

# Многопоточность лекция 2.

Gesamtpunktzahl 18/20 ?

Ваше имя

Alexander Hermann



1. Что произойдёт, если вызвать `t.run()` вместо `t.start()`?

1/1

- ☐ A. Поток будет создан и выполнен в новом потоке
- ☐ B. Поток выполнится дважды
- ☒ C. Метод `run()` выполнится в текущем потоке
- ☐ D. Произойдёт исключение `IllegalThreadStateException`



2. Какой метод следует использовать для корректной остановки потока?

1/1

- ☐ A. `t.stop()`
- ☒ B. `t.interrupt()`
- ☐ C. `t.terminate()`
- ☐ D. `t.shutdown()`



✓ 3. Что делает метод `join()`?

1/1

- ☐ A. Прерывает выполнение потока
- ☐ B. Принудительно завершает поток
- ☐ C. Приостанавливает поток на заданное время
- ☒ D. Ожидает завершения другого потока



✓ 4. Какой приоритет имеет поток по умолчанию?

1/1

- ☐ A. 1
- ☒ B. 5
- ☐ C. 7
- ☐ D. 10



✓ 5. Что произойдет при вызове `Thread.sleep(1000);`?

1/1

- ☐ A. Поток завершит выполнение
- ☒ B. Поток приостановит выполнение на 1 секунду
- ☐ C. Поток перейдёт в состояние ожидания до уведомления
- ☐ D. Поток автоматически получит максимальный приоритет



✓ **6. Что произойдёт с потоком-демоном, если все обычные потоки завершат работу?** 1/1

- ☐ A. Он продолжит работать до конца программы
- ☒ B. Его выполнение будет завершено автоматически ✓
- ☐ C. Он вызовет исключение
- ☐ D. JVM не завершится

✓ **7. Какой метод устанавливает поток как демон?** 1/1

- ☐ A. `makeDaemon(true)`
- ☐ B. `setAsDaemon()`
- ☐ C. `Thread.setDaemon(true)`
- ☒ D. `setDaemon(true)` ✓

✓ **8. Какой результат у вызова `Thread.currentThread().setPriority(11)`?** 1/1

- ☐ A. Поток получит максимальный приоритет
- ☐ B. Поток будет работать быстрее
- ☒ C. Бросится исключение `IllegalArgumentException` ✓
- ☐ D. Приоритет будет сброшен до 5



✓ 9. Что делает метод `isDaemon()`?

1/1

- ☐ A. Завершает демон-поток
- ☐ B. Устанавливает поток как демон
- ☒ C. Проверяет, является ли поток демон-поток ✓
- ☐ D. Возвращает приоритет потока

✓ 10. Что произойдёт, если прервать поток, находящийся в состоянии `sleep()`? 1/1

- ☐ A. Он завершится немедленно
- ☐ B. Он продолжит выполнение
- ☒ C. Будет выброшено `InterruptedException` ✓
- ☐ D. Поток переключится на другой

✗ 11. Какой метод возвращает значение флага прерывания и сбрасывает его?

0/1

- ☐ A. `isInterrupted()`
- ☐ B. `checkInterrupt()`
- ☒ C. `interrupted()` ✗
- ☐ D. `clearInterrupt()`

Richtige Antwort

- ☒ D. `clearInterrupt()`



✓ 12. Какой метод используется для проверки флага прерывания без его сброса? 1/1

- ☐ A. interrupted()
- ☐ B. hasInterrupted()
- ☐ C. checkInterrupt()
- ☒ D. isInterrupted()



✓ 13. Что произойдёт, если вызвать stop() на работающем потоке? 1/1

- ☐ A. Поток корректно завершится
- ☐ B. Поток перейдёт в состояние ожидания
- ☒ C. Поток будет остановлен немедленно, что может привести к ошибкам
- ☐ D. Ничего не произойдёт



✓ 14. Что произойдёт, если поток был прерван, но не обрабатывает исключение? 1/1

- ☒ A. Он продолжит выполнение
- ☐ B. Он будет немедленно завершён
- ☐ C. Он перейдёт в состояние ожидания
- ☐ D. Компилятор выдаст ошибку



✓ 15. Какая пара методов используется для управления флагом прерывания потока?

1/1

- ☐ A. wait() и notify()
- ☐ B. stop() и resume()
- ☒ C. isInterrupted() и interrupt() ✓
- ☐ D. sleep() и join()

✓ 16. Что выведет следующий код?

1/1

```
public class Main {  
    public static void main(String[] args) {  
        Thread t = new Thread(() -> System.out.println("Hello from thread"));  
        t.run();  
        System.out.println("Main done");  
    }  
}
```

- ☒ A. Hello from thread Main done ✓
- ☐ B. Main done Hello from thread
- ☐ C. Hello from thread
- ☐ D. Ошибка компиляции



```
public class Main {  
    public static void main(String[] args) {  
        Thread t = new Thread(() -> {  
            System.out.println("Running");  
            try {  
                Thread.sleep(1000);  
            } catch (InterruptedException e) {  
                System.out.println("Interrupted");  
            }  
        });  
        t.start();  
        t.interrupt();  
    }  
}
```

- ☒ A. Running Interrupted
- ☐ B. Interrupted Running
- ☐ C. Только Running
- ☐ D. Ошибка выполнения



```
public class Main {  
    public static void main(String[] args) throws InterruptedException {  
        Thread t = new Thread(() -> {  
            try {  
                Thread.sleep(2000);  
            } catch (InterruptedException e) {  
                System.out.println("Was interrupted");  
            }  
        });  
  
        t.start();  
        Thread.sleep(1000);  
        t.interrupt();  
        t.join();  
        System.out.println("Main finished");  
    }  
}
```

- ☐ A. Main finished
- ☐ B. Was interrupted Main finished
- ☒ C. Ошибка компиляции
- ☐ D. Поток зависнет



Richtige Antwort

- ☒ B. Was interrupted Main finished





✓ 19. Что выведет следующий код?

1/1

```
public class Main {  
    public static void main(String[] args) {  
        Thread.currentThread().setName("MainThread");  
        System.out.println(Thread.currentThread().getName());  
    }  
}
```

- ☒ A. MainThread ✓
- ☐ B. Thread-0
- ☐ C. main
- ☐ D. java.lang.Thread

✓ 20. Что выведет следующий код?

1/1

```
public class Main {  
    public static void main(String[] args) {  
        Thread t = new Thread(() -> System.out.println(Thread.currentThread().isDaemon()));  
        t.setDaemon(true);  
        t.start();  
    }  
}
```

- ☒ A. true ✓
- ☐ B. false
- ☐ C. Ошибка выполнения
- ☐ D. Ошибка компиляции

Д Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. - [Nutzungsbedingungen](#) - [Datenschutzerklärung](#)

Sieht dieses Formular verdächtig aus? [Bericht](#)

Google Formulare

