Seminar 4

Object-Oriented Design, IV1350

Nemanja Jancic, jancic@kth.se

2021-06-09

Contents

1 Introduction

This seminar is about improving previously created program in seminar 3. Program is improved by using exception handling and polymorphism. The first task is about adding exceptions to the code. Those exceptions are handling situations where invalid item identifier is used as an input and database errors. The second task is about polymorphism and printing to a file. Total revenue should be printed both in console and in a separate text file.

I have worked on my own.

2 Method

The first step while working with exception handling was to create new classes for exceptions and name them after the error condition. After creating exceptions classes I had to make sure that relevant methods throw relevant exceptions. For exception throwing I used both console print and print to a file. I wrote javadoc for each exception and updated old unit tests.

In the second task I have created a saleObserver class for the observer pattern. Observer is implemented in classes TotalRevenueFileOutput and TotalRevenueView. After creating observer interface, I have used addObserver and notifyObservers in the Sale class just as in example provided in the course book. Only task 2 part a is completed.

3 Result

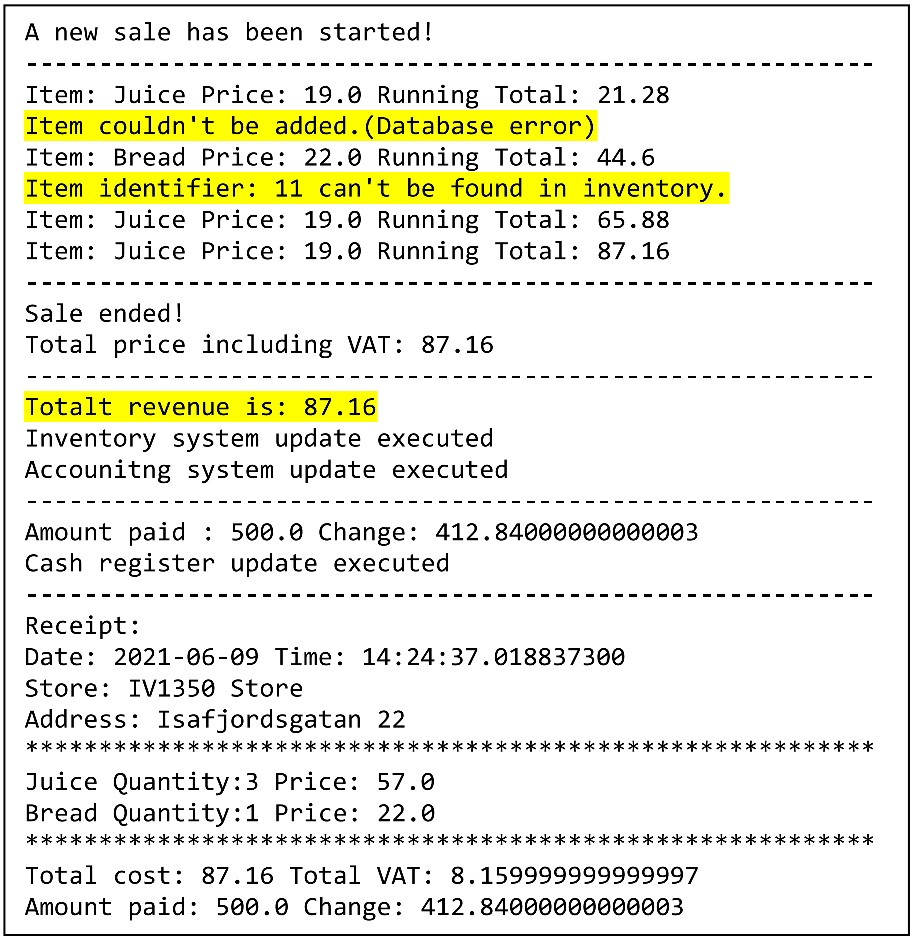
Github repository: https://github.com/janca1/PointOfSale

The program works just the same as before but there are some new functionalities. New functionalities are following: exception handling, printing exceptions to the console and a file named exceptionLog.txt, printing total revenue to the console and a file named RevenueOutputFile.txt.

New classes are AddingItemFailedException.java which is used as an exception for console print when database can't be called. Class DatabaseErrorException.java which is used as an exception for printing to file when database can't be called. ExceptionsLOG.java which is used for logging exceptions to a file. NonExistingIdentifierException. java which is used as an exception when invalid item id is used as an input. SaleObserver.java which is an interface for sale objects. TotalRevenueFileOutput which is used for printing total revenue to a file. And the last one TotalRevenueView which is used for printing total revenue to the console.

Changes have been made in following classes: Controller.java (instances of revenueView and revenueFileOutput have been added and method scanItem throws exceptions). Method findItem in InventorySystem class now throws exceptions. Methods addSaleObserver and notifyObservers have been added to the class Sale.java. Main.java now throws exceptions as well.

Figure 3.1 Console printout (part 1):

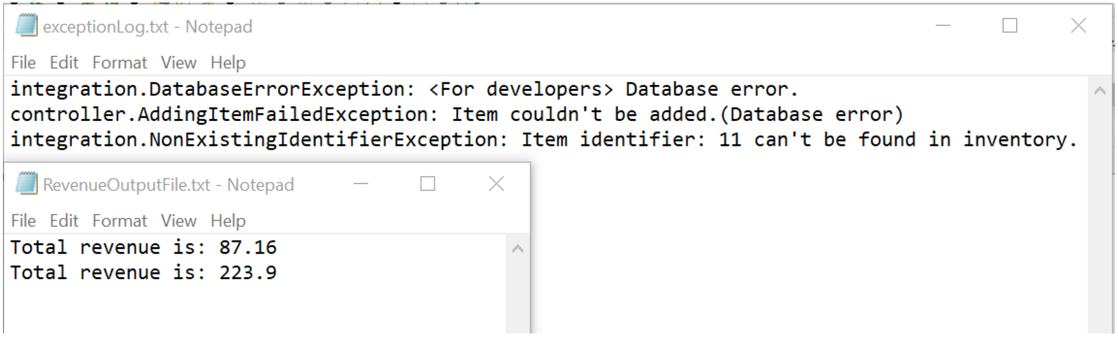


```
A new sale has been started!
----------------------------------------------------------
Item: Juice Price: 19.0 Running Total: 21.28
Item couldn't be added.(Database error)
Item: Bread Price: 22.0 Running Total: 44.6
Item identifier: 11 can't be found in inventory.
Item: Juice Price: 19.0 Running Total: 65.88
Item: Juice Price: 19.0 Running Total: 87.16
----------------------------------------------------------
Sale ended!
Total price including VAT: 87.16
----------------------------------------------------------
Totalt revenue is: 87.16
Inventory system update executed
Accounitng system update executed
----------------------------------------------------------
Amount paid : 500.0 Change: 412.84000000000003
Cash register update executed
----------------------------------------------------------
Receipt:
Date: 2021-06-09 Time: 14:24:37.018837300
Store: IV1350 Store
Address: Isafjordsgatan 22
**********************************************************
Juice Quantity:3 Price: 57.0
Bread Quantity:1 Price: 22.0
**********************************************************
Total cost: 87.16 Total VAT: 8.159999999999997
Amount paid: 500.0 Change: 412.84000000000003
```

Figure 3.2 Console printout (part 2):

```
A new sale has been started!
----------------------------------------------------------
Item: Milk Price: 17.0 Running Total: 18.02
Item: Rice Price: 12.0 Running Total: 30.740000000000002
Item: Book Price: 100.0 Running Total: 136.74
----------------------------------------------------------
Sale ended!
Total price including VAT: 136.74
----------------------------------------------------------
Totalt revenue is: 223.9
Inventory system update executed
Accounitng system update executed
----------------------------------------------------------
Amount paid : 500.0 Change: 363.26
Cash register update executed
----------------------------------------------------------
Receipt:
Date: 2021-06-09 Time: 14:24:37.020587200
Store: IV1350 Store
Address: Isafjordsgatan 22
**********************************************************
Milk Quantity:1 Price: 17.0
Rice Quantity:1 Price: 12.0
Book Quantity:1 Price: 100.0
**********************************************************
Total cost: 136.74 Total VAT: 7.740000000000009
Amount paid: 500.0 Change: 363.26
```

Figure 3.3 Output files:


exceptionLog.txt - Notepad
File Edit Format View Help

```
integration.DatabaseErrorException: <For developers> Database error.
controller.AddingItemFailedException: Item couldn't be added.(Database error)
integration.NonExistingIdentifierException: Item identifier: 11 can't be found in inventory.
```

RevenueOutputFile.txt - Notepad
File Edit Format View Help

```
Total revenue is: 87.16
Total revenue is: 223.9
```

4 Discussion

Errors are handled by exceptions and that is the only use of exceptions. Exceptions are named by error condition. There is a short javadoc explaining each exception. Functionality provided in java.lang.Exception is used for creating and handling exceptions.

Observer pattern is used for informing classes TotalRevenueView and TotalRevenueFileOutput about the payment. Observer is added to the Sale class and classes TotalRevenueView and TotalRevenueFileOutput use the information from the observer. Observer is added to the Sale class by method addSaleObserver. Information about paid amount is passed from Sale class to SaleObserver.