**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА**

**И ГОСУДАРСТВЕННОЙ СЛУЖБЫ**

**ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**НИЖЕГОРОДСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ – филиал РАНХиГС**

**Отчет по программированию №4**

**Выполнил: студент группы Иб-321**

**Комышков Владислав Дмитриевич**

[Задание 1 2](#_Toc1891044204)

[Ход 3](#_Toc895467345)

[Итог 8](#_Toc544600226)

[Задание 2 8](#_Toc1769606553)

[Ход 9](#_Toc1492607090)

[Итог 9](#_Toc2016661705)

[Задание 3 9](#_Toc1925250976)

[Ход 10](#_Toc392745380)

[Итог 11](#_Toc223436546)

[Задание 4 11](#_Toc275261936)

[Ход 12](#_Toc1492657605)

[Итог 13](#_Toc237334857)

[Задание 5 13](#_Toc1552578744)

[Ход 14](#_Toc2122732467)

[Итог 15](#_Toc1421787388)

[Задание 6 15](#_Toc1607194800)

[Ход 16](#_Toc1405542868)

[Итог 16](#_Toc1584950945)

# Задание 1

Текст задания:

* 1. Самостоятельно создать дочерние классы для пикапов и джипов, имеющих собственные специфические характеристики. Эти характеристики можно взять из интернета.
  2. Изменить базовый класс таким образом, чтобы была возможность дополнительно фиксировать и менять одно из двух возможных состояний автомобиля: «выставлен на продажу» или «продан»
  3. Продемонстрировать работу программы на примере работы с двумя экземплярами автомобиля каждого из трёх классов.
  4. Предложить вариант выделения печати в отдельную функцию.

## Ход

Классы “Pickup” и “Jeep” представлены на Рис. 1.3.

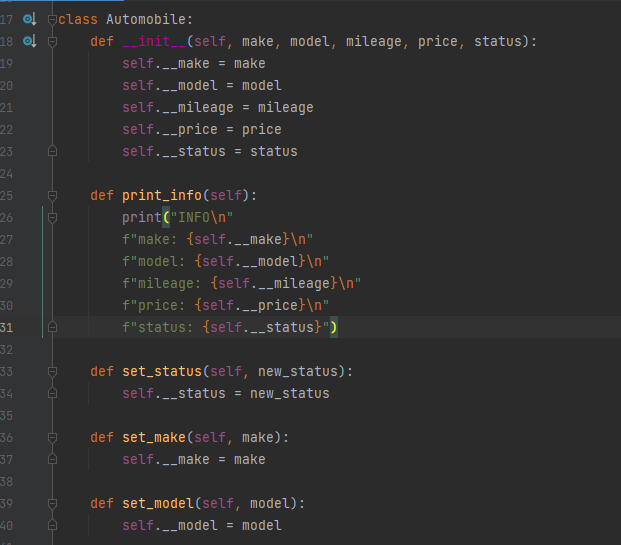


Рис. 1.1 - класс Automobile

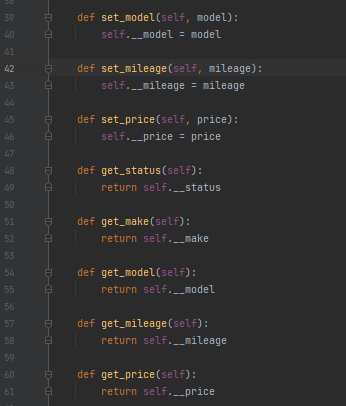


Рис 1.2 - методы класса Automobile

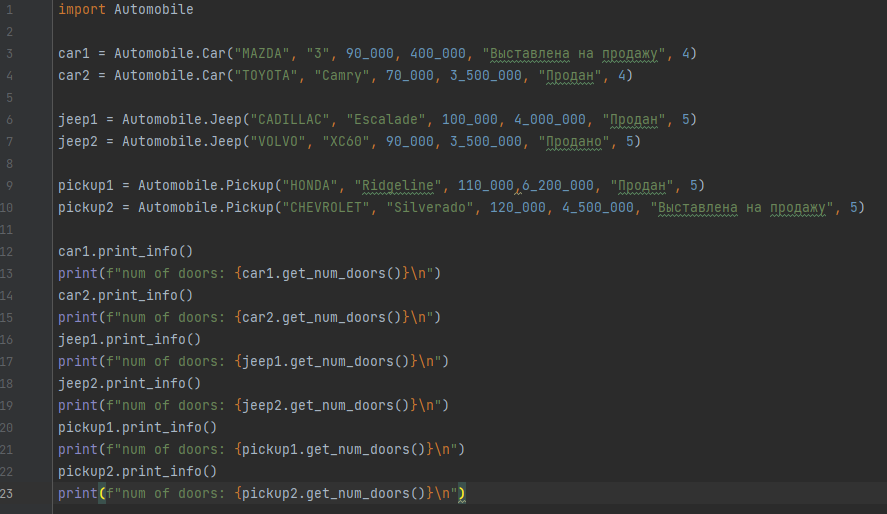


Рис 1.3 - классы Car, Jeep и Pickup

Состояние автомобиля и возможность его изменять (Рис. 1.4)

Рис 1.4 - аргумент status и функция для вывода информации об автомобилях

Пример работы классов Automobile, Car, Jeep и Pickup (Рис. 1.5-1.7).

Рис 1.5 - скрипт для работы с классами

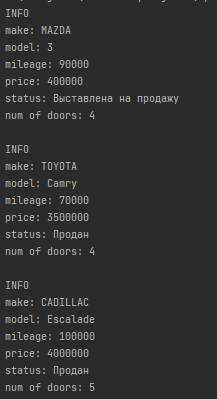


Рис. 1.6 - вывод работы скрипта

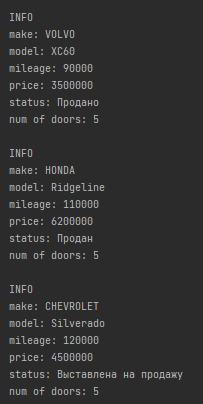


Рис. 1.7 - продолжение вывода работы скрипта

Функция для вывода информации на печать (Рис. 1.4).

Итог

Самостоятельно созданы классы “Pickup” и “Jeep”. Добавлен атрибут фиксирующий состояние автомобиля. Продемонстрирован пример работы классов. Вывод информации об объектах реализован с помощью функции.

# Задание 2

Текст задания:

2.1 Написать скрипт, доказывающий справедливость представленного выше правила работы при совпадении имён методов базового и производного класса.

## Ход

Работа правила о базовых и производных классах представлена на Рис. 2.1.

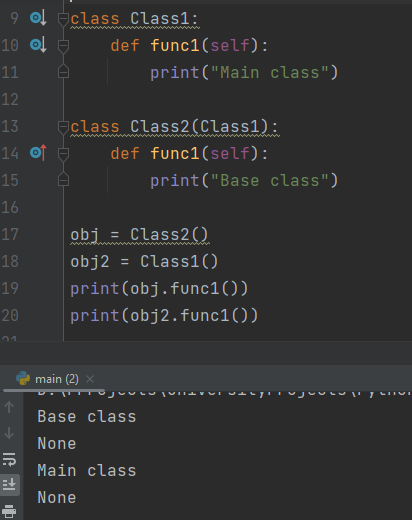


Рис 2.1 - работа правила о базовых и производных классах

Итог

Представлен пример работы правила базовых и производных классов.

# Задание 3

Текст задания:

3.1 Проанализировать работу скрипта, представленного на рисунке 4.6 с закомментированной строкой и раскомментированной строкой 48.

3.2 Проверить, можно ли включить вызов конструктора базового класса из метода экземпляра класса-наследника.

3.3 Определить, как решается вопрос о работе конструкторов в скрипте, представленном на рисунке 4.3.

## Ход

Работа скрипта с закомментированной строкой (Рис. 3.1).

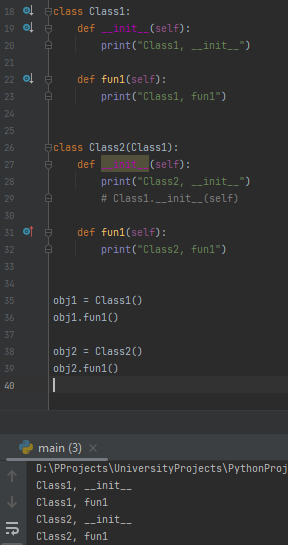


Рис 3.1 - закомментированная строка

Работа скрипта с раскомментированной строкой (Рис. 3.2).

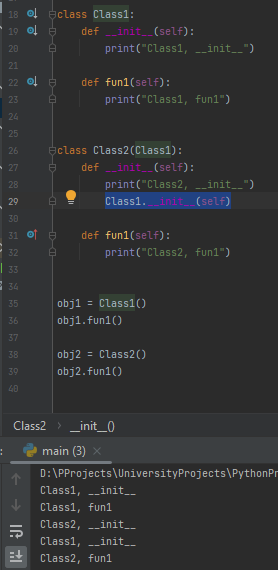


Рис 3.2 - раскомментированная строка

Из вышепредставленных примеров видно что можно включить вызов конструктора класса из метода экземпляра класса наследника.

Итог

Проанализированна работа скрипта из лекции на рис. 4.6. Выявлено, что можно включить вызов конструктора класса из метода экземпляра класса наследника.

# Задание 4

Текст задания:

Проверить варианты наследования конструктора базового класса дочерним на примере рисунков 4.7-4.9 добавив для класса грузовиков ещё одно свойство «цена\_перевозки».

## Ход

Пример работы класса “Truck” (грузовик) с приминением метода “transport\_price” (цена перевозки) представлен на Рис. 4.1-4.2.

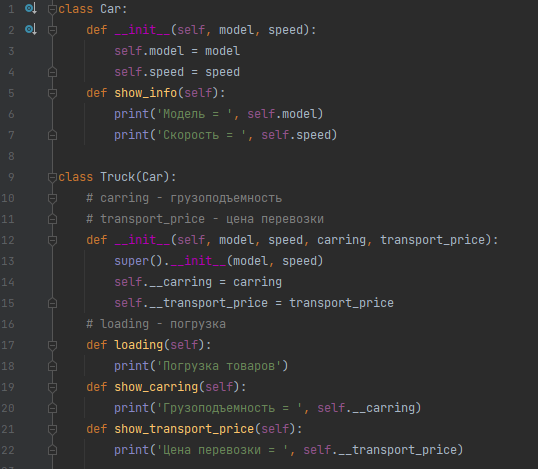
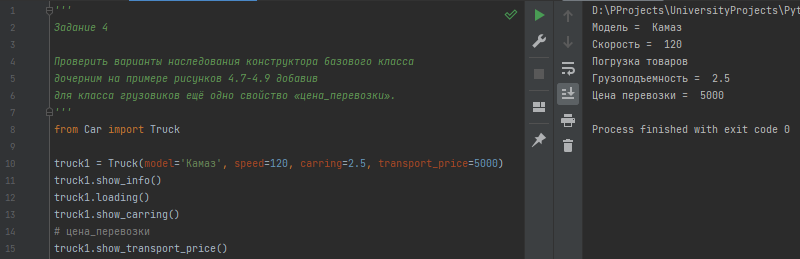


Рис 4.1 - класы Car и Truck

Рис 4.2 - Пример работы класса “Truck”

Итог

Проверен вариант наследования конструкторов базового класса дочерним на примере класса “Truck”.

# Задание 5

Текст задания:

5.1. Создать класс, наследующий все 4 класса из рисунка 4.10. Проверить замену методов, аналогичную указанному выше.

5.2 Проверить на примере, как работает организовано и реализуется наследование конструкторов в трёх поколениях: дедушка → родитель → сын.

## Ход

Пример наследования 4 классов (Рис. 5.1).

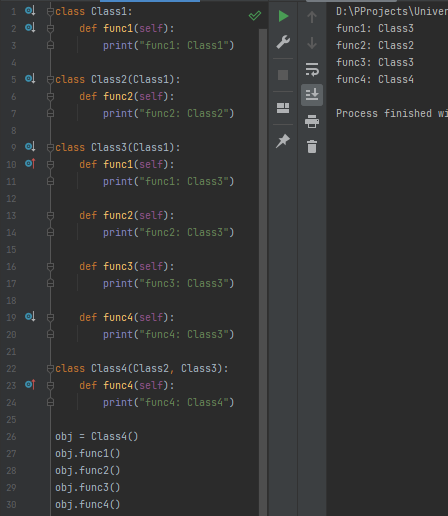


Рис 5.1 - Пример наследования 4 классов

Пример работы наследования классов “Grandfather->Parent->Son” (Дедушка->Родитель->Сын) представлен на Рис.5.2.

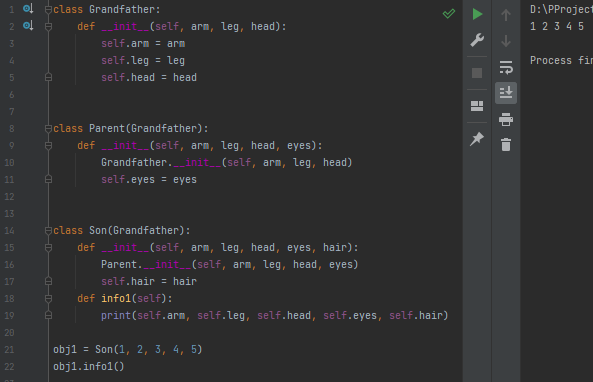


Рис 5.2 - Пример работы наследования классов “Grandfather->Parent->Son”

Итог

Создан класс наследующий 4 класса и представлен пример работы. На примере классов “Grandfather->Parent->Son” проверена работа наследования.

# Задание 6

Текст задания:

В удобном для Вас редакторе создать диаграммы классов по заданию 1. Созданные диаграммы отобразить в документе Word или Writer.

## Ход

Диаграмма классов из задания 1 (Рис. 6.1). Диаграммы выполнены в программе StarUML.

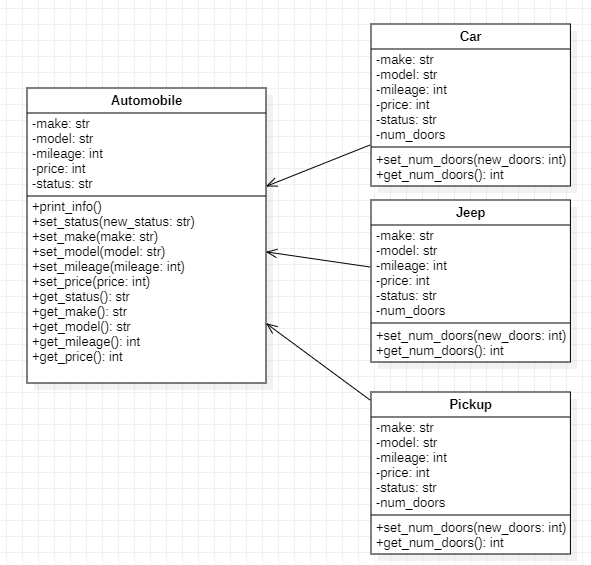


Рис 6.1 - Диаграмма классов из задания 1

Итог

Реализованны диаграммы классов из задания 1.