NAME : SYAHIRAH AQILAH BINTI SULAIMAN

MATRIC NO : BI19110068

TOPIC : ORDERING RECORD

**JAVA CODE**

1 import java.awt.\*;  
 2 import java.awt.event.\*;  
 3 import javax.swing.\*;  
 4 import javax.swing.event.\*;  
 5 import java.time.format.DateTimeFormatter;   
 6 import java.time.LocalDateTime;   
 7   
 8 public class MyOrderingSystem extends JPanel {  
 9   
 10 CBBDrinkSelectIndexChanged cbbDrinkSelectIndexChanged;  
 11 CBBFoodSelectIndexChanged cbbFoodSelectIndexChanged;  
 12 CBBDessertSelectIndexChanged cbbDessertSelectIndexChanged;  
 13 ButtonCalculateClicked buttonCalculateClicked;  
 14 ButtonDoneClicked buttonDoneClicked;  
 15 ButtonNewOrderClicked buttonNewOrderClicked;  
 16 double drinkprice = 0;  
 17 double foodprice = 0;  
 18 double dessertprice = 0;  
 19 String drinkname = "Not Selected";  
 20 String foodname = "Not Selected";  
 21 String dessertname = "Not Selected";  
 22 String PaymentType = "Not Selected";  
 23 String DineInOrTakeAway = "Not Selected";  
 24   
 25 // combobox  
 26 private JComboBox jcomboboxDrink;  
 27 private JComboBox jcomboboxFoods;  
 28 private JComboBox jcomboboxDessert;  
 29   
 30 //label  
 31 private JLabel jlabelDrink;  
 32 private JLabel jlabelHeader;  
 33 private JLabel jlabelFood;  
 34 private JLabel jlabelDessert;  
 35 private JLabel jlabelSubtotal;  
 36 private JLabel jlabelPayment;  
 37 private JLabel jlabelOrder;  
 38 private JLabel jlabelRate;  
 39 private JLabel jlabelStatus;  
 40 private JLabel jlabelFeedback;  
 41 private JLabel jlabelPinNo;  
 42 private JLabel jlabelTotalPayment;  
 43   
 44   
 45 //textField  
 46 private JTextField jtextfieldSubtotal;  
 47 private JTextField jtextfieldTotalPayment;  
 48   
 49   
 50 //textArea  
 51 private JTextArea jtextareaOrder;  
 52 private JTextArea jtextareaFeedback;  
 53   
 54 //RadioButton  
 55 private JRadioButton jradiobuttonDineIn;  
 56 private JRadioButton jradiobuttonTakeAway;  
 57   
 58 //Slider   
 59 private JSlider jsliderRate;  
 60   
 61 //PasswordField  
 62 private JPasswordField jpasswordfieldPinNo;  
 63   
 64 //Button  
 65 private JButton jbuttonDone;  
 66 private JButton jbuttonCalculateTotal;  
 67 private JButton jbuttonNewOrder;  
 68   
 69 //RadioButton  
 70 private JRadioButton JradiobuttonCash;  
 71 private JRadioButton JradiobuttonCredit;  
 72   
 73 // GroupRadioButton  
 74 private ButtonGroup groupTakeAwayOrDineIn;  
 75 private ButtonGroup groupCashOrCredit;  
 76   
 77   
 78 public MyOrderingSystem() {  
 79   
 80 cbbDrinkSelectIndexChanged = new CBBDrinkSelectIndexChanged(this);  
 81 cbbFoodSelectIndexChanged = new CBBFoodSelectIndexChanged(this);  
 82 cbbDessertSelectIndexChanged = new CBBDessertSelectIndexChanged(this);  
 83 buttonCalculateClicked = new ButtonCalculateClicked(this);  
 84 buttonDoneClicked = new ButtonDoneClicked(this);  
 85 buttonNewOrderClicked = new ButtonNewOrderClicked(this);  
 86   
 87   
 88 //construct preComponents  
 89 String[] jcomboboxDrinkItems = {"[Select]", "Iced Chocolate", "Iced Cappucino", "Iced Milk Tea"};  
 90 String[] jcomboboxFoodsItem = {"[Select]", "Spagetti", "Burger", "Fries"};  
 91 String[] jcomboboxDessertItems = {"[Select]", "Chocolate Cake", "Cheese Cake", "Ice Cream"};  
 92   
 93 //construct components  
 94 jcomboboxDrink = new JComboBox (jcomboboxDrinkItems);  
 95 jcomboboxFoods = new JComboBox (jcomboboxFoodsItem);  
 96 jcomboboxDessert = new JComboBox (jcomboboxDessertItems);  
 97   
 98 jlabelDrink = new JLabel ("Drinks");  
 99 jlabelHeader = new JLabel ("Kor's Simple Cafe");  
100 jlabelFood = new JLabel ("Foods");  
101 jlabelDessert = new JLabel ("Dessert");  
102 jlabelSubtotal = new JLabel ("Subtotal (RM):");  
103 jlabelPayment = new JLabel ("Payment");  
104 jlabelOrder = new JLabel ("Order");  
105 jlabelRate = new JLabel ("Rate");  
106 jlabelStatus = new JLabel ("Status");  
107 jlabelFeedback = new JLabel ("Feedback");  
108 jlabelPinNo = new JLabel ("Pin No");  
109 jlabelTotalPayment = new JLabel ("Total Payment (RM):");  
110   
111 jtextfieldSubtotal = new JTextField (5);  
112 jtextfieldTotalPayment = new JTextField (5);  
113   
114 jtextareaOrder = new JTextArea (50, 5);  
115 jtextareaFeedback = new JTextArea (5, 5);  
116   
117 jradiobuttonDineIn = new JRadioButton ("Dine In");  
118 jradiobuttonTakeAway = new JRadioButton ("Take Away");  
119   
120 jsliderRate = new JSlider (0, 10,6);  
121   
122 jpasswordfieldPinNo = new JPasswordField (5);  
123   
124 jbuttonDone = new JButton ("Done");  
125 jbuttonCalculateTotal = new JButton ("Calculate");  
126 jbuttonNewOrder = new JButton ("New Order");  
127   
128 JradiobuttonCash = new JRadioButton ("Cash");  
129 JradiobuttonCredit = new JRadioButton ("Credit/Debit Card");  
130   
131 groupTakeAwayOrDineIn = new ButtonGroup();  
132 groupCashOrCredit =new ButtonGroup();  
133   
134 //set components properties  
135 jsliderRate.setOrientation (JSlider.HORIZONTAL);  
136 jsliderRate.setMinorTickSpacing (1);  
137 jsliderRate.setMajorTickSpacing (1);  
138 jsliderRate.setPaintTicks (true);  
139 jsliderRate.setPaintLabels (true);  
140 jsliderRate.setToolTipText ("Rate");  
141   
142 //adjust size and set layout  
143 setPreferredSize (new Dimension (666, 538));  
144 setLayout (null);  
145   
146 //add components  
147 groupTakeAwayOrDineIn.add(jradiobuttonDineIn);  
148 groupTakeAwayOrDineIn.add(jradiobuttonTakeAway);  
149 groupCashOrCredit.add(JradiobuttonCash);  
150 groupCashOrCredit.add(JradiobuttonCredit);  
151   
152 add (jcomboboxDrink);  
153 add (jlabelDrink);  
154 add (jlabelHeader);  
155 add (jlabelFood);  
156 add (jlabelDessert);  
157 add (jcomboboxFoods);  
158 add (jcomboboxDessert);  
159 add (jtextareaOrder);  
160 add (jlabelPayment);  
161 add (jlabelTotalPayment);  
162 add (jlabelOrder);  
163 add (jsliderRate);  
164 add (jlabelStatus);  
165 add (jlabelFeedback);  
166 add (jlabelPinNo);  
167 add (jtextareaFeedback);  
168 add (jradiobuttonDineIn);  
169 add (jradiobuttonTakeAway);  
170 add (jpasswordfieldPinNo);  
171 add (jbuttonDone);  
172 add (jbuttonCalculateTotal);  
173 add (jbuttonNewOrder);  
174 add (jlabelSubtotal);  
175 add (jtextfieldSubtotal);  
176 add (JradiobuttonCash);  
177 add (JradiobuttonCredit);  
178 add (jlabelRate);  
179 add (jtextfieldTotalPayment);  
180   
181 //set component bounds (only needed by Absolute Positioning)  
182 jcomboboxDrink.setBounds (100, 75, 100, 25);  
183 jlabelDrink.setBounds (35, 75, 100, 25);  
184 jlabelHeader.setBounds (260, 20, 122, 25);  
185 jlabelFood.setBounds (35, 125, 100, 25);  
186 jlabelDessert.setBounds (35, 175, 100, 25);  
187 jlabelRate.setBounds (490, 415, 100, 25);  
188 jcomboboxFoods.setBounds (100, 125, 100, 25);  
189 jcomboboxDessert.setBounds (100, 175, 100, 25);  
190 jtextareaOrder.setBounds (40, 260, 270, 360);  
191 jlabelPayment.setBounds (470, 65, 100, 25);  
192 jlabelOrder.setBounds (150, 230, 100, 25);  
193 jsliderRate.setBounds (430, 450, 150, 50);  
194 jlabelStatus.setBounds (280, 65, 100, 25);  
195 jlabelFeedback.setBounds (470, 230, 100, 25);  
196 jlabelPinNo.setBounds (470, 165, 100, 25);  
197 jtextareaFeedback.setBounds (390, 260, 210, 70);  
198 jradiobuttonDineIn.setBounds (255, 100, 100, 25);  
199 jradiobuttonTakeAway.setBounds (255, 135, 100, 25);  
200 jpasswordfieldPinNo.setBounds (445, 190, 100, 25);  
201 jbuttonDone.setBounds (200, 660, 100, 25);  
202 jbuttonCalculateTotal.setBounds (50, 660, 100, 25);  
203 jbuttonNewOrder.setBounds (100, 45, 100, 25);  
204 jlabelSubtotal.setBounds (270, 165, 100, 25);  
205 jtextfieldSubtotal.setBounds (260, 190, 100, 25);  
206 JradiobuttonCash.setBounds (440, 100, 100, 25);  
207 JradiobuttonCredit.setBounds (440, 135, 128, 25);  
208 jlabelTotalPayment.setBounds (50, 630, 120, 25);  
209 jtextfieldTotalPayment.setBounds (180, 630, 120, 25);  
210   
211   
212 jcomboboxDrink.addActionListener(cbbDrinkSelectIndexChanged);  
213 jcomboboxFoods.addActionListener(cbbFoodSelectIndexChanged);  
214 jcomboboxDessert.addActionListener(cbbDessertSelectIndexChanged);  
215 jbuttonCalculateTotal.addActionListener(buttonCalculateClicked);  
216 jbuttonDone.addActionListener(buttonDoneClicked);  
217 jbuttonNewOrder.addActionListener(buttonNewOrderClicked);  
218   
219 jradiobuttonDineIn.setSelected(true);  
220 JradiobuttonCash.setSelected(true);  
221   
222 }  
223   
224   
225 public static void main (String[] args) {  
226 JFrame frame = new JFrame ("MyOrderingSystem");  
227 frame.setDefaultCloseOperation (JFrame.EXIT\_ON\_CLOSE);  
228 frame.getContentPane().add (new MyOrderingSystem());  
229 frame.pack();  
230 frame.setVisible (true);  
231 frame.setSize(700,750);  
232 }  
233   
234 private class CBBDrinkSelectIndexChanged implements ActionListener   
235 {  
236 MyOrderingSystem MOS;  
237   
238 CBBDrinkSelectIndexChanged( MyOrderingSystem mos)   
239 {   
240 this.MOS=mos;   
241 }   
242   
243 public void actionPerformed(ActionEvent event)   
244 {  
245 JComboBox<String> combo = (JComboBox<String>) event.getSource();  
246 String selectedDrink = (String) combo.getSelectedItem();  
247   
248 if (selectedDrink.equals("Iced Chocolate"))   
249 {  
250 drinkname = "Iced Chocolate";  
251 drinkprice = 3.0;  
252 } else if (selectedDrink.equals("Iced Cappucino"))   
253 {  
254 drinkname = "Iced Cappucino";  
255 drinkprice = 4.0;  
256 } else if (selectedDrink.equals("Iced Milk Tea"))   
257 {  
258 drinkname = "Iced Milk Tea";  
259 drinkprice = 2.5;  
260 }   
261 MOS.jtextfieldSubtotal.setText(Double.toString(drinkprice + foodprice + dessertprice));   
262 }  
263 }  
264   
265 private class CBBFoodSelectIndexChanged implements ActionListener   
266 {  
267 MyOrderingSystem MOS;  
268   
269 CBBFoodSelectIndexChanged( MyOrderingSystem mos)   
270 {   
271 this.MOS=mos;   
272 }   
273   
274 public void actionPerformed(ActionEvent event)   
275 {  
276 JComboBox<String> combo = (JComboBox<String>) event.getSource();  
277 String selectedDrink = (String) combo.getSelectedItem();  
278   
279 if (selectedDrink.equals("Spagetti"))   
280 {  
281 foodname = "Spagetti";  
282 foodprice = 7.0;  
283 } else if (selectedDrink.equals("Burger"))   
284 {  
285 foodname = "Burger";  
286 foodprice = 5.0;  
287 } else if (selectedDrink.equals("Fries"))   
288 {  
289 foodname = "Fries";  
290 foodprice = 3.0;  
291 }   
292 MOS.jtextfieldSubtotal.setText(Double.toString(drinkprice + foodprice + dessertprice));   
293 }  
294 }  
295   
296 private class CBBDessertSelectIndexChanged implements ActionListener   
297 {  
298 MyOrderingSystem MOS;  
299   
300 CBBDessertSelectIndexChanged( MyOrderingSystem mos)   
301 {   
302 this.MOS=mos;   
303 }   
304   
305 public void actionPerformed(ActionEvent event)   
306 {  
307 JComboBox<String> combo = (JComboBox<String>) event.getSource();  
308 String selectedDrink = (String) combo.getSelectedItem();  
309   
310 if (selectedDrink.equals("Chocolate Cake"))   
311 {  
312 dessertname = "Chocolate Cake";  
313 dessertprice = 3.0;  
314 } else if (selectedDrink.equals("Cheese Cake"))   
315 {  
316 dessertname = "Cheese Cake";  
317 dessertprice = 5.0;  
318 } else if (selectedDrink.equals("Ice Cream"))   
319 {  
320 dessertname = "Ice Cream";  
321 dessertprice = 2.0;  
322 }   
323 MOS.jtextfieldSubtotal.setText(Double.toString(drinkprice + foodprice + dessertprice));   
324 }  
325 }  
326   
327 private class ButtonCalculateClicked implements ActionListener  
328 {  
329 MyOrderingSystem MOS;  
330   
331 ButtonCalculateClicked( MyOrderingSystem mos)   
332 {   
333 this.MOS=mos;   
334 }   
335   
336 public void actionPerformed(ActionEvent event)   
337 {  
338 double TotalTax = MOS.jsliderRate.getValue();  
339 double PaymentPercentage = 1+ TotalTax/100;  
340 double Subtotal = (drinkprice + foodprice + dessertprice);  
341 double Total = PaymentPercentage\* Subtotal;  
342 String STRTotal = String.format("%.2f", Total);  
343 if ( MOS.jradiobuttonDineIn.isSelected() == true )  
344 {  
345 DineInOrTakeAway = "Dine In";  
346 } else if (MOS.jradiobuttonTakeAway.isSelected() == true)  
347 {  
348 DineInOrTakeAway = "Take Away";  
349 }  
350   
351   
352 if ( MOS.JradiobuttonCash.isSelected() == true )  
353 {  
354 PaymentType = "CASH";  
355 } else if (MOS.JradiobuttonCredit.isSelected() == true)  
356 {  
357 PaymentType = "Credit/Debit";  
358 }  
359   
360 if(Subtotal != 0)  
361 {  
362 DateTimeFormatter dtfDate = DateTimeFormatter.ofPattern("yyyy/MM/dd");   
363 DateTimeFormatter dtfTime = DateTimeFormatter.ofPattern("HH:mm:ss");   
364   
365 LocalDateTime now = LocalDateTime.now();   
366 jtextareaOrder.setText("");  
367 jtextareaOrder.append("\n");   
368 jtextareaOrder.append(" Kor's Simple Cafe \n");  
369 jtextareaOrder.append("\n");  
370 jtextareaOrder.append(" Date: "+dtfDate.format(now) +"\n");  
371 jtextareaOrder.append(" Time: "+dtfTime.format(now)+"\n");  
372 jtextareaOrder.append("\n");  
373 jtextareaOrder.append(" Drink: "+drinkname+"\n");  
374 jtextareaOrder.append(" RM: "+drinkprice+"\n");  
375 jtextareaOrder.append(" Food: "+foodname+"\n");  
376 jtextareaOrder.append(" RM: "+foodprice+"\n");  
377 jtextareaOrder.append(" Dessert: "+dessertname+"\n");  
378 jtextareaOrder.append(" RM: "+dessertprice+"\n");  
379 jtextareaOrder.append("\n");  
380 jtextareaOrder.append(" Payment Type: "+PaymentType+"\n");  
381 jtextareaOrder.append(" Note: "+DineInOrTakeAway+"\n");  
382 jtextareaOrder.append("\n");   
383 jtextareaOrder.append(" Subtotal (RM):"+Subtotal+"\n");  
384 jtextareaOrder.append(" Tax: "+TotalTax+"\n");  
385 jtextareaOrder.append(" Total (RM): "+STRTotal+"\n");  
386 } else if(Subtotal == 0)  
387 {  
388 jtextareaOrder.setText("");  
389 jtextareaOrder.append("No Item Selected");  
390 }  
391 }  
392   
393 }  
394 private class ButtonDoneClicked implements ActionListener  
395 {  
396 MyOrderingSystem MOS;  
397   
398 ButtonDoneClicked( MyOrderingSystem mos)   
399 {   
400 this.MOS=mos;   
401 }   
402   
403 public void actionPerformed(ActionEvent event)   
404 {  
405 double TotalPayment = Double.parseDouble(MOS.jtextfieldTotalPayment.getText());  
406 double TotalTax = MOS.jsliderRate.getValue();  
407 double PaymentPercentage = 1+ TotalTax/100;  
408 double Subtotal = (drinkprice + foodprice + dessertprice);  
409 double Total = PaymentPercentage\* Subtotal;  
410 double Balance = TotalPayment - Total;  
411 String STRBalance = String.format("%.2f", Balance);  
412 if(TotalPayment != 0 && TotalPayment >=Total)  
413 {  
414 jtextareaOrder.append(" Payment (RM):"+TotalPayment+"\n");  
415 jtextareaOrder.append(" Balance (RM): "+STRBalance+"\n");  
416 jtextareaOrder.append(" THANK YOU !");  
417 }  
418   
419 }  
420   
421 }  
422   
423 private class ButtonNewOrderClicked implements ActionListener  
424 {  
425 MyOrderingSystem MOS;  
426   
427 ButtonNewOrderClicked( MyOrderingSystem mos)   
428 {   
429 this.MOS=mos;   
430 }   
431   
432 public void actionPerformed(ActionEvent event)   
433 {  
434 jcomboboxDrink.setSelectedItem("[Select]");   
435 jcomboboxFoods.setSelectedItem("[Select]");   
436 jcomboboxDessert.setSelectedItem("[Select]");   
437 drinkprice = 0;  
438 foodprice = 0;  
439 dessertprice = 0;  
440 drinkname = "Not Selected";  
441 foodname = "Not Selected";  
442 dessertname = "Not Selected";  
443 PaymentType = "Not Selected";  
444 DineInOrTakeAway = "Not Selected";  
445 jtextfieldTotalPayment.setText("");  
446 jtextareaOrder.setText("");   
447 jtextareaFeedback.setText("");   
448 jpasswordfieldPinNo.setText("");   
449 jtextfieldSubtotal.setText("");   
450 }  
451   
452 }   
453 }  
454

**OBJECT ORIENTED CONCEPT IMPLEMENTATION**

* ABSTRACTION

Abstraction aims to hide complexity from the users and show them only the relevant information. Abstraction in Java hides the underlying complexity of data and also help to avoid repetitive code.

* ENCAPSULATION

Encapsulation allows us to protect the data stored in a class from system-wide access. As its name suggests, it safeguards the internal contents of a class like a real-life capsule. Encapsulation in Java restricts direct access to data members(fields) of a class. Besides, the fields are set to private.

* INHERITANCE

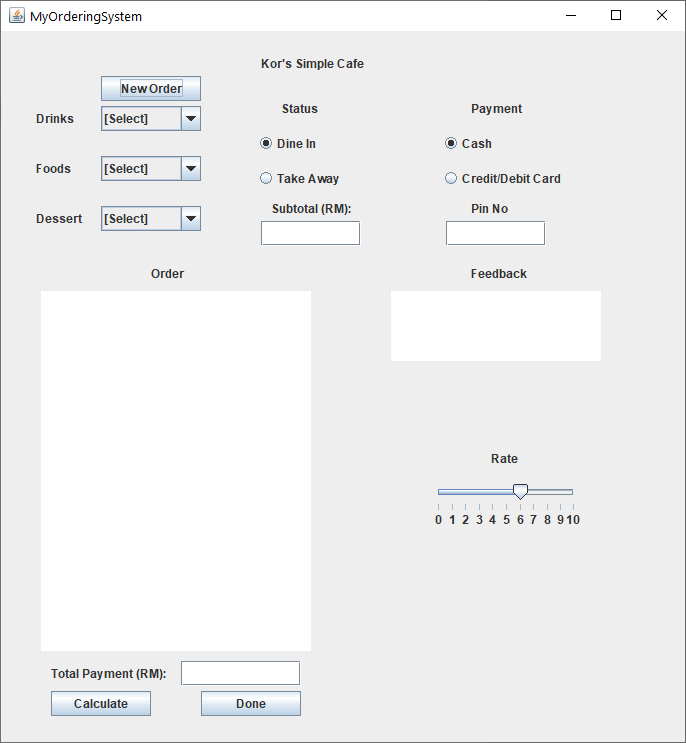
Inheritance make it possible to create a child class that inherits the fields and methods of the parent class. The child class can override the values and the methods of the parent class, however it is not necessary. Inherintance in Java implements “Don’t Repeat Yourself” programming principle and improves code reusability.

* POLYMORPHISM

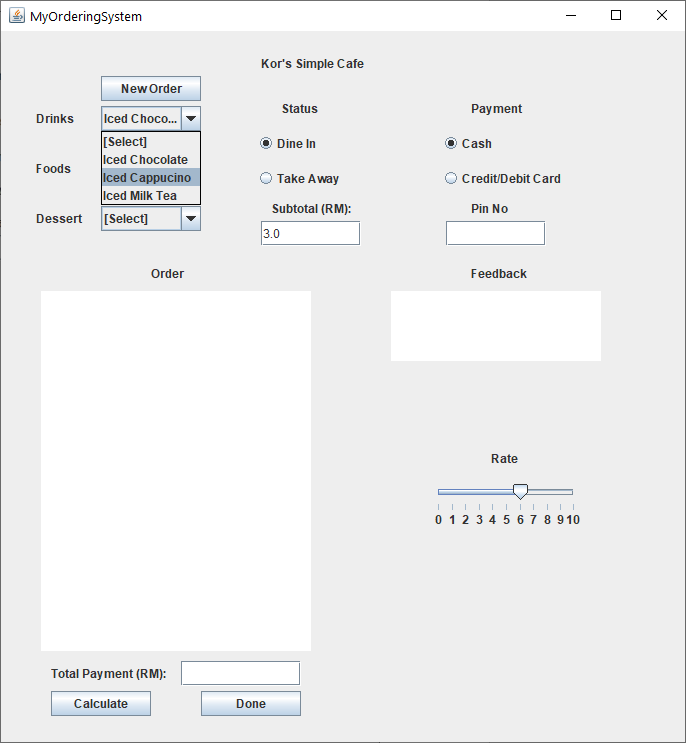
Polymorphism makes it possible to use the name entity in different forms. In Java, this means that you can declare several methods with the same name until they are different in certain characterictics. Java provides us with two ways to implement polymorphism that is method overloading and method overriding.

**USER MANUAL**

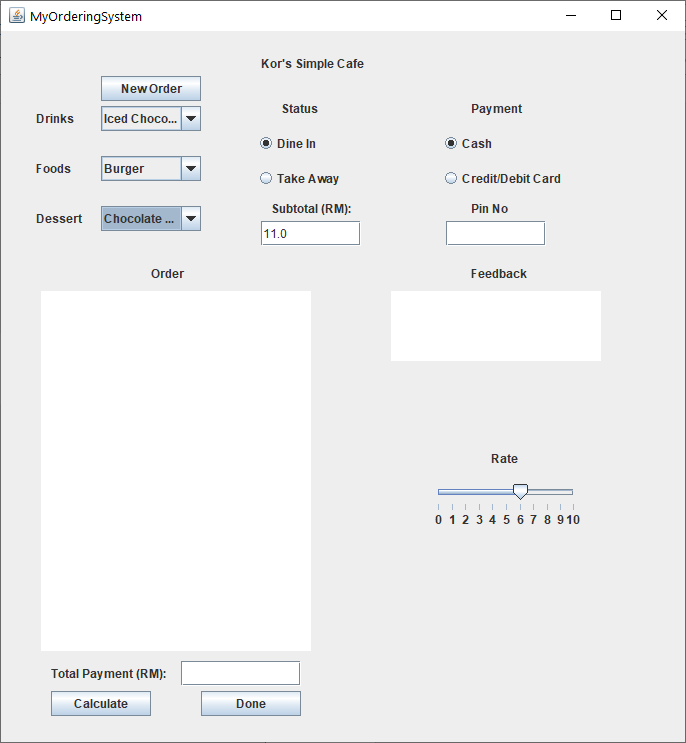
1. User need to click the button “New Order” to reset the data.



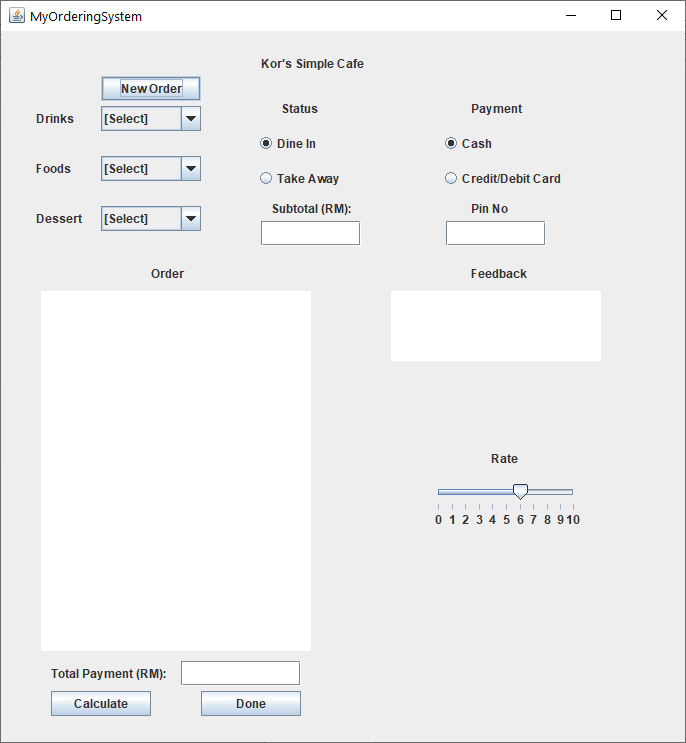
2. Select the drinks and the subtotal will directly changes based on the price.



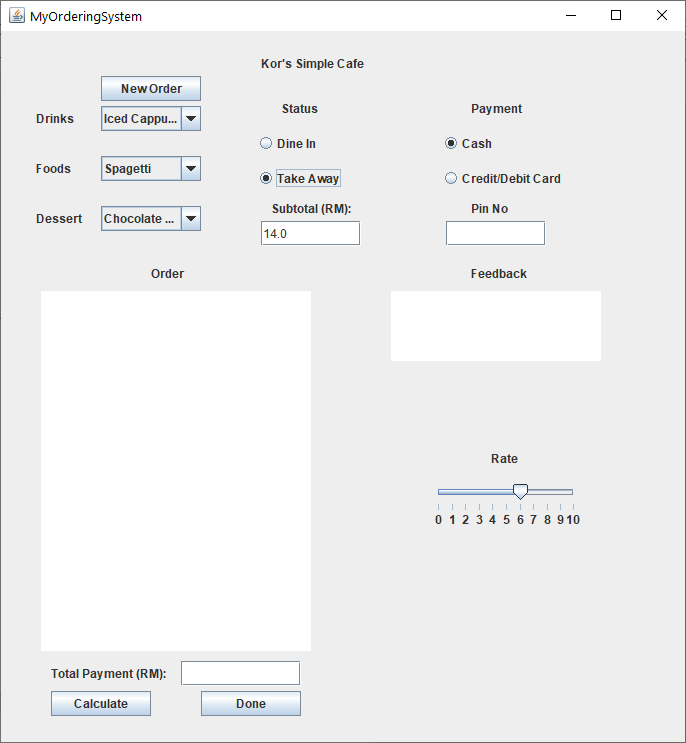
3. Then select the foods and dessert and the subtotal will directly show the current subtotal of the orders.



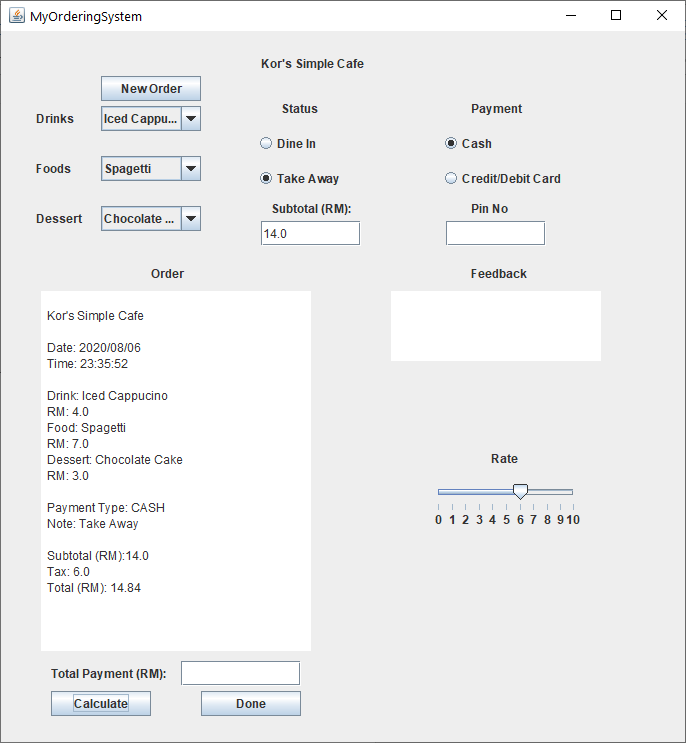
4. When the customer want to cancel order or change the order, user need to click the button “New Order” so that the subtotal will directly changes.



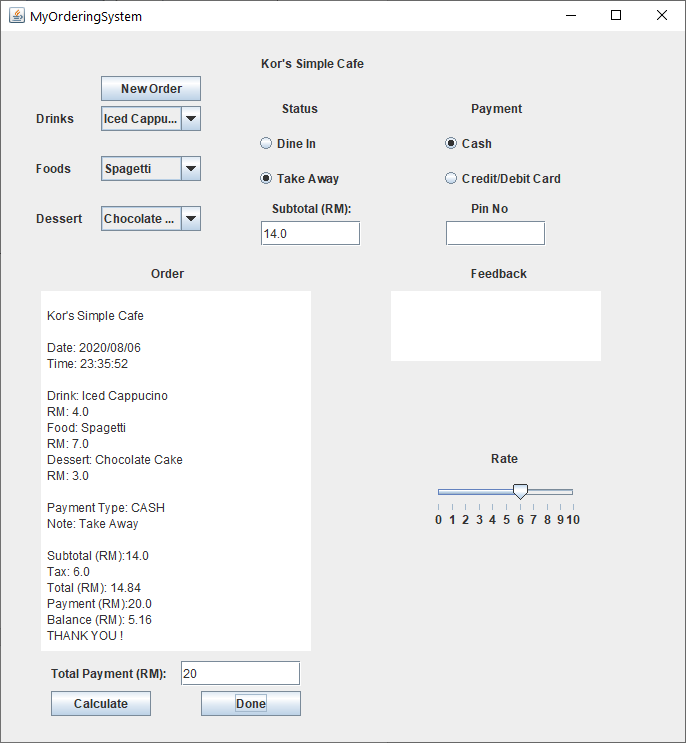
5. Select the status whether dine in or take away.



6. Click the button “Calculate” . The box will show the information such as the date, time, orders , status, payment method and also the total.



7. Insert the total payment only if the payment method is cash. If the payment method is credit/debit card, user need to insert the pin no and click the button “Done”. The payment and balance will be directly added to the box but if the box is empty or the total payment is less than the total, there is nothing will be display in the box



8. Last step, user can comment the feedback and give rating for the serve and foods taste.

