

Niezbędnik programisty

Testy



Aleksandra Konarska

Nie mamy czasu pisać testów

Nie muszę pisać testów.
Wiem, że mój kod działa.

Kim jestem?

Programistka Javy @



Po co piszemy testy?

Żeby mieć pewność

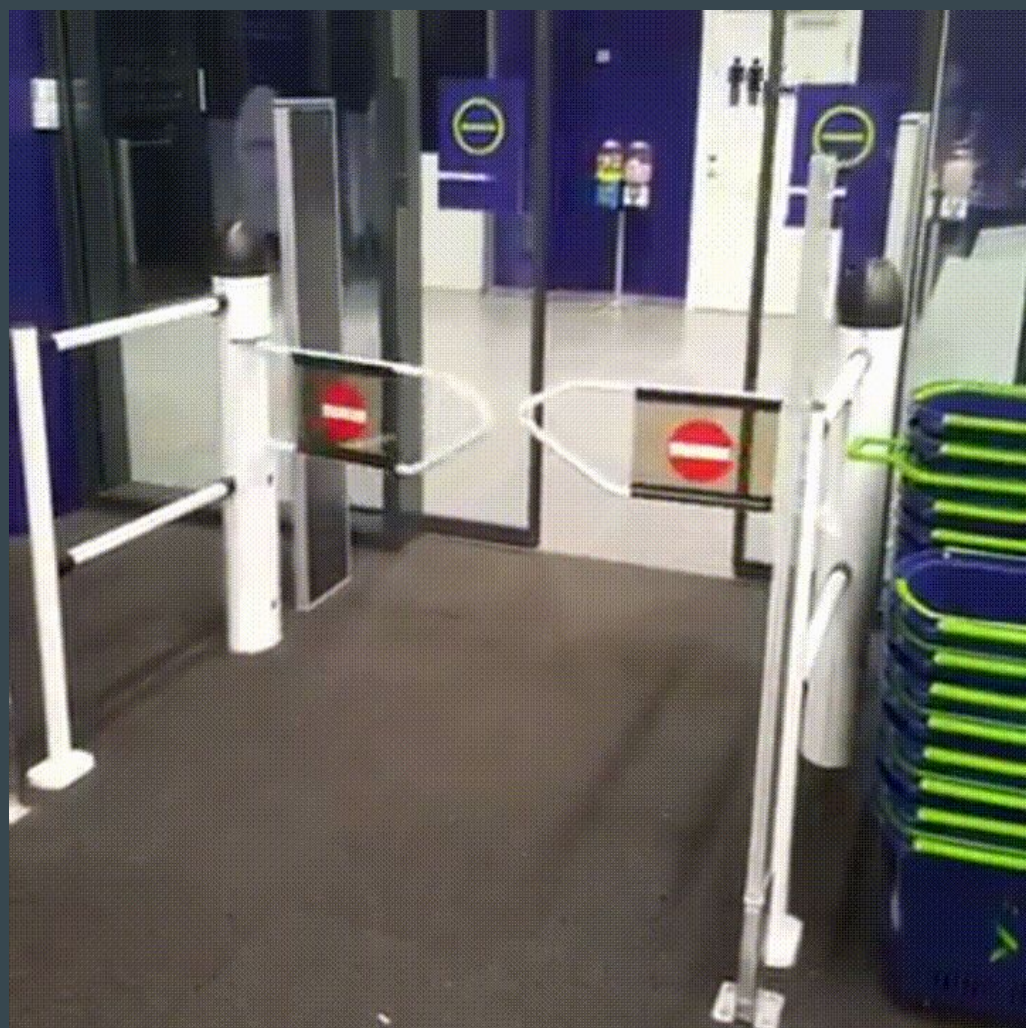
Rodzaje testów



Jednostka

Najmniejsza testowalna część aplikacji

W językach zorientowanych obiektowo najczęściej klasa lub metoda



Testy integracyjne

Testy akceptacyjne

Jak pisać testy?

Testy = Kod

Testy = Specyfikacja

F AST

I NDEPENDENT

R EPEATABLE

S ELF-CHECKING

T IMELY

Dobra nazwa

shouldThrowExceptionWhenArgumentIsNull

testFailToWithdrawMoneyIfAccountIsInvalid

WithdrawMoney_InvalidAccount_ExceptionThrown

When_InvalidAccount_Expect_WithdrawMoneyToFail

Given When Then

(Arrange Act Assert)

```
@Test
public void shouldReverseString() {
    // given
    String input = "abc";

    // when
    String result = Util.reverse(input);

    // then
    assertEquals(result, "cba");
}
```

Używaj najprostszycch możliwych wartości

```
@Test
public void shouldAddTwoNumbers() {
    // given
    int expectedSum = 87;

    // when
    int result = Sumator.sum(54, 33);

    // then
    assertEquals(result, expectedSum);
}
```

```
@Test
public void shouldAddTwoNumbers() {
    // given
    int expectedSum = 3;

    // when
    int result = Sumator.sum(1, 2);

    // then
    assertEquals(result, expectedSum);
}
```

Tylko jedna logiczna asercja

```
@Test
public void shouldSplitByComma() {
    // given
    String input = "a,b,c";
    List<String> expectedOutput = Arrays.asList("a", "b", "c");

    // when
    List<String> result = Util.split(input, ",");

    // then
    assertNotNull(result);
    assertFalse(result.isEmpty());
    assertEquals(result, expectedOutput);
}
```

```
@Test
public void shouldSplitByComma() {
    // given
    String input = "a,b,c";
    List<String> expectedOutput = Arrays.asList("a", "b", "c");

    // when
    List<String> result = Util.split(input, ",");

    // then
    assertEquals(result, expectedOutput);
}
```

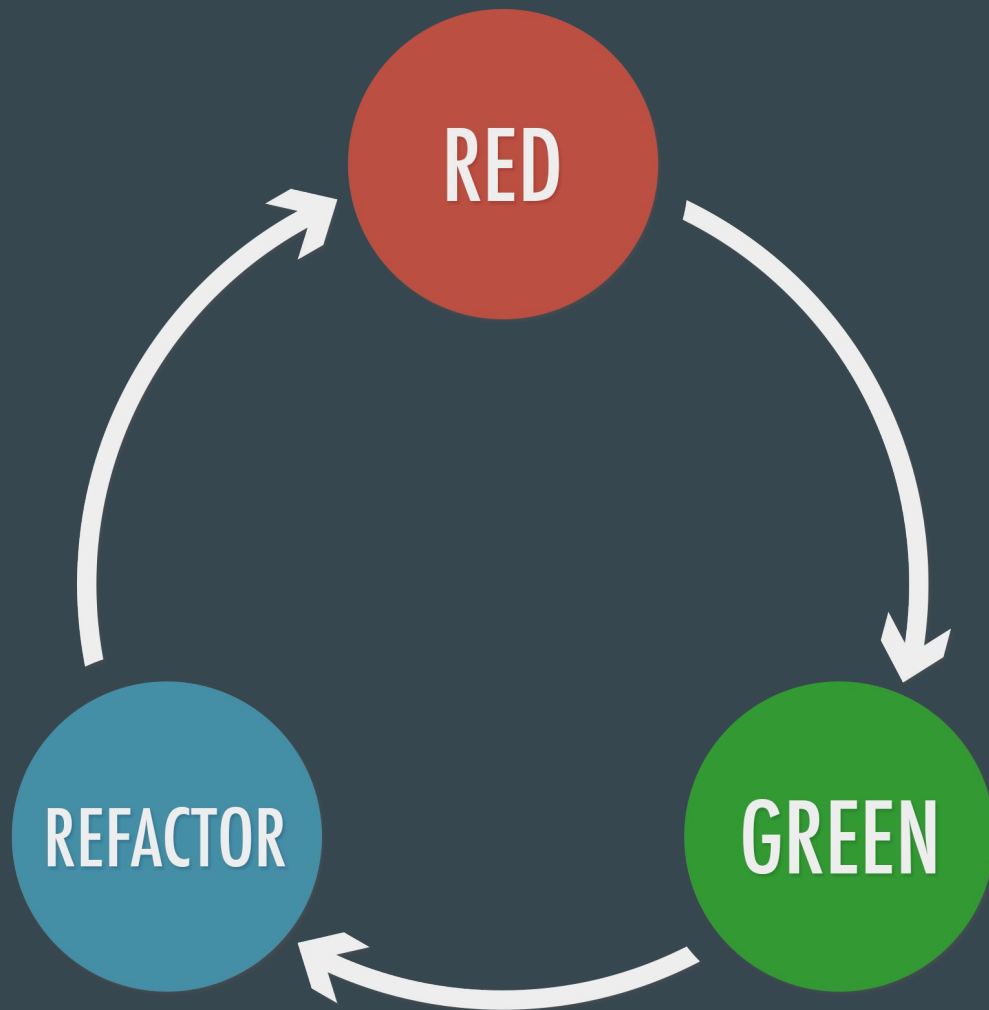
<https://github.com/amkonarska/niezbednik-programisty-testy>

TDD

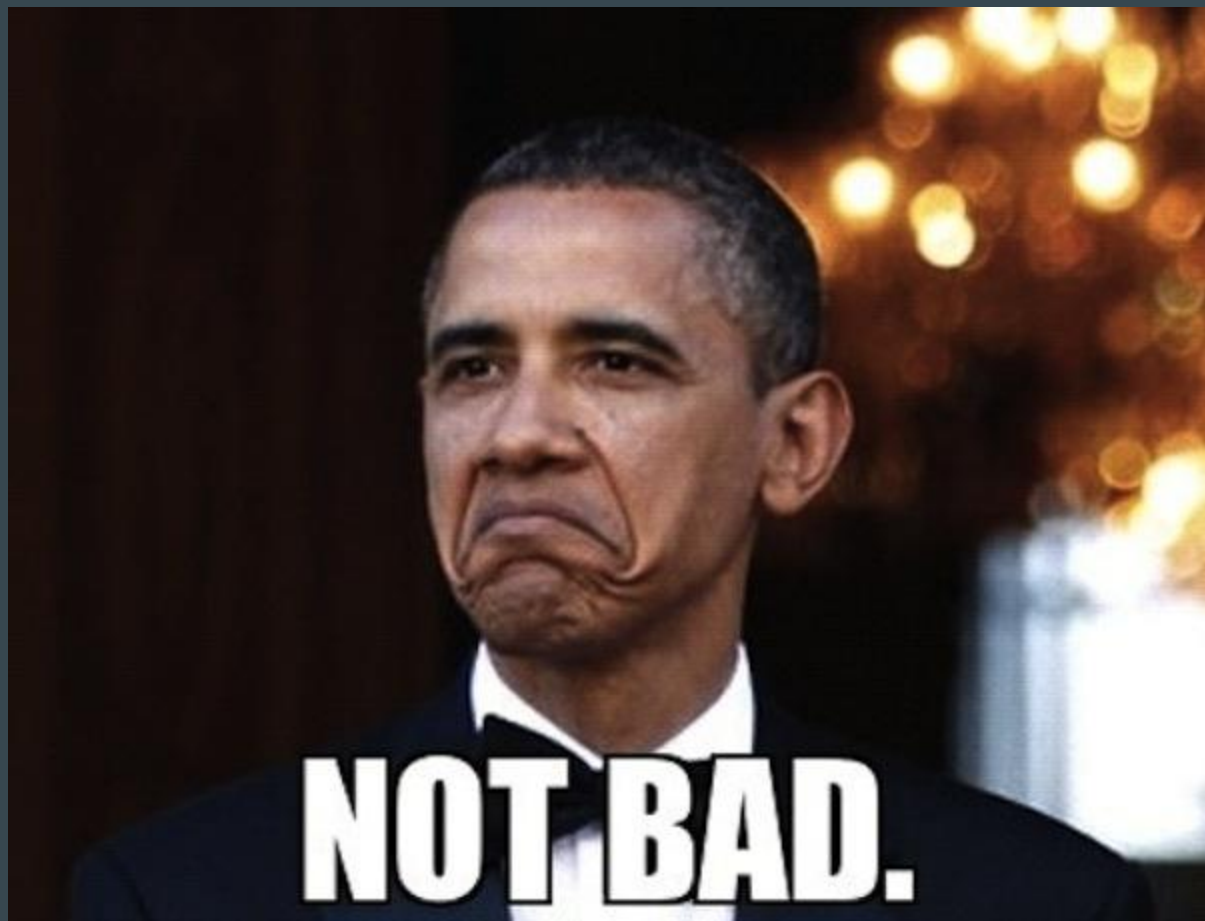
Test Driven Development

First make it work. Then make it right.
Then make it fast and small.

~Kent Beck



Po co TDD?



TDD nie wyleczy raka.

TDD nie przyniesie światu pokoju.

TDD nie ochroni Cię przed pijanymi kierowcami.

TDD nie uzdrowi amerykańskich wyborów.

~Robert C. Martin (Uncle Bob)

Jak?

Naprawianie błędów w stylu TDD

1. Napisz przewracający się test jednostkowy, który odtwarza błąd
2. Napraw błąd
3. Zobacz jak test przechodzi

Code kata

Ćwiczenie, które pomaga programistom doskonalić swoje umiejętności poprzez powtarzanie.

Bądźcie cierpliwi.



amkonarska



amkonarska



aleksandra.konarska@compactbi.com