**Финализируем с finalize**

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

[Уровень 6](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_SYNTAX&level=6), Лекция 3

— И снова привет! Сегодня будет кратенькое знакомство с методом **finalize**(). Этот метод вызывается Java-машиной у объекта перед тем, как объект будет уничтожен. Фактически этот метод – противоположность конструктору. В нем можно освобождать ресурсы, используемые объектом.

— Этот метод есть у класса Object и, следовательно, есть в каждом классе (все классы в Java считаются унаследованными от класса Object и содержат копию его методов). Ты можешь просто написать в твоём классе такой же метод, и он будет вызываться перед уничтожением объектов этого класса.

Пример:

Пример:

class Cat

{

String name;

Cat(String name)

{

**this.**name = name;

}

protected void **finalize**() throws Throwable

{

System.out.println(name + " destroyed");

}

}

2

Задача

Java Syntax,  6 уровень,  3 лекция

Набираем код Ӏ Java Syntax: 6 уровень, 3 лекция

Java Syntax: 6 уровень, 3 лекция. Иногда думать не надо, строчить надо! Как ни парадоксально звучит, порой пальцы «запоминают» лучше, чем сознание. Вот почему во время обучения в секретном центре JavaRush вы иногда встречаете задания на набор кода. Набирая код, вы привыкаете к синтаксису и зарабатываете немного материи. А ещё — боретесь с ленью.

— Понятно.

— Но! Java-машина сама решает – вызвать данный метод или нет. Чаще всего объекты, созданные в методе и объявленные мусором после его завершения, уничтожаются сразу же и без всяких вызовов метода **finalize**(). Этот метод скорее дополнительная страховка, чем надёжное решение. Лучшим вариантом будет освобождать любые используемые ресурсы (обнулять сохраненные ссылки на другие объекты), когда наш объект ещё жив. О пользе этого метода и нюансах его работы я расскажу тебе уровней через 10, или 20. А пока что ты должен знать две вещи: такой метод существует, и (сюрприз!) не всегда вызывается.