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

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

**Лабораторная работа №1.**

**«Классы и объекты. Использование конструкторов.»**

Выполнил студент гр. РИС-24-2б

Молочко Артём Анатольевич

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

**Постановка задачи**

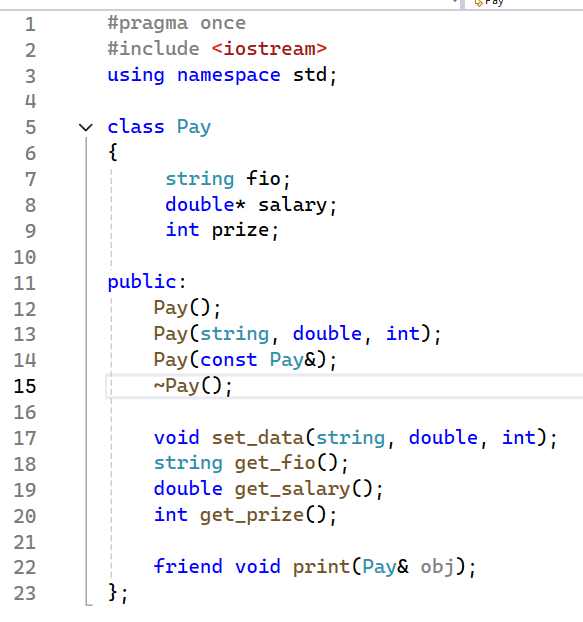
1. **Определение пользовательского класса:**
   * Создать пользовательский класс с именем Зарплата, который будет содержать следующие поля данных:
     + ФИО (тип string) — фамилия, имя и отчество сотрудника.
     + Оклад (тип double) — размер оклада сотрудника.
     + Премия (тип int) — процент премии от оклада.
2. **Реализация конструкторов:**
   * Реализовать в классе следующие конструкторы:
     + Конструктор без параметров (по умолчанию).
     + Конструктор с параметрами для инициализации всех полей класса.
     + Конструктор копирования.
3. **Реализация деструктора:**
   * Определить в классе деструктор для корректного освобождения ресурсов.
4. **Реализация методов доступа:**
   * Определить в классе компоненты-функции (селекторы и модификаторы) для просмотра и установки значений полей данных:
     + Методы для получения значений полей (ФИО, Оклад, Премия).
     + Методы для установки значений полей.
5. **Демонстрационная программа:**
   * Написать демонстрационную программу, в которой:
     + Продемонстрировать все три случая вызова конструктора:
       - Конструктор без параметров.
       - Конструктор с параметрами.
       - Конструктор копирования.
     + Показать работу селекторов и модификаторов для установки и получения значений полей.

**Анализ задачи**

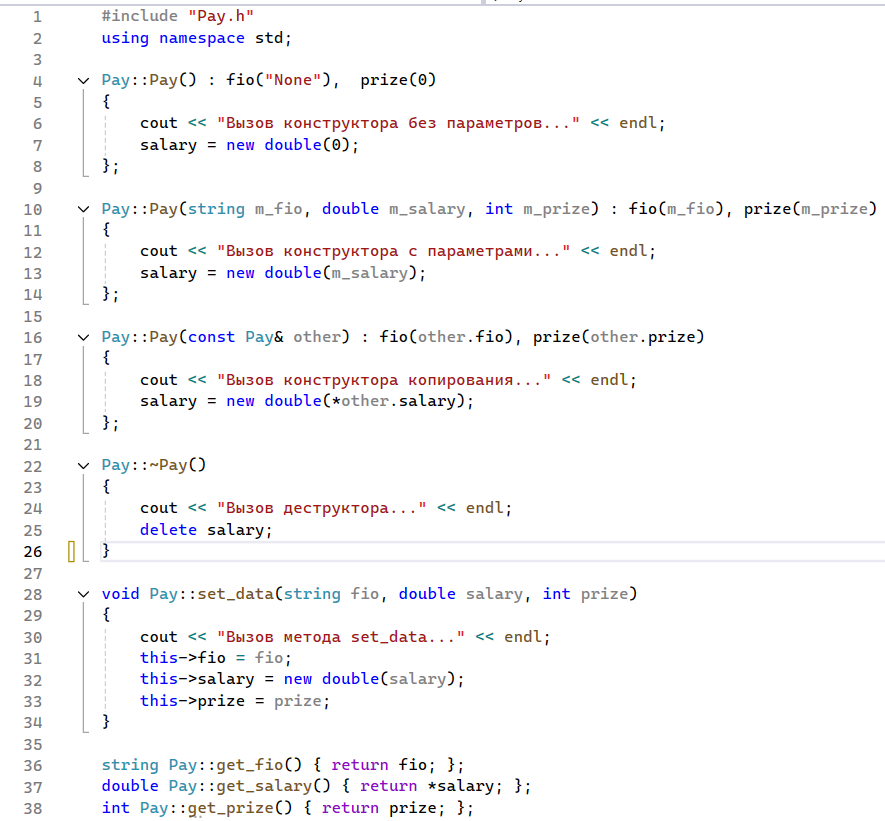
* Реализация класса Pay, который моделирует данные о зарплате сотрудника, включая ФИО, оклад и премию. Задача демонстрирует работу с конструкторами, деструктором, методами доступа и модификации данных.
* Метод set\_data позволяет изменять значения полей объекта после его создания.
* В main.cpp создаются объекты класса Pay с использованием всех трех конструкторов, демонстрируется их работа и вызов методов.
* Функция print выводит значения полей объекта, что позволяет наглядно увидеть результаты работы конструкторов и методов.

**Код программы**

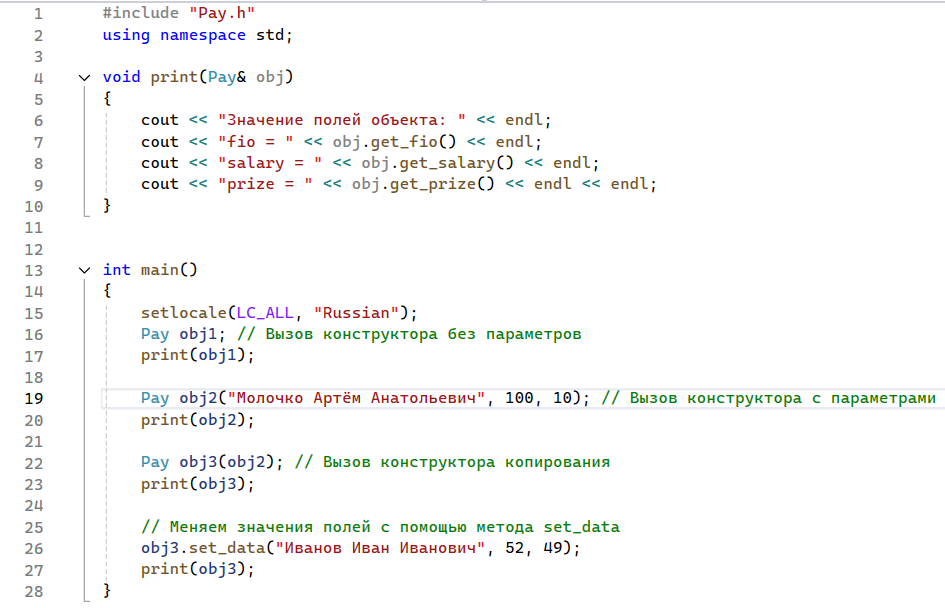
**Pay.h**

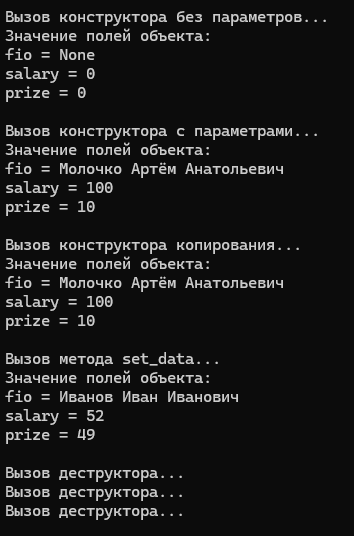


Pay.cpp



main.cpp





**Контрольные вопросы**

1. **Для чего нужен конструктор?**  
   Конструктор используется для инициализации объектов класса при их создании.
2. **Сколько типов конструкторов существует в C++?**  
   В C++ существует конструктор без параметров, конструктор с параметрами, конструктор копирования и делегирующий конструктор.
3. **Для чего используется деструктор? В каких случаях деструктор описывается явно?**  
   Деструктор используется для освобождения ресурсов, выделенных объектом. Он описывается явно, когда требуется освободить динамическую память или выполнить другие действия при уничтожении объекта.
4. **Для чего используется конструктор без параметров? Конструктор с параметрами? Конструктор копирования?**
   * Конструктор без параметров инициализирует объект значениями по умолчанию.
   * Конструктор с параметрами инициализирует объект переданными значениями.
   * Конструктор копирования создает новый объект как копию существующего.
5. **В каких случаях вызывается конструктор копирования?**  
   Конструктор копирования вызывается при создании нового объекта как копии существующего, при передаче объекта в функцию по значению и при возврате объекта из функции по значению.
6. **Перечислить свойства конструкторов.**
   * Имя конструктора совпадает с именем класса.
   * Не имеет возвращаемого типа.
   * Может быть перегружен.
   * Вызывается автоматически при создании объекта.
7. **Перечислить свойства деструкторов.**
   * Имя деструктора совпадает с именем класса с тильдой (~).
   * Не имеет параметров и возвращаемого типа.
   * Не может быть перегружен.
   * Вызывается автоматически при уничтожении объекта.
8. **К каким атрибутам имеют доступ методы класса?**  
   Методы класса имеют доступ ко всем атрибутам класса, включая private и protected.
9. **Что представляет собой указатель this?**  
   Указатель this передается в нестатические методы класса и указывает на текущий объект, для которого вызван метод.
10. **Какая разница между методами, определенными внутри класса и вне класса?**  
    Методы, определенные внутри класса, автоматически считаются встроенными, а методы, определенные вне класса, требуют отдельной реализации и не являются встроенными по умолчанию.
11. **Какое значение возвращает конструктор?**  
    Конструктор не возвращает значения.
12. **Какие методы создаются по умолчанию?**  
    По умолчанию создаются конструктор без параметров, конструктор копирования, оператор присваивания и деструктор.
13. **Какое значение возвращает деструктор?**  
    Деструктор не возвращает значения.
14. **Дано описание класса:**

**class Student**

**{**

**string name;**

**int group;**

**public:**

**student(string, int);**

**student(const student&);**

**~student();**

**};**

**Какой метод отсутствует в описании класса?**

В описании класса отсутствует конструктор без параметров.

1. **Какой метод будет вызван при выполнении следующих операторов:**

**student\* s;**

**s = new student;**

Будет вызван конструктор без параметров.

1. **Какой метод будет вызван при выполнении следующих операторов:**

**student s("Ivanov", 20);**

Будет вызван конструктор с параметрами.

1. **Какие методы будут вызваны при выполнении следующих операторов:**

**student s1("Ivanov", 20);**

**student s2 = s1;**

Будет вызван конструктор копирования.

1. **Какие методы будут вызваны при выполнении следующих операторов:**

**student s1("Ivanov", 20);**

**student s2;**

**s2 = s1;**

Будет вызван конструктор с параметрами и оператор присваивания.

1. **Какой конструктор будет использоваться при передаче параметра в функцию print():**

**void print(student a) {**

**a.show();**

**}**

Будет использован конструктор копирования.

1. **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?**

p.set\_name("Новое имя");