

Functional testing - Practice

For the latest version, please check the Moodle course „Software Quality (Tarkvara kvaliteet)” (or tepandi.ee)

1. Credit card charges

1. If the charge is counted, it is 0.5% from the purchase
2. Price: omin...omax
3. Owners: VIP, non-VIP. VIP does not pay charge
4. Cards:
 - VIP: Diamond
 - non-VIP: Gold (no charge), Classic
5. Location: local (no charge), abroad

Practice: Design functional tests

- 1) Equivalence classes
- 2) Test data (+ boundary)
- 3) Tests
- 4) Specification modifications?

Example (no boundaries, first cases, omin=20, omax=100000, outputs in red).

Equivalence classes and test data (no boundary values given):

	Price	Owner	Card	Location	Charge %	Charge
Equip classes	<0 0...19 20...100000 >100000	VIP n-VIP	Diamond Gold Classic	local abroad	0.5% 0% Error msg	Value Error msg
Test data	-5, 10, 1000, 150000	VIP n-VIP	Diamond Gold Classic	local abroad	0.5% 0% Error msg	Value Error msg

Tests (selection - please complete the table):

Test 1	-5	VIP	Diamond	local	Error msg	Error msg
...
Test k	1000	n-VIP	Classic	abroad	0.5%	5
...

What is the minimum number of tests needed to cover all equivalence classes?

It is desirable to cover all classes with a minimum number of tests. How many new (yet untested) classes are included in each test?

2. Saving and executing tests

1. Create a program that computes the final grade based on three inputs: grade points (GP) for Lab4; GP for semester activities; GP from the exam. Computation algorithm is based on arrangement from [http://tepandi.ee/Jaak Tepandi Sw QS org.pdf](http://tepandi.ee/Jaak_Tepandi_Sw_QS_org.pdf).
2. Write tests based on equivalence classes and boundary values.
3. Store the tests in a testing framework for your selected language and execute them. Example:
 - 3.1. The program is given as a Python function “grade_f” in file “grade_f_m.py”
 - 3.2. Use Python unittest to store the tests (an example is given on [http://tepandi.ee/SQ pract white-box.pdf](http://tepandi.ee/SQ_pract_white-box.pdf))
 - 3.3. Use “from grade_f_m import grade_f” to import the function and test it.

2. Testing a database form

Company name: Saldējums

Country: Latvia

<u>Order ID</u>	<u>Order date</u>	<u>Agreed date</u>	<u>Delivery date</u>
10643	25-sept-00	23-okt-00	03-okt-00
10692	03-nov-00	01-dets-00	13-nov-00
10702	13-nov-00	25-dets-00	21-nov-00
10835	15-veebr-01	14-marts-01	21-mai-01
10952	15-apr-01	27-mai-01	23-apr-01
11011	09-mai-01	06-juuni-01	13-mai-01

<u>Product</u>	<u>Unit price</u>	<u>Volume</u>	<u>Discount</u>	<u>Total</u>
<u>Del. jogurt</u>	1,20 eur	2	25,00%	1,80 eur
<u>Viljandi sai</u>	1,80 eur	5	20,00%	7,20 eur
<u>Eesti juust</u>	4,56 eur	15	25,00%	51,30 eur

Practice: Present requirements and design functional tests

- Select some elements of the form
- Present requirements to the selected elements
- Design functional tests using equivalence partitioning and boundary values

Discussion

- Is it possible to test without the requirements? To which extent?
- Minimum number of tests?

3. Use case based testing

Select three use cases from your project. Select criterion for testing.

Design tests. Possible criteria:

- At least one test for each use case primary scenario
- At least one test for each primary and alternative scenario
- Equivalence partitioning for use case inputs/outputs
- ...

4. Decision table testing

Compose a decision table for a task in your project (or select one, eg Fig 2 from <http://www.ibm.com/developerworks/rational/library/jun06/vauthier/>)

Design tests, possible testing criteria:

- Test every condition at least once
- Test every action at least once
- Test every rule at least once

How many tests are needed?

5. State transition testing

Compose a state transition diagram for a task in your project (or select one, eg <http://www.uml-diagrams.org/bank-atm-uml-state-machine-diagram-example.html>)

Design tests:

- Test typical state sequences
- Test all states
- Test all transitions
- Test specific sequences of transitions
- Test invalid transitions