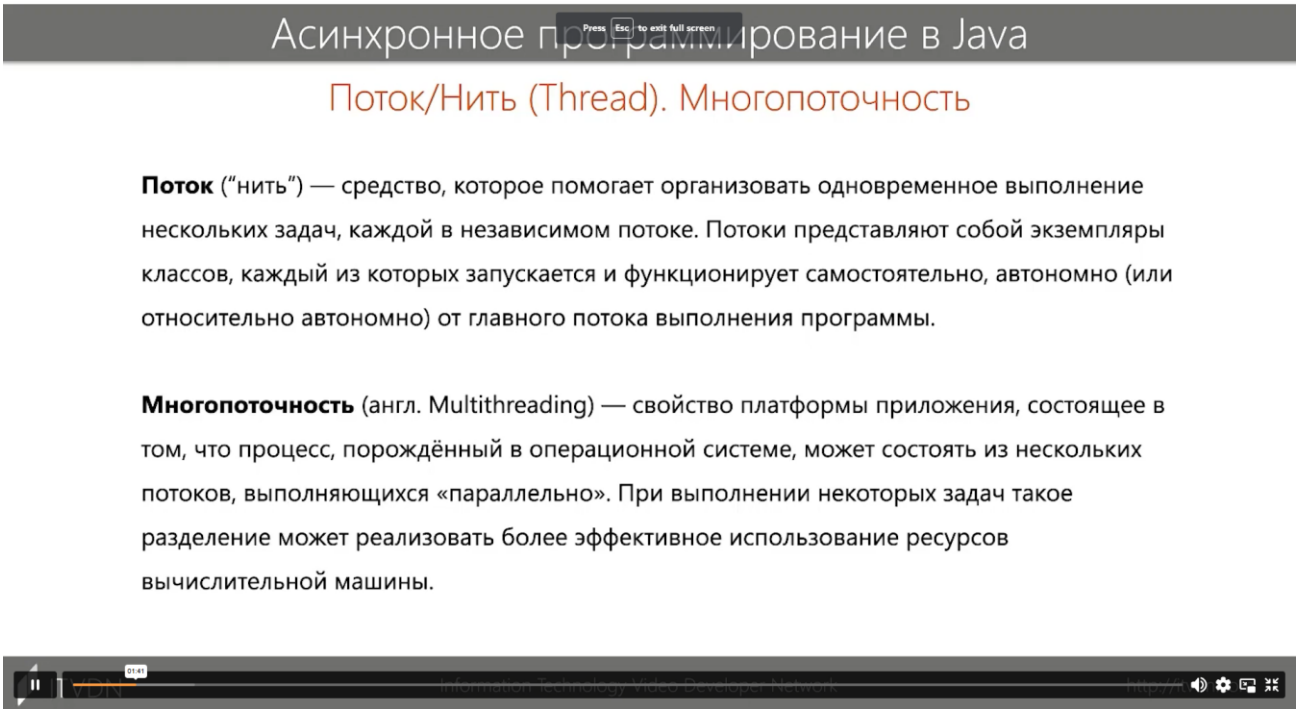
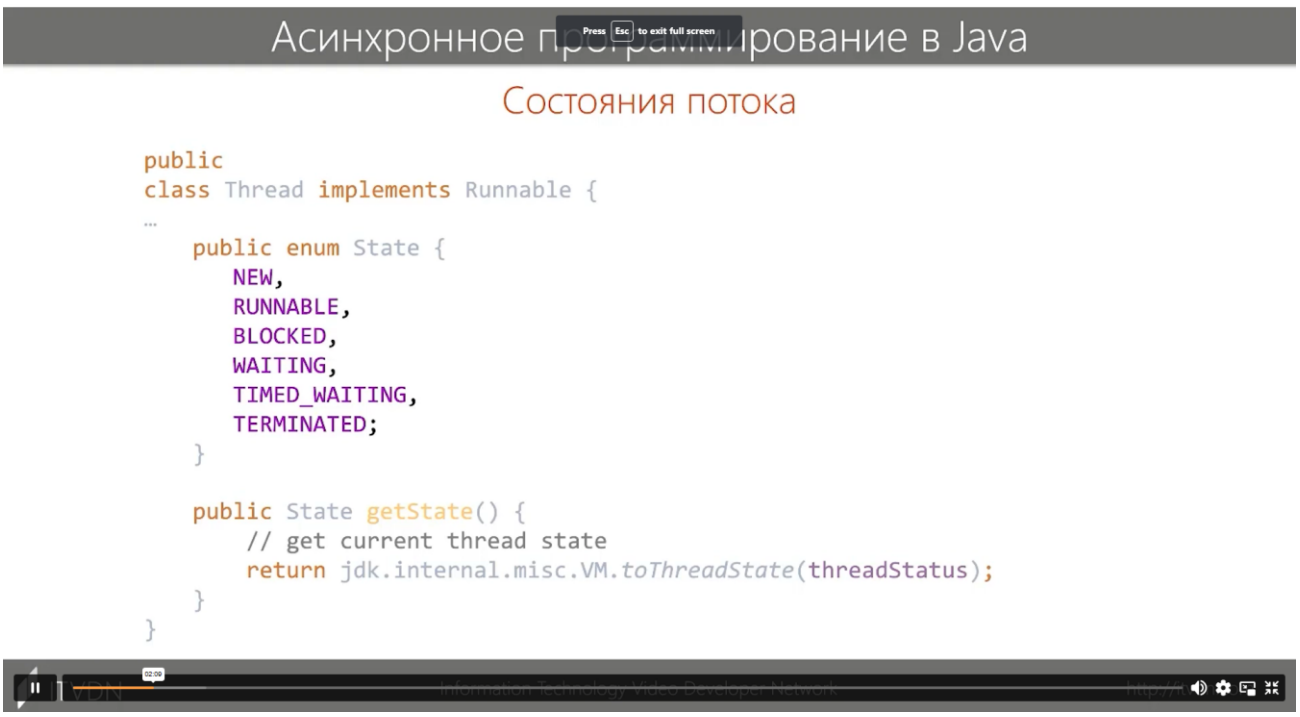
ITVDN_LESSON1





NEW – поток создан, но не запустиля

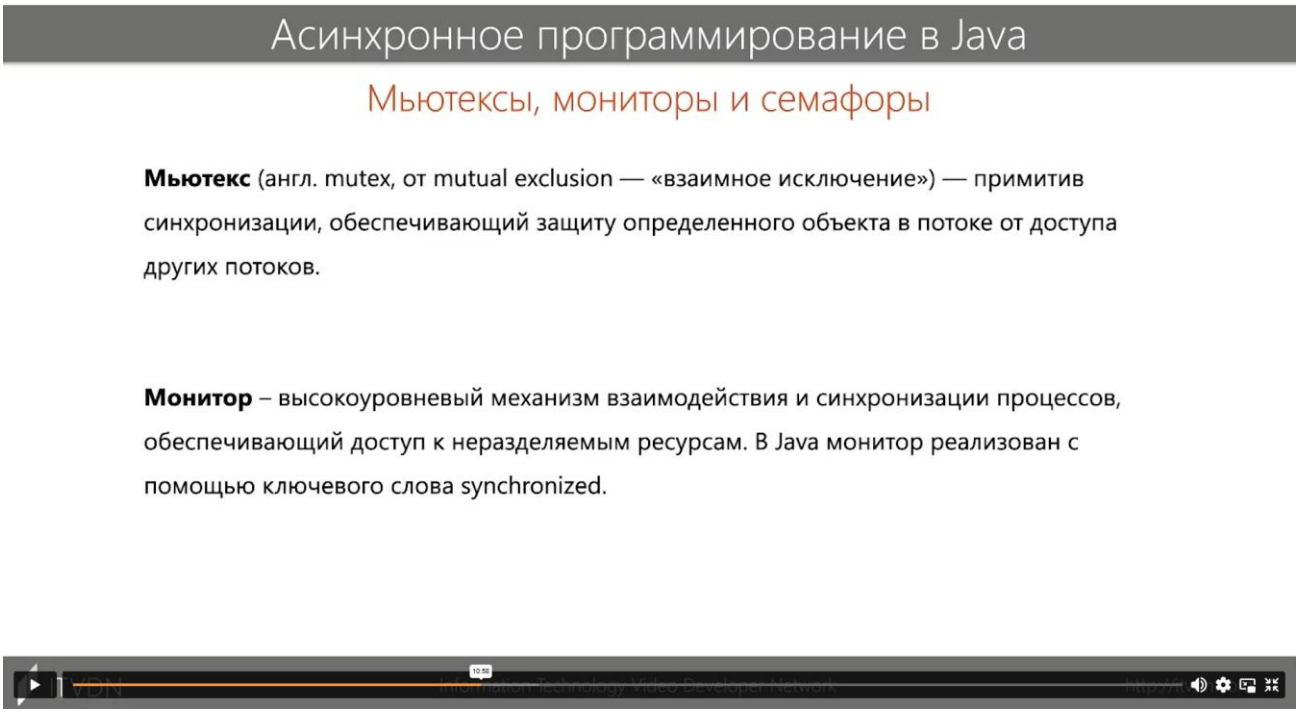
RUNNABLE – запустился и выполняется

BLOCKED – поток ожидает разблокировки монитора

WAITING – поток который ожидает другой поток чтобы выполнить определённые действия

TIMED_WAITING – поток который ожидает другой поток, чтобы выполнить действия за определённое время от какого то момента

TERMINATED – поток завершил свое выполнение



В java Нельзя работать с Мьютексом – работаем через Монитор

У мьютекса только два состояния TRUE and FALSE

Монитор(Synchronize) – блокирует ресурс – пока с ним работает другой поток.

Асинхронное программирование в Java

Мьютексы, мониторы и семафоры

```
public void doSomething() throws InterruptedException {
/*до тех пор, пока мьютекс объекта занят - любой другой поток (кроме того, который его захватил),
ожидает*/
     while (obj.getMutex().isBusy()) {
         Thread.sleep(1);
     }
     //пометить мьютекс объекта как занятый
     obj.getMutex().isBusy() = true;
     /*выполнить важную работу, при которой доступ к объекту должен быть только у одного потока*/
     obj.someImportantMethod();
     //освободить мьютекс объекта
     obj.getMutex().isBusy() = false;
 }
```

Асинхронное программирование в Java

Мьютексы, мониторы и семафоры

Семафор (англ. semaphore) — примитив синхронизации работы процессов и потоков, в основе которого лежит счётчик, над которым можно производить две атомарные операции: увеличение и уменьшение значения на единицу, при этом операция уменьшения для нулевого значения счётчика является блокирующей.

```java
public class SemaphoreSample {
    public static void main(String[] args) {
        Semaphore semaphore = new Semaphore( permits: 1);
        Runnable runnable = () -> {
            try {
                System.out.printf("Поток: %s запрашивает доступ к someMethod.\n", Thread.currentThread().getName());
                semaphore.acquire();
                someMethod();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            semaphore.release();
        };
```