**Статические классы и методы**

[Java Syntax](https://javarush.ru/quests/QUEST_JAVA_SYNTAX)

[Уровень 6](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_SYNTAX&level=6), Лекция 7

— Кроме статических методов есть ещё и статические классы. Что это такое мы рассмотрим в будущем, я лишь покажу пример, что такое может быть:

Пример

public class StaticClassExample

{

private static int catCount = 0;

public static void main(String[] args) throws Exception

{

Cat vaska = new Cat("Bella");

Cat murka = new Cat("Tiger");

System.out.println("Cat count " + catCount);

}

public static class Cat

{

private String name;

public Cat(String name)

{

this.name = name;

StaticClassExample.catCount++;

}

}

}

2

Задача

Java Syntax,  6 уровень,  7 лекция

Набираем код Ӏ Java Syntax: 6 уровень, 7 лекция

Java Syntax: 6 уровень, 7 лекция. Внимание! Объявляется набор кода на JavaRush. Для этого включите режим повышенной внимательности, расслабьте пальцы, читайте код и… набирайте его в соответствующем окошке. Набор кода — вовсе не бесполезное занятие, как может показаться на первый взгляд: благодаря ему новичок привыкает к синтаксису и запоминает его (современные IDE редко дают ему это сделать).

— Объектов класса Cat можно создавать сколько угодно. В отличие от, например, статической переменной, которая существует в единственном экземпляре.

— Основной смысл модификатора **static** перед объявлением класса — это регулирование отношения класса **Cat** к классу **StaticClassExample**. Смысл примерно такой: класс Cat не привязан к объектам класса StaticClassExample, и не может обращаться к обычным (нестатическим) переменным класса StaticClassExample.

— Значит, я могу создавать классы внутри других классов?

— Да. Java такое позволяет. Не слишком задумывайся об этом сейчас. В будущем я объясню еще некоторые вещи, и все станет немного проще.

— Надеюсь.