Homework 07
Sorting Laundry
Due 11/17/2023 by 11:55pm

Objective:
Write a program that will sort a basket of clothes into their proper drawers. If you were not aware you are sort clothes by their type in this order:

- Top Drawer – Undergarments
- Next Drawer – Socks or Stockings
- The Following Drawer – Tops
- The Subsequent Drawer – Bottoms
- The Cape Drawer – Capes

Requirements:

- Functionality. (80pts)
  - No Syntax Errors. (80pts*)
    - *Code that cannot be compiled due to syntax errors is nonfunctional code and will receive no points for this entire section.
  - Write a class called **Clothing** with the following (10pts):
    - Instance Variables (1pt)
      - Type: A value that represents the type of clothing, and can only have the value “Undergarment”, “Socks”, “Stockings”, “Top”, “Bottom”, or “Cape”. Its default value should be “Socks”
      - Color: A value that represents the color of the clothing, and can only have the value “Brown”, “Red”, “Pink”, “Black”, “White”, “Orange”, “Green”, “Blue”, “Purple”, or “Grey”. Its default value should be “Black”.
      - All must have the private scope, and all above must apply for full credit.
    - Constructors (2pts)
      - Default: Must set all properties to their default values mentioned in the “Instance Variables” section.
      - Parameterized: Must take in a parameter for each instance variable in the order named above. This means the first instance variable is the first parameter, the second instance variable is the second parameter, and so on. This must set the instance variable values only if the given values are valid, but otherwise it must set the instance variables to their default values.
      - All above must apply for full credit.
    - Methods
      - Accessors for the instance variables. (1pt)
        - All must be written for full credit.
      - Mutators for the instance variables. (2pts)

        - Make sure in the mutators check for valid values named in the “Instance Variables” Section.
        - If the value that is being set is not valid, then set the instance variable to its default value.
        - All above must apply for full credit.
    - toString (2pts): This method returns a String with all of the instance variable values concatenated together with the format:

      [Clothing] Type: <<Type>> Color: <<Color>>

      Where values in “<<>>” correspond to instance variable values.
    - Equals (2pts): This method takes in another instance of Clothing and only returns true if all of the instance variables match.
- Write a class called **Dresser** with the following: (60pts)
  - Instance Variables (5pts)
    - Clothes: This is a 2D array of type clothing where it has 5 Drawers that can only hold 10 items of clothing.
    - All must have the private scope, and all above must apply for full credit.
  - Constructors (5pts)
    - Default: Must construct the clothes array given the correct dimensions.
  - Methods
    - No Accessors of Mutators.
    - Add (20pts): Takes in an instance of Clothing as a parameter and returns nothing. The parameter is then sorted in their proper drawers by its type as mentioned above. If a drawer is full make sure to tell the user.
    - Remove (20pts): Takes in an instance of Clothing as a parameter and returns nothing. This method searches for a piece of clothing, and if it exists it is removed. This can be done by either setting the value to *null* or shifting memory addresses.
    - Print (10pts): This prints out every piece of clothing’s information in the dresser to the console.
- Write a class called **DresserFrontEnd** with the following (10pts):
  - Include the main method.
  - The program must prompt the user with the options
    - Add Clothes to the Dresser
    - Remove Clothes from the Dresser
    - Quit
  - If the user chooses to add clothes to the dresser, then the user must be prompted to enter the clothes’ information. Once entered the program must add the new article of clothing to the correct dresser drawer. If that drawer is full, then the program may ignore the request.

        - If the user chooses to remove clothes from the dresser, then the user must be prompted to enter the clothes' information. Once entered the program must search and remove that article of clothing from the dresser. If the item being removed does not exist in the dresser, then the program may ignore the request.
        - After every operation, the program must clearly print out the items contained in the dresser.
        - All above must apply for full credit.

- Coding Style. (10pts)
    - Code functionality organized within multiple methods other than the main method, and methods organized within multiple classes where appropriate. (5pts)
    - Readable Code. (5pts)
        - Meaningful identifiers for data and methods.
        - Proper indentation that clearly identifies statements within the body of a class, a method, a branching statement, a loop statement, etc.
        - All the above must apply for full credit.
- Comments. (10pts)
    - Your name in every file. (5pts)
    - At least 5 meaningful comments in addition to your name. These must describe the function of the code it is near. (5pts)

Example Dialog:
*The following Example Dialog demonstrates the interactions between a user and ONE possible implementation of the required software's front-end / user interface. The software's front-end / user interface may be implemented in MANY different ways and will receive full credit as long as it meets the most minimal of the above requirements. While you may use the example dialog as a guide, it is strongly encouraged to create the front-end / user interface in your own way. *

Key	
Unhighlighted Text	**Program's Output**
Highlighted Text	**User's Input**

```
Welcome to the dresser!
Enter 1: to add an item
Enter 2: to remove an item
Enter 3: to print out the dresser contents
Enter 9: to quit
1
Enter the type
```

```
It may be undergarment, socks, stockings,
top, bottom, or cape
top
Enter a color
It may be brown, pink, orange, green, blue,
purple, or grey
red

Enter 1: to add an item
Enter 2: to remove an item
Enter 3: to print out the dresser contents
Enter 9: to quit
1
Enter the type
It may be undergarment, socks, stockings,
top, bottom, or cape
cape
Enter a color
It may be brown, pink, orange, green, blue,
purple, or grey
purple

Enter 1: to add an item
Enter 2: to remove an item
Enter 3: to print out the dresser contents
Enter 9: to quit
1
Enter the type
It may be undergarment, socks, stockings,
top, bottom, or cape
socks
Enter a color
It may be brown, pink, orange, green, blue,
purple, or grey
grey
```

```
Enter 1: to add an item
Enter 2: to remove an item
Enter 3: to print out the dresser contents
Enter 9: to quit
1
Enter the type
It may be undergarment, socks, stockings,
top, bottom, or cape
cape
Enter a color
It may be brown, pink, orange, green, blue,
purple, or grey
blue

Enter 1: to add an item
Enter 2: to remove an item
Enter 3: to print out the dresser contents
Enter 9: to quit
1
Enter the type
It may be undergarment, socks, stockings,
top, bottom, or cape
undergarment
Enter a color
It may be brown, pink, orange, green, blue,
purple, or grey
pink

Enter 1: to add an item
Enter 2: to remove an item
Enter 3: to print out the dresser contents
Enter 9: to quit
3
Drawer 0
undergarment pink
```

```
Drawer 1
socks grey

Drawer 2
top red

Drawer 3

Drawer 4
cape purple
cape blue

Enter 1: to add an item
Enter 2: to remove an item
Enter 3: to print out the dresser contents
Enter 9: to quit
2
Enter the type
It may be undergarment, socks, stockings,
top, bottom, or cape
socks
Enter a color
It may be brown, pink, orange, green, blue,
purple, or grey
grey

Enter 1: to add an item
Enter 2: to remove an item
Enter 3: to print out the dresser contents
Enter 9: to quit
3
Drawer 0
undergarment pink

Drawer 1
```

```
Drawer 2
top red

Drawer 3

Drawer 4
cape purple
cape blue

Enter 1: to add an item
Enter 2: to remove an item
Enter 3: to print out the dresser contents
Enter 9: to quit
9
Goodbye
```

Finally:
Upload all java source files (.JAVA extension) to the CSCE Dropbox