



## Проблемы многопоточности: обращение к общему ресурсу

Java Core (/quests/QUEST\_JAVA\_CORE)  
7 уровень (/quests/lectures/?quest=QUEST\_JAVA\_CORE&level=7), 1 лекция (/quests/lectures/questcore.level07.lecture01)

ОТКРЫТА

— Привет, Амиго! Вчера мы обсудили преимущества и удобства, которые несет с собой многопоточность (multithreading). Теперь пора взглянуть и на минусы. А они, к сожалению, не маленькие.

Раньше мы смотрели на программу, как на набор объектов, которые вызывают методы друг друга. Теперь все стало немного сложнее. Программа – это скорее набор объектов, по которым лазает несколько «маленьких роботиков» – нитей – и выполняют команды, содержащиеся в методах.

Формально – второе не отменяет первое. Это все еще объекты, и они все еще вызывают методы друг у друга. **Но нужно не забывать, что нитей – несколько, и каждая нить выполняет свою работу – свое задание.**

Программа становится сложнее. **Разные нити меняют состояние разных объектов в соответствии с задачей, которую каждая из них выполняет. И могут мешать друг другу.**

Но самое страшное происходит глубоко внутри Java-машины. Как я уже рассказывала, видимая одновременность работы нитей достигается за счет того, что процессор постоянно переключается с одной нити на другую. Переключился на нить, поработал 10 миллисекунд, переключился на следующую нить, там поработал 10 миллисекунд и так далее. И тут возникает проблема: переключение может произойти в самый неподходящий момент. Пример:

### Код первой нити

### Код второй нити

```
1 System.out.print ("Коле");
2 System.out.print ("");
3 System.out.print ("15");
4 System.out.print ("");
5 System.out.print ("лет");
6 System.out.println ();
```

```
1 System.out.print ("Лене");
2 System.out.print ("");
3 System.out.print ("21");
4 System.out.print ("");
5 System.out.print ("год");
6 System.out.println ();
```

### Ожидаемый вывод на консоль

Коле 15 лет  
Лене 21 год

**Итоговый порядок****Код первой нити****Код второй нити**

```
1 System.out.print ("Коле");
2 System.out.print ("Лене");
3 System.out.print (" ");
4 System.out.print (" ");
5 System.out.print ("15");
6 System.out.print ("21");
7 System.out.print (" ");
8 System.out.print (" ");
9 System.out.print ("лет");
10 System.out.println ();
11 System.out.print ("год");
12 System.out.println ();
```

```
1 System.out.print ("Коле");
2 //исполняется другая нить
3 //исполняется другая нить
4 System.out.print (" ");
5 System.out.print ("15");
6 //исполняется другая нить
7 //исполняется другая нить
8 System.out.print (" ");
9 System.out.print ("лет");
10 System.out.println ();
11 //исполняется другая нить
12 //исполняется другая нить
```

```
1 //исполняется другая нить
2 System.out.print ("Лене");
3 System.out.print (" ");
4 //исполняется другая нить
5 //исполняется другая нить
6 System.out.print ("21");
7 System.out.print (" ");
8 //исполняется другая нить
9 //исполняется другая нить
10 //исполняется другая нить
11 System.out.print ("год");
12 System.out.println ();
```

**Реальный вывод на консоль**

Коле Лене 15 21 лет  
год

Или вот еще пример:

**Код****Описание**

```
1 class MyClass
2 {
3     private String name1 = "Оля";
4     private String name2 = "Лена";
5     public void swap()
6     {
7         String s = name1;
8         name1 = name2;
9         name2 = s;
10    }
11 }
```

Метод `swap` меняет местами значения переменных `name1` & `name2`.

Что же будет если его вызывать из двух нитей одновременно?

исполняется другая нить выполняется другая нить `String s2 = name1; name1 = name2;` выполняется другая нить `name2 = s2;`

**Итоговый порядок****Код первой нити****Код второй нити**

```
1 String s1 = name1; //Оля
2 name1 = name2; //Лена
3 String s2 = name1; //Лена(!)
4 name1 = name2; //Лена
5 name2 = s1; //Оля
6 name2 = s2; //Лена
```

```
1 String s1 = name1;
2 name1 = name2;
3 //исполняется другая нить
4 //исполняется другая нить
5 name2 = s1;
6 выполняется другая нить
```

```
1 //исполняется другая нить
2 //исполняется другая нить
3 String s2 = name1;
4 name1 = name2;
5 //исполняется другая нить
6 name2 = s2;
```

**Итог**

Обе переменных имеют значение «Лена».  
Объект «Оля» был перезаписан и потерян.

— Кто бы мог подумать, что при элементарном присваивании возможны такие ошибки?

— Да, но для этой проблемы есть решение. Но об этом немного позже – у меня горло пересохло.

< (/quests/lectures/questcore.level07.lecture00)

×17 > (/quests/lectures/questcore.level07.lecture02)

vk 0 facebook 0 in 0 twitter 0 G+ 0 +8

Комментарии (32)

популярные

новые

старые

Никита

sh.ru/), **G+** (<https://plus.google.com/114772402300089087607>), **Twitter** ([https://twitter.com/javarush\\_ru](https://twitter.com/javarush_ru)),



Программистами не рождаются  
© 2018