

Cell Phone Algorithm

1. Declare variables for the cellphone options.
2. Set the base price for the cellphone before customization
3. Declare variables for the agent(manager) user of the system
4. Declare variables for the business sales report.
5. Create the following functions:
 - a. Create a login function to validate the password (3 attempts)
 - b. Create a process order function to capture the user choices
 - c. Display a receipt for each order
 - d. Display a summary of all sales
 - e. Create a change color function for error messages
 - f. Write the sales report to a text file.
6. Create classes
 - a. Customer
 - b. Order
 - c. CellPhone
 - d. Manufacturer
 - e. ManufacturerSystem (FabTechSystem)
7. Create text files
 - a. customers.dat (input)
 - b. orders.dat (output)

```
1 package pkgCustomPhones;
2
3
4 import java.io.IOException;
15
16 public class ManufacturerSystem {
17
18     /*****MAIN METHOD*****/
19     *
20     * @throws IOException
21     *****/
22
23     public static void main(String[] args) throws IOException
24     {
25         // TODO Auto-generated method stub
26         Manufacturer motorola = new Manufacturer();
27         int pi = -1;
28
29         motorola.loadAgents();
30         pi = login(motorola.purchasers, motorola.PURCHASERSIZE);
31
32         motorola.customizePhone();
33         motorola.printSales();
34
35     } // end of main
36
37     /*****LOGIN METHOD*****/
38     *
39     *****/
40
41
42     public static int login(Customer purchasers[], int PURCHASERSIZE)
43     {
44
45         //Working Variables
46
47         Scanner keyboard = new Scanner(System.in);
48         String entpass = "";
49         boolean isPassValid = false;
50         int numattempts = 0;
51         int pi = -1;
```

```
52
53     purchasers[0] = new Customer(111,"Mahatma Ghandi","ghandi","mghandi@gmail.com");
54     purchasers[1] = new Customer(222,"Mother Theresa","theresa","mtheresa@gmail.com");
55     purchasers[2] = new Customer(333,"Kathy Hochul","hochul","khochul@gmail.com");
56
57     System.out.println("Welcome to KM Custom Phones Sales System");
58     //validate password
59     do
60     {
61         System.out.print("Please enter your password: ");
62         entpass = keyboard.next();
63
64         for (int i = 0; i < PURCHASERSIZE; i++ )
65             {   if (entpass.equals(purchasers[i].password))
66                 {   isPassValid = true;
67                     pi = i;
68                 }
69             }
70
71         if (!isPassValid)
72         {   System.out.println("Password Invalid, try again!");}
73         numattempts++;
74
75     } while (numattempts < 3 && !isPassValid);
76
77     if (!isPassValid)
78     {   System.out.println("Number of attempts exceeded, program terminated!");
79         System.exit(0);
80     }
81     keyboard.close();
82     return pi;
83 } //end of login
84 } // end of class
85
```

```

1 package pkgCustomPhones;
2
3
4 import java.text.SimpleDateFormat;
5
6
7
8 /*****MANUFACUTURER CLASS*****/
9 *
10 *****/
11
12 public class Manufacturer
13 {
14
15     //Class Variables
16     CellPhone customphone;
17     final int PURCHASERSIZE; //size of the purchaser array
18     Customer purchasers[];
19     int pi;
20
21     //keep an array list of cell phones sold
22     ArrayList<Order> sales;
23
24     //Working Variables
25     BufferedReader br;
26     Scanner keyboard;
27     PrintWriter pw;
28     final float BASEPRICE;
29     char morephones;
30     int numphones;
31     float grandtotal;
32
33     final int COLORSIZE; //size of the colors array
34     String colors[];
35
36     /*****MANUFACUTURER
37     CONSTRUCTOR*****/
38     *
39     *
40     *
41     *
42     *
43     *
44     *
45     *
46     *
47     *
48     *
49     *
50     *
51     *
52     *
53     *
54     *
55     *
56     *
57     *
58     *
59     *
60     *
61     *
62     *
63     *
64     *
65     *
66     *
67     *
68     *
69     *
70     *
71     *
72     *
73     *
74     *
75     *
76     *
77     *
78     *
79     *
80     *
81     *
82     *
83     *
84     *
85     *
86     *
87     *
88     *
89     *
90     *
91     *
92     *
93     *
94     *
95     *
96     *
97     *
98     *
99     *
100    */
101
102     Manufacturer()
103     {

```

```

42         //Class Variables
43         customphone = new CellPhone();
44         PURCHASERSIZE = 3; //size of the purchaser array
45         purchasers = new Customer[PURCHASERSIZE];
46
47
48         //keep an array list of cell phones sold
49         sales = new ArrayList<Order>();
50
51         //Working Variables
52         br = new BufferedReader(new InputStreamReader(System.in));
53         keyboard = new Scanner(System.in);
54         BASEPRICE = 550.00f;
55         morephones = 'Y';
56         numphones = 0;
57         grandtotal = 0;
58
59
60         COLORSIZE = 5; //size of the colors array
61         colors = new String[COLORSIZE];
62         colors[0] = "Red";
63         colors[1] = "Blue";
64         colors[2] = "Black";
65         colors[3] = "White";
66         colors[4] = "Gold";
67
68     }
69
70     /*****CUSTOMIZEPHONE
71     METHOD*****/
72     *
73     *
74     */
75     void customizePhone()
76     {
77         int input = 0;
78         System.out.println("Welcome, " + purchasers[pi].fullname);
79         do
80         {
81             customphone.setTotalprice(BASEPRICE);
82             System.out.println("Choose the options for your phone.\n");
83             System.out.print("Operating System (1 - Android 2 - iOS): " );

```

```
81     input = keyboard.nextInt();
82     if (input == 1)
83     {
84         customphone.setOs("Android");
85     }
86     else if (input == 2)
87     {
88         customphone.setOs("iOS");
89         customphone.setTotalprice(customphone.getTotalprice()+100.00f);
90     }
91     else
92     {
93         System.out.println("The operating system choice is invalid!");
94     }
95
96     //Display each color
97     System.out.print("Color (");
98     for (int i = 0; i < colors.length; i++)
99     {
100         System.out.print(i+1 + " - " + colors[i] + " ");
101     }
102     System.out.print("): ");
103     input = keyboard.nextInt();
104     if (input < 1 || input > 5)
105     {
106         System.out.println("The color choice is invalid!");
107     }
108     else
109     {
110         customphone.setColor(colors[input -1]);
111     }
112
113     System.out.print("Screen Size (1 - 6.5 in 2 - 7.5 in): ");
114     input = keyboard.nextInt();
115     if (input == 1)
116     {
117         customphone.setScreen_size(6.5f);
118     }
119     else if (input == 2)
120     {
121         customphone.setScreen_size(7.5f);
122         customphone.setTotalprice(customphone.getTotalprice()+50.00f);
123     }
124     else
125     {
126         System.out.println("The screen size choice is invalid!");
127     }
128
129     System.out.print("Storage (1 - 64gb 2 - 128 gb): ");
130     input = keyboard.nextInt();
131     if (input == 1)
132     {
133         customphone.setStorage(128);
134     }
135     else if (input == 2)
136     {
137         customphone.setStorage(128);
138         customphone.setTotalprice(customphone.getTotalprice()+100.00f);
139     }
140 }
```

```
123         else
124             System.out.println("The storage choice is invalid!");
125
126         System.out.print("Camera (1 - Front Only 2 - Dual): " );
127         input = keyboard.nextInt();
128         if (input == 1)
129             { customphone.setCam(false); }
130         else if (input == 2)
131             { customphone.setCam(true);
132               customphone.setTotalprice(customphone.getTotalprice()+50.00f);
133             }
134         else
135             System.out.println("The camera choice is invalid!");
136
137         System.out.print("Battery Life (1 - 24 2 - 48): " );
138         input = keyboard.nextInt();
139         if (input == 1)
140             { customphone.setBatt_life(24); }
141         else if (input == 2)
142             { customphone.setBatt_life(48);
143               customphone.setTotalprice(customphone.getTotalprice()+25.00f);
144             }
145         else
146             System.out.println("The battery choice is invalid!");
147
148
149         System.out.print("Waterproof (1 - No 2 - Yes): " );
150         input = keyboard.nextInt();
151         if (input == 1)
152             { customphone.setWaterproof(false); }
153         else if (input == 2)
154             { customphone.setWaterproof(true);
155               customphone.setTotalprice(customphone.getTotalprice()+50.00f);
156             }
157         else
158             System.out.println("The waterproof choice is invalid!");
159
160         System.out.print("Port (1 - No 2 - Yes): " );
161         input = keyboard.nextInt();
162         if (input == 1)
163             { customphone.setPort(false); }
164         else if (input == 2)
```

```

165         {    customphone.setPort(true);
166             customphone.setTotalprice(customphone.getTotalprice()+50.00f);
167         }
168     else
169     System.out.println("The port choice is invalid!");
170
171     numphones++;
172     grandtotal += customphone.getTotalprice();
173     customphone = new CellPhone();
174
175     //add the sale
176
177     printReceipt();
178
179     System.out.println("Would you like another phone (Y/N)? : ");
180     morephones = keyboard.next().charAt(0);
181
182     }while(morephones == 'Y');
183     //end of while loop
184
185 } //end of customizePhones function
186
187 //*****PRINT RECEIPT
METHOD*****
188 *
189 *****
*/
190
191 void printReceipt()
192 {
193     //CAPTURE THE DATE TIME - https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html
194     String pattern = "MM-dd-yyyy hh:mm:ss aa";
195     Calendar calendar = Calendar.getInstance();
196     String orderTime = new SimpleDateFormat(pattern).format(calendar.getTime());
197     System.out.println(orderTime);
198
199     //Summary display for all phone sold
200     String tabs2 = "\t\t";
201     String tabs3 = "\t\t\t";
202     String tabs4 = "\t\t\t\t";
203     String starst = "*****";

```

```

204     String starsb = "*****";
205     String stars4 = "****";
206     System.out.println("Thank you for your purchase. You ordered the following customizations:");
207
208     System.out.println(starst + "RECEIPT" + starst);
209     System.out.println(stars4 + tabs2 + tabs4 + purchasers[pi].fullname + tabs3 + tabs4 + stars4);
210     System.out.println(stars4+ tabs2 + tabs4 + orderTime + tabs2+ tabs4 + stars4);
211     System.out.println(stars4+ tabs2 + tabs4 + numphones + " cell phone(s)" + tabs3+ tabs4 + stars4);
212     System.out.println(stars4+ tabs2 + tabs4 + "Total Price $" + grandtotal + tabs2+ tabs4 + stars4 );
213     System.out.println(starsb + "END OF RECEIPT" + starsb);
214
215 } //end of printReceipt function
216
217
218 /*****PRINT SALES METHOD*****/
219 *
220 *****/
221
222 void printSales()
223 {
224     String tabs4 = "\t\t\t\t";
225     String stars4 = "****";
226     String starsb = "*****";
227     //Loop for each order
228     grandtotal = 0;
229     for (int i = 0; i < sales.size(); i++)
230     {
231         System.out.print(stars4 + " ");
232         System.out.print(sales.get(i).getOrdernum());
233         System.out.print(sales.get(i).getOrdertotal());
234         System.out.print(sales.get(i).getOrderdate());
235         grandtotal += sales.get(i).getOrdertotal();
236         System.out.println(tabs4+stars4);
237     }
238     System.out.println("Grand Total $" + grandtotal);
239     System.out.println(starsb + "END OF REPORT" + starsb);
240 } //end of printSales function
241
242 /*****LOAD AGENTS METHOD*****/
243 *
244 *****/
245

```

```

246 void loadAgents() throws IOException
247 {
248     //create the file for reading from
249     FileReader fr=new FileReader("agents.txt");
250     BufferedReader br = new BufferedReader(fr);
251
252     //create the variables for the file
253     String f = "";
254     String u = "";
255     String p = "";
256     int id = 0;
257     int ai = 0;
258
259     String eachLine = "";
260     StringTokenizer st;
261     eachLine = br.readLine(); //read the first line
262
263     while (eachLine != null) //tests for the eof
264     {
265         st = new StringTokenizer(eachLine, ",");
266         while (st.hasMoreTokens())
267         { //remember the order of the text file
268
269             id = Integer.parseInt(st.nextToken());
270             f = st.nextToken();
271             u = st.nextToken();
272             p = st.nextToken();
273
274
275             purchasers[ai] = new Customer(id,f,u,p);//add to the array
276             ai++;
277             eachLine = br.readLine(); //read the next line
278         } //end of reading one line
279     } //end of reading the file
280     br.close();
281     System.out.println("Agents Loaded");
282 } //end of loadCars()
283
284
285 /*****GET SYSTEM DATE METHOD*****/
286 *
287 *****/

```

```
288
289
290 String getSystemDate()
291 {
292     //CAPTURE THE DATE TIME - https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html
293     String pattern = "MM-dd-yyyy hh:mm:ss aa";
294     Calendar calendar = Calendar.getInstance();
295     String orderTime = new SimpleDateFormat(pattern).format(calendar.getTime());
296     System.out.println(orderTime);
297     return orderTime;
298 }
299
300
301 /*****WRITE SALES*****/
302 *
303 *****/
304
305
306 void writeSales() throws IOException
307 {
308     //sales is an array list - get() method gets the full car transaction
309
310     FileWriter fw = new FileWriter("sales.dat", true);
311     BufferedWriter bw = new BufferedWriter(fw);
312
313     // Write each sales record to the file
314     for (int i = 0; i < sales.size(); i++)
315     {
316         bw.write(sales.get(i).getOrdernum() + "," +
317                 sales.get(i).getOrdertotal() + "," +
318                 sales.get(i).getOrderdate() + "\n");
319     }
320     System.out.println("File written Successfully");
321     bw.close();
322
323 } //end of WriteSales()
324
325
326
327
328
329 } // end of Manufacturer class
```

Manufacturer.java

Monday, November 22, 2021, 12:33 AM

330

```
1 package pkgCustomPhones;
2
3 public class Order {
4
5     private int ordernum;
6     private float ordertotal;
7     private String orderdate;
8
9     public Order(int on, float ot, String os)
10    {
11        ordernum = on;
12        ordertotal = ot;
13        orderdate = os;
14    }
15
16    public int getOrdernum() {
17        return ordernum;
18    }
19    public void setOrdernum(int ordernum) {
20        this.ordernum = ordernum;
21    }
22    public float getOrdertotal() {
23        return ordertotal;
24    }
25    public void setOrdertotal(float ordertotal) {
26        this.ordertotal = ordertotal;
27    }
28    public String getOrderdate() {
29        return orderdate;
30    }
31    public void setOrderdate(String orderdate) {
32        this.orderdate = orderdate;
33    }
34 }
35
36
37
38
39 }
40
```

```
1 package pkgCustomPhones;
2
3 public class Customer {
4
5     int custid;
6     String username;
7     String password;
8     String fullname;
9
10    Customer()
11    {
12        custid = 0;
13        fullname = "";
14        password = "";
15        username = "";
16    }
17    Customer(int c, String f, String p, String u)
18    {
19        custid = c;
20        fullname = f;
21        password = p;
22        username = u;
23    }
24    public int getCustid() {
25        return custid;
26    }
27    public void setCustid(int custid) {
28        this.custid = custid;
29    }
30    public String getUsername() {
31        return username;
32    }
33    public void setUsername(String username) {
34        this.username = username;
35    }
36    public String getPassword() {
37        return password;
38    }
39    public void setPassword(String password) {
40        this.password = password;
41    }
42    public String getFullname() {
```

Customer.java

Monday, November 22, 2021, 12:36 AM

```
43     return fullname;
44 }
45 public void setFullname(String fullname) {
46     this.fullname = fullname;
47 }
48
49 }
50
```

```
1 package pkgCustomPhones;
2
3 public class CellPhone
4 {
5     private String os;
6     private String color;
7     private float screen_size;
8     private int storage;
9     private boolean cam;
10    private int batt_life;
11    private boolean waterproof;
12    private boolean port;
13    private float totalprice;
14
15    //CellPhone customphone = new CellPhone("Android", 6.5, 64, "Front Only Camera", 1, false, false, false);
16
17    public CellPhone() {
18        super();
19        this.os = "";
20        this.color = "";
21        this.screen_size = 0;
22        this.storage = 0;
23        this.cam = false;
24        this.batt_life = 0;
25        this.waterproof = false;
26        this.port = false;
27        this.totalprice = 0;
28    }
29
30
31    public CellPhone(String os, String color, float screen_size, int storage, boolean cam, int batt_life,
32        boolean waterproof, boolean port, float totalprice) {
33        super();
34        this.os = os;
35        this.color = color;
36        this.screen_size = screen_size;
37        this.storage = storage;
38        this.cam = cam;
39        this.batt_life = batt_life;
40        this.waterproof = waterproof;
41        this.port = port;
42        this.totalprice = totalprice;
```

```
43     }
44
45
46     public String getOs() {
47         return os;
48     }
49
50     public void setOs(String os) {
51         this.os = os;
52     }
53
54     public String getColor() {
55         return color;
56     }
57
58     public void setColor(String color) {
59         this.color = color;
60     }
61
62     public float getScreen_size() {
63         return screen_size;
64     }
65
66     public void setScreen_size(float screen_size) {
67         this.screen_size = screen_size;
68     }
69
70     public int getStorage() {
71         return storage;
72     }
73
74     public void setStorage(int storage) {
75         this.storage = storage;
76     }
77
78     public boolean isCam() {
79         return cam;
80     }
81
82     public void setCam(boolean cam) {
83         this.cam = cam;
84     }
```

```
85
86     public int getBatt_life() {
87         return batt_life;
88     }
89
90     public void setBatt_life(int batt_life) {
91         this.batt_life = batt_life;
92     }
93
94     public boolean isWaterproof() {
95         return waterproof;
96     }
97
98     public void setWaterproof(boolean waterproof) {
99         this.waterproof = waterproof;
100     }
101
102     public boolean isPort() {
103         return port;
104     }
105
106     public void setPort(boolean port) {
107         this.port = port;
108     }
109
110     public float getTotalprice() {
111         return totalprice;
112     }
113
114     public void setTotalprice(float totalprice) {
115         this.totalprice = totalprice;
116     }
117
118
119
120 }
121
```

```
1 package pkgCustomPhones;
2
3 public class CellPhone
4 {
5     private String os;
6     private String color;
7     private float screen_size;
8     private int storage;
9     private boolean cam;
10    private int batt_life;
11    private boolean waterproof;
12    private boolean port;
13    private float totalprice;
14
15    //CellPhone customphone = new CellPhone("Android", 6.5, 64, "Front Only Camera", 1, false, false, false);
16
17    public CellPhone() {
18        super();
19        this.os = "";
20        this.color = "";
21        this.screen_size = 0;
22        this.storage = 0;
23        this.cam = false;
24        this.batt_life = 0;
25        this.waterproof = false;
26        this.port = false;
27        this.totalprice = 0;
28    }
29
30
31    public CellPhone(String os, String color, float screen_size, int storage, boolean cam, int batt_life,
32        boolean waterproof, boolean port, float totalprice) {
33        super();
34        this.os = os;
35        this.color = color;
36        this.screen_size = screen_size;
37        this.storage = storage;
38        this.cam = cam;
39        this.batt_life = batt_life;
40        this.waterproof = waterproof;
41        this.port = port;
42        this.totalprice = totalprice;
```

```
43     }
44
45
46     public String getOs() {
47         return os;
48     }
49
50     public void setOs(String os) {
51         this.os = os;
52     }
53
54     public String getColor() {
55         return color;
56     }
57
58     public void setColor(String color) {
59         this.color = color;
60     }
61
62     public float getScreen_size() {
63         return screen_size;
64     }
65
66     public void setScreen_size(float screen_size) {
67         this.screen_size = screen_size;
68     }
69
70     public int getStorage() {
71         return storage;
72     }
73
74     public void setStorage(int storage) {
75         this.storage = storage;
76     }
77
78     public boolean isCam() {
79         return cam;
80     }
81
82     public void setCam(boolean cam) {
83         this.cam = cam;
84     }
```

```
85
86     public int getBatt_life() {
87         return batt_life;
88     }
89
90     public void setBatt_life(int batt_life) {
91         this.batt_life = batt_life;
92     }
93
94     public boolean isWaterproof() {
95         return waterproof;
96     }
97
98     public void setWaterproof(boolean waterproof) {
99         this.waterproof = waterproof;
100    }
101
102    public boolean isPort() {
103        return port;
104    }
105
106    public void setPort(boolean port) {
107        this.port = port;
108    }
109
110    public float getTotalprice() {
111        return totalprice;
112    }
113
114    public void setTotalprice(float totalprice) {
115        this.totalprice = totalprice;
116    }
117
118
119
120 }
121
```

111,Mahatma Ghandi,ghandi,mghandi@gmail.com
222,Mother Theresa,theresa,mtheresa@gmail.com
333,Kathy Hochul,hochul,khochul@gmail.com