Option #1: Program 1 Pre-Planning (Clothing Boutique)

Micah J. Courey

ITS350 – Information Systems and Security

Colorado State University – Global Campus

Nathan Braun

March 20th, 2017

Program 1 Pre-Planning (Clothing Boutique)

A bag is a abstract data type that can store multiple different objects as well as duplicate objects in no particular order and objects can be added to the bag until the bag reaches it's maximum capacity. In this paper I will plan and design an inventory program for a clothing boutique which will store clothing items and their prices. The program will consist of a Java class which will implement a bag ADT to store each clothing item object and the item's corresponding price value.

Class Design

The first step in the program design process is to outline the responsibilities and collaborations of the Java class. A technique to use in the first stage of designing Java classes is to create a CRC (class-responsibility-collaboration) card for each class. The CRC card lists the classes responsibilities and collaborations with other classes.

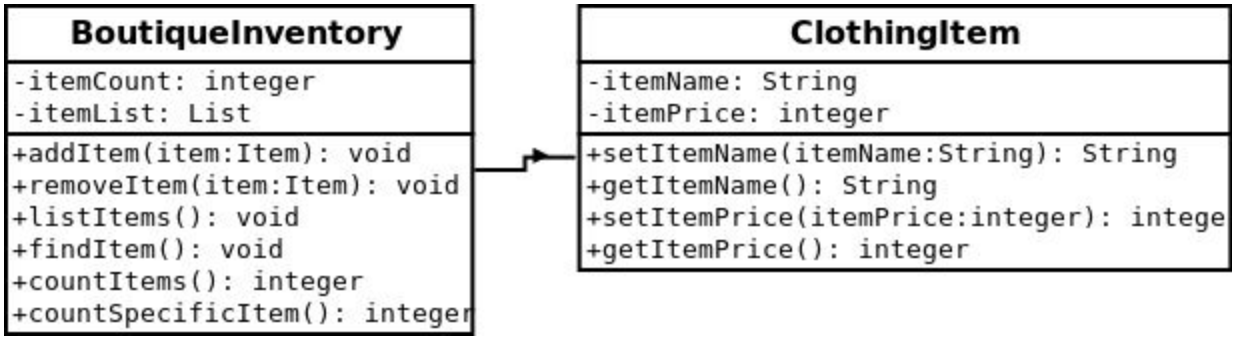
BoutiqueInventory CRC Card

BoutiqueInventory
Responsibilities
Add a clothing item and value
Remove a clothing item and value
Search for a clothing item
Count how many of a specific clothing item are in stock
Count how many total items are in the inventory system
List all items that are in the inventory system
Collaborations
ClothingItem

UML Class Diagram

The next step in the design process is to create a UML class diagram. UML (Unified Modeling Language) is a design language that is used to visually represent

relationships and interactions between different parts of a software program. A class diagram contains the class name, attributes, and methods. Each class has it's own box and different connectors are used between classes to specify the relationship between the classes. The Class Diagram below contains the BoutiqueInventory class as well as a second class ClothingItem. The arrow between the classes is used to represent the association between the two classes.



Bag Interface Implementation

Rather than recreate functionality in the BoutiqueInventory class that has already been created we can implement a reusable bag interface. Since a bag class and interface already contains methods to handle all of the methods in the BoutiqueInventory class, using the bag interface will greatly reduce the amount of programming code needed to write the BoutiqueInventory class.

Conclusion

The design process is an important aspect of engineering a software program. Clearly outlining the classes and interfaces beforehand can save time and reduce frustration during the programming process. Implementing an existing ADT class and interface into your software program can save time by maximizing code reuse.

References

Carrano, F. M., Henry, T., & Tahiliani, M. P. (2016). Data Structures and Abstractions with Java (4th ed.). Harlow, Essex: Pearson Education Limited.