

Классы и Объекты

№ урока: 2 Курс: Java Essential

Средства обучения: Компьютер с установленной IntelliJ IDEA.

Обзор, цель и назначение урока

Рассмотрение понятия «Конструктор», keyword «this». Перегрузка конструкторов. Конструктор по умолчанию. Приватный конструктор.
Keyword «static».

Изучив материал данного занятия, учащийся сможет:

- Оперировать знаниями по конструкторам.
- Создавать конструкторы по умолчанию и «default».
- Знание и понимание ключевых слов «this», «static».

Содержание урока

1. Рассмотреть понятие конструктора.
2. Объявление конструктора.
3. Ключевое слово **this**.
4. Рассмотреть значение и применение конструктора по умолчанию.
5. Рассмотреть, когда конструктор по умолчанию не создается.
6. Рассмотреть создание и использование приватного конструктора.
7. Перегрузка конструкторов.
8. Создание конструкторов из конструкторов.
9. Статические элементы

Резюме

Конструктор – объявление конструктора выглядит, как объявление метода, только для этого, мы используем имя Класа. По сути, когда после ключевого слова **new** мы пишем имя класса со скобками, то на самом деле мы вызываем конструктор класса с теми параметрами, которые укажем в скобках (передадим на вход конструктора).

Есть два вида конструкторов «с параметрами» и «по умолчанию». Конструкторы не могут быть synchronized, final, abstract, native, static.

this – это ссылка на нынешний объект (объект, чей конструктор был вызван), то есть на экземпляр класса, в котором разворачивается действие.

Конструктор:

- 1) Конструктор создается из полей, которые объявлены в классе.
- 2) Если у класса явно не создан конструктор, то компилятор создаст конструктор по умолчанию сам.
- 3) Если мы создали конструктор с параметрами, то конструктор по умолчанию не создается. В случае если он нам нужен будет для работы, то его надо будет создавать руками.
- 4) Так же Конструктор может иметь модификатор доступа **private**, это значит, что объекты с использованием параметров данного конструктора могут создаваться только в рамках Класа, где конструктор объявлен.

Перегрузка конструкторов. Перегрузка – это когда сигнатура конструкторов отличается друг от друга.

Так же возможно создавать конструкторы из конструкторов при помощи к.с. **this**.

При выполнении конструктора **this()** сначала выполняется перегруженный конструктор, который соответствует списку параметров, затем выполняются операторы, находящиеся внутри исходящего конструктора, если таковые существуют. Вызов конструктора **this()** должен быть первым оператором в конструкторе.

static – обозначает связь с Классом, а не с его объектом.

Слово **static** применимо с:

- полями;
- методами;
- классами;
- блоками инициализации;
- импортами;

Если нужно, что бы переменные являлись общими для всех Объектов, то применяется **static**. Поля, которые имеют статический модификатор в объявлении – называются статическими полями или переменными класса. Каждый экземпляр класса разделяет переменную класса, которая находится в одном определенном месте в памяти.

static модификатор, в сочетании с **final** модификатора, используется также для определения констант.

Статические методы имеют преимущество в применении, т.к. отсутствует необходимость каждый раз создавать новый объект для доступа к таким методам. Статический метод можно вызвать, используя тип класса, в котором эти методы описаны. Именно поэтому, подобные методы как нельзя лучше подходят в качестве методов-фабрик (factory), и методов-утилит (utility). Класс `java.lang.Math` — замечательный пример, в котором почти все методы статичны, по этой же причине классы-утилиты в Java финализированы (final).

final модификатор указывает, что значение этого поля не может измениться.

Java Naming Convention гласит, что константы должны писаться в верхнем регистре, и отделять слова надо нижним подчеркиванием «_».

Статические методы, должны быть вызваны именем класса, без необходимости создания экземпляра класса:

ClassName.methodName(args);

Вы так же можете обратиться к статическим методам с объектной ссылкой, но это не рекомендуется, потому что становится не ясно, что методы являются Методами класса.

Обычно статические методы используются, чтобы обратиться к статическим полям. Так же **static** не может использоваться с **this** (потому что this это ссылка на нынешний объект).

Итог:

1. static для классов значит, что класс вспомогательным и будет оставаться таким как есть (так званые статические Вложенные классы. За исключением классов верхнего уровня)
2. Статические методы не имеют зависимости от данных экземпляра.
3. Некоторые данные, которые необходимо использовать несколькими экземплярами обозначаются статическими. Вместо создания их каждый раз, мы создаем общий для всех.
4. Статические методы не могут быть **overridden**.

Закрепление материала

- Что такое Конструктор?
- Что такое Конструктор по умолчанию, Приватный?
- Что значит **this**?
- Что такое перегрузка конструкторов?
- Как создавать конструктор из конструктора.
- Как применяется статика с полями?
- Как используется статика в методах?
- Перечислить где может статика использоваться.

Дополнительное задание

Задание

Используя IntelliJ IDEA создать проект, пакет.

Создать класс `MyArea`, в нем объявить константу $\pi = 3.14$ и статический метод `areaOfCircle`, который должен принимать радиус и используя π посчитать площадь круга.

Создать класс `Main`, где запустить данный метод.

Самостоятельная деятельность учащегося

Задание 1

В любой из профильных **книг (Хорстман, Эккель)** найти соответствующие темы и закрепить материал. Использование **YouTube, Quizful** приветствуется.

Задание 2

Используя IntelliJ IDEA создать проект, пакет.

Создать класс Машина с полями год(int), цвет(String).

Создать конструктор по умолчанию, конструктор с одним и 2-я параметрами.

Создать класс Main в котором создать экземпляры вызывая разные конструкторы.

Задание 3

Используя IntelliJ IDEA создать проект, пакет.

(Заново!) Создать класс Машина с полями год(int), скорость(double), вес(int) цвет(String).

Создать конструктор по умолчанию, конструктор с 1-м параметром, 2-я, 3-я, 4-я.

Перегрузить конструкторы.

Создать класс Main, в котором создать экземпляры класса Машина с разными параметрами.

Задание 4

Используя IntelliJ IDEA создать проект, пакет.

(Заново!) Создать класс Машина с полями год(int), скорость(double), вес(int) цвет(String).

Создать конструктор по умолчанию, конструктор с 1-м параметром, 2-я, 3-я, 4-я.

Перегрузить конструкторы вызывая конструктор из конструктора.

Создать класс Main, в котором создать экземпляры класса Машина с разными параметрами.

Рекомендуемые ресурсы

<https://docs.oracle.com/javase/tutorial/java/javaOO/thiskey.html>

<https://docs.oracle.com/javase/tutorial/java/javaOO/constructors.html>

<https://docs.oracle.com/javase/tutorial/java/javaOO/classvars.html>