

Многопоточность в Java

Профессия Java-разработчик на Hexlet

Преподаватель: Яковлев Егор

Вопросы к лекции

- Как устроены потоки в Java
- Как создавать потоки в Java
- Как правильно завершать потоки?

План

1. Потоки в Java
2. Создание потоков в Java
3. Завершение потоков
4. join

Потоки в Java

- Каждый процесс имеет хотя бы один выполняющийся поток
- Выполнение главного потока начинается с метода `main()`
- В заданных программистом местах запускаются другие, побочные потоки

Создание потоков в Java

- Интерфейс Runnable
- Наследник класса Thread

Потоки в программе

- Программа заканчивает свою работу, когда заканчивают работу все её потоки
- Утверждение выше не относится к потокам-демонам

```
setDaemon(true); // поток-демон  
isDaemon(); // определяем, является ли поток демоном
```

Завершение потоков

- НЕ ИСПОЛЬЗОВАТЬ `Thread.stop()`, `Thread.suspend()`, `Thread.resume()`
- Поток получает сигнал об остановке – сам решает, когда завершиться – `InterruptedException`
- Каждый поток (как наследник класса `Thread`) содержит метод `isInterrupted()`

join() у потоков

- В Java предусмотрен механизм, позволяющий одному потоку ждать завершения выполнения другого - join()
- Метод join() имеет перегруженную версию, которая получает в качестве параметра время ожидания