ: ко всем.

9. Что представляет собой запись: this

Ответ: это скрытый константный для адреса указатель, содержащий адрес объекта.

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

Ответ: для метода, определённого вне класса, нужно прописать пространство имен класса.

11. Какое значение возвращает конструктор?

Ответ: никакое.

12. Какие методы создаются по умолчанию?

Ответ: конструкторы, деструктор.

13. Какое значение возвращает деструктор?

Ответ: Никакое.

14. Дано описание класса

```
class Student
{
    string name;
    int group;
public:
    student(string, int);
    student(const student&)
    ~student();
};
```

Какой метод отсутствует в описании класса?

Ответ: конструктор без параметров

15. Какой метод будет вызван при выполнении следующих операторов:  
`student*s;`  
`s=new student;`

Ответ: конструктор без параметров

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

Ответ: конструктор с параметрами

17. Какие методы будут вызваны при выполнении следующих операторов:  
`student s1("Ivanov",20);`  
`student s2=s1;`

Ответ: конструктор с параметрами, копирования

18. Какие методы будут вызваны при выполнении следующих операторов:  
`student s1("Ivanov",20);`  
`student s2;`  
`s2=s1;`

Ответ: конструктор с параметрами, без параметров, копирования.

19. Какой конструктор будет использоваться при передаче параметра в функцию `print()`:  
`void print(student a)`  
`{a.show();}`

Ответ: 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 `p`? р?

Ответ: с помощью сеттера `set_name(string);`