Министерство высшего образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (ПНИПУ)

Электротехнический факультет

Дисциплина: основы алгоритмизации и программирования, 2 семестр

**ОТЧЁТ**

Тема: «Классы и объекты. Использование конструкторов»

Выполнил

Студент РИС-22-2б

Зубов Р.А.

Проверил

Доцент кафедры ИТАС

Полякова О. А.

Пермь 2023

# **Содержание**

Постановка задачи**4**

Коды программ**5**

Результаты программ**6**

**Постановка задачи**

Определить пользовательский класс. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования. Определить в классе деструктор. Определить в классе компоненты – функции для просмотра и установки полей данных(селекторы и модификаторы). Написать демонстрационную программу, в которой продемонстрировать все три случая вызова конструктора – копирования, вызов конструктора с параметрами и конструктора без параметров.

**Задача**

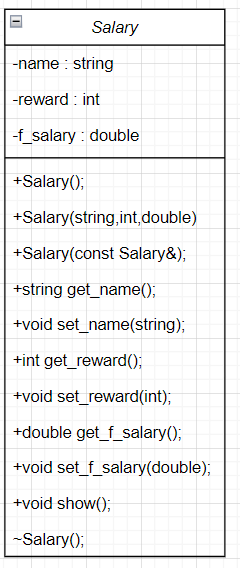
Пользовательский класс ЗАРПЛАТА

ФИО – string

Оклад – double

Премия(% от оклада) – int

**UML диаграмма**

****

**Описание класса**

Класс Salary содержит следующие поля:

* name - строка;
* reward - целое число;
* f\_salary - вещественное число.

Класс содержит следующие методы:

* конструктор по умолчанию;
* конструктор с параметрами;
* конструктор копирования;
* деструктор;
* get\_name() - возвращает имя объекта класса;
* set\_name(string) - устанавливает имя объекта класса;
* get\_reward() - возвращает значение поля reward объекта класса;
* set\_reward(int) - устанавливает значение поля reward объекта класса;
* get\_f\_salary() - возвращает значение поля f\_salary объекта класса;
* set\_f\_salary(double) - устанавливает значение поля f\_salary объекта класса;
* show() - выводит на экран значения всех полей объекта класса.

**Определение компонентных функций**

Определены следующие компонентные функции:

* Конструктор по умолчанию;
* Конструктор с параметрами;
* Конструктор копирования;
* Деструктор;
* get\_name() - возвращает имя объекта класса;
* set\_name(string) - устанавливает имя объекта класса;
* get\_reward() - возвращает значение поля reward объекта класса;
* set\_reward(int) - устанавливает значение поля reward объекта класса;
* get\_f\_salary() - возвращает значение поля f\_salary объекта класса;
* set\_f\_salary(double) - устанавливает значение поля f\_salary объекта класса;
* show() - выводит на экран значения всех полей объекта класса.

**Определение функций make\_ () и print\_()**

Функция make\_Salary() принимает значения полей объекта класса Salary с помощью функции cin и возвращает объект класса Salary с этими значениями. Функция print\_salary(Salary t) принимает объект класса Salary и выводит на экран значения его полей с помощью метода show().

**Объяснение результатов работы программы**

Результатом работы программы являются все три случая вызова конструктора с различными параметрами – копирования, вызов конструктора с параметрами и конструктора без параметров.

**Контрольные вопросы**

* 1. Конструктор в С++ - специальная функция, направленная на создание, подготовку к работе, определение параметров, экземпляров класса

class fraction

{

public:

double first;

int second;

int j;

void Init(double, int, int);

void Read();

};

* 1. Конструктор с параметрами, конструктор без параметров, конструктор копирования

class Salary

{

string name;

int reward;

double f\_salary;

public:

Salary(); // конструктор без параметров

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

Salary(const Salary&);//конструктор копирования

~Salary();

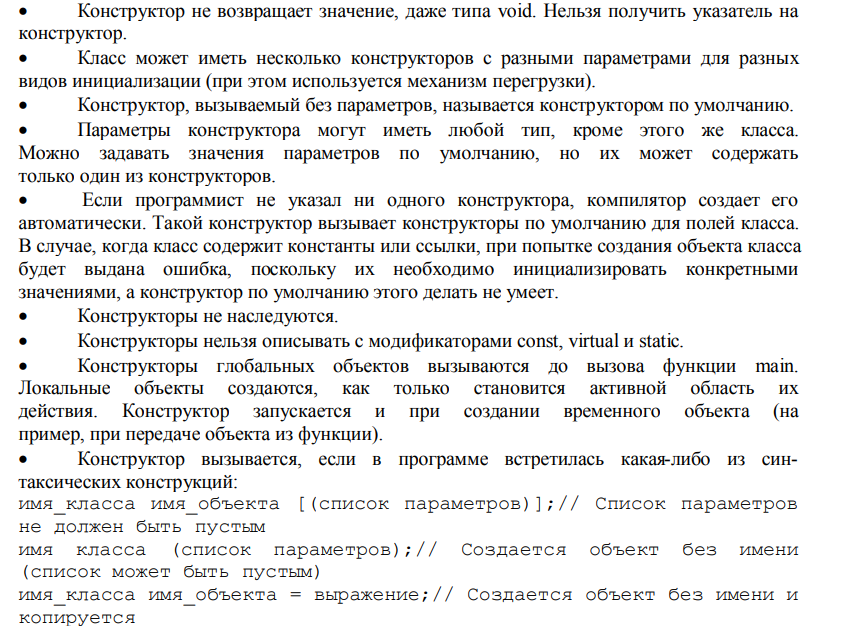
};

* 1. Деструктор применяется для освобождения ресурсов, выделенных конструктору объекту. Вызывается явно, когда объект имеет указатель на память, выделенную динамически.
  2. 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.

6. 

7. Свойства деструктора:

- не имеет аргументов возвращаемого значения;

- не наследуется;

- не может быть объявлен как const или static;

- может быть виртуальным;

8) К каким атрибутам имеют доступ методы класса

Ко всем элементам класса вне зависимости от спецификатора доступа

9) Что представляет собой запись: this

Указатель на текущий объект

10) Как разница между методами, определенными внутри класса и вне

класса

Нет разницы.

11) Какое значение возвращает конструктор?

Конструктор ничего не возвращает

12) Какие методы создаются по умолчанию?

Конструктор без параметров, копирования, деструктор

13) Какое значение возвращает деструктор?

Деструктор ничего не возвращает

14) Дано описание класса class Student

{

string name; int group; public:

student(string, int); student(const student&)

~student();

};

Какой метод отсутствует в описании класса?

Конструктор без параметров

15) Какой метод будет вызван при выполнении следующих операторов:

student\*s;

s=new student;

Конструктор без параметров

16) Какой метод будет вызван при выполнении следующих операторов:

student s(“Ivanov”,20);

Конструктор с параметрами

17) Какие методы будут вызваны при выполнении следующих

операторов: student s1(“Ivanov”,20);

student s2=s1;

Конкструктор с параметрами, копирования

18) Какие методы будут вызваны при выполнении следующих

операторов: student s1(“Ivanov”,20);

student s2; s2=s1;

Конструктор с параметрами и без, копирования

19)Какой конструктор будет использоваться при передаче параметра в

функцию print():

void print(student a)

{

a.show();

}

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 р?

С помощью сеттера p.set\_name(string YOUR TEXT);