2018-05-09

Implementation and testing

Assigment in the course PA1435 Objektorienterad Design

|  |  |  |  |
| --- | --- | --- | --- |
| **Author Name** | **Personal identity Number** | **Thinking** | **Writing** |
| Andersson Jacob | 960221-8134 | 35% | 20% |
| Johansson Tim | 970718-3472 | 40% | 50% |
| Wikström Leo | 970523-6611 | 5% | 5% |
| Åsbrink Anton | 970428-0135 | 20% | 15% |

# Testing

For our testing-phase we planned to go through our detailed use cases of the first iteration, exploring our different use cases in a usage based manner.

Our usage based testing were successful in 8 out of our 10 detailed use cases. The Remove Truck and Remove Good use cases were the ones that were unsuccessful.

# Changes to Original Design

When implementing our first iteration we had to make certain changes to the original design in order to have the program work in the way we wanted it to. Starting from the left in the Class Diagram, these are the major changes that were made to the design:

  Because of issues concerning memory handling, a WarehouseHandler and TruckHandler object pointer had to be placed in the Menu class. Other changes in this class includes changing the type of runOptions()  to a bool in order to check if the program should exit. From the function displayOptions() the string parameter was removed as we opted to hard code the options to be displayed in the function itself.

  In the different interface classes the parameters were removed from certain functions since we decided to instead put the user input inside those functions, for example in the select functions. A goBack() function was added to each of these classes to be able to return to the previous interface.

  The User and UserHandler classes are implemented but do not have any real function in the program. This is due to us not having time to properly implement them with the rest of the program, but we left them in since we had already created them and as a a bit of a “nod to the future”.

  Moving on to the Warehouse package, the WarehouseHandler looks mostly the same, but a method has been added to be able to determine the size of the vector containing the Warehouses. In the Warehouse class itself, the former getTruck() method has been changed to getTruckHandler(), since the TruckHandler is what we actually want to access from the warehouse. A TruckHandler and GoodSpaceHandler object that were not present before have also been added to the class.

  The TruckHandler has had several methods added: a getTruck() method, taking an int as a parameter has been added, as well as a function to find how many trucks are currently in the warehouse. There is also a selectTruck() and a getSelectedTruck() function since a truck will need to be selected in order to move into the TruckScreen. Trucks as well have a few changes. The getGood() and setGood() methods have been changed to bool to avoid using exceptions and stop a truck from replacing a piece of goods it is carrying until that piece of goods is put down. A setCarries() function is also added in order to change the truck's status between carrying and not carrying a piece of goods.

  In the GoodSpaceHandler the function getCurrentGoodSpace() has been changed to return a GoodSpace pointer. A similar function, getGoodSpace(), which takes an array of two integers as parameters, has been added as well as a function to get the position of the current selected GoodSpace. The getGood() function in GoodSpace has been changed to return a Good pointer in order to accomodate for polymorphism. A getCurrentGood() function and methods for getting and setting the temperature of the GoodSpace have been added.

  In the Good class an = operator overload has been added so that adding and editing Goods works as intended. Just as with the User related classes, the classes BigBox, Small\_Item and GoodsCollection have been implemented but are not really used by the program as it is.