

Введение в многопоточность (Java)

Gesamtpunktzahl 9/10

✓ 1. Что из ниже перечисленного является определением потока (thread)? 1/1

- ☐ A) Класс, запускающий программу
- ☐ B) Отдельная программа, выполняющаяся на компьютере
- ☒ C) Наименьшая единица выполнения внутри процесса ✓
- ☐ D) Секция кода, вызываемая из метода main

✗ 2. В чём отличие Thread.run() от Thread.start()? 0/1

- ☒ A) run() запускает поток, start() вызывает метод run() ✗
- ☐ B) run() выполняется в текущем потоке, start() — в новом
- ☐ C) start() нельзя вызывать у Runnable
- ☐ D) Разницы нет

Richtige Antwort

- ☒ B) run() выполняется в текущем потоке, start() — в новом



✓ **3. Что делает метод join()?**

1/1

- ☐ A) Завершает поток
- ☐ B) Останавливает выполнение потока навсегда
- ☒ C) Приостанавливает текущий поток, пока не завершится другой ✓
- ☐ D) Очищает память, занятую потоком

✓ **4. Какой интерфейс можно реализовать для создания потока без наследования от Thread?**

1/1

- ☒ A) Runnable ✓
- ☐ B) Executor
- ☐ C) Callable
- ☐ D) Sleepable

✓ **5. Метод sleep(...) класса Thread вызывает checked-исключение. Какое именно?**

1/1

- ☐ A) InterruptedException
- ☐ B) IOException
- ☒ C) InterruptedException ✓
- ☐ D) ExecutionException



✓ 6. Что выведет следующий код?

1/1

```
Thread t = new Thread(() -> System.out.println("Hello"));
t.run();
System.out.println("Done");
```

- ☒ A) Hello Done
- ☐ B) Done Hello
- ☐ C) Hello (потом исключение)
- ☐ D) Ничего не выведется



✓ 7. Что выведет следующий код?

1/1

```
Runnable r = () -> {
    for (int i = 0; i < 3; i++) {
        System.out.print(i + " ");
    }
};
r.start();
```

- ☐ A) 0 1 2
- ☐ B) 0 1 2 0 1 2
- ☒ C) Ошибка компиляции
- ☐ D) Исключение во время выполнения



```
Thread t = new Thread(() -> {  
    try {  
        Thread.sleep(1000);  
    } catch (Exception e) {  
        System.out.println("Interrupted");  
    }  
    System.out.println("Thread finished");  
});  
t.start();  
t.join();
```

- ☐ A) Программа зависнет
- ☒ B) Ошибка компиляции ✓
- ☐ C) Всё отработает: Thread finished
- ☐ D) Исключение InterruptedException

Feedback

join() вызывает InterruptedException, а он не обработан → ошибка компиляции.



✓ 9. Что выведет следующий код?

1/1

```
new Thread(() -> {
    try {
        Thread.sleep(500);
    } catch (InterruptedException e) {
        System.out.println("Interrupted");
    }
}).start();
System.out.println("Main done");
```

- ☐ A) Main done затем Interrupted
- ☐ B) Interrupted затем Main done
- ☐ C) Main done, потом может быть Interrupted

☒ D) Main done, и больше ничего



✓ 10. Что будет, если вызвать start() дважды на одном и том же объекте Thread?

1/1

```
Thread t = new Thread(() -> System.out.println("Run"));
t.start();
t.start();
```

- ☐ A) Поток запустится дважды
- ☐ B) Ничего не произойдёт
- ☐ C) Компилятор выдаст предупреждение

☒ D) Исключение IllegalStateException



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

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

Google Formulare

