

Примитивы синхронизации

Профессия Java-разработчик на Hexlet

Преподаватель: Яковлев Егор

Вопросы к лекции

- Что такое синхронизация?
- Какие есть возможности блокировки в Java?

План

1. Синхронизация
2. Блокировки
3. Ньюансы

Синхронизация

Синхронизация – это процесс, который позволяет выполнить все параллельные потоки в программе синхронно. Синхронизация позволяет избежать ошибок согласования памяти, возникшие из-за непоследовательного доступа к общей памяти

Блокировка

1. на уровне объекта
2. на уровне класса

Блокировка на уровне объекта

Это механизм синхронизации не статического метода или не статического блока кода, такой, что только один поток сможет выполнить данный блок или метод на данном экземпляре класса

Блокировка на уровне объекта

```
public class Demo1 {  
    public synchronized void demoMethod() {}  
}  
  
public class Demo2 {  
    public synchronized void demoMethod() {  
        synchronized(this) {  
            // other thread safe code  
        }  
    }  
}
```

Блокировка на уровне класса

```
public class Demo1 {  
    public synchronized static void demoMethod() {}  
}
```


Нюансы

А что будет, если поток выполнил метод с ошибкой?

JVM снимает блокировку всегда после выхода потока из метода,
даже если поток завершил его выполнение из-за ошибки или
исключения

Нюансы

А что будет, если из одного `synchronized` метода вызвать другой `synchronized` метод?

`synchronized` в Java реентерабельно – если синхронизированный метод вызовет другой синхронизированный метод, который требует такой-же замок, то текущий поток, который держит замок, может войти в этот метод, не приобретая замок

Нюансы

Насколько эффективно использовать ключевое слово
`synchronized`?

Синхронизированные методы в Java вносят дополнительные
затраты на производительность

Нюансы

Как использовать `synchronized` для конструктора?

В соответствии со спецификацией языка вы не можете использовать `synchronized` в конструкторе – это приведёт к ошибке компиляции

Блокировки

А есть ли другие способы синхронизации в Java?

Да – блокировки (Lock)

Вначале поток пытается получить доступ к общему ресурсу. Если он свободен, то на него накладывают блокировку. После завершения работы блокировка с общего ресурса снимается. Если же ресурс не свободен и на него уже наложена блокировка, то поток ожидает, пока эта блокировка не будет снята

Демо