Auditoria, Calidad y Gestion de Sistemas software

ACG

Cambio testing

Pasos

1. Identificar clases validas e invalidas
   - Distinción a groso modo
2. Refinar clases validas
   - Para probar todas las monedas
3. Definir criterio de cobertura
4. Definir casos de test

1. Identify valid and invalid parts

- What can be inserted in the fields for *p* and *d*?
  - Integers seem a good option, just as floating point numbers like 23.95.
  - Can we insert negative numbers?
  - Can we insert, for instance, 09, or 00123, or 238746293478356?
  - What happens when we insert something that is not a number?

ANSWER: it is not possible to insert anything other than a non-negative integer in the range 0 . . . 99999999 for p, with the possibility of leading zeros. The same holds for d. Moreover, all inputs that contain something other than characters from 0 . . . 9 are treated the same way.

Invalid Parts

- Invalid Parts

input	part		comment
	ID	values	
p	iP_1	$p \notin \mathbb{N}$	FM "Character not allowed"
	iP_2	$p \in \mathbb{N}, p > 99999999$	impossible
d	iP_3	$d \notin \mathbb{N}$	FM "Character not allowed"
	iP_4	$d \in \mathbb{N}, d > 99999999$	impossible

Invalid Parts

- Invalid Parts: test cases

TestCase	p	d	part
1	3,5	2	iP1,vP2
2	1000000000		iP2,vP2
3	5	Hola	vP1,iP3
4	5	1000000000	VP1,iP4

Valid Parts I

input	part		comment
	ID	values	
p	vP_1	$\{0,\ldots,99999999\}$	valid
d	vP_2	$\{0,\ldots,99999999\}$	valid
$d-p$	vP_3	< 0	valid, no change *
	vP_4	$= 0$	valid, no change
	vP_5	> 0	valid, specific change

Valid Parts I

input	part		comment
	ID	values	
p	vP_1	$\{0,\ldots,99999999\}$	valid
d	vP_2	$\{0,\ldots,99999999\}$	valid
$d-p$	vP_3	<0	valid, no change *
	vP_4	$=0$	valid, no change
	vP_5	>0	valid, specific change

* Strictly speaking, this is correct, if the question is just "how much change do we get?". But is that really what the specification says? We need to check if according to specification

Valid Parts II

input	part		comment
	ID	values	
p	vP1	{0, . . . , 99999999}	Valid
d	vP2	{0, . . . , 99999999}	Valid
d-p	vP3	<0	valid, no change
	vP4	{0}	valid, no change
	vP5	{1,...,4}	valid, 1-euro coin plus rest
	vP6	{5,...,9}	valid, 5-euro note plus rest
	vP7	{10,...,19}	valid, 10-euro note plus rest
	vP8	{20,...,49}	valid, 20-euro note plus rest
	vP9	{50,...,99}	valid, 50-euro note plus rest
	vP10	>=100	valid, 100-euro note plus rest

Selecting A Criterion

- EC
  - 8 text cases
- AC
  - 2^8-1

Testing Cases EC coverage

- vP1, vP2, vP3
- vP1, vP2, vP4
- vP1, vP2, vP5
- vP1, vP2, vP6
- vP1, vP2, vP7
- vP1, vP2, vP8
- vP1, vP2, vP9
- vP1, vP2, vP10

Testing Cases EC coverage

test case	p	d	d-p	expected outcomes						cover parts
				100	50	20	10	5	1	
1	0	0	0	0	0	0	0	0	0	vP1, vP2, vP3
2	15	0	-15	0	0	0	0	0	0	vP1, vP2, vP4
3	14	15	1	0	0	0	0	0	1	vP1, vP2, vP5
4	15	21	6	0	0	0	0	1	2	vP1, vP2, vP6
5	54	67	13	0	0	0	1	0	3	vP1, vP2, vP7
6	25	49	24	0	0	1	0	0	4	vP1, vP2, vP8
7	71	146	75	0	1	1	0	1	0	vP1, vP2, vP9
8	153	300	147	1	0	2	0	1	2	vP1, vP2, vP10

Testing Cases EC coverage

test case	p	d	d-p	expected outcomes						cover parts
				100	50	20	10	5	1	
1	0	0	0	0	0	0	0	0	0	vP1, vP2, vP3
2	15	0	-15	0	0	0	0	0	0	vP1, vP2, vP4
3	14	15	1	0	0	0	0	0	1	vP1, vP2, vP5
4	15	21	6	0	0	0	0	1	2	vP1, vP2, vP6
5	54	67	13	0	0	0	1	0	3	vP1, vP2, vP7
6	25	49	24	0	0	1	0	0	4	vP1, vP2, vP8
7	71	146	75	0	0	1	0	1	0	vP1, vP2, vP9
8	153	300	147	1	0	2	0	1	2	vP1, vP2, vP10

This column can contain zeros, ones and twos. Our test suite includes all of these possibilities at least once

Testing Cases EC coverage

test case	p	d	d-p	expected outcomes						cover parts
				100	50	20	10	5	1	
1	0	0	0	0	0	0	0	0	0	vP1, vP2, vP3
2	15	0	-15	0	0	0	0	0	0	vP1, vP2, vP4
3	14	15	1	0	0	0	0	0	1	vP1, vP2, vP5
4	15	21	6	0	0	0	0	1	2	vP1, vP2, vP6
5	54	67	13	0	0	0	1	0	3	vP1, vP2, vP7
6	25	49	24	0	0	1	0	0	4	vP1, vP2, vP8
7	71	146	75	0	1	1	0	1	0	vP1, vP2, vP9
8	153	300	147	1	0	2	0	1	2	vP1, vP2, vP10

This column can contain zeros, ones, twos, threes and fours. Our test suite includes all of these possibilities at least once

Testing Cases EC coverage

test case	p	d	d-p	expected outcomes						cover parts
				100	50	20	10	5	1	
1	0	0	0	0	0	0	0	0	0	vP1, vP2, vP3
2	15	0	-15	0	0	0	0	0	0	vP1, vP2, vP4
3	14	15	1	0	0	0	0	0	1	vP1, vP2, vP5
4	15	21	6	0	0	0	0	1	2	vP1, vP2, vP6
5	54	67	13	0	0	0	1	0	3	vP1, vP2, vP7
6	25	49	24	0	0	1	0	0	4	vP1, vP2, vP8
7	71	146	75	0	1	1	0	1	0	vP1, vP2, vP9
8	153	300	147	1	0	2	0	1	2	vP1, vP2, vP10

The 100 euros note column can hold any non-negative integer, and again we probably need to try more values here

Testing Cases EC coverage

				expected outcomes						
test case	p	d	d-p	100	50	20	10	5	1	cover parts
1	0	0	0	0	0	0	0	0	0	vP1, vP2, vP3
2	15	0	-15	0	0	0	0	0	0	vP1, vP2, vP4
3	14	15	1	0	0	0	0	0	1	vP1, vP2, vP5
4	15	21	6	0	0	0	0	1	2	vP1, vP2, vP6
5	54	67	13	0	0	0	1	0	3	vP1, vP2, vP7
6	25	49	24	0	0	1	0	0	4	vP1, vP2, vP8
7	71	146	75	0	1	1	0	1	0	vP1, vP2, vP9
8	153	300	147	1	0	2	0	1	2	vP1, vP2, vP10
9	3	1005	1002	10	0	0	0	0	2	vP1, vP2, vP10
10	12	400	388	3	1	2	1	1	3	vP1, vP2, vP10
11	1	1000	999	9	5	2	0	1	4	vP1, vP2, vP10

It seems like a good idea to make sure that d - p ends in as many different digits as possible (in the case that d - p > 0), because the implementation probably uses the modulus or remainder operator to calculate how to give back the change