CIS211: Data Structures

Delaware Tech, CT Department

Wilmington

(Instructor: Lee Hsu)

**Project #1: Bag 1 -- Interfaces, Generics, Exception and File Processing**

1. **Introduction:**

This is the 3rd programming course for CIS and CSC students. It is assumed that all students taking this course are able to write programs using Object-Oriented programming constructs such as Class, Inheritance, and Polymorphism. The purpose of this project is to introduce the concept of Data Structures and also brush up your essential JAVA programming skills that need to be used for this course. For this project and the next, you should read pages 1 to 128 of your textbook.

1. **Part A.**
   1. Description:

The sample programs in Chapter 1 of your textbook are not complete. They are used for illustration purpose only. The implementation of Listing 1-1 on page 39 is explained in Chapter 2. In order to see the result of using it, we will need the following set of files:

1. BagInteface.java – the specification only.
2. ArrayBag.java – the implementation of BagInerface.java.
3. ArrayBagDemo.java – a sample program to show us how to use the Bag Interface.
   1. Assignment:

Download and run the above-mentioned programs to ensure they work properly. Study the code and make necessary changes so that the program will run according to the following:

1. Default capacity is 5.
2. When the current capacity is full, increment the capacity by 3 and resize the array accordingly.
3. There is no capacity limit.
4. **Part B.**
   1. Description:
5. In this part, we are going to apply the concepts of Bag Interface to Online Shopping Cart. If you examine the Listing 1-2 program, you will find that the program is not complete and the type Item is missing.
6. In designing interactive application like shopping activities, it is desirable to have the test data stored in a file to speed up the testing time. Therefore, to prepare for the future project, we need to get used to file I/O in Java.
7. There is a tab-delimited text file named “p1artists.txt” which is to be used as an input file. Each record consists of the following two fields:
   1. artistID: integer.
   2. artistName: String;
   3. Assignment:
      1. Refer to the sample program of Listing 1-2 on pages 41 and 42, instead of Item class, implement the Artist class that contains the following:
         1. The 2 fields mentioned above in “p1artists.txt” file.
         2. The Constructor that accepts the 2 fields when the record of an artist is created.
         3. Getters and setters for the 2 fields.
         4. The toString() method.
         5. Test the program with the following data.

|  |  |
| --- | --- |
| ArtistID | Artist Name |
| 1 | Acconci |
| 2 | Ames |
| 3 | Aserty |
| 4 | Baron |
| 5 | Battenberg |

* + 1. Modify the above program according to the following:
       1. Read the input file “p1artists.txt” instead of hardcoding the values.
       2. Use the exception handler to ensure that artist ID is numeric.
       3. If the input is incorrect, print the line number and its contents, but do not stop.
       4. Send the output to a text file named “p1artists\_out1.txt”.
    2. Enhance the above program for the following:
       1. A tab-delimited text file named “p1arts.txt” contains the following record formats:

|  |  |  |  |
| --- | --- | --- | --- |
| ArtID | Title | ArtistID | Appraised Value |
| 1001 | Red Rock Mountain | 50 | 18000 |
| 1002 | Offerings | 52 | 10000 |
| 1003 | Spring Flowers | 12 | 2400 |

* + - 1. Assume that this file contains no error. You are going to read “p1artists.txt” into an array, and then process “p1arts.txt” to add the artist name to the output. Name this output file “p1arts\_out.txt”. Example follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ArtID | Title | Artist | ArtistID | Appraised Value |
| 1001 | Red Rock Mountain | Mogan | 50 | 18000 |
| 1002 | Offerings | Novarre | 52 | 10000 |
| 1003 | Spring Flowers | Chico | 12 | 2400 |

* + - 1. Use an exception handler if the ArtistID does not exist in the “p1artists.txt” file. However, you still need to write this record to the output file.
      2. In the end, include the total number of the Art Works and Artists, and the total appraised value.