

Assignment 1

Group 10:

Members: Badrit Bin Imran, Punyaja Mishra, Gowri Nandana

Question#1:

In the GoF book, List interface is defined as follows:

```
interface List {  
    int count();                //return the current number of elements in the list  
    Object get(int index);      //return the object at the index in the list  
    Object first();             //return the first object in the list  
    Object last();              //return the last object in the list  
    boolean include(Object obj); //return true is the object in list  
    void append(Object obj);     //append the object to the end of the list  
    void prepend(Object obj);    //insert the object to the front of the list  
    void delete(Object obj);     //remove the object from the list  
    void deleteLast();           //remove the last element of the list  
    void deleteFirst();          //remove the first element of the list  
    void deleteAll();            //remove all elements of the list  
}
```

(a) Write a class adapter to adapt Java ArrayList to GoF List interface.

Ans: The code is attached separately.

(b) Write a main program to test your adapters through List interface.

Ans: Output Window:

```
<terminated> Question_1 [Java Application] C:\Program Files\Java\jdk-11.0.5\bin\javaw.exe (Feb. 6, 2021, 3:08:07 a.m.)
Creating a new ClassAdapter class called 'birdAdapter'! We add a few variables in the beginning to continue testing the Adapter
Now, we print out our birdAdapter arrayList : [Duck, Chicken, Eagle, Turkey]
1. Testing count method : 4
2. Testing get method for index = 2 : Eagle
3. Testing first method: Duck
4. Testing last method: Turkey
5. Testing include method for eagle: true
6. Testing append method for Ostrich:
   New birdAdapter arrayList: [Duck, Chicken, Eagle, Turkey, Ostrich]
7. Testing prepend method for Falcon:
   New birdAdapter arrayList: [Falcon, Duck, Chicken, Eagle, Turkey, Ostrich]
8. Testing delete method for Duck:
   New birdAdapter arrayList: [Falcon, Chicken, Eagle, Turkey, Ostrich]
9. Testing delete last :
   New birdAdapter arrayList: [Falcon, Chicken, Eagle, Turkey]
10. Testing delete first :
   New birdAdapter arrayList: [Chicken, Eagle, Turkey]
11. Testing delete all :
   New birdAdapter arrayList: []
So we see that our new birdAdapter class has access to all methods of the interface List from Gof book and all methods native to the java ArrayList
```

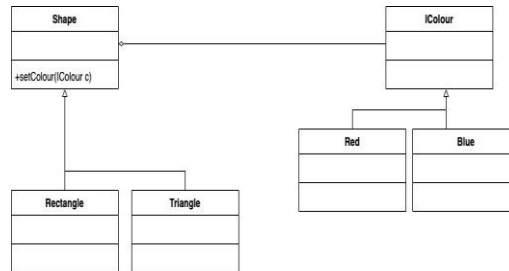
(c) Same requirement as (a) and (b), but write an object adapter to adapt Java ArrayList to GoF List interface.

Ans: Code attached separately, output window:

```
<terminated> testA1 [Java Application] C:\Program Files\Java\jdk-11.0.5\bin\javaw.exe (Feb. 6, 2021, 3:06:46 a.m.)
Creating a new ObjectAdapter class called 'birdAdapter'! We add a few variables in the beginning to continue testing the Adapter
Now, we print out our birdAdapter arrayList : [Duck, Chicken, Eagle, Turkey]
1. Testing count method : 4
2. Testing get method for index = 2 : Eagle
3. Testing first method: Duck
4. Testing last method: Turkey
5. Testing include method for eagle: true
6. Testing append method for Ostrich:
   New birdAdapter arrayList: [Duck, Chicken, Eagle, Turkey, Ostrich]
7. Testing prepend method for Falcon:
   New birdAdapter arrayList: [Falcon, Duck, Chicken, Eagle, Turkey, Ostrich]
8. Testing delete method for Duck:
   New birdAdapter arrayList: [Falcon, Chicken, Eagle, Turkey, Ostrich]
9. Testing delete last :
   New birdAdapter arrayList: [Falcon, Chicken, Eagle, Turkey]
10. Testing delete first :
   New birdAdapter arrayList: [Chicken, Eagle, Turkey]
11. Testing delete all :
   New birdAdapter arrayList: null
So we see that our new birdAdapter class has access to all methods of the interface List from Gof book and all methods native to the java ArrayList
```

Question#2:

In class, we studied that Bridge pattern and discussed the Shape example below. Implement this example in Java. Assume that we want to control the thickness of Shape, i.e., to set the thickness to either Thick or Thin. Add a getter method for Colour and getters and setters to control the thickness. Then write a main method, in which you instantiate four instances of Shape, 2 Rectangle and 2 Triangle: a red thick and a blue thin triangle, and a red thin and a blue thick rectangle. Call the toString() method on all instances to print their colour and thickness.



```
Problems  Javadoc  Declaration  Console X
<terminated> Program [Java Application] C:\Program Files\zulu_jdk_fx_1.8\bin\javaw.exe (Feb. 3, 2021, 1:17:43 a.m.)
Shape Color is Red and thickness is Thick
Shape Color is Blue and thickness is Thin
Shape Color is Red and thickness is Thin
Shape Color is Blue and thickness is Thick
```

Question#3: Design and implement a Java program using Abstract Factory and Singleton design patterns.

The program displays date and time in one of the following two formats:

Format 1:

Date: MM/DD/YYYY

Time: HH:MM:SS

Format 2:

Date: DD-MM-YYYY

Time: SS,MM,HH

The following is how the program works. In the beginning, the program asks the user what display format that she wants. Then the program continuously asks the user to give one of the following commands, and performs the corresponding task. Note that the program gets the current date and time from the system clock (use the appropriate Java date and time operations for this).

'd' : display current date

't': display current time

'q': quit the program.

- In the program, there should be 2 product hierarchies: “DateObject” and “TimeObject”. Each hierarchy should have format1 and format2 described above.
- Implement the concrete factories as singleton classes.
- Draw a UML class diagram for the program.

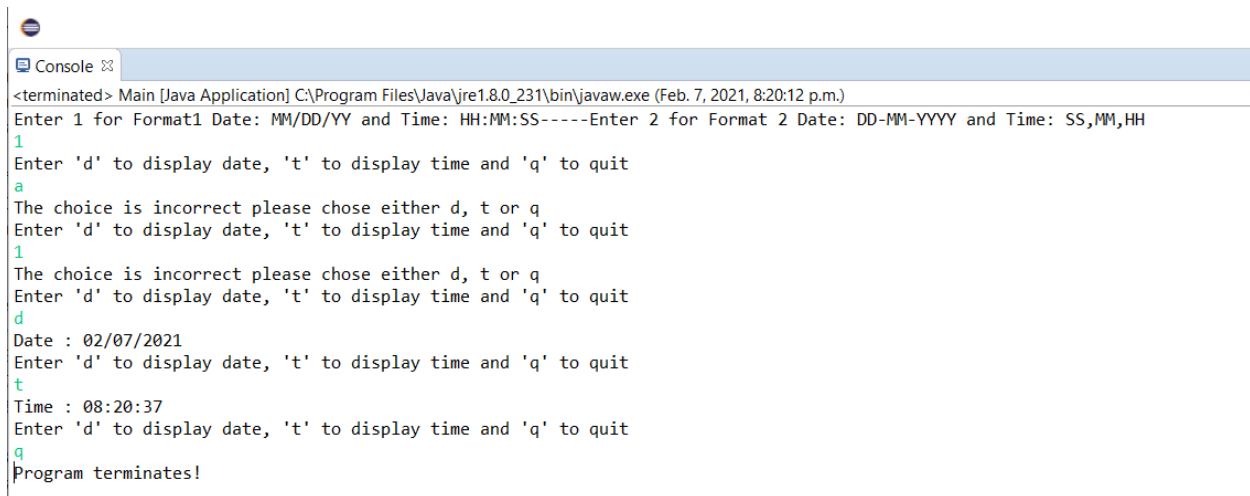
Ans: Code attached separately. Testing and UML Diagram done from next page.

Testing:*Format 2 :*

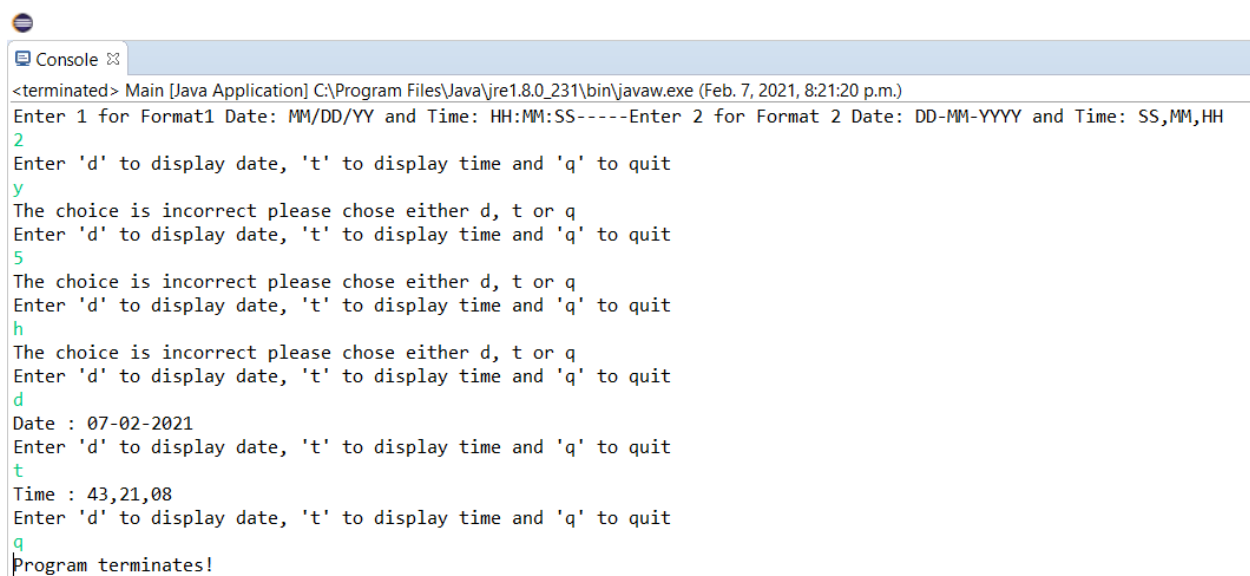
```
Console
<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe (Feb. 7, 2021, 8:16:28 p.m.)
Enter 1 for Format1 Date: MM/DD/YY and Time: HH:MM:SS-----Enter 2 for Format 2 Date: DD-MM-YYYY and Time: SS,MM,HH
2
Enter 'd' to display date, 't' to display time and 'q' to quit
d
Date : 07-02-2021
Enter 'd' to display date, 't' to display time and 'q' to quit
t
Time : 42,16,08
Enter 'd' to display date, 't' to display time and 'q' to quit
q
Program terminates!
```

Format 1 :

```
Console
<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe (Feb. 7, 2021, 8:18:53 p.m.)
Enter 1 for Format1 Date: MM/DD/YY and Time: HH:MM:SS-----Enter 2 for Format 2 Date: DD-MM-YYYY and Time: SS,MM,HH
1
Enter 'd' to display date, 't' to display time and 'q' to quit
d
Date : 02/07/2021
Enter 'd' to display date, 't' to display time and 'q' to quit
t
Time : 08:19:06
Enter 'd' to display date, 't' to display time and 'q' to quit
q
Program terminates!
```

Invalid input :

```
<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe (Feb. 7, 2021, 8:20:12 p.m.)
Enter 1 for Format1 Date: MM/DD/YY and Time: HH:MM:SS-----Enter 2 for Format 2 Date: DD-MM-YYYY and Time: SS,MM,HH
1
Enter 'd' to display date, 't' to display time and 'q' to quit
a
The choice is incorrect please chose either d, t or q
Enter 'd' to display date, 't' to display time and 'q' to quit
1
The choice is incorrect please chose either d, t or q
Enter 'd' to display date, 't' to display time and 'q' to quit
d
Date : 02/07/2021
Enter 'd' to display date, 't' to display time and 'q' to quit
t
Time : 08:20:37
Enter 'd' to display date, 't' to display time and 'q' to quit
q
Program terminates!
```



```
<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe (Feb. 7, 2021, 8:21:20 p.m.)
Enter 1 for Format1 Date: MM/DD/YY and Time: HH:MM:SS-----Enter 2 for Format 2 Date: DD-MM-YYYY and Time: SS,MM,HH
2
Enter 'd' to display date, 't' to display time and 'q' to quit
y
The choice is incorrect please chose either d, t or q
Enter 'd' to display date, 't' to display time and 'q' to quit
5
The choice is incorrect please chose either d, t or q
Enter 'd' to display date, 't' to display time and 'q' to quit
h
The choice is incorrect please chose either d, t or q
Enter 'd' to display date, 't' to display time and 'q' to quit
d
Date : 07-02-2021
Enter 'd' to display date, 't' to display time and 'q' to quit
t
Time : 43,21,08
Enter 'd' to display date, 't' to display time and 'q' to quit
q
Program terminates!
```

UML diagram :

