

Exercise 3

sqa@swc.rwth-aachen.de

Issued: 21.05.2019

Submission: 04.06.2019

Discussion: 07.06.2019

3.1. Task: Boundary-Value Analysis

E-Mail is one of the oldest and heavily used communication medium in the internet. E-Mail addresses are used to identify the recipients. An E-Mail address has a specific structure and comprises three essential parts:

- A local part (local-part)
- A domain part (domain-part)
- A top-level domain (tld)

Together these three form an email address: local-part@domain-part.tld

For this format certain rules and constraints exists:

1. The local-part has a minimum length of 1 character.
2. The local-part has maximum length of 64 characters.
3. The domain-part has a minimum length of 1 character.
4. The domain-part has maximum length of 128 characters.
5. The top-level domain has a minimum of 1 character.
6. The top-level domain has a maximum of 63 character.

A character is defined as the character class: word character [a-zA-Z_0-9]

To verify if an email is valid, a software component has been developed. Your task is to provide suitable test suites to test the functionality of the component.

An initial project is available: <https://git.rwth-aachen.de/swc-sqa/emailvalidator-tests>

In this project you will find an initial test class. You can run the tests via the maven goal "test" (mvn clean test)

- a) Please use the Boundary-Value Analysis to identify the input space. Please name all necessary boundary values.
Create and implement a test suite with JUnit5 using **Boundary Value Testing**.
- b) Create and implement a test suite with JUnit5 using **Robustness BV Testing**. Explain the changes and additions you made to the test suite.
- c) Create and implement a test suite with JUnit5 using the **Worst Case BV Testing**.
Explain the changes and additions you made to the test suite.
For this provide a **generator** which computes the needed input/output values. Explain how your generator works.
- d) Create and implement a test suite with JUnit5 representing the **Robustness Worst Case BV Testing**. Explain the changes and additions you made to the test suite.
- e) Please list all bugs found in the EMailValidator Component.

All test suites need to be executable.

Hint: Use JUnit5 Features like Parameterized Tests to minimize the test coded needed.

3.2. Task: Equivalence Class Testing

DHL is a German logistics company which has a price model for their services. They also have a calculator on their website for your convenience to calculate the price for an actual delivery. Here you can find a description of their price model: <https://www.dhl.de/content/dam/images/pdf/dhl-paket-pk-preisuebersicht-032018.pdf>. Unfortunately, it is only available in German. Please use Google Translator to translate the PDF if you don't speak German. If something is still unclear, please drop us an email.

To reduce the scope, please only consider information written on page 2 and page 3. We also limit the number of characteristics you should take into account. For this exercise, a package is composed of the following characteristics.

$\text{Package} \in \text{ProductType} \times \text{Weight} \times \text{Size} \times \text{PointOfSale} \times \text{DeliveryAddress}$

- ProductType: Different types of products (Päckchen, Paket, ...)
- Weight: Weight of the package in kg
- Size: Different sizes of package
- PointOfSale: Do you pay online or in a store
- DeliveryAddress: Only consider the target country* and ignore streets etc.

* Consider "National/German" as zone 0 and the international zones from 1 to 8

The calculator is a function *price* that calculates the package price in cents for a given package.

price: Package → Integer

- a) As a first step, derive suitable equivalence classes from DHL's price model and take the different package characteristics into account. Please provide some information about the equivalence relation you've used for each characteristic.
- b) Select representatives and create a test suite which satisfies the weak equivalence class test criterion (This can be done manually). Enrich the test cases with expected results. You can derive them manually from the document or from the DHL online service. Store the resulting test suite as a CSV file.
- c) Implement the price calculation as a function on your own. Write a parameterized test that executes the created test suite (via importing the CSV file). Use the expected results to test the results of your implementation.

```
public static int calculatePrice(ProductType type, int weight, ...)
```

Hint: You can use `enum` to model discrete value sets such as `ProductType`. It is sufficient if all test cases pass. You don't have to implement a full-fledged calculation.

- d) How can the equivalence classes be extended by boundary values? Please note: You don't have to implement this.
- e) How many test cases are necessary for strong equivalence class testing and how many are necessary for strong equivalence class testing including boundary values?

We would like to ask you to use the templates provided in the L²P room for answering the exercises.

Your results must be handed in as a single PDF file or as a compressed ZIP file containing all necessary files and data named „**SQA2019_AssignmentX_GroupX.zip**“; replace the **X** with your group identifier. Your submission should be submitted via L2P Learning Room.