Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

Тема: “Классы и объекты. Использование конструкторов"

Семестр 2

Выполнил работу

Студент группы РИС-22-1Б

Мальчиков Дмитрий Григорьевич

Проверил

Доцент кафедры ИТАС

Полякова Ольга Андреевна

Г. Пермь-2023

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

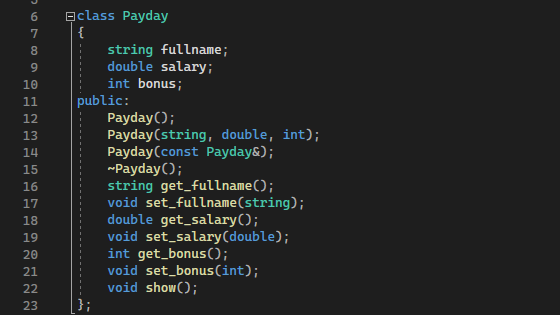
ФИО - string

Оклад - double

Премия (% от оклада) - int

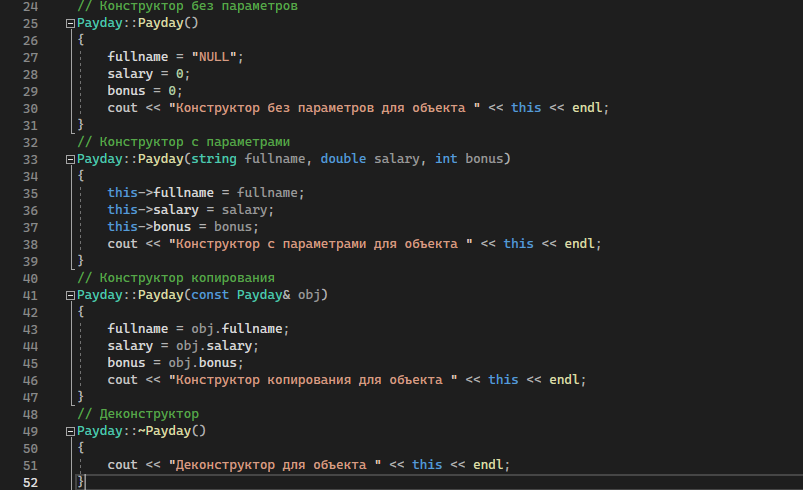
Описание класса

Класс Payday - Зарплата сотрудника, поля класса: string fullname - ФИО сотрудника, double salary - оклад, int bonus - премия, бонус. Также в классе описаны методы:  
 Payday() - конструктор без параметров, Payday(string, double, int) - конструктор с параметрами, Payday(const Payday&) - конструктор копирования, ~Payday() - деструктор.  
 get\_fullname(), get\_salary(), get\_bonus() - селекторы.  
 set\_fullname(string), set\_salary(double), set\_bonus(int) - модификаторы.

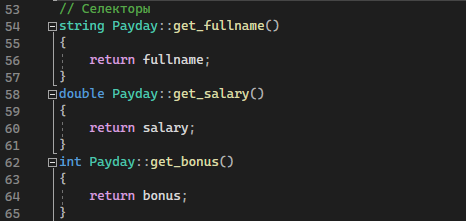


Определение компонентных функций

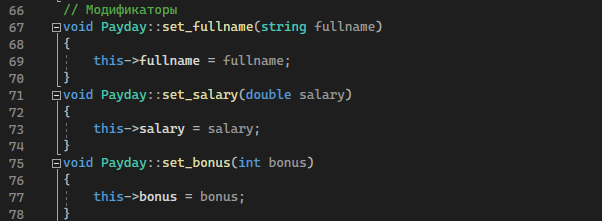
3 вида конструкторов + деструктор



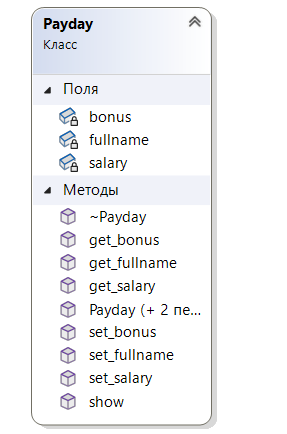
Селекторы

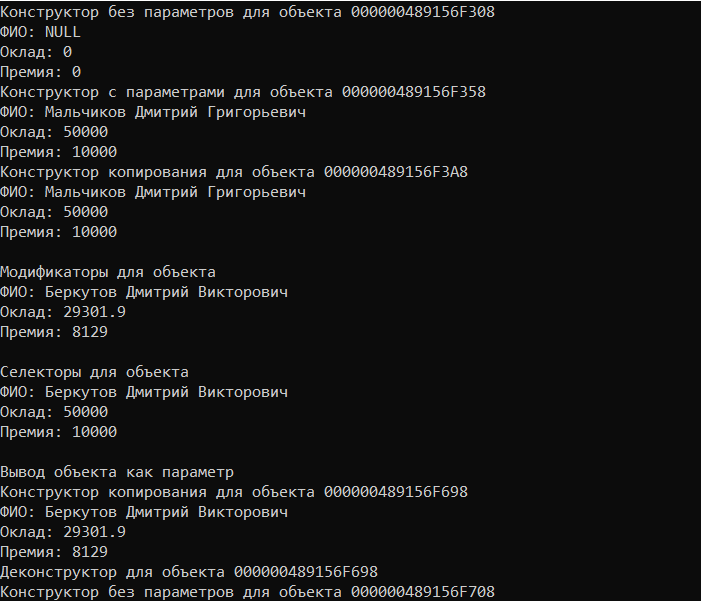
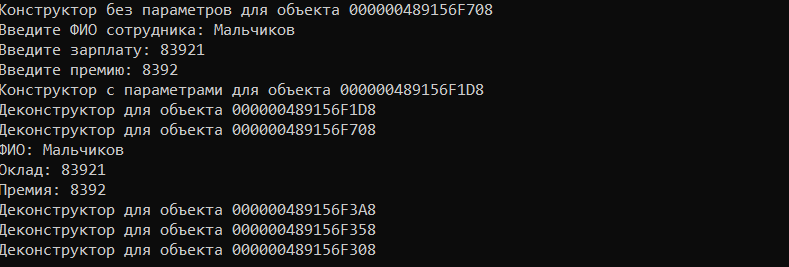


Модификаторы



UML-диаграмма классов



Результаты работы программы  


Ответы на вопросы:

1. Конструктор необходим для инициализации полей класса.

2. 3 - Без параметров, с параметрами, конструктор копирования.

3. Для завершения работы с конструктором класса после выхода из области видимости ф-ии. Если конструктор описан явно, то необходимо и деструктор описать явно.

4. Без параметров – для инициализации полей по умолчанию(заранее заданными значениями). С параметрами – для инициализации полей переданными в конструктор значениями. Копирования – для инициализации полей с помощью копирования их из другого обьекта.

5. Конструктор копирования вызывается в случае, когда нужно заполнить n обьектов одинаковыми значениями полей.

6. Инициализация полей переданными параметрами, инициализация полей по умолчанию, инициализация полей с помощью копирования, перегрузка конструкторов, конструктор списка по умолчанию, конструктор списка инициализаторов.

7. Не могут иметь аргументов; Не возвращают значение (или void ); Не может быть объявлен как const , volatile или static . Однако они могут вызываться для уничтожения объектов, объявленных как const , volatile или static ; Может быть объявлен как virtual . Используя виртуальные деструкторы, можно уничтожать объекты, не зная их тип — правильный деструктор для объекта вызывается с помощью механизма виртуальных функций. Обратите внимание, что для абстрактных классов деструкторы также могут объявляться как чисто виртуальные функции; не описаны явно – автоматически вызовутся.

8. Private, Public, Protected;

9. Адрес обьекта класса;

10. Внутри класса можно не передать параметры, вне – обязательно. Использование шаблонов ф-ий порой обязательно.

11. Конструкторы не возвращают значений.

12. Конструктор и деструктор.

13. Аналогично, деструктор не возвращает значений.

14. Сеттер.

15. Никакой.

16. Конструктор с параметрами.

17. Конструктор с параметрами для первого обьекта и конструктор копирования для второго.

18. Аналогично предыдущему, только сначала второй обьект вызовет конструктор без параметров.

19. С параметрами.

20. p.set\_name(“name”);