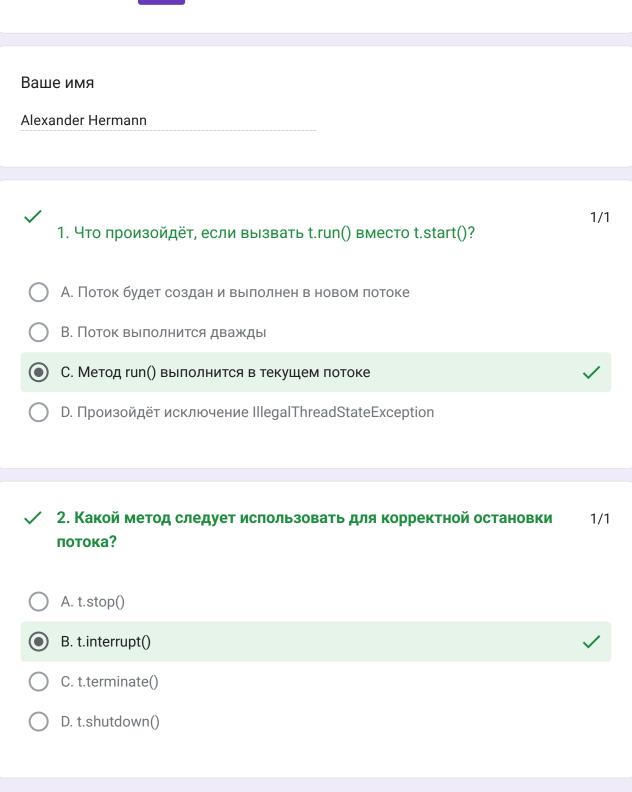
## Многопоточность лекция 2.

| Gesamt | punk | ktzał | ٦ĺ |
|--------|------|-------|----|

18/20 ?





| ✓ 3. Что делает метод join()?   | 1/1      |
|---|----------|
| <ul><li>А. Прерывает выполнение потока</li></ul>                          |          |
| В. Принудительно завершает поток  |          |
| С. Приостанавливает поток на заданное время                               |          |
| <ul><li>D. Ожидает завершения другого потока</li></ul>                    | <b>~</b> |
|   |          |
| 4. Какой приоритет имеет поток по умолчанию?                              | 1/1      |
| O A. 1  |          |
| <ul><li>B. 5</li></ul>  | <b>✓</b> |
| O C. 7  |          |
| O D. 10   |          |
|   |          |
| ✓ 5. Что произойдет при вызове Thread.sleep(1000);?                       | 1/1      |
| <ul><li>А. Поток завершит выполнение</li></ul>                            |          |
| В. Поток приостановит выполнение на 1 секунду                             | <b>✓</b> |
| С. Поток перейдёт в состояние ожидания до уведомления                     |          |
| <ul> <li>D. Поток автоматически получит максимальный приоритет</li> </ul> |          |

| 6. Что произойдёт с потоком-демоном, если все обычные потоки завершат работу? | 1/1      |
|---|----------|
| <ul> <li>А. Он продолжит работать до конца программы</li> </ul>               |          |
| В. Его выполнение будет завершено автоматически                               | <b>✓</b> |
| С. Он вызовет исключение  |          |
| О 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)  | <b>✓</b> |
|   |          |
| ✓ 8. Какой результат у вызова Thread.currentThread().setPriority(11)?         | 1/1      |
| <ul> <li>А. Поток получит максимальный приоритет</li> </ul>                   |          |
| В. Поток будет работать быстрее   |          |
| © С. Бросится исключение IllegalArgumentException                             | <b>✓</b> |
| <ul><li>D. Приоритет будет сброшен до 5</li></ul>                             |          |

| <b>✓</b> | 9. Что делает метод isDaemon()?   | 1/1      |
|----------|---|----------|
| 0        | А. Завершает демон-поток  |          |
| 0        | В. Устанавливает поток как демон  |          |
| •        | С. Проверяет, является ли поток демон-потоком                             | <b>✓</b> |
| 0        | D. Возвращает приоритет потока  |          |
|          |   |          |
| <b>~</b> | 10. Что произойдёт, если прервать поток, находящийся в состоянии sleep()? | 1/1      |
| 0        | А. Он завершится немедленно   |          |
| 0        | В. Он продолжит выполнение  |          |
| •        | C. Будет выброшено InterruptedException                                   | <b>✓</b> |
| 0        | D. Поток переключится на другой   |          |
|          |   |          |
| ×        | 11. Какой метод возвращает значение флага прерывания и сбрасывает его?    | 0/1      |
| 0        | A. isInterrupted()  |          |
| 0        | B. checkInterrupt()   |          |
| •        | C. interrupted()  | ×        |
| 0        | D. clearInterrupt()   |          |
| Richt    | ige Antwort   |          |
| •        | D. clearInterrupt()   |          |

| <b>✓</b> | 12. Какой метод используется для проверки флага прерывания без его сброса? | 1/1      |
|----------|--|----------|
| 0        | A. interrupted()   |          |
| 0        | B. hasInterrupted()  |          |
| 0        | C. checkInterrupt()  |          |
| •        | D. isInterrupted()   | <b>✓</b> |
|          |  |          |
| <b>/</b> | 13. Что произойдёт, если вызвать stop() на работающем потоке?              | 1/1      |
| 0        | А. Поток корректно завершится  |          |
| 0        | В. Поток перейдёт в состояние ожидания                                     |          |
| •        | С. Поток будет остановлен немедленно, что может привести к ошибкам         | <b>✓</b> |
| 0        | D. Ничего не произойдёт  |          |
| <b>/</b> | 14. Что произойдёт, если поток был прерван, но не обрабатывает исключение? | 1/1      |
| •        | А. Он продолжит выполнение   | <b>✓</b> |
| 0        | В. Он будет немедленно завершён  |          |
| 0        | С. Он перейдёт в состояние ожидания  |          |
| 0        | D. Компилятор выдаст ошибку  |          |

| 15. Какая пара методов используется для управления флагом прерывания потока?   | 1/1      |
|--|----------|
| A. wait() и notify()   |          |
| В. stop() и resume()   |          |
| © C. isInterrupted() и interrupt()   | <b>✓</b> |
| D. sleep() и join()  |          |
| <pre> ✓ 16. Что выведет следующий код?  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");     } }  </pre> A Hello from thread Main done | 1/1      |
| A. Hello from thread Main done   |          |
| B. Main done Hello from thread   |          |
| C. Hello from thread   |          |
| О В. 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

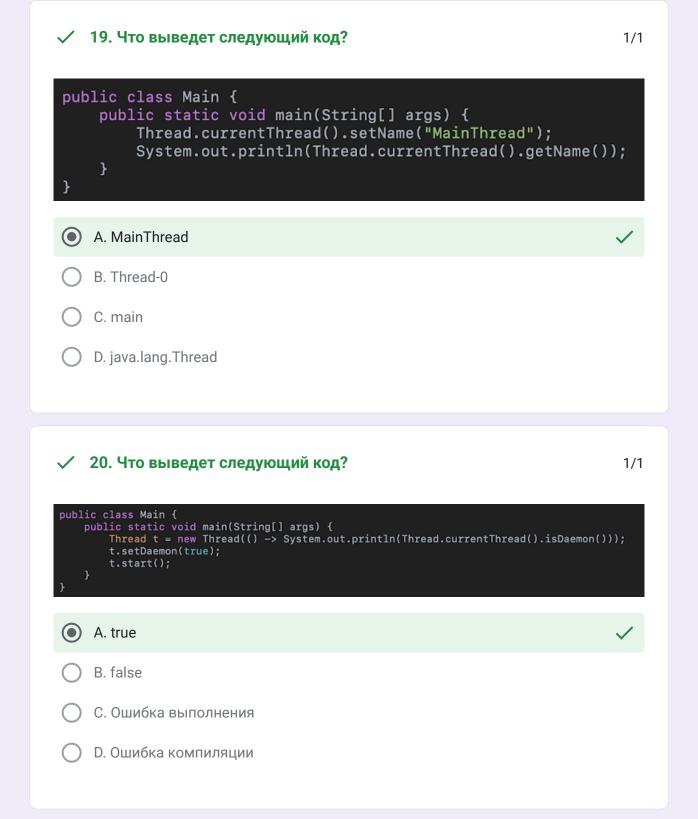
- A. Main finished
- B. Was interrupted Main finished
- С. Ошибка компиляции

X

D. Поток зависнет

Richtige Antwort

B. Was interrupted Main finished



Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. - <u>Nutzungsbedingungen</u> - <u>Datenschutzerklärung</u>

Sieht dieses Formular verdächtig aus? Bericht

## Google Formulare