

СПб АУ ИОЦНТ РАН

Java 06

11.11.2015

InterruptedException

```
try {  
    Thread.sleep(100);  
} catch (InterruptedException e) {  
    // ?  
}
```

Причина: вызов метода “interrupt” на объекте текущего потока

Причина: вызов метода “interrupt” на объекте текущего потока

- ▶ Завершение приложения

Причина: вызов метода “interrupt” на объекте текущего потока

- ▶ Завершение приложения
- ▶ Отмена текущей задачи

Причина: вызов метода “interrupt” на объекте текущего потока

- ▶ Завершение приложения
- ▶ Отмена текущей задачи
- ▶ Отмена текущей задачи (через интерфейс Future.cancel)

- ▶ Если ваш код где-то в другом месте может вызвать “interrupt”, то завершаем текущую задачу

InterruptedException. Варианты поведения

- ▶ Если ваш код где-то в другом месте может вызвать “interrupt”, то завершаем текущую задачу
- ▶ Если вы ничего такого не ожидали:

```
try {  
    Thread.sleep(100);  
} catch (InterruptedException e) {  
    // LOG.error(e.getMessage)  
    // throw new YourRuntimeInterruptedException(e)  
    Thread.currentThread().interrupt();  
}
```


С помощью флага `Thread.interrupted` (см. `interrupted` / `isInterrupted`):

- ▶ На самом деле мало методов могут бросить `InterruptedException` => Все равно очень часто придется проверять руками
- ▶ Все-таки отменяются именно задачи, а не потоки

Для каждой отдельной задачи заводим специальный флаг — отменена ли она

- ▶ В коде, который выполняется в рамках задачи необходимо с определенной периодичностью проверять флаг

Паттерны для отменяемых задач

Для каждой отдельной задачи заводим специальный флаг — отменена ли она

- ▶ В коде, который выполняется в рамках задачи необходимо с определенной периодичностью проверять флаг
- ▶ Можно завести специальный метод “checkCanceled”, который проверяет, установлен ли флаг. Если задача отменена, кидается unchecked исключение, которое ловится на самом верхнем уровне менеджера задач

Паттерны для отменяемых задач

Для каждой отдельной задачи заводим специальный флаг — отменена ли она

- ▶ В коде, который выполняется в рамках задачи необходимо с определенной периодичностью проверять флаг
- ▶ Можно завести специальный метод “checkCanceled”, который проверяет, установлен ли флаг. Если задача отменена, кидается unchecked исключение, которое ловится на самом верхнем уровне менеджера задач
- ▶ Остается только аккуратно разместить по коду задачи вызовы “checkCanceled()”