

Цели и смысл тестирования

Семинар 1

Давайте знакомиться!



Вячеслав Савенок

Роль: Разработчик в АО "Российские космические системы".

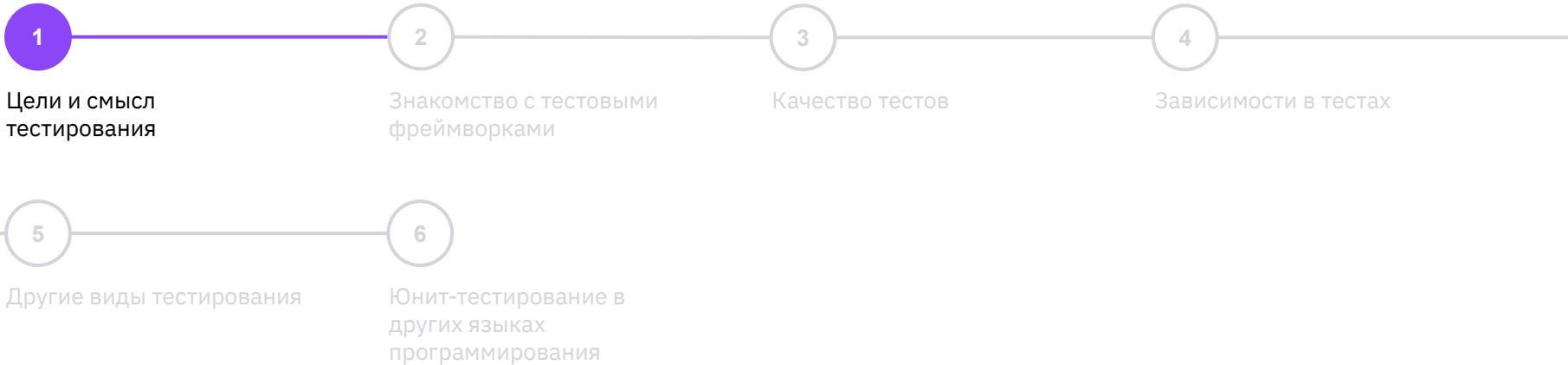
Стек: Java, Spring Boot, Hibernate, PostgreSQL, JUnit

Обучение: Автор образовательных программ в GeekBrains.

Достижения: Руководитель команды программистов



План курса





Что будет на уроке сегодня

- 📌 Что такое тестирование
- 📌 Цели тестирования
- 📌 Ошибки
- 📌 Принципы тестирования программного обеспечения
- 📌 Цикл разработки
- 📌 Попробуем написать свой первый тест на Java





Повторение теории с лекции





Вопрос

Что такое тестирование ПО?

20 секунд на размышление

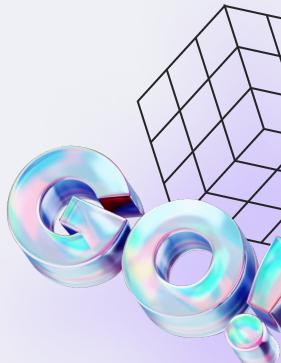




Вопрос

Какие есть виды ошибок и в чем
они заключаются?

20 секунд на размышление

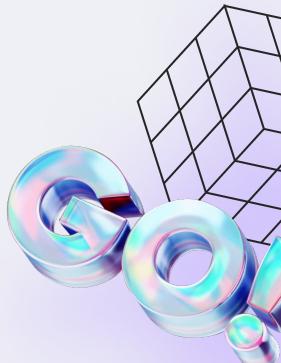




Вопрос

Что определяет затраты на исправление ошибки в коде

20 секунд на размышление





Вопрос

Перечислите 7 принципов
тестирования ПО

20 секунд на размышление





Вопрос

Зачем нам нужны asserts и библиотека AssertJ?

20 секунд на размышление





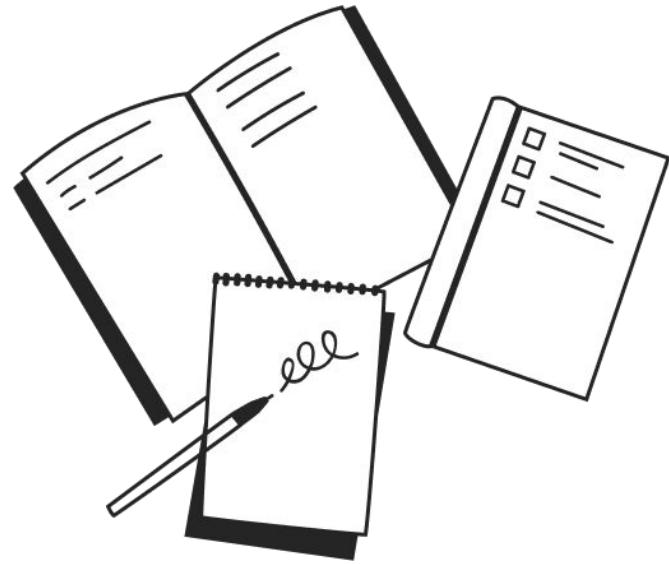
Задание с
лекции





Задание

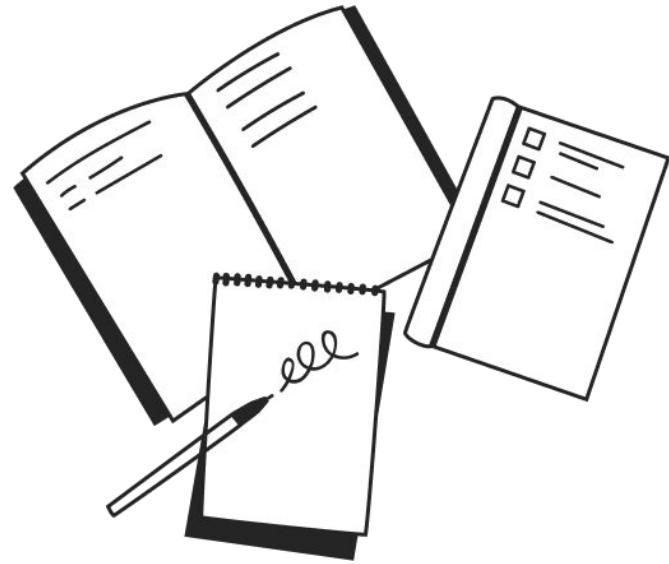
1. Придумайте и опишите (можно в псевдокоде) функцию извлечения корня и необходимые проверки для него используя граничные случаи
2. Письменно ответьте: Как будет выглядеть проверка для случая деления на ноль? (с использованием AssertJ)
3. Письменно ответьте: Сравните одну и ту же проверку с использованием условий, ассертов, AssertJ в каком случае стандартное сообщение об ошибке будет более информативным?
4. (Устно) Протестируйте системный калькулятор на вашем ПК на граничные условия, как он реагирует? Как на нем обрабатывается деление на ноль?





Задание

1. Придумайте и опишите (можно в псевдокоде) функцию извлечения корня и необходимые проверки для него используя граничные случаи
2. Письменно ответьте: Как будет выглядеть проверка для случая деления на ноль?
(с использованием AssertJ)
3. Письменно ответьте: Сравните одну и ту же проверку с использованием условий, ассертов, AssertJ в каком случае стандартное сообщение об ошибке будет более информативным?
4. (Устно) Протестируйте системный калькулятор на вашем ПК на граничные условия, как он реагирует? Как на нем обрабатывается деление на ноль?





Задание

1. Придумайте и опишите (можно в псевдокоде) функцию извлечения корня и необходимые проверки для него используя граничные случаи
2. Письменно ответьте: Как будет выглядеть проверка для случая деления на ноль? (с использованием AssertJ)
3. Письменно ответьте: Сравните одну и ту же проверку с использованием условий, ассертов, AssertJ в каком случае стандартное сообщение об ошибке будет более информативным?
4. (Устно) Протестируйте системный калькулятор на вашем ПК на граничные условия, как он реагирует? Как на нем обрабатывается деление на ноль?





Задание

1. Придумайте и опишите (можно в псевдокоде) функцию извлечения корня и необходимые проверки для него используя граничные случаи
2. Письменно ответьте: Как будет выглядеть проверка для случая деления на ноль? (с использованием AssertJ)
3. Письменно ответьте: Сравните одну и ту же проверку с использованием условий, ассертов, AssertJ в каком случае стандартное сообщение об ошибке будет более информативным?
4. Протестируйте системный калькулятор на вашем ПК на граничные условия, как он реагирует? Как на нем обрабатывается деление на ноль?





Вопросы?



Вопросы?

Вопросы?





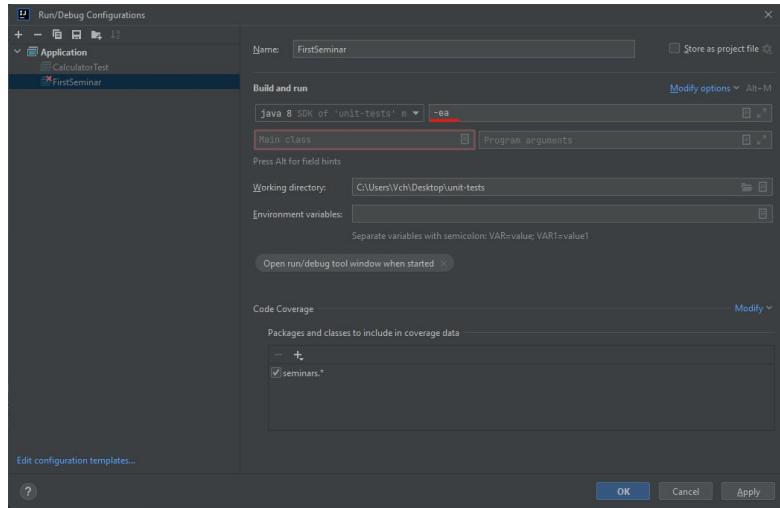
Практика





Задание №0

- 📌 IDEA (JDK) [Download IntelliJ IDEA – The Leading Java and Kotlin IDE](#)
- 📌 Флаг -ea
- 📌 [AssertJ Core](#)



	AssertJ Fluent Assertions » 3.23.1
Rich and fluent assertions for testing for Java	
License	Apache 2.0
Categories	Assertion Libraries
Tags	testing assert assertion quality
HomePage	\${project.parent.url}#assertj-core
Date	May 31, 2022
Files	pom (23 KB) jar (1.2 MB) View All
Repositories	Central
Ranking	#22 in MvnRepository (See Top Artifacts) #1 in Assertion Libraries
Used By	14,414 artifacts
Vulnerabilities	Vulnerabilities from dependencies: CVE-2022-42004 CVE-2022-42003



Задание №1

Давайте представим, что вы работаете над календарным приложением и вам нужно убедиться, что количество выходных в неделе правильно установлено.

Исправьте следующий код так, чтобы утверждение было верно и код не выбрасывал ошибку

```
public static void assertConditionA() {  
    String[] weekends = {"Суббота", "Воскресенье"};  
    assert weekends.length == 3;  
    System.out.println("В неделе " + weekends.length + " дня выходных");  
}
```



Задание №2

Разберемся с использованием условий в утверждениях на примере переменной x.

Нужно исправить код так, чтобы метод не выбрасывал ошибку.

```
public static void assertConditionB() {  
    int x = -1;  
    assert x >= 0;  
}
```



Задание №3

Попробуем сделать наш код более надежным, предотвратив переполнение переменной.

Исправьте метод сложения двух чисел так, чтобы при переполнении переменной выбрасывалось предупреждение.

```
public static int sum(int a, int b) {  
    return a+b;  
}
```



Задание №4

The screenshot shows a GitHub commit card for a tweet from the account @windowsdev. The tweet, posted 3 hours ago, reads "Run code 🐦 : Happy New Year". The commit itself contains the following C# code:

```
1 if (DateTime.Now.ToString() == "01/01/2022 00:00:00")
2 {
3     Console.WriteLine("Happy New Year");
4 }
5 else
6 {
7     Console.WriteLine("It's still 2021...");
8 }
```

The card includes a "Windows 11" icon at the bottom left and standard GitHub social sharing icons at the bottom right.



Задание №4

Найдите и исправьте логическую ошибку в коде, который планирует поздравление с Новым Годом в полночь.

```
public static void happyNY() {
    Calendar calendar = Calendar.getInstance();
    DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
    String currentTime = dateFormat.format(calendar.getTime());

    assert currentTime.equals("01/01/2023 00:00:00") : "Еще 2023 год :(";
    System.out.println("С новым годом!");
}
```



Задание №5

Исправьте код метода, предполагая что условие утверждения написано верно.

```
public static void checkingShoppingCart() {  
    ArrayList<String> productCategories = new ArrayList<>();  
    productCategories.add("fruits");  
    productCategories.add("vegetables");  
    productCategories.add("bakery");  
  
    ArrayList<String> products = new ArrayList<>();  
    products.add("apple");  
    products.add("tomato");  
    products.add("bread");  
    products.add("water");  
  
    for (String product : products) {  
        if (product.equals("apple")) {  
            System.out.println("category: " + productCategories.get(0));  
        } else if (product.equals("tomato")) {  
            System.out.println("category: " + productCategories.get(1));  
        } else if (product.equals("bread")) {  
            System.out.println("category: " + productCategories.get(2));  
        } else {  
            assert false : "Unknown category for the product " + product;  
        }  
    }  
}
```

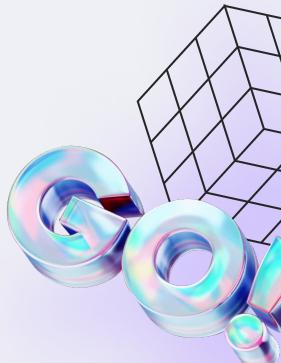


Вопрос

Какой будет результат выполнения следующего кода,
если утверждение не выполняется?

```
assert x > 0 : "x is negative";
```

- A. Код выполнится без ошибок
- B. Будет выброшено исключение AssertionError с сообщением "x is negative"
- C. Код просто прекратит выполнение
- D. В консоль будет выведено сообщение "x is negative"



20 секунд на размышление



Задание №6

Найдите и исправьте ошибку

```
public static void expectedValue() {  
    assertThat(5).isEqualTo(sum(2, 3));  
}
```



Задание №7

У вас есть следующий метод, который использует assert-выражения из библиотеки AssertJ для проверки содержания массива строк:

```
public static void testingJavaCollectionsAssertJ(String[] colors) {  
    assertThat(colors)  
        .isNotEmpty()  
        .hasSize(7)  
        .doesNotHaveDuplicates()  
        .contains("orange", "green", "violet")  
        .endsWith("gold")  
        .startsWith("aqua")  
        .containsSequence("yellow", "blue")  
        .doesNotContain("red", "black");  
  
    String[] colors = {"...", "...",};  
  
    testingJavaCollectionsAssertJ(colors);
```

Создайте массив строк colors, который удовлетворяет всем этим условиям.



Задание №8

Используйте заготовку класса Hero и напишите следующие проверки для объекта Hero с использованием assert:

1. Проверить, что герой создался с именем Emmett
2. Проверить, что значение прочности брони героя равно 50
3. Проверить, что у героя оружие типа sword
4. Проверить содержимое инвентаря героя (размер 3, содержимое "Bow", "Axe", "Gold", порядок не важен)
5. Проверить, что герой это человек (свойство isHuman)

```
public void checkingHero() {  
    List<String> heroBag = Arrays.asList("Bow", "Axe", "Gold");  
    Hero emmett = new Hero("Emmett", 50, "sword", heroBag, true);  
}
```



Домашнее задание





Домашнее задание

Задание 1.

В классе Calculator создайте метод calculateDiscount, который принимает сумму покупки и процент скидки и возвращает сумму с учетом скидки.

Ваша задача - проверить этот метод с использованием библиотеки AssertJ. Если метод calculateDiscount получает недопустимые аргументы, он должен выбрасывать исключение ArithmeticException.

Не забудьте написать тесты для проверки этого поведения.

```
public class Calculator {  
    ...  
    public static double calculatingDiscount(double purchaseAmount, int  
discountAmount) {  
        // purchaseAmount - сумма покупки  
        // discountAmount - размер скидки  
        return 0; // Метод должен возвращать сумму покупки со скидкой  
    }  
}
```



Домашнее задание

Задание 2. (*)

Мы хотим улучшить функциональность нашего интернет-магазина. Ваша задача - добавить два новых метода в класс Shop:

- Метод `sortProductsByPrice()`, который сортирует список продуктов по стоимости.
- Метод `getMostExpensiveProduct()`, который возвращает самый дорогой продукт.

Напишите тесты, чтобы проверить, что магазин хранит верный список продуктов (правильное количество продуктов, верное содержимое корзины).

Напишите тесты для проверки корректности работы метода `getMostExpensiveProduct`.

Напишите тесты для проверки корректности работы метода `sortProductsByPrice` (проверьте правильность сортировки).

Используйте класс `Product` для создания экземпляров продуктов и класс `Shop` для написания методов сортировки и тестов.



Подведем итоги

- 📌 Узнали, что такое тестирование
- 📌 Вспомнили про цели тестирования
- 📌 Разобрали, какие бывают ошибки
- 📌 Поговорили про принципы тестирования программного обеспечения и Цикл разработки
- 📌 Написали свой первый тест на Java





Спасибо за работу!

