M226a Project

Topic: SnackMachine in Java

Documentinformations

Dateiname: M226a-Documentation-SnackMachine

Speicherdatum: 07.11.2021

Author-informations

Autor: Matijas Polazarov

E-Mail: matijas.polazarov@edu.tbz.ch

Tel: +41 76 323 97 84

Table of contents

[1 Introduction 3](#_Toc87217212)

[1.1 Task 3](#_Toc87217213)

[2 Planning 4](#_Toc87217214)

[2.1 Class Diagram 5](#_Toc87217215)

[2.2 Thoughts about my class diagram 6](#_Toc87217216)

[3 SnackImporter Class 6](#_Toc87217217)

[3.1 Thoughts while programming 6](#_Toc87217218)

[4 Sequence Diagram 7](#_Toc87217219)

# Introduction

## Task

Our task for this java project was to build an application with some criterias. My idea was to program the SnackMachine Challenge we programmed as a team at noser young. I have reprogrammed the whole program from scratch, i.e. there is no element that is the same as in the challenge at Noser Young.

# Planning

In my planning, the first thing I did was create my class diagram. In the end, two classes were added to the class diagram. One was the SnackImporter class that connects to the database and the other was the UserHandler class to have the Manager functions only in the ManagerHandler class and the Customer functions only in the CustomerHandler class.

## Class Diagram

Diagram

Description automatically generated

## Thoughts about my class diagram

In my class diagram, it is so that the Main class is at the top. From the Main class then an object is created from the UserHandler class in the Main. The UserHandler class has a connection with the classes IO, SnackImporter, PrintSnackMachine, CustomerHandler, ManagerHandler and of course the Main class. The IO class has only one connection to the UserHandler class. The PrintSnackMachine class has a connection to the UserHandler class and to the CustomerHandler class. The CustomerHandler class has a connection to the ManagerHandler, UserHandler, SnackImporter and PrintSnackMachine. The Snacks and IO class are the only classes that have only one connection. The Snacks class has only three variables and one constructor to create the snacks. The attributes of the snacks are "snackName", "snackPrice", "numberOfSnacks".

# SnackImporter Class

In this class I handle the Connection to the Database.

## Thoughts while programming

The class that took me the most time is the SnackImporter class because I had to build it with SQL statements and especially because the elements of the class were new to me (Connection Class from Java, etc.). First I wrote the code for the connection. I first created 3 variables to connect the program to the database. The first variable is called userName and has the value "stduser", because my user in the database is called "stduser". After that I created the variable "password" with the value "stduserpw", because it is the password of my user "stduser". As last variable I created the variable "URL" which has the value "jdbc:mariadb://localhost:3306/user". After that I created the function "testConnection" which has a try catch to check if you are connected to the database. If you have successfully connected to the database, my program prints "Successfully connected to Database". If you could not connect, my program returns a SQLException. Next I created the function "grabData" which returns an ArrayList. In the parameter I have String query to insert the SQL statement. I called this function in the constructor of the UserHandler class. Since I create an object of the UserHandler class in the main, the content is executed in the constructor, i.e. the function I called in the constructor is executed. In the constructor it looks like this: "mh.setSnacks(si.grabData("SELECT \* FROM user.snack;"));". I set the Arraylist (setSnacks) to "SELECT \* FROM user.snack", meaning the whole content of my database was stored in my ArrayList. This ArrayList is printed in a grid in the PrintSnackMachine class.

# Sequence Diagram

Graphical user interface

Description automatically generated with low confidence