Introduction

This application was build in Java and serves as an inventory management system tailored for a makeup retail company. Its primary function is to record stock, specifically focusing on facial and eye makeup products, and save this inventory data within a text file for staff to have easy access for record-keeping.

The software lets users have control over the inventory dataset, offering functionalities to access, append, delete, and review product records within the text file. To achieve organised data management, the program utilises two distinct data structures: FaceMakeup and EyeMakeup arrays. These arrays facilitate the classification and handling of inventory items, ensuring efficient data retrieval and manipulation.

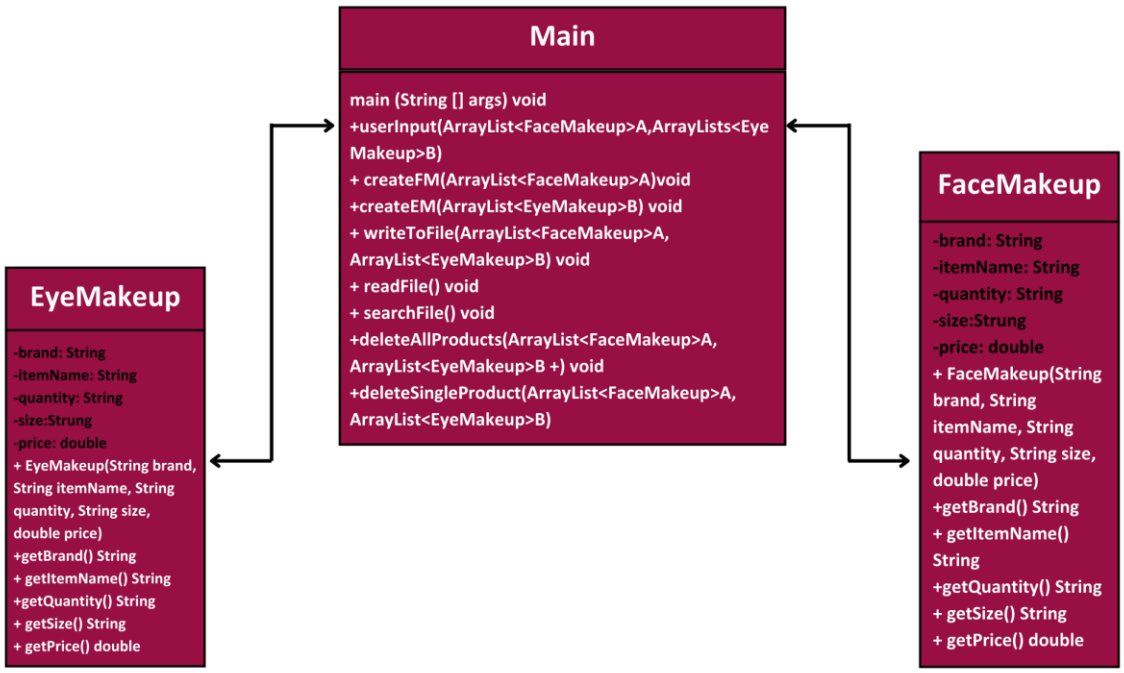
Object Oriented Design and UML Diagrams

Unified Modelling Language (UML) is a standardized modelling language widely used in software engineering to visualise, specify, construct, and document the components of software systems. By providing a common framework, UML facilitates clear communication among developers and stakeholders, aiding in the detection of bugs, identification of incomplete sections, and tracking of pending tasks within a program.

Among the various UML diagrams the two most popular types are:

- **Class Diagrams**: These diagrams are fundamental in object-oriented design, representing the static structure of a system by illustrating its classes, attributes, methods, and the relationships among objects. They provide a high-level overview of a system's architecture, detailing how different classes interact and are associated with one another.
- **Object Diagrams**: Serving as instances of class diagrams, object diagrams depict a snapshot of the system at a particular point in time. They showcase specific objects and their relationships, offering insights into the system's state and helping to validate the accuracy of class diagrams. By highlighting real-world examples of class interactions, object diagrams assist in identifying potential design flaws that may require attention.

In my program, two distinct arrays (“`FaceMakeup`” and “`EyeMakeup`”) are used to organise data into two categories. Organising related data and functions into specific groups, called classes, makes managing and accessing information easier. This approach follows object-oriented programming principles, where each class represents a specific category with its own data. By structuring a program this way, it can work more efficiently and have less errors.


Main
main (String [] args) void
+userInput(ArrayList<FaceMakeup>A,ArrayLists<EyeMakeup>B)
+ createFM(ArrayList<FaceMakeup>A)void
+createEM(ArrayList<EyeMakeup>B) void
+ writeToFile(ArrayList<FaceMakeup>A, ArrayList<EyeMakeup>B) void
+ readFile() void
+ searchFile() void
+deleteAllProducts(ArrayList<FaceMakeup>A, ArrayList<EyeMakeup>B +) void
+deleteSingleProduct(ArrayList<FaceMakeup>A, ArrayList<EyeMakeup>B)
EyeMakeup
-brand: String
-itemName: String
-quantity: String
-size:Strung
-price: double
+ EyeMakeup(String brand, String itemName, String quantity, String size, double price)
+getBrand() String
+ getItemName() String
+getQuantity() String
+ getSize() String
+ getPrice() double
FaceMakeup
-brand: String
-itemName: String
-quantity: String
-size:Strung
-price: double
+ FaceMakeup(String brand, String itemName, String quantity, String size, double price)
+getBrand() String
+ getItemName() String
+getQuantity() String
+ getSize() String
+ getPrice() double

Program Details

FaceMakeup.java | EyeMakeup.java | *Main.java | .project

```
1   package inventory;
2
3   import java.util.ArrayList;
4   import java.util.Scanner;
5   import java.io.IOException;
6   import java.io.File;
7   import java.io.FileReader;
8   import java.io.BufferedReader;
9   import java.io.PrintWriter;
10  import java.io.FileWriter;
11  import java.util.ArrayList;
12  import java.time.LocalDateTime;
13  import java.time.format.DateTimeFormatter;
14  import java.util.Scanner;
15  import java.util.InputMismatchException;
16
17
18  public class Main {
19      public static void main(String[] args) throws IOException
38
39  public static void userInput(ArrayList<FaceMakeup> A, ArrayList<EyeMakeup> B) throws IOException{
71
72  public static void createFM(ArrayList<FaceMakeup>A){
96
97  public static void createEM(ArrayList<EyeMakeup> B ){
121
122 public static void writeToFile(ArrayList<FaceMakeup> A,ArrayList<EyeMakeup> B) throws IOException{
141
142 public static void readFile () throws IOException {
164
165 public static void main1(String[] args) {
169  * Allows the user to search for products containing a specific keyword and displays their contents.
171 public static void searchFile() {
204
206  * Deletes all files within the specified directory.
208 public static void deleteAllProducts(ArrayList<FaceMakeup> A,ArrayList<EyeMakeup> B) throws IOException
223
224 public static void deleteSingleProduct(ArrayList<FaceMakeup> A,ArrayList<EyeMakeup> B) throws IOException{
276
277 }
```

userInput:

Is used to input any data given by the user

```
39 public static void userInput(ArrayList<FaceMakeup> A, ArrayList<EyeMakeup> B) throws IOException{
40
41
42      Scanner input = new Scanner (System.in);
43      System.out.println("Please type FM or EM to input a new product record \n Type finished when you have completed this task" );
44      String x = input.next();
45
46      if (x.equalsIgnoreCase("Exit")) {
47          System.exit(0);
48      }
49      while (!x.equalsIgnoreCase("finished"))
50      {
51          if (x.equalsIgnoreCase("FM"))
52          {
53              createFM(A);
54
55          }
56          else if (x.equalsIgnoreCase("EM"))
57          {
58              createEM(B);
59              }
60          else
61          {
62              System.out.println("you have not selected a valid entry - try again\n");
63
64          }
65
66          System.out.println("Type FM or EM to input the next type of product you wish to capture \n Type finished when have completed this task" );
67          x = input.next();
68
69            }
70  }
```

createFM:

Create an array for face makeup.

```java
public static void createFM(ArrayList<FaceMakeup>A){

    Scanner input = new Scanner (System.in);
    boolean correctValue = false;

    System.out.println("Please provide the brand name\n");
    String brand = input.next();

    System.out.println("Please provide the name of the item\n");
    String itemName = input.next();

    System.out.println("Please provide the quatity\n");
    String quantity = input.next();

    System.out.println("Please provide the size od the product\n");
    String size = input.next();

    System.out.println("Provide the price of the product\n");
    double price = input.nextInt();

    FaceMakeup FM1= new FaceMakeup(brand, itemName, quantity, size, price );
    A.add(FM1);

}
```

createEM:

Create an array for eye makeup.

```java
public static void createEM(ArrayList<EyeMakeup> B ){


    Scanner input = new Scanner (System.in);

    System.out.println("Please provide the brand name\n");
    String brand = input.next();

    System.out.println("Please provide the name of the item\n");
    String itemName = input.next();

    System.out.println("Please provide the quatity\n");
    String quantity = input.next();

    System.out.println("Please provide the size od the product\n");
    String size = input.next();

    System.out.println("Provide the price of the product\n");
    double price = input.nextDouble();

    EyeMakeup EM1= new EyeMakeup(brand, itemName, quantity, size, price );
    B.add(EM1);

}
```

writeToFile:

Write to a file names “Products.txt”

```
122 public static void writeToFile(ArrayList<FaceMakeup> A,ArrayList<EyeMakeup> B) throws IOException{
123
124     File file = new File ("Product.txt");
125     if (!file.exists()){
126         file.createNewFile();
127     }
128
129     FileWriter writer = new FileWriter(file);
130
131     for (int i=0;i<A.size();i++){
132         writer.write((A.get(i).getBrand()+" "+A.get(i).getItemName()+" "+A.get(i).getQuantity()+" "+A.get(i).getSize()+" "+A.get(i).getPrice()+"\n"));
133     }
134
135     for (int j=0; j<B.size();j++){
136
137         writer.write((B.get(j).getBrand()+" "+B.get(j).getItemName()+" "+B.get(j).getQuantity()+" "+B.get(j).getSize()+" "+B.get(j).getPrice()+"\n"));
138     }
139
140     writer.close();}
```

readFile:

Reads the data from the file “Products.txt”

```
142 public static void readFile () throws IOException {
143
144     System.out.println("Do you wish to review the data you have added? Type Yes or No please\n");
145     Scanner input = new Scanner(System.in);
146     String x = input.next();
147
148     if (x.equalsIgnoreCase("yes")){
149
150     FileReader file = new FileReader("Product.txt");
151     BufferedReader y = new BufferedReader(file);
152
153     String z = y.readLine();
154
155     while (z!=null)
156     {
157         System.out.println(z);
158         z =y.readLine();
159     }
160
161     y.close();
162     }
163 }
```

searchFile:

allows the user to search the files and displays the data

```java
public static void searchFile() {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter the keyword to search for in product names:");
    String keyword = Scanner.nextLine();
    File directory = new File(directory);

    if (!directory.exists() || !directory.isDirectory()) {
        System.out.println("Directory does not exist.");
        return;
    }

    File[] matchingProducts = directory.listFiles((dir, itemName) -> itemName.contains(keyword));

    if (matchingProducts == null || matchingProducts.length == 0) {
        System.out.println("No files found matching the keyword.");
        return;
    }

    for (File file : matchingProducts) {
        System.out.println("Found product: " + file.getName());
        System.out.println("Contents:");
        try (BufferedReader reader = new BufferedReader(new FileReader(file))) {
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            System.out.println("Error reading file: " + file.getName());
            e.printStackTrace();
        }
        System.out.println("--------------------------------------------------");
    }
}
```

deleteAllProducts:

allows the user to delete all the products saved in the file

```java
public static void deleteAllProducts(ArrayList<FaceMakeup> A,ArrayList<EyeMakeup> B) throws IOException
{
    Scanner input = new Scanner (System.in);
    System.out.println("Would you like to delete all the products? Please type Yes or No");
    String x = input.nextLine();

    if (x.equalsIgnoreCase("Yes"))
    {
        //this clears the data from the file by overwriting it
    FileWriter file = new FileWriter ("device.txt");
    file.close();

    System.out.println("The data has been deleted from the file now\n");
    }
}
```

deleteSingleProduct:

Allows the user to delete a single product in the file

```
224 public static void deleteSingleProduct(ArrayList<FaceMakeup> A,ArrayList<EyeMakeup> B) throws IOException{
225     // we also need to store a create a copy of this file while we are copying
226
227     File temp = new File("Temp.txt");
228     if (!temp.exists())
229     {
230         temp.createNewFile();
231     }
232     // call a previous method to reload all the data to the device.txt file
233     writeToFile(A,B);
234
235     // need to start reading the existing file and copy
236     // it to the temp file except for line with the "searchvalue";
237     File file = new File("Product.txt");
238     FileReader x = new FileReader(file);
239     BufferedReader y = new BufferedReader(x);
240     String z = y.readLine();
241
242     // this is the filewriter needed for the copy
243     FileWriter tempWrite = new FileWriter(temp);
244
245     /// firstly the user needs to input the value they wish to delete;
246     System.out.println("You will be asked to select a product. To help with the selection you have the chance to review all data");
247     // also going to allow the user a refreshed view of the data
248     readFile();
249
250     System.out.println("Provide some detail on the product you wish to delete now");
251     Scanner input = new Scanner (System.in);
252     String searchValue =input.next();
253
254     while(z!=null )
255     {
256         // while the line doesnt contain the product the user is deleting
257         //copy it to the file
258         if (!z.trim().contains(searchValue))
259         {
260
261         tempWrite.write(z);}
262         z=y.readLine();
263     }
264
265     // close the reader & writers;
266     System.out.println("That product has now been deleted from the file");
267     y.close();
268     tempWrite.close();
269
270         // to overwrite the file - we need to delete the existing
271         // device.txt file & rename the temp file to device.txt;
272         file.delete();
273
274         temp.renameTo(file);
275 }
276
277 }
```

The program has the following imports:

FaceMakeup.java | EyeMakeup.java | *Main.java | .project

```
1  package inventory;
2
3  import java.util.ArrayList;
4  import java.util.Scanner;
5  import java.io.IOException;
6  import java.io.File;
7  import java.io.FileReader;
8  import java.io.BufferedReader;
9  import java.io.PrintWriter;
10 import java.io.FileWriter;
11 import java.util.ArrayList;
12 import java.time.LocalDateTime;
13 import java.time.format.DateTimeFormatter;
14 import java.util.Scanner;
15 import java.util.InputMismatchException;
```

Buffered Reader:

Is used to read the text from an input stream, like out .txt file.

File:

This import is used to get information about files and directories. It also allows the creation of files and directories.

FileReader:

Is used for reading files.

FileWriter:

Allows you to write to the file

IOException:

Is used to handle errors that are related to input/output.

LocalDateTime:

Used to represent date and time

DateTimeFormatter:

Used to represent date and time

ArrayList:

Allows the creation of arrays and arraylists.

Scanner:

Is used to create a scanner object to allow user input.

Methods Details

Main Method:

The main method calls the other methods and is where the program starts from. It also creates the arrays “FaceMakeup” and “EyeMakeup”

```
18 public class Main {
19     public static void main(String[] args) throws IOException
20     {
21
22     System.out.println("Welcome to Stock Manager[SM]. This creates a record of your Face Makeup[FM] and Eye Makeup[EM]\n");
23
24         ArrayList<FaceMakeup> A = new ArrayList<FaceMakeup>();
25         ArrayList<EyeMakeup> B = new ArrayList<EyeMakeup>();
26
27         userInput(A,B);
28         writeToFile(A,B);
29         System.out.println("Thank you for the data - it has been added to the file\n");
30
31         readFile();
32         searchFile();
33         deleteAllProducts(A,B);
34         deleteSingleProduct(A,B);
35         System.out.println("Program Terminated. Thank You");
36
37     }
38
```

Scenario Tested	Expected Results	Actual Results
Inserting a foundation into the “Products.txt” file	The data input will be saved in the file *Products.txt*	Working

Inserting a mascara into the *Products.txt* file	The data input will be store in the *Products.txt* file	Working
Searching data in the System	The program will display the records that match the query	Working
Deleting Records	The program will either delete all the records if told to or it will delete a single record that was chosen.	Working

The program functions properly, and the database performs as it was intended to. It gathers user information that can be accessed later for review, and the data can also be deleted/edited.