

Тестируем код

Метод `main` или JUnit



ПРЕПОДАВАТЕЛЬ



Артем Гордийчук

Full-stack software engineer

- Более 8 лет опыта работы
- Java, Spring, Hibernate, AWS, Oracle, PostgreSQL
- Проекты связанные с банковской, финансовой деятельностью, e-commerce

artemsgor@gmail.com

www.linkedin.com/in/artem-g-48071a61



ВАЖНО:

- Камера должна быть включена на протяжении всего занятия.
- Если у Вас возник вопрос в процессе занятия, пожалуйста, поднимите руку и дождитесь, пока преподаватель закончит мысль и спросит Вас, также можно задать вопрос в чате или когда преподаватель скажет, что начался блок вопросов.
- Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях.
- Вести себя уважительно и этично по отношению к остальным участникам занятия.
- Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя.

ПЛАН ЗАНЯТИЯ

1. Повторение
2. Вопросы по повторению
3. Основной блок
4. Задание для закрепления
5. Задание для закрепления
6. Вопросы по основному блоку
7. Практическая работа
8. Оставшиеся вопросы

1

ПОВТОРЕНИЕ ИЗУЧЕННОГО

Повторение

- Строки
- StringBuilder
- StringBuffer





TEL-RAN
by Starta Institute

2

ВОПРОСЫ ПО ПОВТОРЕНИЮ

Введение

- Метод `main()` для теста
- JUnit для теста





TEL-RAN
by Starta Institute

3

ОСНОВНОЙ БЛОК

Метод `main()`, как точка входа и метод для теста

- `main()` является отправной точкой для JVM для начала выполнения Java-программы.
- Без метода `main()` JVM не будет выполнять программу.

```
public static void main(String[] args) {  
    int a = 3;  
    int b = 4;  
    System.out.println(a+b);  
}
```

← Старт программы

← Визуальный тест программы



TEL-RAN
by Starta Institute

4

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

ЗАДАНИЕ

1. Создайте метод `getNumbers()`;
2. Метод должен возвращать массив из трех целочисленных значений: `[1, 2, 3]`
3. Вызовите метод в `main()`
4. Убедитесь, что метод работает верно

JUnit

- Тестирование — это процесс проверки функциональности приложения, чтобы убедиться, что оно работает в соответствии с требованиями.
- Модульное тестирование - это тестирование одной сущности (класса или метода).
- Модульное тестирование может быть выполнено двумя способами:
 - ручным тестированием (как пример – посмотреть результат в методе `main()`)
 - автоматизированным тестированием (напишем код, который не будет зависеть от человеческих ресурсов)



JUnit

- JUnit - это фреймворк модульного тестирования для языка программирования Java.
- JUnit:
 - повышает производительность программиста и стабильность программного кода
 - снижает нагрузку на программиста и время, затрачиваемое на отладку
- Модульный тестовый случай - является частью кода, которая гарантирует, что другая часть кода (метод) работает должным образом.



JUnit

Перед тем, как начать писать тесты:

1. Найдите pom.xml в своем проекте
2. Добавьте зависимость в блок dependencies

```
<dependencies>
```

```
  <dependency>
```

```
    <groupId>org.junit.jupiter</groupId>
```

```
    <artifactId>junit-jupiter</artifactId>
```

```
    <version>5.9.2</version>
```

```
    <scope>test</scope>
```

```
  </dependency>
```

```
</dependencies>
```

3. Ок. Все готово!



JUnit – основные понятия

- Test Class – класс в котором разрабатываются тесты
- Test method – метод в котором тестируется функционал другого кода: помечается аннотацией @Test
- @BeforeEach - указывает на то, что, что-то должно выполняться перед каждым методом @Test
- @AfterEach - указывает на то, что, что-то должно выполняться после каждого @Test



JUnit – основные понятия

Утверждения и предположения:

- `assertEquals()` - проверяет равное условие результата
- `assertTrue()` - проверяет, является ли данное условие истинным или нет
- `assertFalse()` - проверяет, является ли данное условие ложным
- `assertNull()` - проверяет, имеет ли переменная данных `null` значение или нет
- `assertNotNull()` - проверяет, имеет ли переменная данных значение или нет





TEL-RAN
by Starta Institute

5

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

ЗАДАНИЕ

1. Написать тест для метода `getNumbers()`;
2. Используйте метод `Assertions.assertEquals`
3. Проанализируйте вывод
4. Используйте метод `Assertions.assertArrayEquals`
5. Проанализируйте результат
6. Напишите тест, который проверяет размер массива
7. Создайте метод `void init()`, определите в нем ожидаемый результат и примените аннотацию `@BeforeEach`
8. Проанализируйте работу тестов

Экспресс-опрос

- **Вопрос 1.**

Расскажите о преимуществах и недостатках тестов при помощи метода `main()`

- **Вопрос 2.**

Что делает аннотация `@Test`?

- **Вопрос 2.**

Назовите все аннотации, которые усвоили.



6

ВОПРОСЫ ПО ОСНОВНОМУ БЛОКУ



TEL-RAN
by Starta Institute

7

ПРАКТИЧЕСКАЯ РАБОТА

Практическое задание 1

1. Написать метод `findMax()`, который возвращает максимальное число из массива
2. Протестировать с помощью метода `main()`
3. Написать тесты используя библиотеку JUnit
 - a. тест, который проверяет пустой массив или нет
 - b. тест, который проверяет правильность поиска максимального значения



Реализация задания 1

```
public static int findMax(int[] arr) {  
    int max = arr[0];  
    for (int i = 1; i < arr.length; i++) {  
        if (max < arr[i])  
            max = arr[i];  
    }  
    return max;  
}
```


Практическое задание 2

Проверка номера банковского счета

Создайте метод `isValidAccountNumber(String accountNumber)`, который возвращает `true` в случае успешной проверки.

1. БС может содержать только 14 цифр
2. Все 14 цифр не могут быть нулями
3. Номер счета не может быть `null` или пустым.

Проверьте с помощью метода `main()`

Напишите тесты используя библиотеку JUnit



Реализация задания 2

```
public static boolean isValidAccountNumber(String accountNumber) {  
    if (accountNumber == null || accountNumber.equalsIgnoreCase( anotherString: "")) {  
        return false;  
    }  
    if (accountNumber.length() == 14) {  
        int count = 0;  
        for (int i = 0; i < accountNumber.length(); i++) {  
            if (!Character.isDigit(accountNumber.charAt(i))) {  
                return false;  
            }  
            if (accountNumber.charAt(i) == '0') {  
                count += 1;  
            }  
        }  
        return count != 14;  
    } else {  
        return false;  
    }  
}
```



TEL-RAN
by Starta Institute

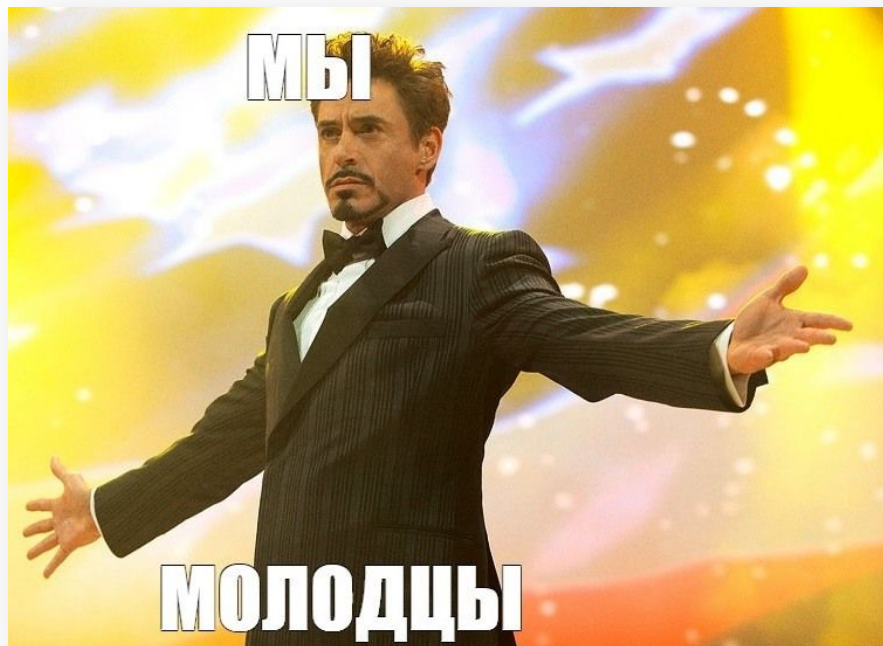
8

ОСТАВШИЕСЯ ВОПРОСЫ

Полезные ссылки

- [JUnit - Wikipedia](#)
- [JUnit 5 User Guide](#)
- [Maven Repository: org.junit.jupiter \(mvnrepository.com\)](#)

ЗАКЛЮЧЕНИЕ



Дополнительная практика

1. Создайте метод: `getElementByIndex()`, который принимает, массив и индекс в массиве
2. Метод возвращает элемент по индексу
3. Напишите тесты
 - a. Элемент действительно возвращается
 - b. **При передаче неверного индекса, вы получаете ошибку `ArrayIndexOutOfBoundsException`