

#2 Project, package, Class, method

- Проект
- Пакет
- Класс
- Метод

Проект и пакет

- Пакет в Java — это механизм для инкапсуляции группы классов, подпакетов и интерфейсов.

Пакеты используются для:

- Предотвращение конфликтов имен. Например, может быть два класса с именем cars в двух пакетах: `com.auto.berlin.cars` и `com.auto.finland.cars`.
- Упрощение поиска/нахождения и использования классов
- Пакет — это контейнер группы связанных классов, где некоторые из доступных классов выставлены, а другие хранятся для внутренних целей.
- Мы можем повторно использовать существующие классы из пакетов столько раз, сколько нам нужно в нашей программе.
- Соглашения об именах пакетов: Пакеты именуются в обратном порядке доменных имен, например, `org.telran.berlin.lesson.practice`.

Class, Object and method in Java

- Класс в Java - это шаблон для создания объекта, а объект - это экземпляр класса.
- Класс содержит переменные и методы, которые называются элементами класса.
- Методы используются для описания того, что объект класса умеет делать или что можно с ним сделать.
- Переменные - для описания свойств или характеристик объекта.

Переменные в Java

- Переменная в Java — это контейнер данных, который сохраняет значения данных во время выполнения программы Java.
- Каждой переменной назначается **тип данных**, определяющий тип и количество значений, которые она может содержать.
- Переменная — это имя, присвоенное ячейке памяти.

Типы переменных в Java

- Локальные переменные
- Переменные экземпляра
- Статические переменные

Scope of variables

Локальные переменные
Переменные экземпляра
Статические переменные

Локальные переменные

1. Переменная, определенная в блоке, методе или конструкторе, называется локальной переменной.
2. Эти переменные создаются при входе в блок или при вызове функции и уничтожаются после выхода из блока или при возврате вызова из функции.
3. Область действия этих переменных существует только внутри блока, в котором переменные объявлены, т. е. мы можем получить доступ к этим переменным только внутри этого блока.
4. Инициализация локальной переменной обязательна перед ее использованием в определенной области

Scope of variables

Переменные экземпляра

1. Переменные экземпляра являются нестатическими переменными и объявляются в классе вне какого-либо метода, конструктора или блока.
2. Поскольку переменные экземпляра объявляются в классе, эти переменные создаются при создании объекта класса и уничтожаются при уничтожении объекта.
3. В отличие от локальных переменных, мы можем использовать модификаторы доступа для переменных экземпляра. Если мы не укажем модификатор доступа, будет использоваться модификатор доступа по умолчанию.
4. Инициализация переменной экземпляра не является обязательной. Его значение по умолчанию равно 0.
5. Доступ к переменным экземпляра можно получить только путем создания объектов.

Scope of variables

Статические переменные

1. Статические переменные также известны как переменные класса.
2. Эти переменные объявляются аналогично переменным экземпляра. Разница в том, что статические переменные объявляются с использованием ключевого слова `static` внутри класса вне какого-либо метода, конструктора или блока.
3. В отличие от переменных экземпляра, у нас может быть только одна копия статической переменной для каждого класса, независимо от того, сколько объектов мы создаем.
4. Статические переменные создаются в начале выполнения программы и автоматически уничтожаются по окончании выполнения.
5. Инициализация статической переменной не является обязательной. Его значение по умолчанию равно 0.
6. Доступна без создания экземпляра

Static variables VS Instance variables

Static variables	Instance variables
Одна копия	Собственная копия
Изменения отражаются	Изменения не отражаются
Доступ на прямую	Доступ через ссылку

Модификаторы доступа

Modifier	Class	Package	Subclass	Global
Public	Yes	Yes	Yes	Yes
Protected	Yes	Yes	Yes	No
Default	Yes	Yes	No	No
Private	Yes	No	No	No

Соглашения об именах в Java

Class: если вы называете какой-либо класс, то это должно быть **существительное**, и поэтому оно должно быть названо в соответствии с целью, которая должна быть достигнута в программе.

```
class Student { } class Scanner { }
```

Interface: если вы называете интерфейс, оно должно выглядеть как **прилагательное**.

```
interface Runnable() interface Flying()
```

Method: предполагается, что метод должен делать что-то, что он содержит в своем теле, это должен быть **глагол**.

Constant: как следует из названия, это должно выглядеть так, как мы читаем, как будто оно **фиксировано** PI, MAX_INT, MIN_INT.

Типы данных

- Java - типизированный язык, в котором каждая переменная уже известна во время компиляции.
- Как только переменная объявлена определенным типом данных, она не может содержать значения других типов данных.
- Прimitивный тип данных:
например -> boolean, char, int, short, byte, long, float и double
- Непрimitивный тип данных или объектный тип данных:
например -> строка, массив, Объект

Примитивные типы данных

- boolean - Логический тип данных представляет только (true, false) - один бит информации: либо истина, либо ложь
- byte - представляет собой число (-128 to 127) - 8 bits.
- short - представляет собой число (-32 768 до 32 767) – 16 bits
- int - представляет собой число (-2147483648 до 2147483647) – 32 bits.
- long – большое целое число (-9223372036854775808 до 9223372036854775807) – 64 bits.
- float - число с плавающей запятой (до 7 десятичных цифр) – 32 bits.
- double - число с плавающей запятой (до 16 десятичных цифр) – 64 bits.
- char - символ Unicode (от '\u0000' (0) до '\uffff' (65535)) – 16 bits.