UMS
UNIVERSITI MALAYSIA SABAH

INDIVIDUAL PROJECT 2

OBJECT ORIENTED PROGRAMMING

SEMESTER 2 2019/2020

DUEDATE:	**06/08/2020**

SECTION : 1

SUBJECT CODE : KK14203

SUBJECT : OBJECT ORIENTED PROGRAMMING

LECTURER : DR. MOHD SHAMRIE SAININ

NO.	STUDENT NAME	MATRIC NO.
1.	ZHARIFF BIN OMARHATTA	BI19110069

MANUAL FOR USER

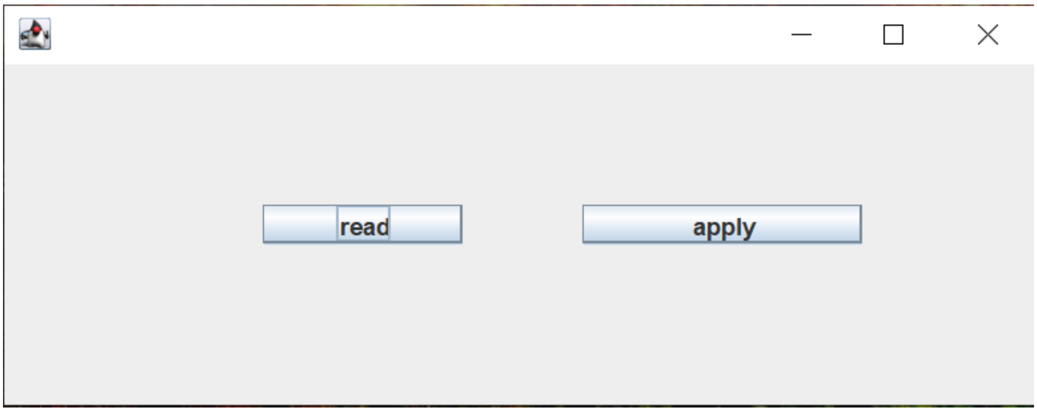
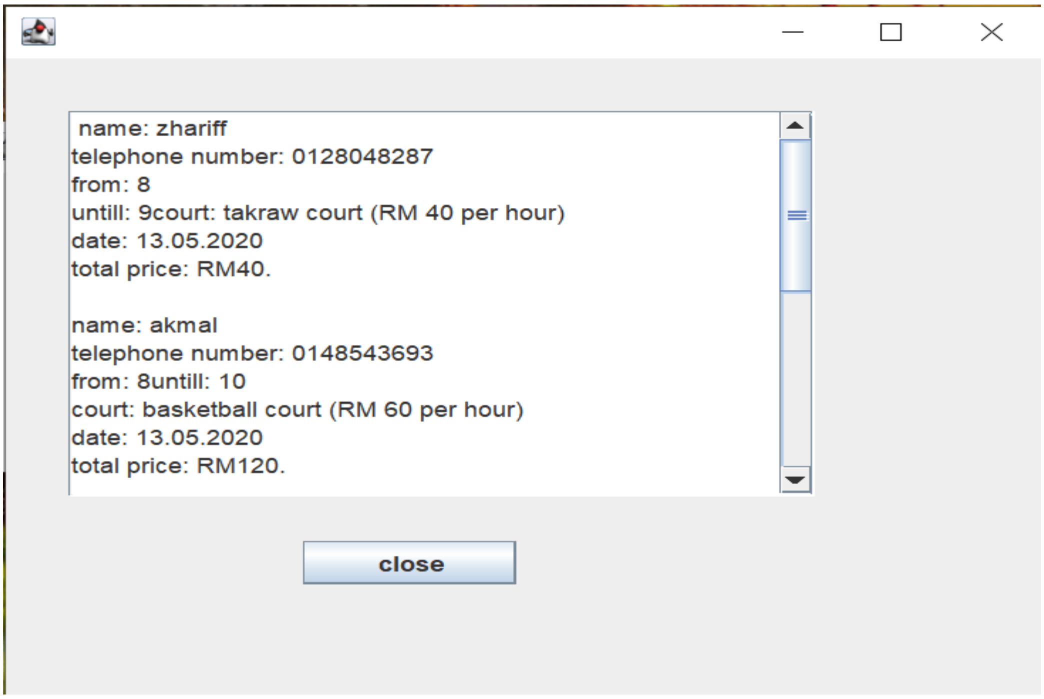
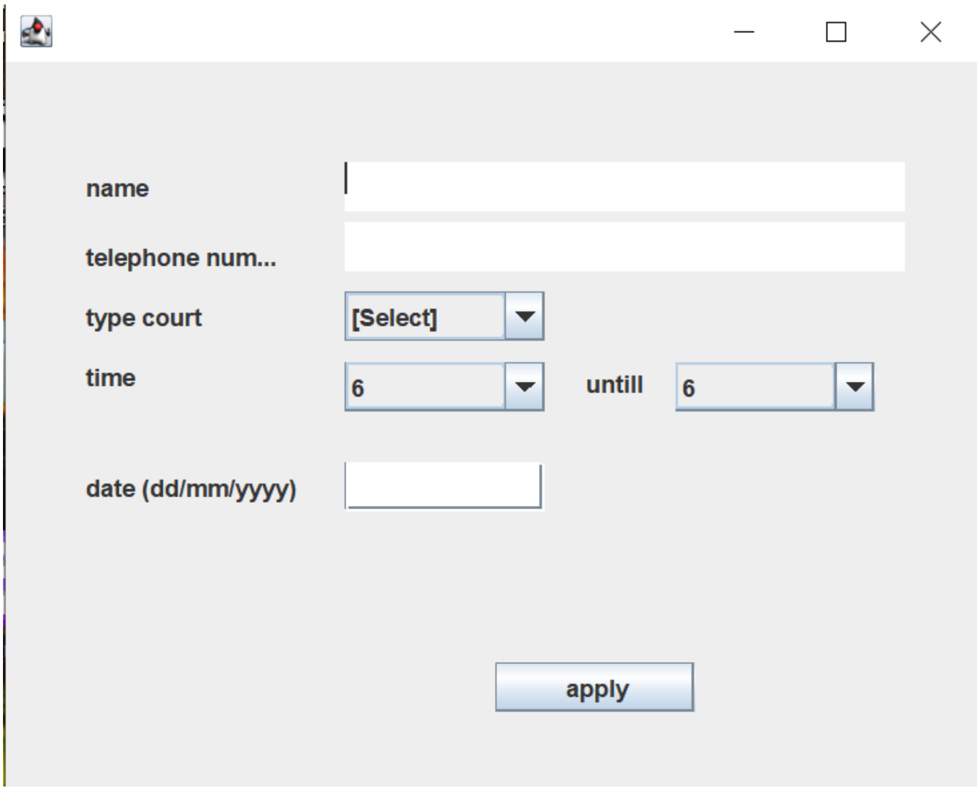


Figure 1: user is given option whether to read all successful application or make a new application.



```
 name: zhariff
telephone number: 0128048287
from: 8
untill: 9court: takraw court (RM 40 per hour)
date: 13.05.2020
total price: RM40.

name: akmal
telephone number: 0148543693
from: 8untill: 10
court: basketball court (RM 60 per hour)
date: 13.05.2020
total price: RM120.
```

Figure 2: If user chooses to read, all successful application will be shown.



name

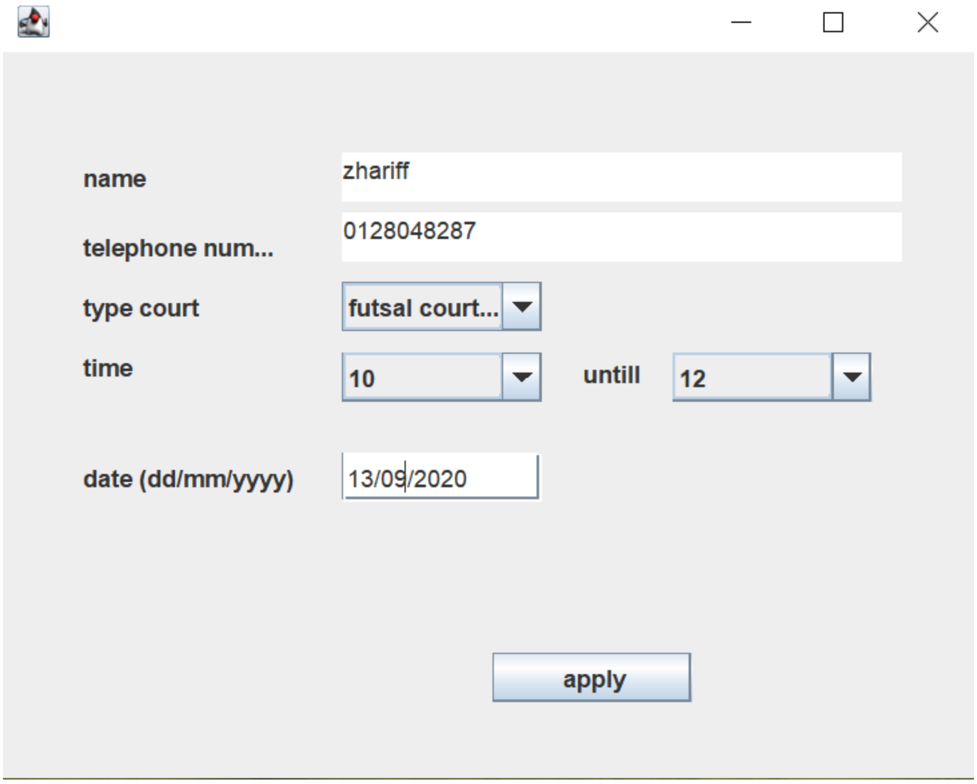
telephone num...

type court [Select]

time 6 untill 6

date (dd/mm/yyyy)

Figure 3: If user chooses to make a new application a form will be pop up.



name zhariff

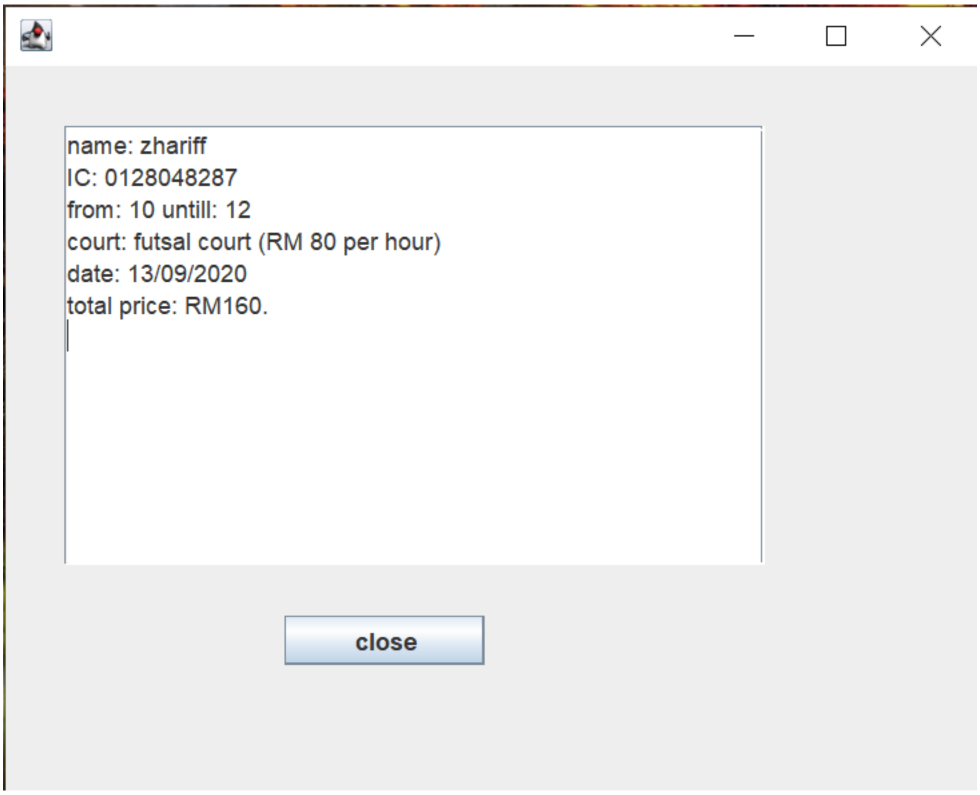
telephone num... 0128048287

type court futsal court...

time 10 untill 12

date (dd/mm/yyyy) 13/09/2020

Figure 4: This is an example of complete application form.

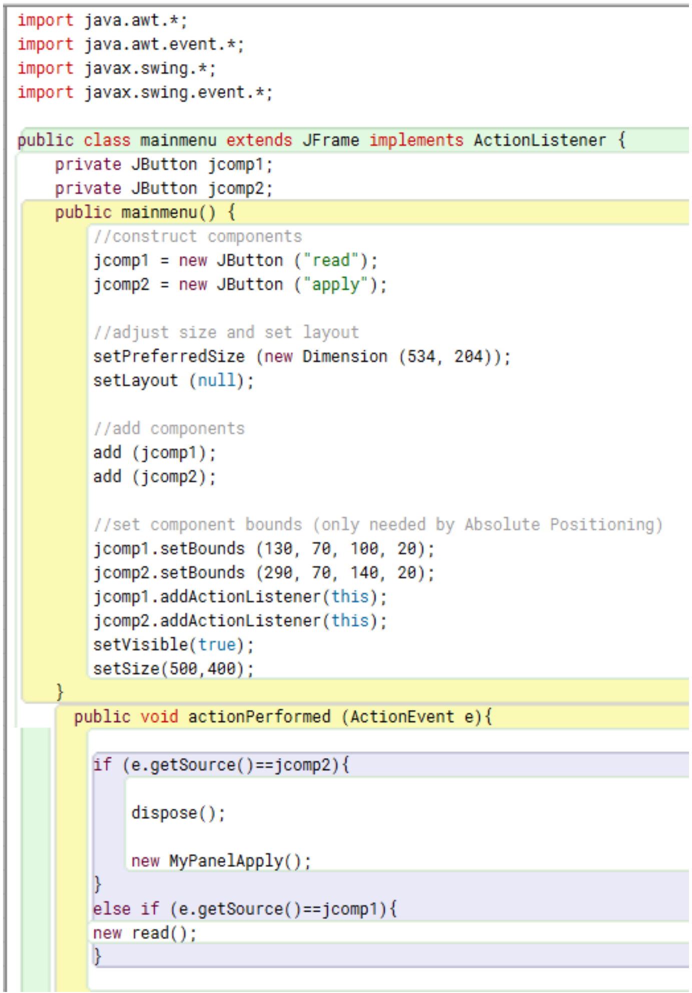


```
name: zhariff
IC: 0128048287
from: 10 untill: 12
court: futsal court (RM 80 per hour)
date: 13/09/2020
total price: RM160.
```

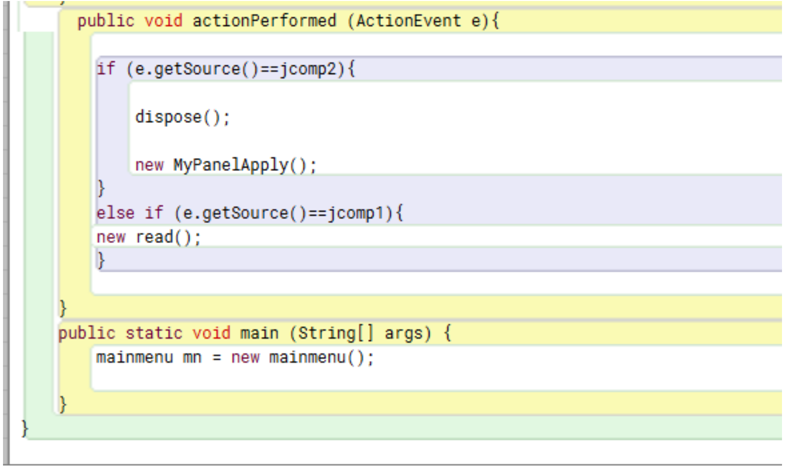
Figure 5: After the application is successful a receipt will be shown including the total price.

JAVA CODE

Main class

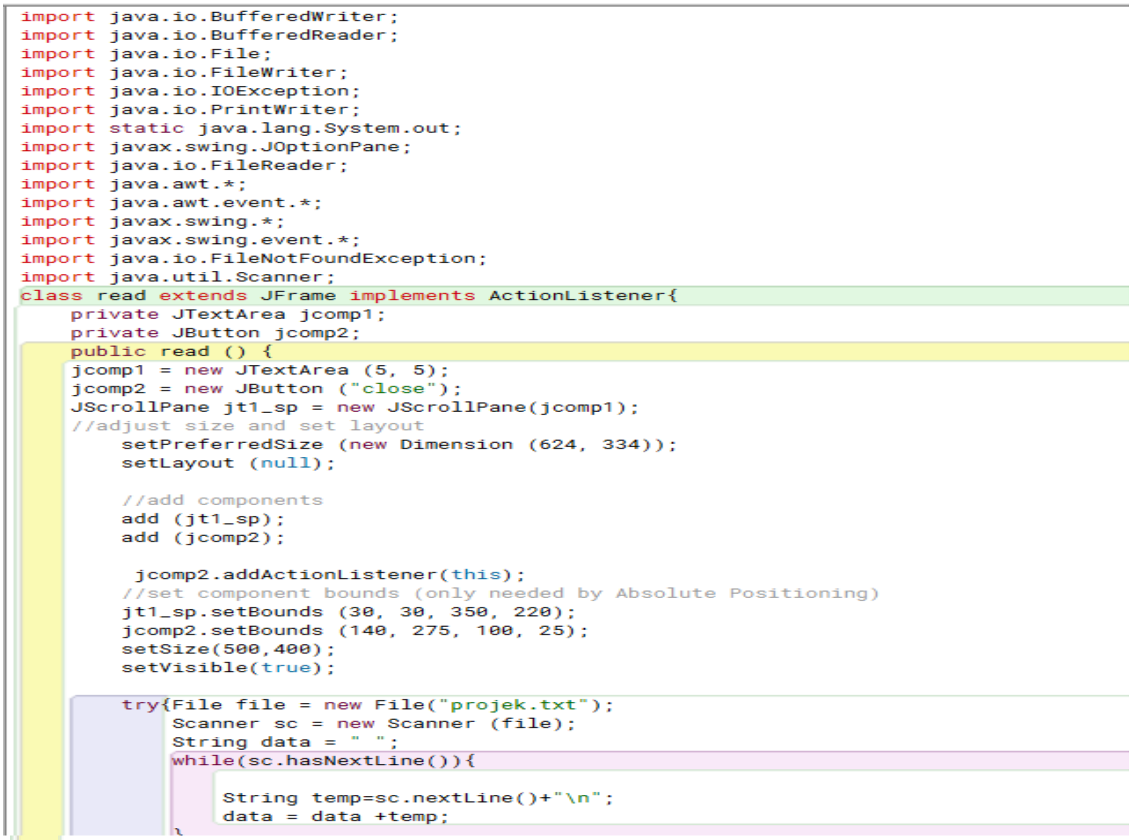


```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;

public class mainmenu extends JFrame implements ActionListener {
    private JButton jcomp1;
    private JButton jcomp2;
    public mainmenu() {
        //construct components
        jcomp1 = new JButton ("read");
        jcomp2 = new JButton ("apply");

        //adjust size and set layout
        setPreferredSize (new Dimension (534, 204));
        setLayout (null);

        //add components
        add (jcomp1);
        add (jcomp2);

        //set component bounds (only needed by Absolute Positioning)
        jcomp1.setBounds (130, 70, 100, 20);
        jcomp2.setBounds (290, 70, 140, 20);
        jcomp1.addActionListener(this);
        jcomp2.addActionListener(this);
        setVisible(true);
        setSize(500,400);
    }
      public void actionPerformed (ActionEvent e){

        if (e.getSource()==jcomp2){

            dispose();

            new MyPanelApply();
        }
        else if (e.getSource()==jcomp1){
        new read();
        }
```

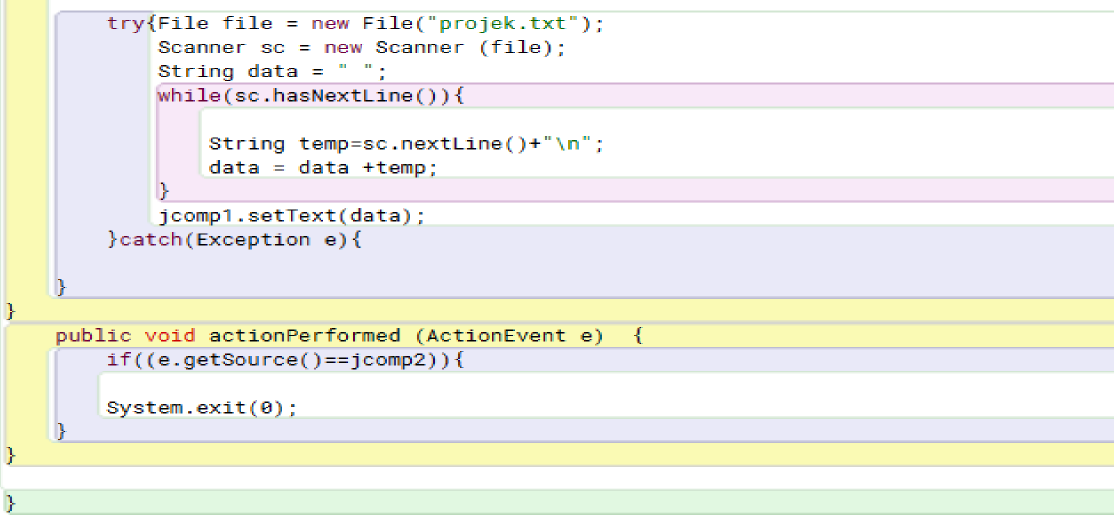


```
      public void actionPerformed (ActionEvent e){

        if (e.getSource()==jcomp2){

            dispose();

            new MyPanelApply();
        }
        else if (e.getSource()==jcomp1){
        new read();
        }

    }
    public static void main (String[] args) {
        mainmenu mn = new mainmenu();

    }
}
```

Read class

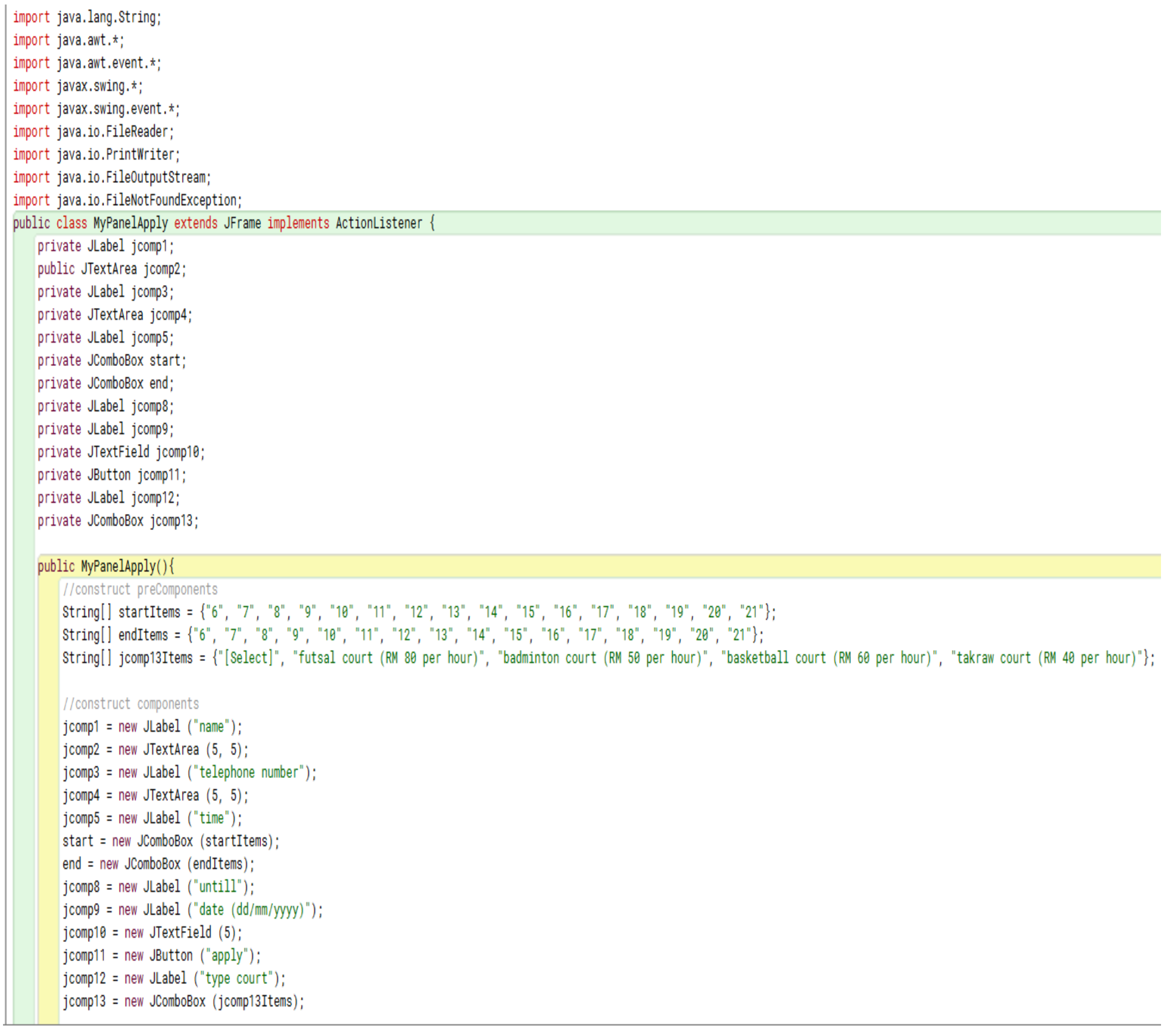


```
import java.io.BufferedWriter;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import static java.lang.System.out;
import javax.swing.JOptionPane;
import java.io.FileReader;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import java.io.FileNotFoundException;
import java.util.Scanner;
class read extends JFrame implements ActionListener{
    private JTextArea jcomp1;
    private JButton jcomp2;
    public read () {
    jcomp1 = new JTextArea (5, 5);
    jcomp2 = new JButton ("close");
    JScrollPane jt1_sp = new JScrollPane(jcomp1);
    //adjust size and set layout
        setPreferredSize (new Dimension (624, 334));
        setLayout (null);

        //add components
        add (jt1_sp);
        add (jcomp2);

         jcomp2.addActionListener(this);
        //set component bounds (only needed by Absolute Positioning)
        jt1_sp.setBounds (30, 30, 350, 220);
        jcomp2.setBounds (140, 275, 100, 25);
        setSize(500,400);
        setVisible(true);

        try{File file = new File("projek.txt");
            Scanner sc = new Scanner (file);
            String data = " ";
            while(sc.hasNextLine()){

                String temp=sc.nextLine()+"\n";
                data = data +temp;
            }
```

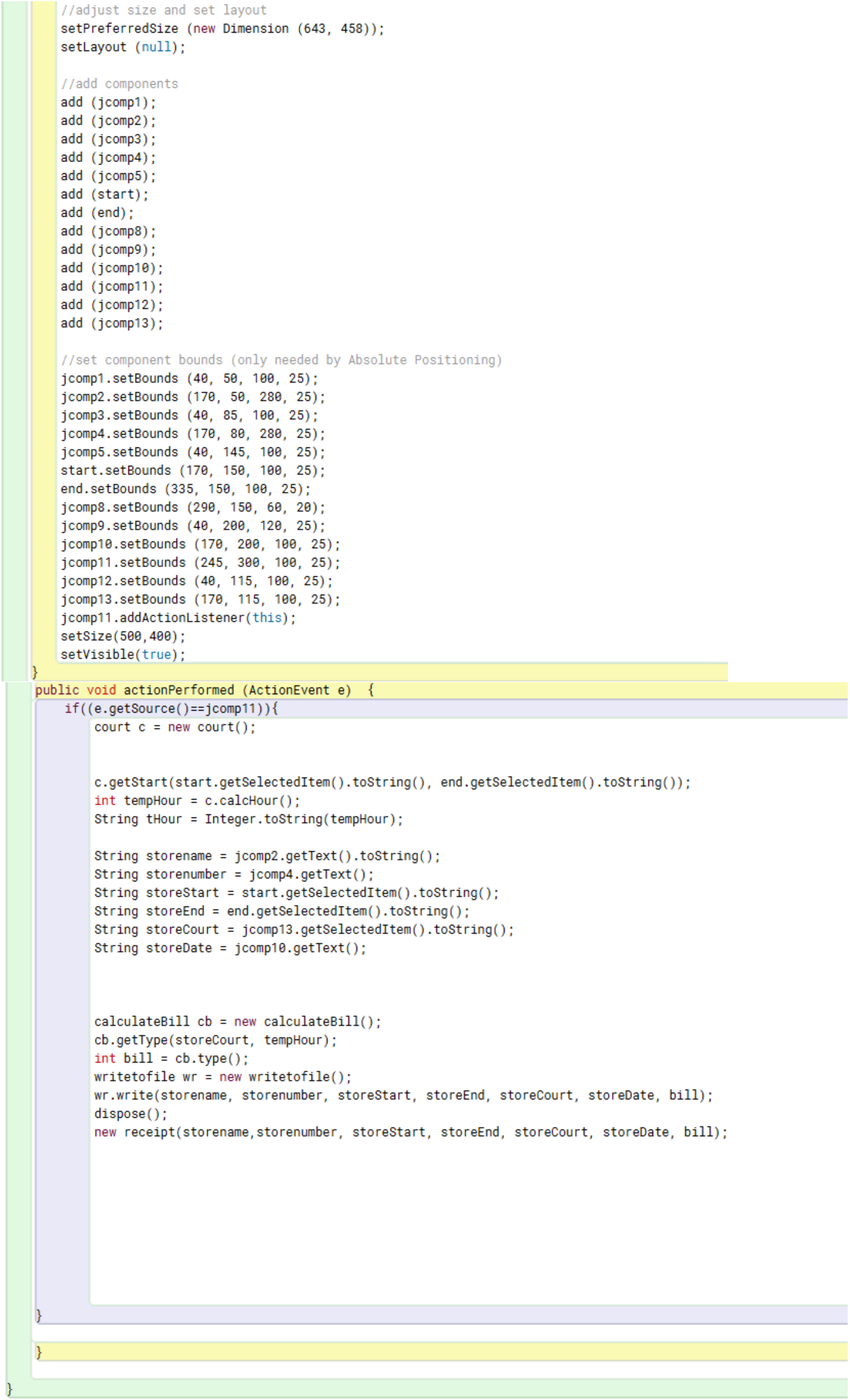


```
        try{File file = new File("projek.txt");
            Scanner sc = new Scanner (file);
            String data = " ";
            while(sc.hasNextLine()){

                String temp=sc.nextLine()+"\n";
                data = data +temp;
            }
            jcomp1.setText(data);
        }catch(Exception e){

    }
}
    public void actionPerformed (ActionEvent e)   {
        if((e.getSource()==jcomp2)){

        System.exit(0);
    }
}

}
```

MyPanelApply class

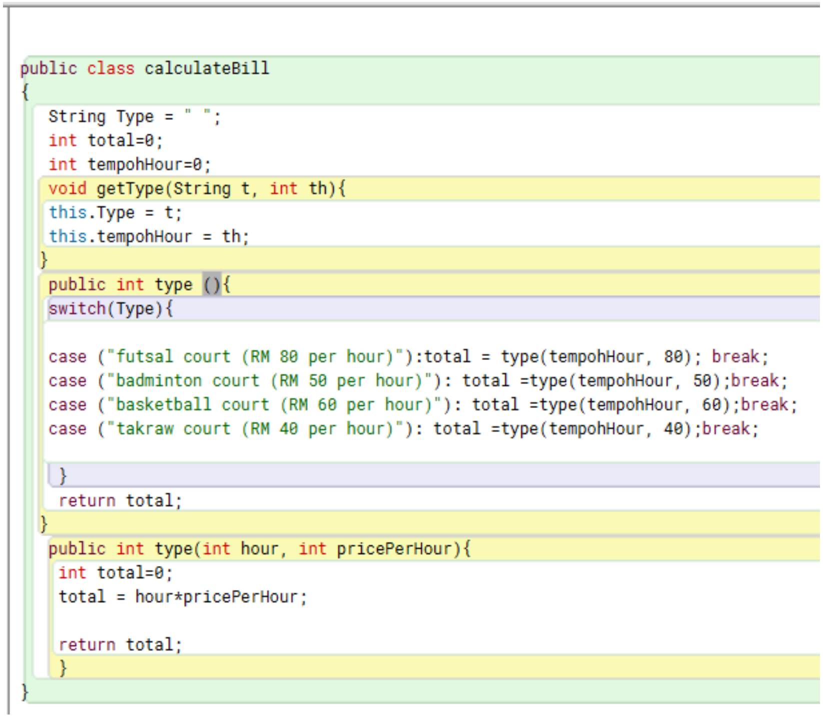


```java
import java.lang.String;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import java.io.FileReader;
import java.io.PrintWriter;
import java.io.FileOutputStream;
import java.io.FileNotFoundException;
public class MyPanelApply extends JFrame implements ActionListener {
    private JLabel jcomp1;
    public JTextArea jcomp2;
    private JLabel jcomp3;
    private JTextArea jcomp4;
    private JLabel jcomp5;
    private JComboBox start;
    private JComboBox end;
    private JLabel jcomp8;
    private JLabel jcomp9;
    private JTextField jcomp10;
    private JButton jcomp11;
    private JLabel jcomp12;
    private JComboBox jcomp13;

    public MyPanelApply(){
        //construct preComponents
        String[] startItems = {"6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21"};
        String[] endItems = {"6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21"};
        String[] jcomp13Items = {"[Select]", "futsal court (RM 80 per hour)", "badminton court (RM 50 per hour)", "basketball court (RM 60 per hour)", "takraw court (RM 40 per hour)"};

        //construct components
        jcomp1 = new JLabel ("name");
        jcomp2 = new JTextArea (5, 5);
        jcomp3 = new JLabel ("telephone number");
        jcomp4 = new JTextArea (5, 5);
        jcomp5 = new JLabel ("time");
        start = new JComboBox (startItems);
        end = new JComboBox (endItems);
        jcomp8 = new JLabel ("untill");
        jcomp9 = new JLabel ("date (dd/mm/yyyy)");
        jcomp10 = new JTextField (5);
        jcomp11 = new JButton ("apply");
        jcomp12 = new JLabel ("type court");
        jcomp13 = new JComboBox (jcomp13Items);
```



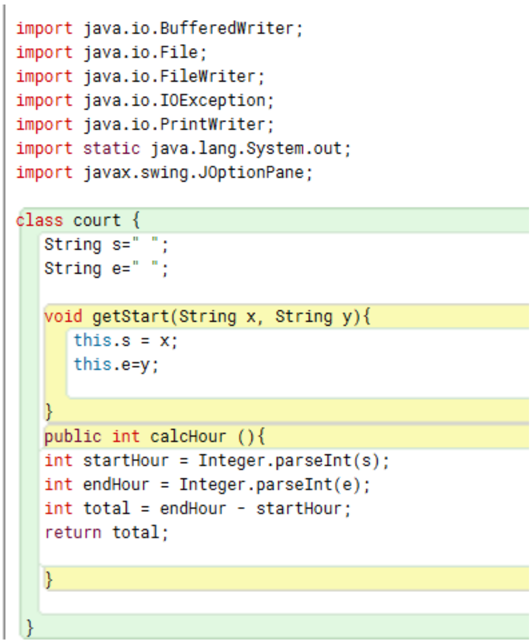
```
        //adjust size and set layout
        setPreferredSize (new Dimension (643, 458));
        setLayout (null);

        //add components
        add (jcomp1);
        add (jcomp2);
        add (jcomp3);
        add (jcomp4);
        add (jcomp5);
        add (start);
        add (end);
        add (jcomp8);
        add (jcomp9);
        add (jcomp10);
        add (jcomp11);
        add (jcomp12);
        add (jcomp13);

        //set component bounds (only needed by Absolute Positioning)
        jcomp1.setBounds (40, 50, 100, 25);
        jcomp2.setBounds (170, 50, 280, 25);
        jcomp3.setBounds (40, 85, 100, 25);
        jcomp4.setBounds (170, 80, 280, 25);
        jcomp5.setBounds (40, 145, 100, 25);
        start.setBounds (170, 150, 100, 25);
        end.setBounds (335, 150, 100, 25);
        jcomp8.setBounds (290, 150, 60, 20);
        jcomp9.setBounds (40, 200, 120, 25);
        jcomp10.setBounds (170, 200, 100, 25);
        jcomp11.setBounds (245, 300, 100, 25);
        jcomp12.setBounds (40, 115, 100, 25);
        jcomp13.setBounds (170, 115, 100, 25);
        jcomp11.addActionListener(this);
        setSize(500,400);
        setVisible(true);
    }
    public void actionPerformed (ActionEvent e)  {
        if((e.getSource()==jcomp11)){
            court c = new court();


            c.getStart(start.getSelectedItem().toString(), end.getSelectedItem().toString());
            int tempHour = c.calcHour();
            String tHour = Integer.toString(tempHour);

            String storename = jcomp2.getText().toString();
            String storenumber = jcomp4.getText();
            String storeStart = start.getSelectedItem().toString();
            String storeEnd = end.getSelectedItem().toString();
            String storeCourt = jcomp13.getSelectedItem().toString();
            String storeDate = jcomp10.getText();



            calculateBill cb = new calculateBill();
            cb.getType(storeCourt, tempHour);
            int bill = cb.type();
            writetofile wr = new writetofile();
            wr.write(storename, storenumber, storeStart, storeEnd, storeCourt, storeDate, bill);
            dispose();
            new receipt(storename,storenumber, storeStart, storeEnd, storeCourt, storeDate, bill);

        }

    }

}
```

calculateBill class



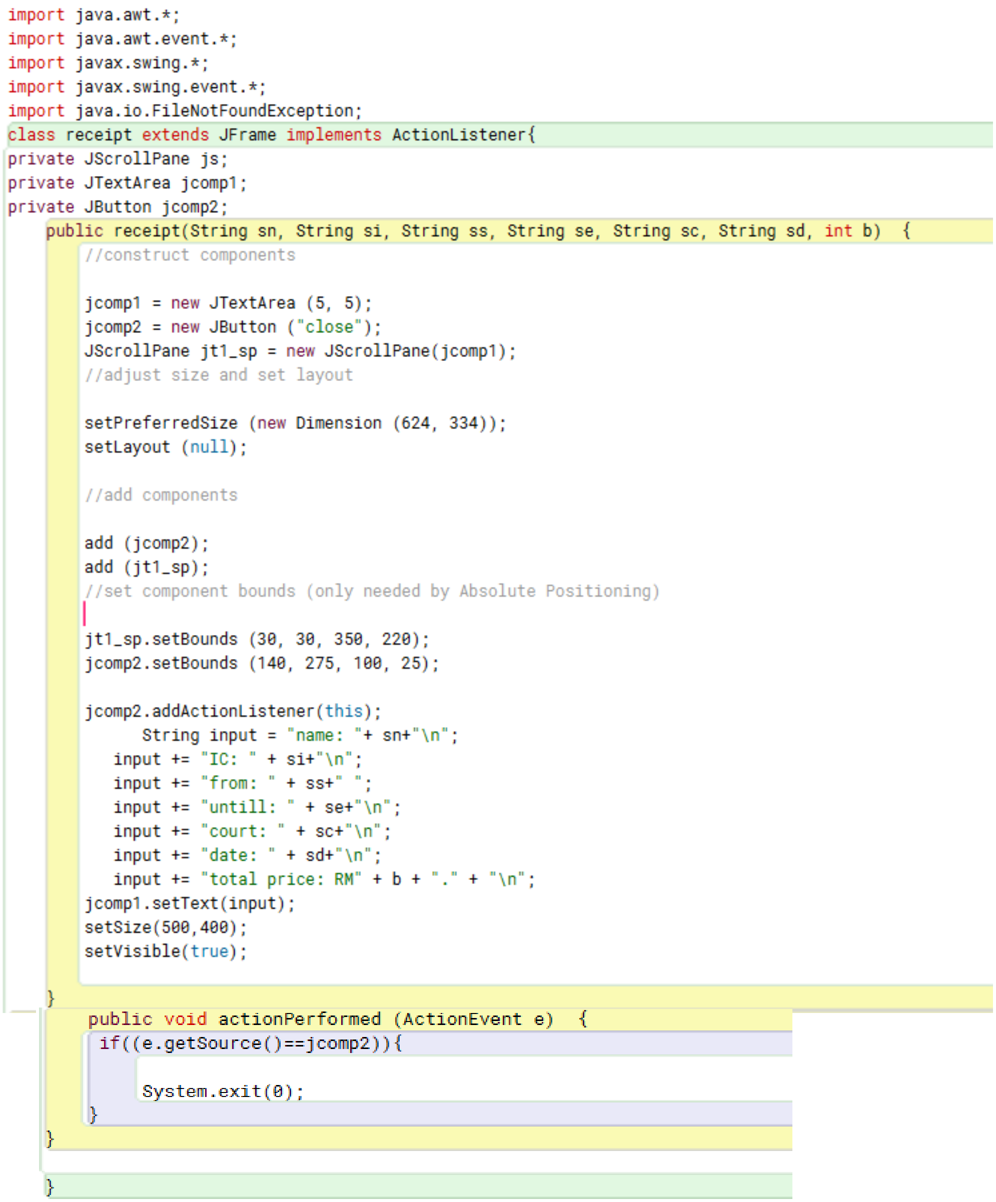
```
public class calculateBill
{
   String Type = " ";
   int total=0;
   int tempohHour=0;
   void getType(String t, int th){
   this.Type = t;
   this.tempohHour = th;
  }
   public int type (){
   switch(Type){

   case ("futsal court (RM 80 per hour)"):total = type(tempohHour, 80); break;
   case ("badminton court (RM 50 per hour)"): total =type(tempohHour, 50);break;
   case ("basketball court (RM 60 per hour)"): total =type(tempohHour, 60);break;
   case ("takraw court (RM 40 per hour)"): total =type(tempohHour, 40);break;

    }
    return total;
  }
   public int type(int hour, int pricePerHour){
    int total=0;
    total = hour*pricePerHour;

    return total;
    }
}
```

Court class



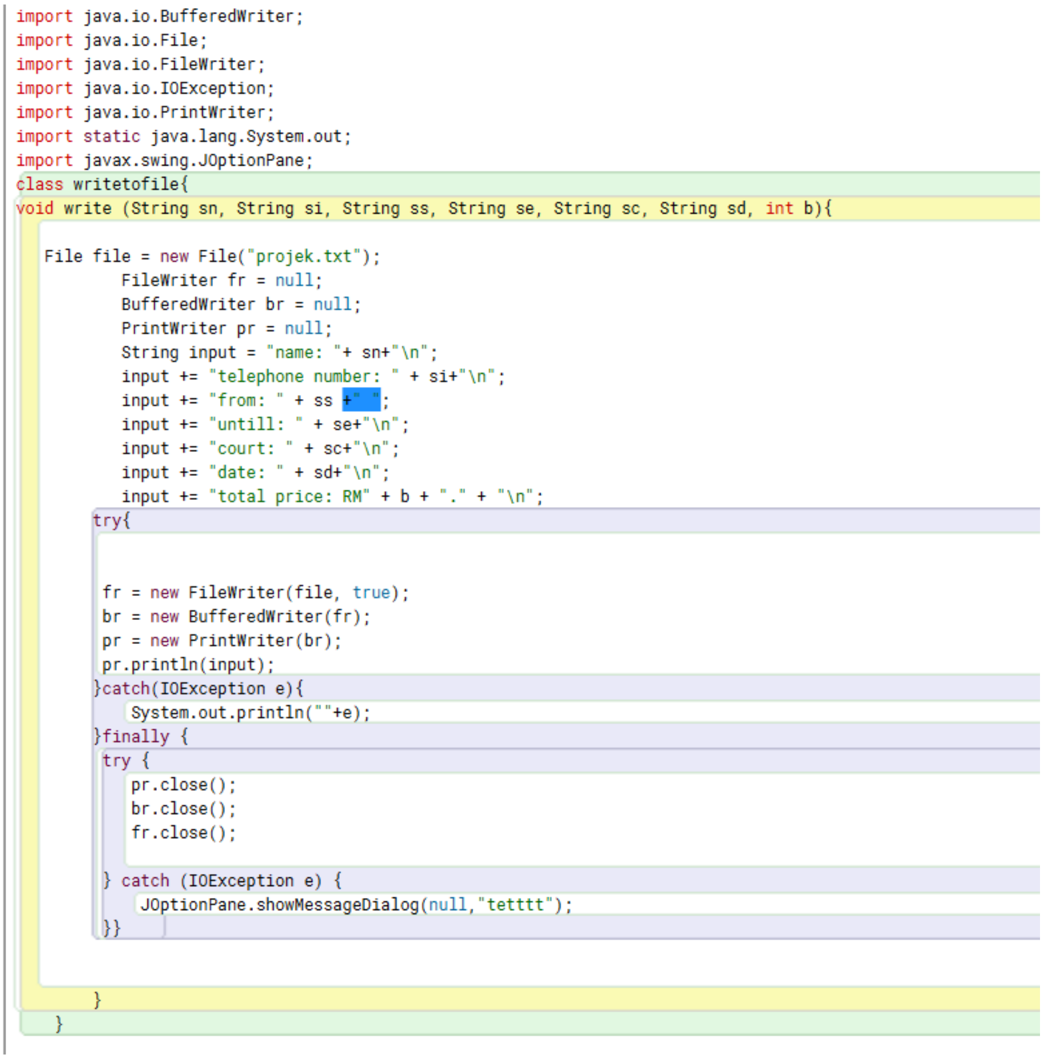
```
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import static java.lang.System.out;
import javax.swing.JOptionPane;

class court {
   String s=" ";
   String e=" ";

   void getStart(String x, String y){
      this.s = x;
      this.e=y;

   }
   public int calcHour (){
   int startHour = Integer.parseInt(s);
   int endHour = Integer.parseInt(e);
   int total = endHour - startHour;
   return total;

   }

 }
```

Receipt class



```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import java.io.FileNotFoundException;
class receipt extends JFrame implements ActionListener{
private JScrollPane js;
private JTextArea jcomp1;
private JButton jcomp2;
    public receipt(String sn, String si, String ss, String se, String sc, String sd, int b)  {
        //construct components

        jcomp1 = new JTextArea (5, 5);
        jcomp2 = new JButton ("close");
        JScrollPane jt1_sp = new JScrollPane(jcomp1);
        //adjust size and set layout

        setPreferredSize (new Dimension (624, 334));
        setLayout (null);

        //add components

        add (jcomp2);
        add (jt1_sp);
        //set component bounds (only needed by Absolute Positioning)

        jt1_sp.setBounds (30, 30, 350, 220);
        jcomp2.setBounds (140, 275, 100, 25);

        jcomp2.addActionListener(this);
              String input = "name: "+ sn+"\n";
           input += "IC: " + si+"\n";
           input += "from: " + ss+" ";
           input += "untill: " + se+"\n";
           input += "court: " + sc+"\n";
           input += "date: " + sd+"\n";
           input += "total price: RM" + b + "." + "\n";
        jcomp1.setText(input);
        setSize(500,400);
        setVisible(true);

    }
        public void actionPerformed (ActionEvent e)  {
         if((e.getSource()==jcomp2)){

              System.exit(0);
        }
    }

    }
```

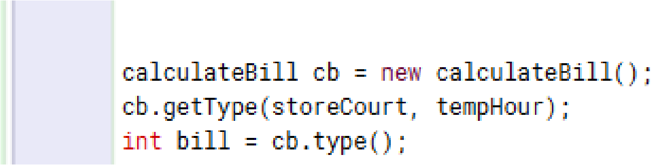
Writetofile class



```
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import static java.lang.System.out;
import javax.swing.JOptionPane;
class writetofile{
void write (String sn, String si, String ss, String se, String sc, String sd, int b){

   File file = new File("projek.txt");
           FileWriter fr = null;
           BufferedWriter br = null;
           PrintWriter pr = null;
           String input = "name: "+ sn+"\n";
           input += "telephone number: " + si+"\n";
           input += "from: " + ss +" ";
           input += "untill: " + se+"\n";
           input += "court: " + sc+"\n";
           input += "date: " + sd+"\n";
           input += "total price: RM" + b + "." + "\n";
         try{


          fr = new FileWriter(file, true);
          br = new BufferedWriter(fr);
          pr = new PrintWriter(br);
          pr.println(input);
         }catch(IOException e){
             System.out.println(""+e);
         }finally {
          try {
             pr.close();
             br.close();
             fr.close();

          } catch (IOException e) {
              JOptionPane.showMessageDialog(null,"tetttt");
          }}


         }
     }
```

OBJECT ORIENTED PROGRAM CONCEPTS

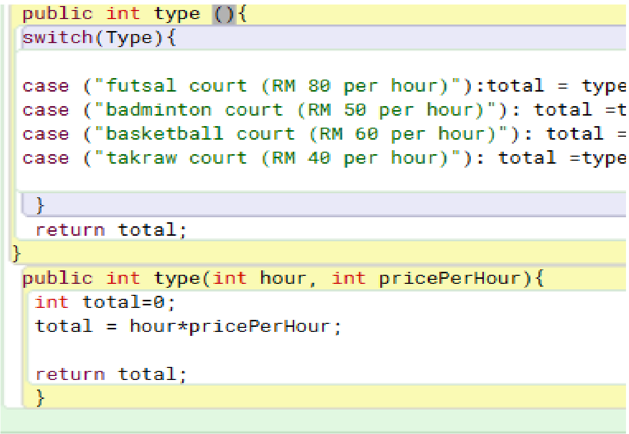
1. Object and classes



```
calculateBill cb = new calculateBill();
cb.getType(storeCourt, tempHour);
int bill = cb.type();
```

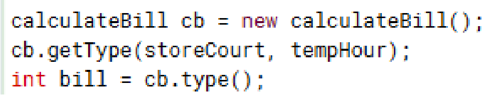
I have implement object and class concept to calculate bill of the court booking cost. Picture above shown that object cb has been created to call method from calculateBill class.

2. Polymorphism



```
public int type (){
switch(Type){

case ("futsal court (RM 80 per hour)"):total = type
case ("badminton court (RM 50 per hour)"): total =t
case ("basketball court (RM 60 per hour)"): total =
case ("takraw court (RM 40 per hour)"): total =type

 }
 return total;
}
public int type(int hour, int pricePerHour){
 int total=0;
 total = hour*pricePerHour;

 return total;
 }
```

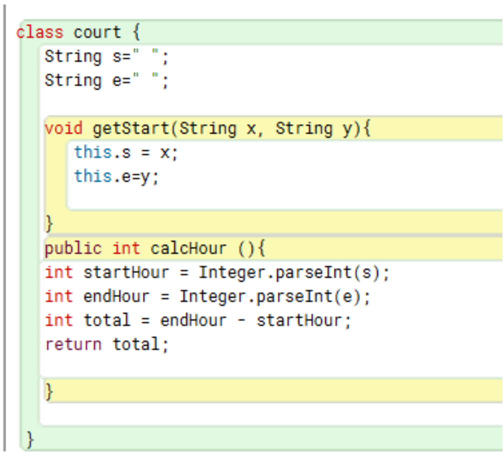
I have created two method that have same method name but different parameter. Which method will be called is determined by the passing argument.



```
calculateBill cb = new calculateBill();
cb.getType(storeCourt, tempHour);
int bill = cb.type();
```

Here are an example of calling the method type. The method type that have switch will be executed because it satisfy the parameter received.

3. Encapsulation



```
class court {
   String s=" ";
   String e=" ";

   void getStart(String x, String y){
      this.s = x;
      this.e=y;

   }
   public int calcHour (){
   int startHour = Integer.parseInt(s);
   int endHour = Integer.parseInt(e);
   int total = endHour - startHour;
   return total;

   }

 }
```

Encapsulation is process of wrapping code and data in single unit. Data is hidden from users. I have hide the process of total of hour of the booking in class court.