Testing Report

PayStation TDD

CIS 3296 Section 05

Spring 2022

Team Members:

**- Roberto Nano**

**- Sofia Drachuk**

**- Arthur Kozhevnik**

Repository URL:

* **https://github.com/cis3296s22/paystationtdd-05-drachuk-kozhevnik-nano-team-4**

Table of Contents

[Document Overview 3](#_gjdgxs)

[Report 3](#_1fob9te)

[Appendix 4](#_3znysh7)

[Unit Test Output 4](#_2et92p0)

[Coverage Report 5](#_tyjcwt)

## 

## Document Overview

This Test Report is a record that the tests were run and a documentation of their results.

Link to repository URL:

https://github.com/cis3296s22/paystationtdd-05-drachuk-kozhevnik-nano-team-4

**Report:**

**The requirements** for this assignment were:

* Add an empty() method that returns the total amount of money collected since the last call and resets values for coins, time, and total money collected to zero
* Modify the cancel() method so that it cancels the present transaction, returns coins that were inserted using HashMap, and resets the values for coins and time to zero.
* Create the following JUnit tests and test others that have been already created:

emptyReturnsTotal()

cancelDoesNotAdd()

emptyShouldResetToZero()

cancelReturnsOneCoin()

cancelReturnTwoCoin()

cancelDoesNotHaveKey()

cancelShouldClearMap()

buyShouldClearMap()

**Implementation:**

Our group implemented two HashMaps so we can easily copy one map to another and clear the old map. We have created 8 JUnit tests and tested 9 JUnit tests that have been already created.

We created the following methods in PayStationImpl.java:

* getInsertedSoFar() to get private variable for inserted so far coins
* populateMap() that clears the previous hashmap and creates a new one based on inserted so far coins
* cutMap() method copies coinsInserted hashmap to the previousCoinMap and then clears coinsInserted hashmap
* getOriginalMap() getter method that returns coinsInserted hashmap

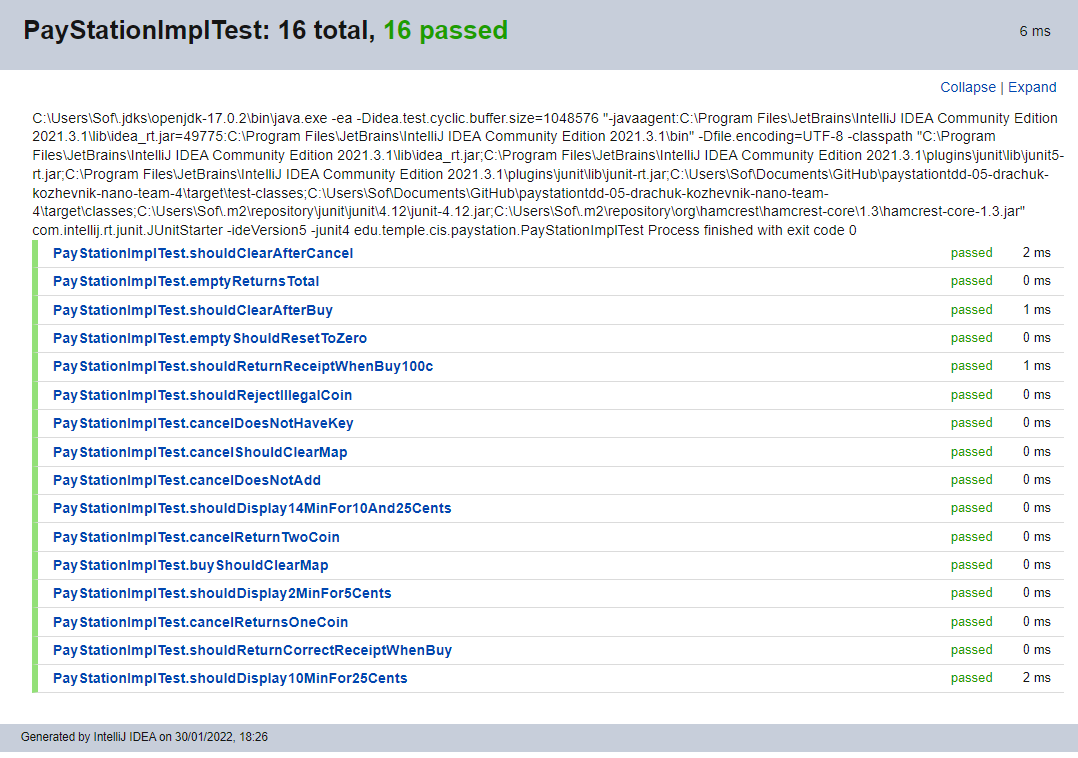
**Problems:**

The first problem that we encountered was that we had all the values in the HashMap but in order to return them and then clear the map, it wasn’t enough for us to have only one map. We created a second HashMap and 2 methods to copy the map and to get the original map that solved our problem. Also, we had problems with Test Case #1(emptyReturnsTotal() test) as we were trying to return the expected value from buy() not the total amount that was inserted. We fixed it by removing buy() and changing empty() so it would return the correct total amount. Another problem we encountered was in test #4, where our initial idea was to test the size of the HashMap instead of checking on how many of the same coin are in it. We modified our test and we used the built-in function “Integer.valueOf() and a “get()” method that would return the value of a specific key(coin). In test #5(cancelReturnsOneCoin()) we faced the same problem as I the previous test (#4) with the expected value and the usage of the “size()” method. Finally, we realized that out cancel() method in the Implementation class was returning wrong values when used and so we had to modify in order to get the expected and correct ones.

## Appendix

### Unit Test Output

(Example output, yours can vary)



### Coverage Report

