Ask me anything

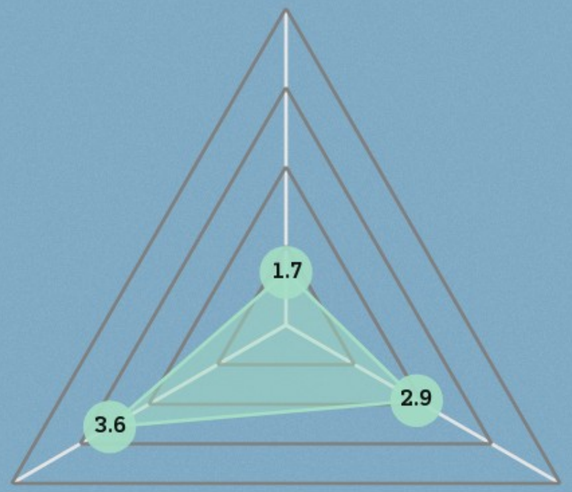
Mentimeter

0 questions
0 upvotes

My tests cover every line of code, and all the tests are green. Can I be sure there are no bugs?

Mentimeter

Yes. The code is correct.



Maybe. Tests can't cover all cases.

No. There are always bugs.

21

Should you write tests for private or protected methods? Why or why not?

Mentimeter

Private methods are called from public methods, so test the public methods that call these private methods. Protected methods should be tested as they are used in the subclasses aswell

1

Why are bugs common at boundaries?

Mentimeter

because sometime we forget trivial things like a +1 or a -1

tests special and unexpected user input, to see if the program can deal with those

Because they are special cases and are not always thought about when writing code

Most implementations are vulnerable to strange input

sometimes we count from 0 and we are sometimes not familiar with that

boundaries are edgecases which are sometimes hard to understand what really happens. Example when handling integer.MAX_VALUE, it could overflow to a negative integer

When writing the method we tend to forget about such input, as it's unusual.

7

What would be a suitable class invariant for ArrayStack?

Mentimeter

```
public class ArrayStack<T>
        implements StackInterface<T> {
    protected T store [];
    protected int capacity;
    protected int size;

    public ArrayStack() {
        store = null;    // default value
        capacity = 0;    // available slots
        size = 0;        // used slots
    }
```

size <= store	size > capacity	size <= store.size()
capacity>=0	capacitiy >=0	size >=0
size < capacity	return size>=0 && capacity >0;	size <= capacity

What would be a suitable class invariant for ArrayStack?

Mentimeter

```
public class ArrayStack<T>
        implements StackInterface<T> {
    protected T store [];
    protected int capacity;
    protected int size;

    public ArrayStack() {
        store = null;    // default value
        capacity = 0;    // available slots
        size = 0;        // used slots
    }
```

store.size == capacity

if not empty, store.size==capacity

store.size <= capacity

if (store == null && capacity = = 0) return trueelse if (store.size() == capacity) return true

0 <= size <= capacity

store.size==capacity || store.size==null

when you think the bug is in that method you should step into

How would you implement the ArrayStack.grow() method?

Mentimeter

Init new Array of Size 2x old ArrayCopy the elements using a for-loop

if you think that there is a problem in this line you should step intootherwise not

2

Demo — debugging ParenMatchTest with a broken LinkStack.pop()

Where should I set a breakpoint?

When should I step into rather than over code?

Mentimeter

if the problem occurs after that line

If a line calls multiple methods

when you suspect a bug or dont understand what that part of the code does

When you expect the bug to be close

if you need more information

when you don't know where the problem is and don't know what it do's

6

How do you explain the benchmarks?

Mentimeter

Popping from the ArrayStack is very fast as there isn't any pointer relocation

wrapperStack has more return statements

Stack Implementation	*10M pushes*	*10M pops*
p2.stack.LinkStack	909	98
p2.stack.ArrayStack	342	24
p2.stack.**WrappedStack**	816	529
java.util.Stack	253	175

2

Last chance for questions