Пермский национальный исследовательский политехнический университет

Кафедра «Информационные технологии и автоматизированные системы»

Лабораторная работа 2

**Классы и объекты. Использование конструкторов.**

Выполнил

Студент гр. РИС-24-3б Рунтов Иван Андреевич

Проверил

Доцент кафедры ИТАС Ольга Андреевна Полякова

г. Пермь, 2025

**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**

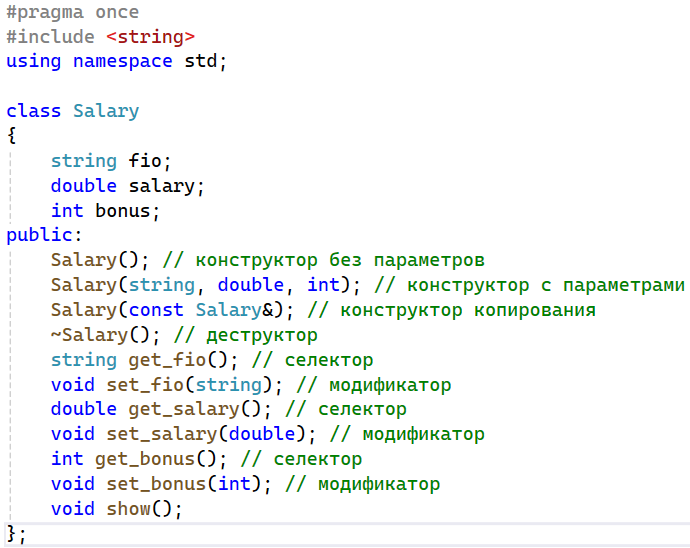
Пользовательский класс ЗАРПЛАТА

ФИО – string

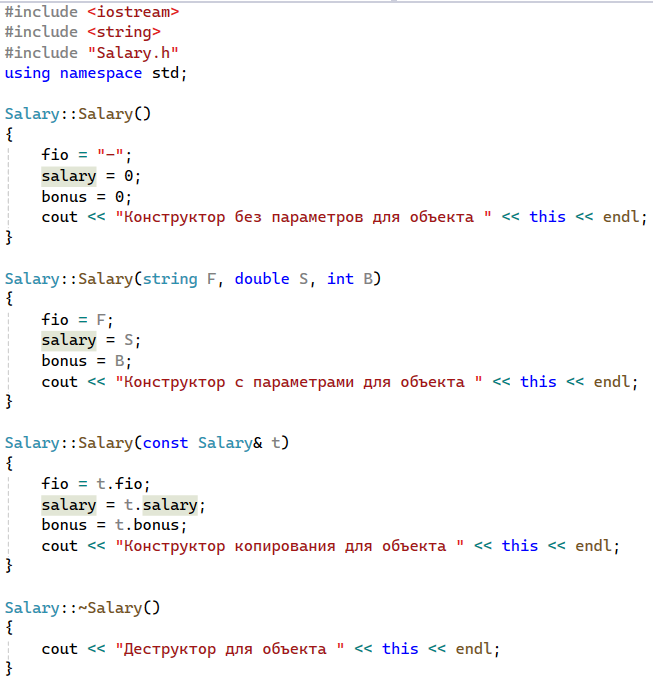
Оклад – double

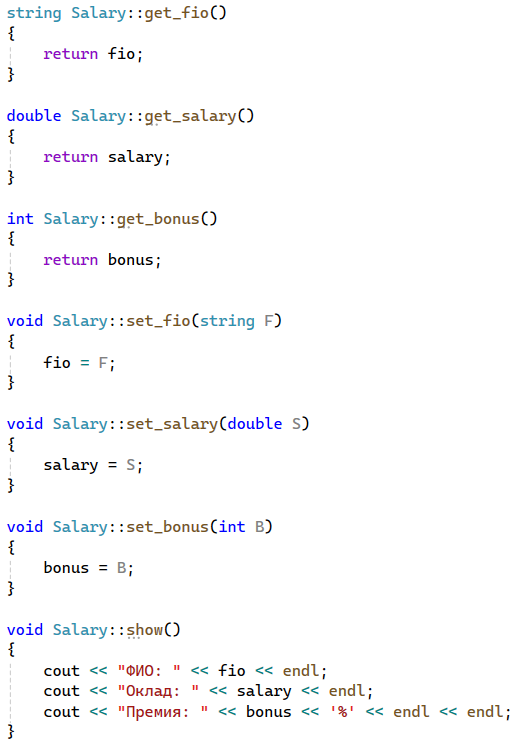
Премия (% от оклада) – int

**Описание класса Salary**

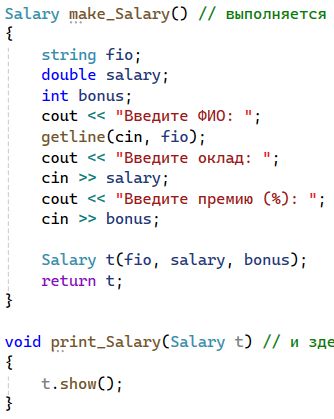


Определение методов класса Salary

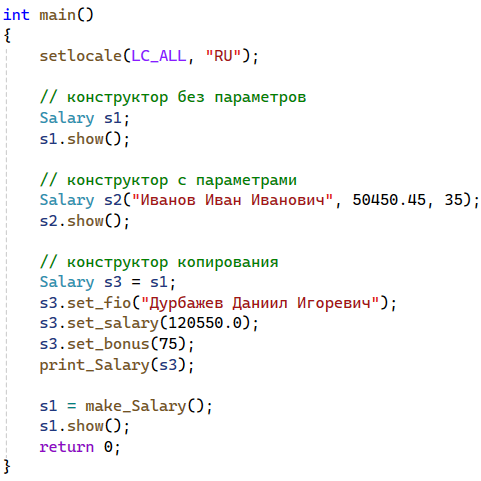




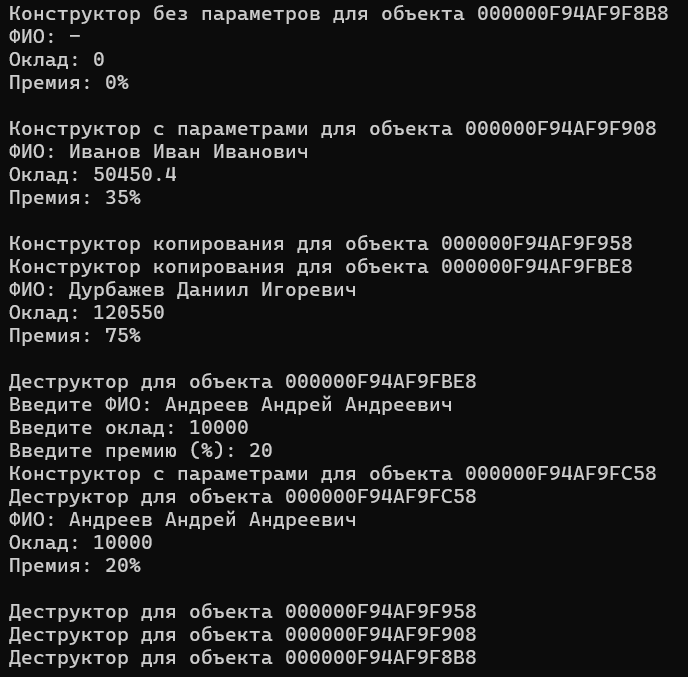
**Определение функций make\_Salary и print\_Salary**



**Функция main**



**Пример выполнения программы**



**Объяснение результатов работы программы**

Сначала создается объект s1, при этом выполняется конструктор без параметров, так как мы не указали их явно.

Далее создается объект s2, при этом выполняется конструктор с параметрами, так как мы явно указали их в скобках.

Далее создается объект s3, при этом выполняется конструктор копирования, в объект s3 копируются данные из объекта s1. Данные объекта s3 изменяются с помощью методов-модификаторов класса. При входе в функцию print\_Salary создается временный объект класса Salary, при этом выполняется конструктор копирования данных из объекта, переданного в функцию (s3). К временному объекту применяется метод вывода данных, и он удаляется.

Далее в функции make\_Salary создается временный объект класса Salary, при этом выполняется конструктор с параметрами. Этот временный объект возвращается из функции, и данные из него записываются в объект s1, после чего временный объект удаляется.

**Ответы на контрольные вопросы**



Конструкторы необходимы для инициализации объектов данными при их создании.

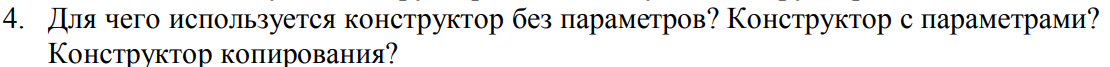


В C++ существует 3 типа конструкторов:

* конструктор с параметрами, используется для инициализации объекта конкретными значениями
* конструктор без параметров, используется для создания пустого объекта
* конструктор копирования, используется для создания объекта, аналогичного тому, который уже существует



Деструктор используется для очищения памяти, выделенной под объект. Явно описывать деструктор следует только тогда, когда память под данные класса выделяется динамически.



Конструктор с параметрами используется для инициализации объекта конкретными значениями. Конструктор без параметров используется для создания пустого объекта. Конструктор копирования используется для создания объекта, аналогичного другому объекту.



Этот конструктор 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 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).



* 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:

При выходе за пределы блока, в котором объявлен экземпляр класса

При использовании операторов **delete** или **delete[].**



Методы класса имеют доступ ко всем атрибутам класса.



Указатель this является указателем на объект, для которого был вызван метод.



Внутри класса метод описывается как обычная функция. При описании метода класса вне его указывается область видимости, которая определяет принадлежность метода к конкретному классу.



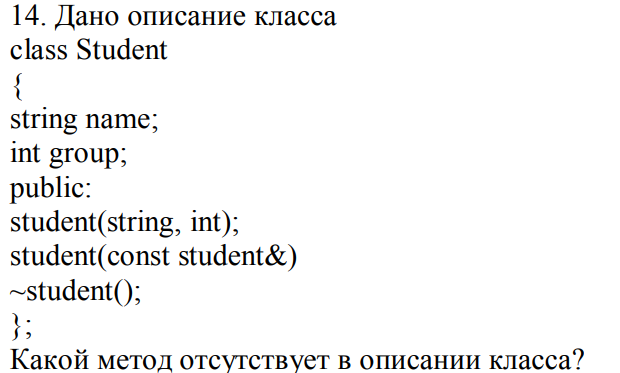
Конструктор не возвращает никакого значения.



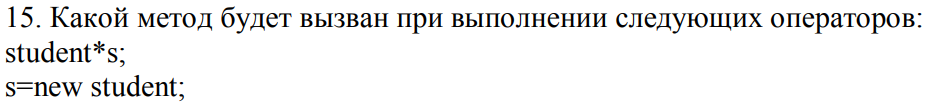
По умолчанию в классе создаются конструкторы без параметров и копирования, а также деструктор.



Деструктор не возвращает никакого значения.



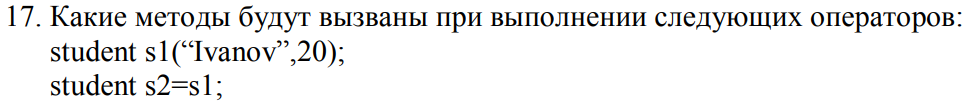
В описании класса отсутствует конструктор без параметров.



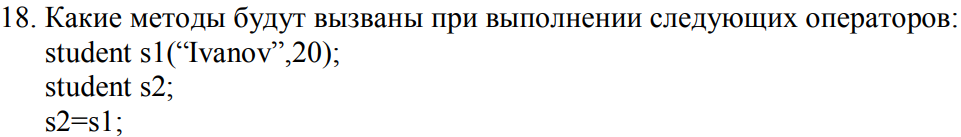
Применяется конструктор без параметров.



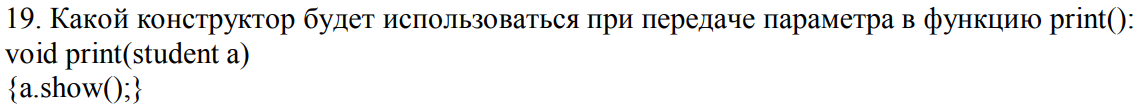
Выполняется конструктор с параметрами.



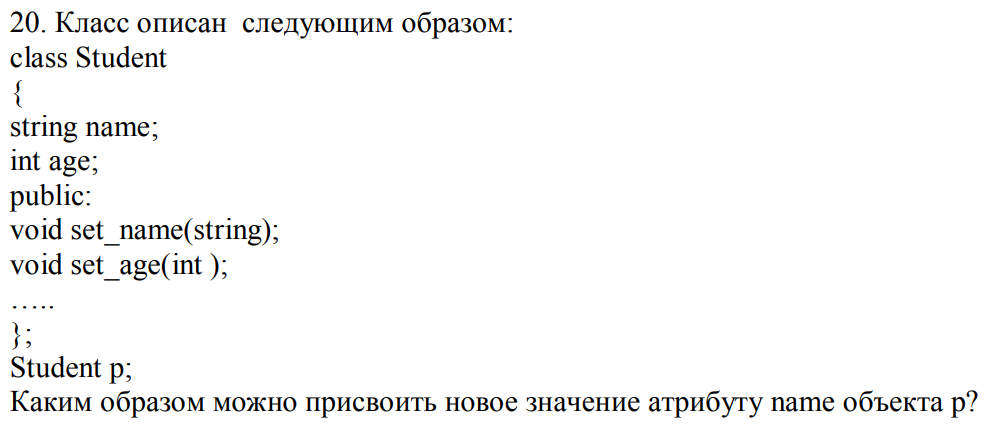
Для объекта s1 выполняется конструктор с параметрами. Для объекта s2 выполняется конструктор копирования.



Для объекта s1 выполняется конструктор с параметрами. Для объекта s2 выполняется конструктор без параметров.



При передаче объекта в функцию будет выполняться конструктор копирования.



Присвоить новое значение атрибуту name объекта p можно с помощью метода-модификатора set\_name, передав в него нужное значение.

Ссылка на github: <https://github.com/RuntovIvan/Informatika>