

ÅBO AKADEMI UNIVERSITY

SOFTWARE TESTING

Assignment 2



FILIPE FELÍCIO (2004623)

MAY 3, 2021

Task 1

Design unit tests for CalculateReturningCoins()

1) Create the CFG of the method

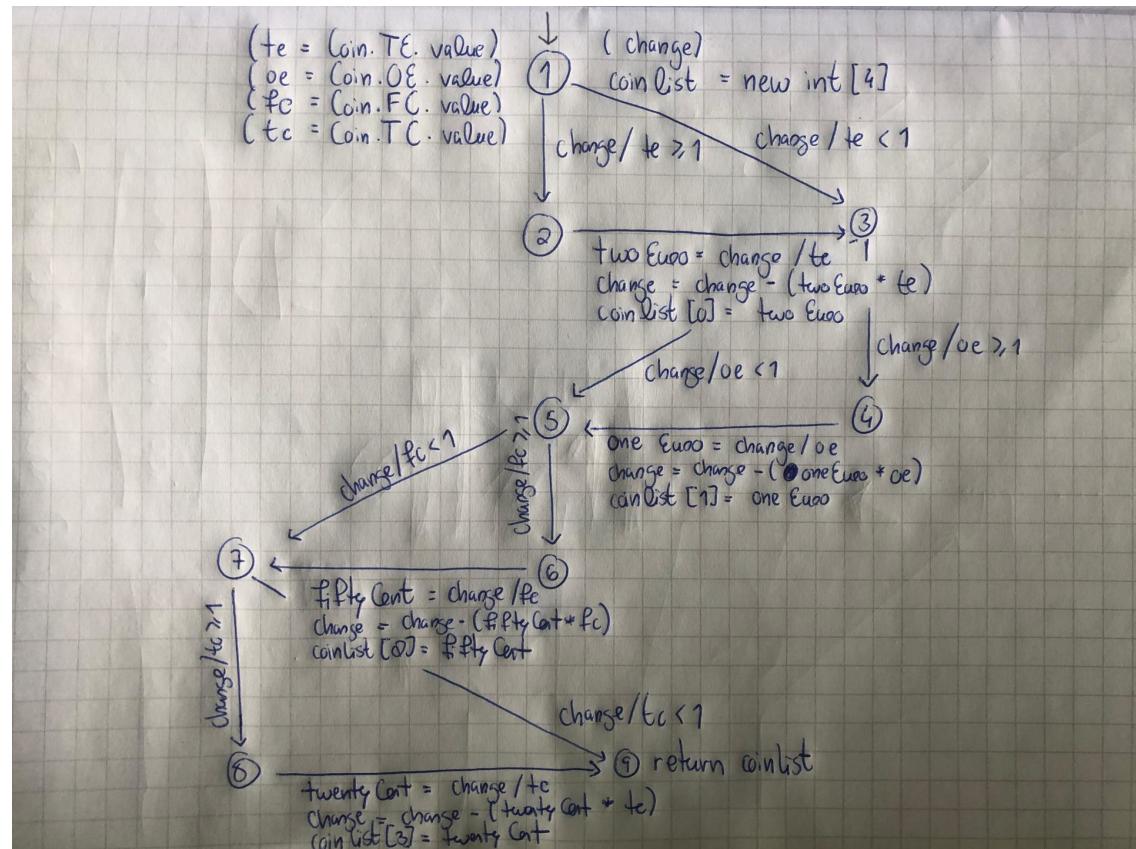


Figure 1: CalculateReturningCoins() CFG

2) Annotate CFG with defs and uses for variables change and coinList

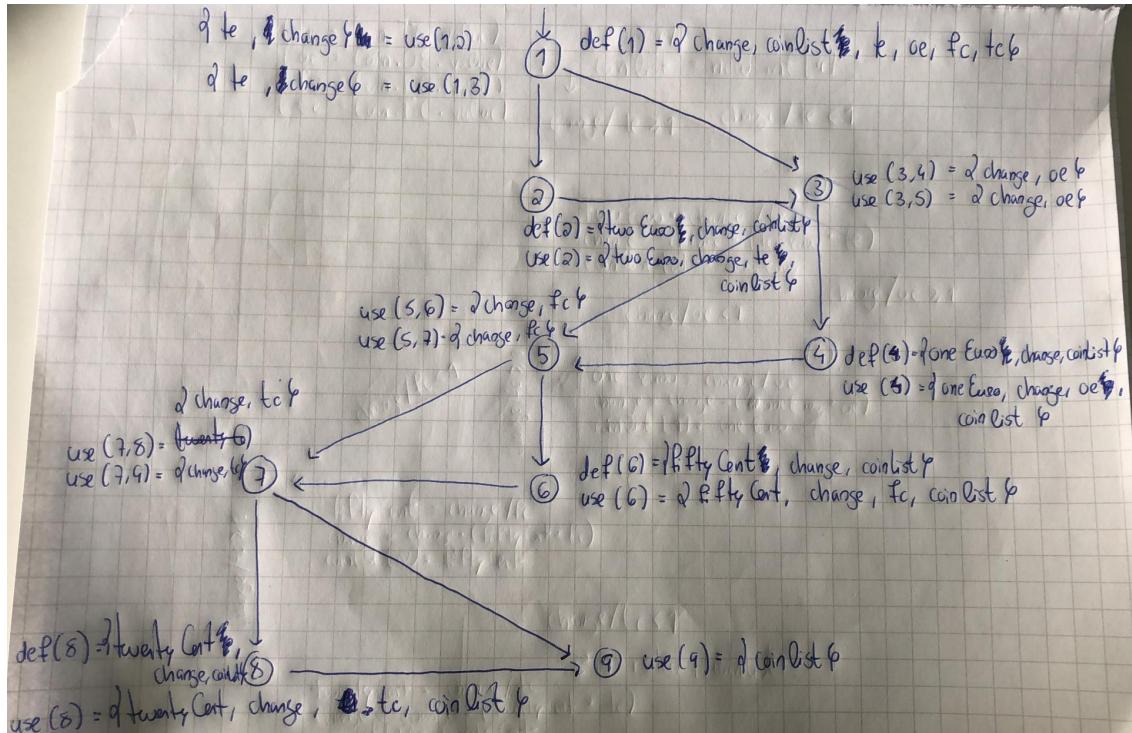


Figure 2: CalculateReturningCoins() CFG with defs and uses

Node's Defs and Uses Table

Node	Def	Use
1	{change, coinList, te, oe, fc, tc}	
2	{change, coinList, twoEuro}	{twoEuro, change, te, coinList}
3		
4	{change, coinList, oneEuro}	{oneEuro, change, oe, coinList}
5		
6	{change, coinList, fiftyCent}	{fiftyCent, change, fc, coinList}
7		
8	{change, coinList, twentyCent}	{twentyCent, change, tc, coinList}
9		{coinList}

Edge's Uses Table

Edge	Use
(1,2)	{te,change}
(1,3)	{te,change}
(2,3)	
(3,4)	{oe,change}
(3,5)	{oe,change}
(4,5)	
(5,6)	{fc,change}
(5,7)	{fc,change}
(6,7)	
(7,8)	{tc,change}
(7,9)	{tc,change}
(8,9)	

4) List Test requirements for Node, Edge and Edge pair coverage

NC: 1, 2, 3, 4, 5, 6, 7, 8, 9
EC: (1,2), (1,3), (2,3), (3,4), (3,5), (4,5),
 (5,6), (5,7), (6,7), (7,8), (7,9), (8,9)
 - 12 TR
EPC: (1,2,3), (1,3,4), (1,3,5), (2,3,4),
 (2,3,5), (3,4,5), (3,5,6), (3,5,7),
 (4,5,6), (4,5,7), (5,6,7), (5,7,8),
 (5,7,9), (6,7,8), (6,7,9), (7,8,9)

Figure 3: Test requirements for Node, Edge and Edge pair coverage

5) List test requirements for All-Defs coverage for variable change

<ul style="list-style-type: none"> • <u>change</u> - Defs: 1, 2, 4, 6, 8 - Used: 2, 4, 6, 8, $(1,2), (1,3), (3,4),$ $(3,5), (5,6), (5,7),$ $(7,8), (7,9)$ 	<u>Every def \rightarrow a use, one path</u> $1 \rightarrow 4: 1, 2, 3, 4$ $2 \rightarrow 6: 2, 3, 4, 5, 6$ $4 \rightarrow 8: 4, 5, 6, 7, 8$ $6 \rightarrow 8: 6, 7, 8$ $8 \rightarrow 8: 8$
--	--

Figure 4: test requirements for All-Defs coverage for variable change

6) List Test requirements for All Use coverage for variable coinList

<ul style="list-style-type: none"> • <u>coinList</u> - Defs: 1, 2, 4, 6, 8 - Used: 2, 4, 6, 8, 9 	<u>Every def \rightarrow Every use, one path</u> $1 \rightarrow 2: 1, 2$ $2 \rightarrow 2: 2$ $4 \rightarrow 6: 4, 5, 6$ $8 \rightarrow 8: 8$ $1 \rightarrow 4: 1, 2, 3, 4$ $2 \rightarrow 4: 2, 3, 4$ $4 \rightarrow 8: 4, 5, 7, 8$ $1 \rightarrow 6: 1, 3, 5, 6$ $2 \rightarrow 6: 2, 3, 5, 6$ $4 \rightarrow 9: 4, 5, 7, 9$ $1 \rightarrow 8: 1, 3, 5, 7, 8$ $2 \rightarrow 8: 2, 3, 5, 7, 8$ $6 \rightarrow 8: 6, 7, 8$ $1 \rightarrow 9: 1, 3, 5, 7, 9$ $2 \rightarrow 9: 2, 3, 5, 7, 8$ $6 \rightarrow 9: 6, 7, 9$
---	--

Figure 5: Test requirements for All Use coverage for variable coinList

8) Create test paths to cover all test requirements specified above (first remove duplicated test requirements)

• <u>Node Coverage</u>
$[1, 2, 3, 4, 5, 6, 7, 8, 9] \rightarrow 9/9$
• <u>Edge Coverage</u>
$[1, 3, 5, 7, 9] \rightarrow 4/12 \quad \{ (1,3), (3,5), (5,7), (7,9) \}$
$[1, 2, 3, 4, 5, 6, 7, 8, 9] \rightarrow 8/12 \quad \{ (1,2), (2,3), (3,4), (4,5), (5,6), (6,7), (7,8), (8,9) \}$
• <u>Edge - Pair Coverage</u>
$[1, 2, 3, 4, 5, 6, 7, 8, 9] \rightarrow 6/16 \quad \{ (1,2,3), (2,3,4), (4,5,6), (5,6,7), (6,7,8), (7,8,9) \}$
$[1, 3, 4, 5, 7, 9] \rightarrow 4/16 \quad \{ (1,3,4), (3,4,5), (4,5,7), (5,7,9) \}$
$[1, 2, 3, 5, 6, 7, 9] \rightarrow 3/16 \quad \{ (1,2,3), (2,3,5), (3,5,6), (6,7,9) \}$
$[1, 3, 5, 7, 8] \rightarrow 1/16 \quad \{ (1,3,5), (3,5,7), (5,7,8) \}$

Figure 6: Test paths to cover all test requirements

9) Create test inputs for each test path and assign the expected output

• Test Paths	Input	Output
NC $[1, 2, 3, 4, 5, 6, 7, 8, 9]$	5, 7	[0, 1, 1, 1]
EC $[1, 3, 5, 7, 9]$	0	[0, 0, 0, 0]
$[1, 2, 3, 4, 5, 6, 7, 8, 9]$	5, 7	[0, 1, 1, 1]
EPC $[1, 2, 3, 4, 5, 6, 7, 8, 9]$	3, 7	[0, 1, 1, 1]
$[1, 3, 4, 5, 7, 9]$	1	[0, 1, 0, 0]
$[1, 2, 3, 5, 6, 7, 9]$	2, 5	[1, 0, 1, 0]
$[1, 3, 5, 7, 8]$	0, 0	[0, 0, 0, 1]

Figure 7: Test inputs for each test path and the expected output

Task 2

Design integration tests for displayReturningCoins (double change) and calculateReturningCoins(double change) methods

- 1) Create CFG for displayReturningCoins(double change) and annotate it with last defs and first uses for the call and return parameters

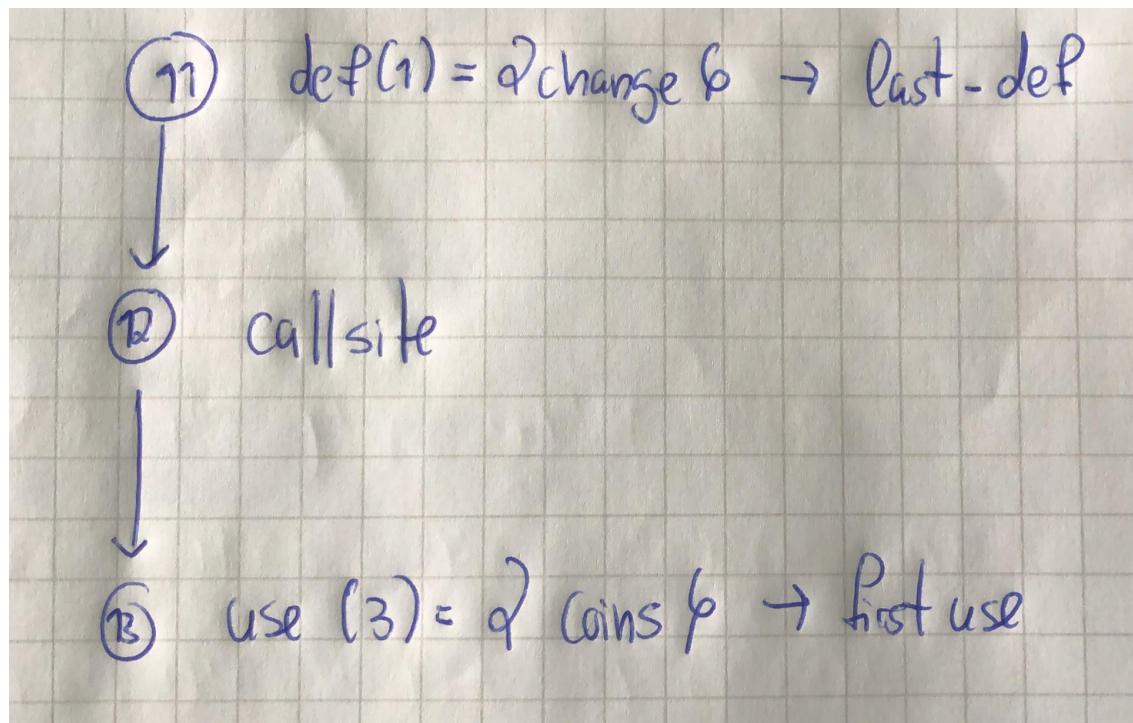


Figure 8: CFG last defs and first uses

2) Annotate the CFG of calculateReturningCoins(double change) defined in Task 1 with last defs and first uses for call parameters

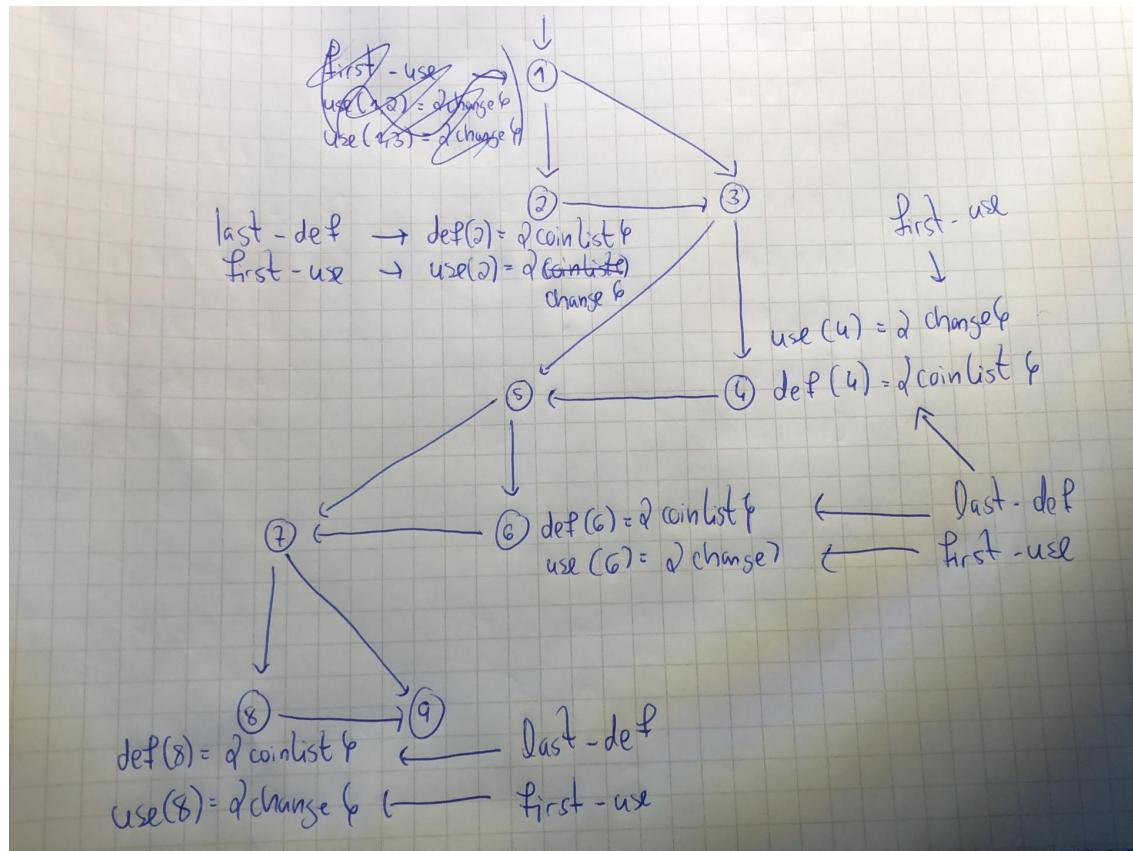


Figure 9: CFG last defs and first uses

3) List test requirements for All-Coupling-Defs coverage for the direct call

- All - Coupling - Defs
Cover a path from every last def to at least a first use
 $(11, \text{change}) \rightarrow (8, \text{change}) : \cancel{(1, 2, 3, 4, 5, 6, 7)} [1, 3, 5, 7, 8]$
~~1, 2, 3, 4, 5, 6, 7~~
 $(2, \text{coinlist}) \rightarrow (13, \text{coins}) : [2, 3, 5, 7, 9, 13]$
 $(4, \text{coinlist}) \rightarrow (13, \text{coins}) : [4, 5, 7, 9, 13]$
 $(6, \text{coinlist}) \rightarrow (13, \text{coins}) : [6, 7, 9, 13]$
 $(8, \text{coinlist}) \rightarrow (13, \text{coins}) : [8, 9, 13]$

Figure 10: Test requirements for All-Coupling-Defs coverage for the direct call

4) List test requirements for All-Coupling-Use coverage for the return

• <u>All-Coupling-Use</u>	Cover a path every las- def to every first-use
(11, change → 2, change): [11, 12, 1, 2]	(6, coinlist → 8, change): [6, 7, 8]
(11, change → 4, change): [11, 12, 1, 3, 4]	(6, coinlist → 13, coins): [6, 7, 9, 13]
(11, change → 6, change): [11, 12, 1, 3, 5, 6]	(8, coinlist → 13, coins): [8, 9, 13]
(11, change → 8, change): [11, 12, 1, 3, 5, 7, 8]	
(11, change → 13, coins): [11, 12, 1, 3, 5, 7, 9, 13]	
(2, coinlist → 4, change): [2, 3, 4]	
(2, coinlist → 6, change): [2, 3, 5, 6]	
(2, coinlist → 8, change): [2, 3, 5, 7, 8]	
(2, coinlist → 13, coins): [2, 3, 5, 7, 9, 13]	
(4, coinlist → 6, change): [4, 5, 6]	
(4, coinlist → 8, change): [4, 5, 7, 8]	
(4, coinlist → 13, coins): [4, 5, 7, 9, 13]	

Figure 11: Test requirements for All-Coupling-Use coverage for the return

5) List test paths to satisfy all test requirements above

In the previous question I already added the paths, so I will just use the same figure:

• <u>All-Couplings - Use</u>	Cover a path every las-def to every first-use
(11, change → 2, change): [11, 12, 1, 2]	(6, coinlist → 8, change): [6, 7, 8]
(11, change → 4, change): [11, 12, 1, 3, 4]	(6, coinlist → 13, coins): [6, 7, 9, 13]
(11, change → 6, change): [11, 12, 1, 3, 5, 6]	(8, coinlist → 13, coins):
(11, change → 8, change): [11, 12, 1, 3, 5, 7, 8]	[8, 9, 13]
(11, change → 13, coins): [11, 12, 1, 3, 5, 7, 9, 13]	
(2, coinlist → 4, change): [2, 3, 4]	
(2, coinlist → 6, change): [2, 3, 5, 6]	
(2, coinlist → 8, change): [2, 3, 5, 7, 8]	
(2, coinlist → 13, coins): [2, 3, 5, 7, 9, 13]	
(4, coinlist → 6, change): [4, 5, 6]	
(4, coinlist → 8, change): [4, 5, 7, 8]	
(4, coinlist → 13, coins): [4, 5, 7, 9, 13]	

Figure 12: Test paths to satisfy all test requirements above

6) Create test inputs for each test path and the expected output

• Test Paths	Input	Output	Expected
[11, 12, 1, 2]	2	Your change is	1x 2Euro 0x 1Euro 0x 50 Cent
[11, 12, 1, 3, 4]	1	" "	0 m 1 m 0 0x20 Cent
[11, 12, 1, 3, 5, 0]	0,5	" "	0 " 0 " 1 " 0 "
[11, 12, 1, 3, 5, 17, 8]	0,2	" "	0 " 0 " 0 " 1 "
Paths			
[2, 3, 4]	3	" "	1 " 1 " 0 " 0 "
[2, 3, 5, 6]	3,5	" "	1 " 1 " 1 " 0 "
[2, 3, 5, 7, 8]	3,7	" "	1 " 1 " 2 " 1 "
[2, 3, 5, 7, 9, 13]	6	" "	3 " 0 " 0 " 0 "
[4, 5, 6]	7,5	" "	3 " 1 " 1 " 0 "
[4, 5, 7, 8]	1,2	" "	0 " 1 " 0 " 1 "
[6, 5, 7, 9, 13]	7	" "	3 " 1 " 0 " 0 "
[6, 7, 8]	0,7	" "	0 " 0 " 1 " 1 "
[6, 7, 9, 13]	0,5	" "	0 " 0 " 1 " 0 "
[8, 9, 13]	0,2	" "	0 " 0 " 0 " 1 "

Figure 13: Test inputs for each test path and the expected output

Task 3

```
Run: Task1
Task1
  ✓ Tests passed: 3 of 3 tests - 4 ms
    /home/felicio/jds/adopt-openjdk-14.0.2/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:/snap/intelliJ-idea-ultimate/205/lib/idea_rt.jar=40461:/snap/intelliJ-idea-ultimate/205/lib/idea_rt.jar
      ✓ testEdgesPairs
      ✓ testEdges
      ✓ testNodes
      Process finished with exit code 0

Tests passed: 3
```

```
Run: Task1
Task1
  ✓ Tests passed: 3 of 3 tests - 5 ms
    /home/felicio/jds/adopt-openjdk-14.0.2/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:/snap/intelliJ-idea-ultimate/205/lib/idea_rt.jar=40795:/snap/intelliJ-idea-ultimate/205/lib/idea_rt.jar
      ✓ testEdgesPairs
      ✓ testEdges
      ✓ testNodes
      Process finished with exit code 0

Tests passed: 3
```

