Black-box testing

November 29th, 2023
Ana Paiva, José Campos

1. Equivalence Class Partitioning & Boundary Value Analysis

For each of the following exercises, perform ‘Category-Partition’ and ‘Boundary Value Analysis’.

In a nutshell, apply Category-Partition and Boundary Value Analysis as described in the lecture classes to the input/outputs of the *programs* described below. Then, write down (1) the partitions you managed to find according to the Category-Partition’s procedure and (2) the inputs you are going to use to test each partition according to the Boundary Value Analysis’ procedure.

1.1

Suppose there is a software that allows one to add new identifiers to a database. The specification of such software says that (1) the length of each identifier must be between 3 and 15 alphanumeric characters, and (2) the first two characters must be letters.

1.2

Suppose there is a program that calculates the change due to pay for a given product, as it would be prepared by a counter employee or a vending machine.

The program receives three types of quantities:

- The price of the product in cents, e.g., 500 cents.
- The total value of coins for payment in cents, e.g., 550 cents.
- A list of available coins, e.g., [1, 10, 100] meaning there is 1 coin of 1 cent, 10 coins of 2 cents, and 100 coins of 5 cents available to the counter employee or a vending machine.

and computes the list of the coins that make up the change as [A, B, C], where A, B, and C would represent the number of 1 cent coins, 2 cent coins, and 5 cent coins, respectively.

The program generates an error when:

- It is not possible to prepare change with the values entered.
- The payment amount was insufficient to cover the price of the product.

1.3

Consider the method `mediaMParesExceptoN` that computes the average of `m` even integers (of a given array of integers) ignoring the first `n` even integers.

```java
public double mediaMParesExceptoN(int[] vector, int m, int n);
```

For example,

```java
mediaMParesExceptoN([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], 3, 2);
```

ignores the first 2 even integers (i.e., 2 and 4) and computes the average of the following 3 even integers (i.e., 6, 8, and 10), 8.0.