SOFTWARE QUALITY & TESTING

Marija Grncharovska

ASSIGNMENT No 1

```
/*
 * if sets1 or set2 is null throw NullPointerException
 * else return a non-null set of elements that are in set1, but not in set2
 * else, if there are no such elements return null
 */

public Set setDifference(Set set1, Set set2) {
}
```

1. INPUT DOMAIN MODELING

Parameters : *set1*
set2

- Interface-based approach

	Characteristics	Blocks and values		
A	state of set1	*null*	*empty*	*non - empty*
B	state of set2	*null*	*empty*	*non - empty*

- Functionality-based approach

	Characteristics	Blocks and values			
A	existence of intersection	*True*	*False*		
B	being subset	$set1 \not\subset set2$ & $set2 \not\subset set1$	$set1 \subseteq set2$ & $set2 \not\subset set1$	$set1 \not\subset set2$ & $set2 \subseteq set1$	$set1 \subseteq set2$ & $set2 \subseteq set1$

2. Q&A

- Disjointness

partitions are disjoint

$b_i \cap b_j = \varnothing,\ \forall i \neq j,\ b_i, b_j, \in B_q$

- Completness

partitions are complete

$\cup b = D,\ b \in B_{q}$.

3. BASE CASE COVERAGE

3.1. INTERFACE – BASED APPROACH

	CHARACTERISTICS	*BLOCKS/VALUES*		
A	state set1	A1 [null]	A2 [empty]	A3 [non - empty]
B	state set2	B1 [null]	B2 [empty]	B3 [non - empty]

Base	**(A3, B3)**
Tests	(A2, B3)
	(A1, B3)
	(A3, B1)
	(A3, B2)
Nº of tests	**5**

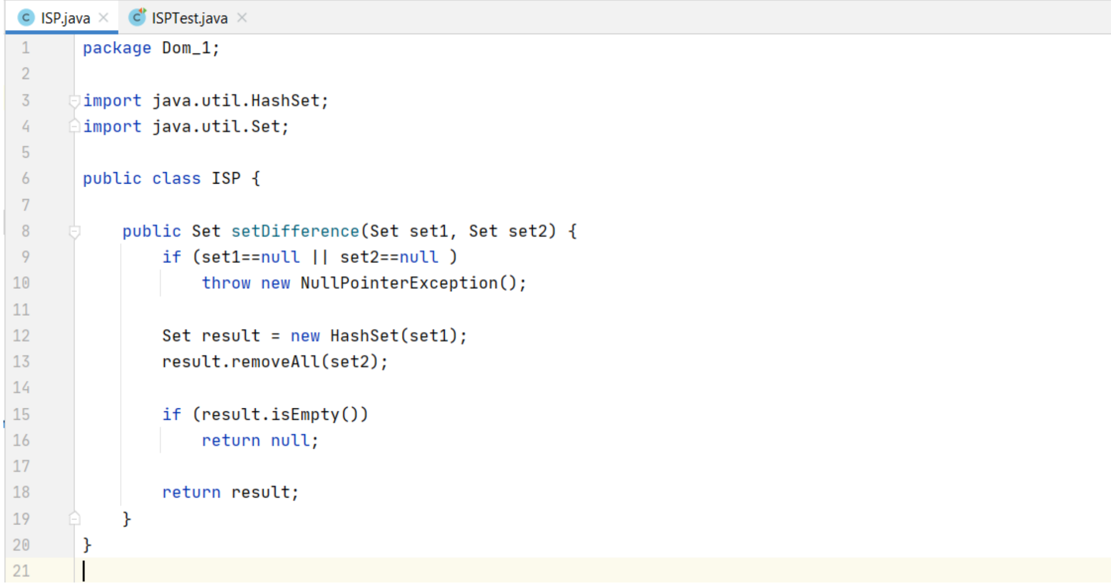
3.2. FUNCTIONALITY – BASED APPROACH

	CHARACTERISTICS	*BLOCKS/VALUES*			
A	existence of intersection	A1 = T	A2 = F		
B	being subset	B1 [none of them]	B2 [A⊆B & B⊄A]	B3 [B⊆A & A⊄B]	B4 [equal sets]

Base	**(A1, B1)**
Tests	(A2, B1)
	(A1, B2)
	(A1, B3)
	(A1, B4)
Nº of tests	**5**

4. jUnit TESTS

Function implementation:

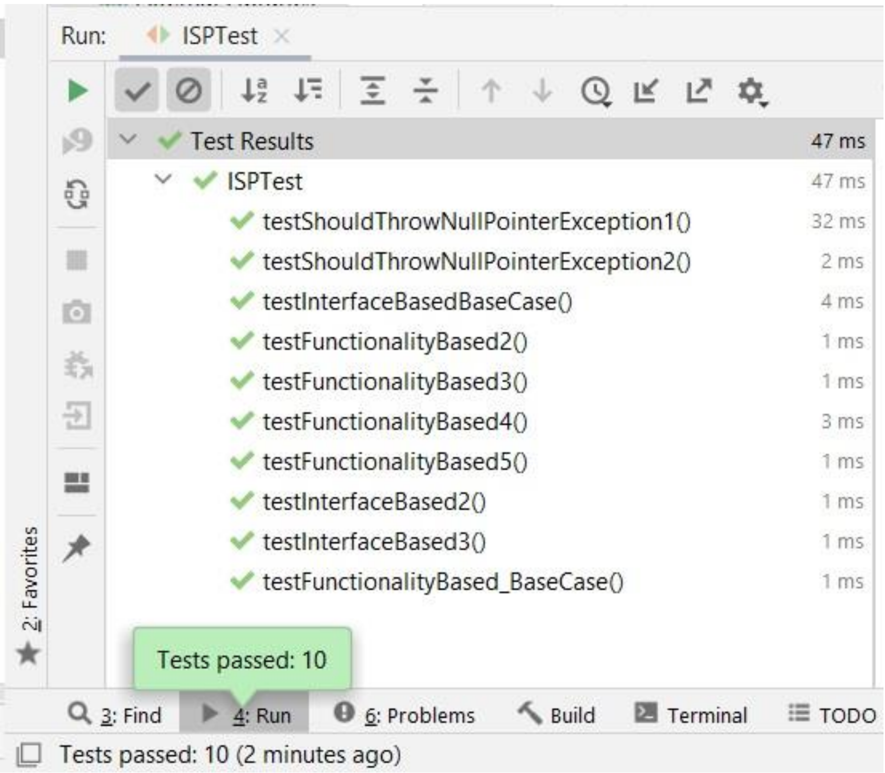


```java
package Dom_1;

import java.util.HashSet;
import java.util.Set;

public class ISP {

    public Set setDifference(Set set1, Set set2) {
        if (set1==null || set2==null )
            throw new NullPointerException();

        Set result = new HashSet(set1);
        result.removeAll(set2);

        if (result.isEmpty())
            return null;

        return result;
    }
}
```

Test cases:

	set1	*set2*
Interface - based	{ “str1” , 2 , 3 }	{ 3 , 4 , “str2” }
	∅	{ 3 , 4 , “str2” }
	null	{ 3 , 4 , “str2” }
	{ “str1” , 2 , 3 }	∅
	{ “str1” , 2 , 3 }	null
Functionality - based	{ 1 , 2 , 3 }	{ 3 , 4 , 5 }
	{ 1 , 2 , 3 }	{ 4 , 5 , 6 }
	{ 1 , 2 }	{ 1 , 2 , 3 }
	{ 1 , 2 , 3 }	{ 1 }
	{ 1 , 2 , 3 }	{ 1 , 2 , 3 }

5. TEST RESULTS



```
Run:  ISPTest
Test Results                                   47 ms
  ISPTest                                      47 ms
    testShouldThrowNullPointerException1()     32 ms
    testShouldThrowNullPointerException2()      2 ms
    testInterfaceBasedBaseCase()                4 ms
    testFunctionalityBased2()                   1 ms
    testFunctionalityBased3()                   1 ms
    testFunctionalityBased4()                   3 ms
    testFunctionalityBased5()                   1 ms
    testInterfaceBased2()                       1 ms
    testInterfaceBased3()                       1 ms
    testFunctionalityBased_BaseCase()           1 ms

Tests passed: 10
Tests passed: 10 (2 minutes ago)
```