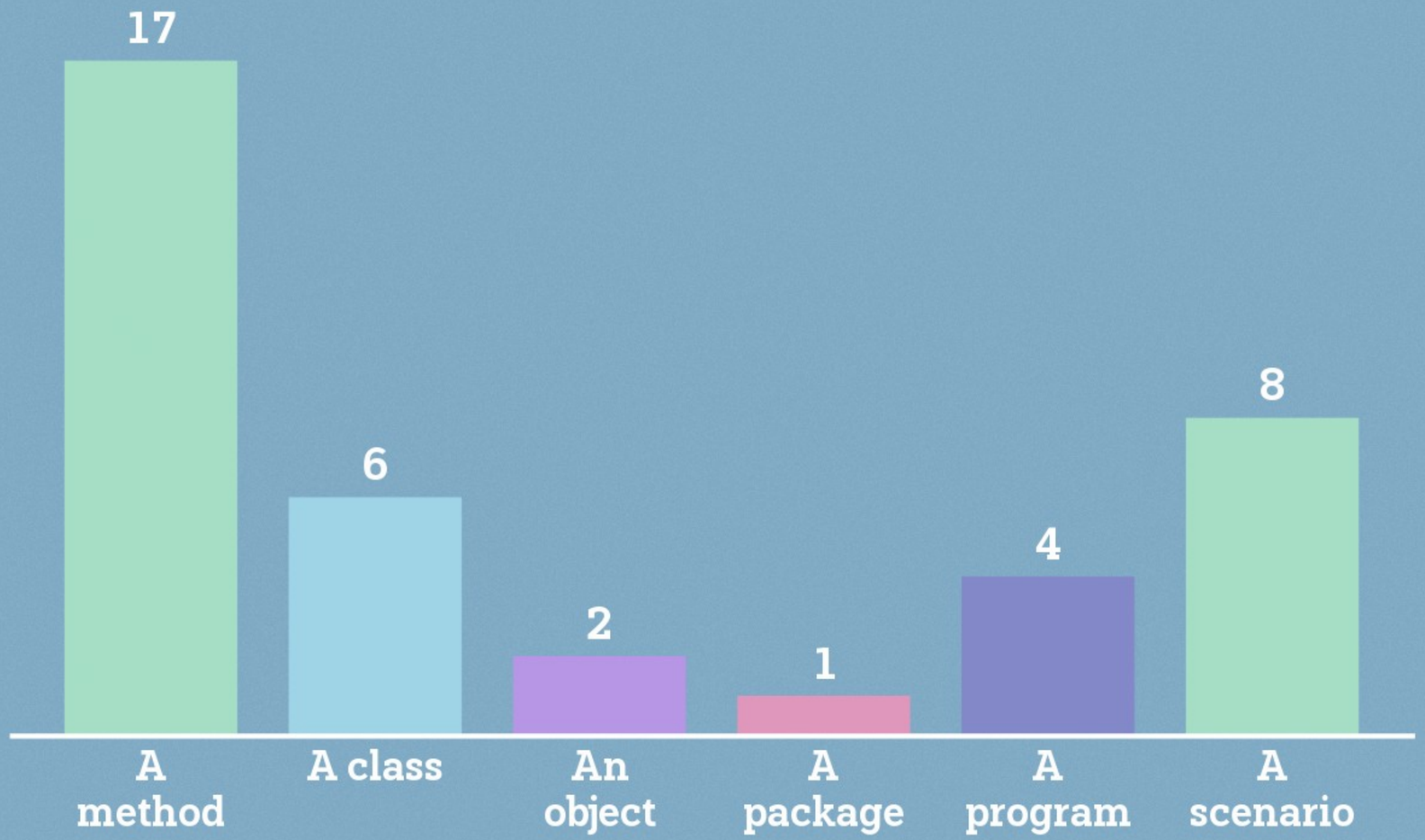
Ask me anything

Mentimeter

0 questions
0 upvotes

What is the “unit” tested by a Unit Test?

Mentimeter



19

What are the advantages of JUnit 5 over JUnit 3?

The "rules" got easier.

Hunit 5 is more flexible

use of @

Having anotations is easier to understand

simpler

Easier implementation

no need to have "test" in the name

tests dont need to have "test" in their names

we do not need to extend a test case, and do not need to import the framwork.TestCase

What are the advantages of JUnit 5 over JUnit 3?

JU5 has no longer relies on inheritanceto plug concrete tests into the framework and it use annotations for flag test Methodes

running before the first tes in the class is invoked

What would be boundary conditions for testing the paren matcher?

Mentimeter

an empty String

null

When there are more right brackets than left ones.

a String only with parent charachters

a to large String?

starting with a right bracket as a first char

if last char in the string is a left bracket

7

How would you design tests to pass through points (2) and (4)?

Mentimeter

```
public boolean parenMatch() {
  for (int i=0; i<line.length(); i++) {
    char c = line.charAt(i);
    if (isLeftParen(c)) {
      stack.push(matchingRightParen(c)); // (1)
    } else {
      if (isRightParen(c)) {
        if (stack.isEmpty()) {
          return false; // (2)
        }
        if (stack.top().equals(c)) {
          stack.pop(); // (3)
        } else {
          return false; // (4)
        }
      } // else not a paren char (5)
    }
  }
  return stack.isEmpty(); // (6)
}
```

To reach point (2), we must have just read a right parenthesis, but the stack must be empty.

To reach point (4), we must have read a right parenthesis, but it does not match the top of the stack.

a empty string(2)

")"

only a single right parantheses as input

String without any parenthesis

4

())

and for point 4 something like (}

12

How would you design tests to pass through points (2) and (4)?

Mentimeter

```
public boolean parenMatch() {
  for (int i=0; i<line.length(); i++) {
    char c = line.charAt(i);
    if (isLeftParen(c)) {
      stack.push(matchingRightParen(c)); // (1)
    } else {
      if (isRightParen(c)) {
        if (stack.isEmpty()) {
          return false; // (2)
        }
        if (stack.top().equals(c)) {
          stack.pop(); // (3)
        } else {
          return false; // (4)
        }
      } // else not a paren char (5)
    }
  }
  return stack.isEmpty(); // (6)
}
```

directly to 6

it would bring us to return stack.istEmpty()

an empty will bring to use 6

12

FizzBuzz TDD Coding Kata

Use TDD to write a class that outputs the numbers 1 to 100, but prints “Fizz” for multiple of 3, “Buzz” for multiples of 5, and “FizzBuzz” for multiples of 15.

What tests would you write for FizzBuzz? What are the boundary tests?

Mentimeter

if 15 is a fizzbuzz

assert 3 prints fizz

2+1 Fizz

0?

5 is buzz

A test Case for:
1,3(Fizz),5(Buzz),12(Fizz),15(fizzbuzz)
,30(fizzbuzz) and 101

7 is 7

45 (3*15)

0 should raise exception

21

What tests would you write for FizzBuzz? What are the boundary tests?

Mentimeter

0 could be fizzbuzz or throw an exception

at 0 we should become nothing, because its not in the boundary, or a exception

i mod 3 == 0 returns fizz

if(i%15 == 0) return Fizzbuzz;

i % 2 == 0

you could also save the lines to be printed before printing them, and the fizzbuzz method returns it.in the for loop listxy.add()

if(i%3==0 &&i%5==0){return "FizzBuzz"}

if(i%3 == 0 && i % 15 !=0) return "fuzz";

35

21

What tests would you write for FizzBuzz? What are the boundary tests?

Mentimeter

if (i/10 == 3 || i%10 == 3) return {Fizz}if (i/10 == 5 || i%10 == 5) return {Buzz}

53 --> buzzfizz?

if (Integer.parseInt(i).contains("3")...

21

FizzBuzz v2

Modify the program to print “Fizz” also for numbers containing “3” and “Buzz” for numbers containing “5”.

What additional tests would you write for FizzBuzz v2?

Mentimeter

13 should return buzz

*fizz

Is 35 fizzbuzz or buzz?

13 (Fizz) 51 (BUZZ) they will be false

*fizz : i wrote buzz instead of fizz. 13 should return fizz

53 -> buzzfizz?

if ((3/10 == 3) || (3%10 == 3)) {return (fizz)}

same with 5

we need to take a look at the chars.

12

What additional tests would you write for FizzBuzz v2?

Mentimeter

if i.toString.contains("3")

you can divide it by 3

51 should give Buzz because it containing "5"

12

Mentimeter

Last chance for questions