

Taxpayer.java

```
2 * @Taxpayer.java
14
15 package data.Manager;
16
17 import java.util.ArrayList;
18
19
21 * This class holds information about a taxpayer and is a superclass that other
26
27 public abstract class Taxpayer {
28
29     protected String fullName;
30     protected int afm;
31     protected String financialState;
32     protected int income;
33
34     private ArrayList<Receipt> collectedReceipts = new ArrayList<>(); // holds a list of
taxpayer's collected receipts
35     private String taxpayerInfo; // holds taxpayer's info
36
37     private int receiptID = 1; // holds an id for each of the taxpayers receipts
38
39     public Taxpayer(String fullName, int afm, String financialState,
40         int income) {
41         this.fullName = fullName;
42         this.afm = afm;
43         this.financialState = financialState;
44         this.income = income;
45
46         taxpayerInfo = "-----TAXPAYER INFO-----\n\n" + "Name: "
47             + fullName + "\nAFM: " + afm + "\nStatus: " + financialState
48             + "\nIncome: " + income;
49     }
50
51     /*
52     * Processes the input values from the submission form and if they are
53     * valid, it creates a new receipt
54     */
55     public Receipt submitReceipts(String date, String kind,
56         JTextField... fields) {
57         double amount = Double.parseDouble(fields[0].getText());
58         Enterprise publisher = createEnterprisePublisher(fields[1].getText(),
59             fields[2].getText(), fields[3].getText(), fields[4].getText(),
60             fields[5].getText());
61
62         resetFields(fields);
63         Receipt receipt = createReceipt(receiptID, date, kind, amount, publisher);
64         collectedReceipts.add(receipt);
65         receiptID++;
66
67         return receipt;
68     }
69     /*
70     * checks in which category the receipt belongs and creates it
71     */
72     private Receipt createReceipt(int id, String date, String kind, double amount,
73         Enterprise publisher) {
74         if (kind.equals("Entertainment")) {
75             return new EntertainmentReceipt(receiptID, date, amount, publisher);
76         } else if (kind.equals("Basic")) {
77             return new BasicReceipt(receiptID, date, amount, publisher);
78         } else if (kind.equals("Travel")) {
79             return new TravelReceipt(receiptID, date, amount, publisher);
80         }
81     }
82 }
```

Taxpayer.java

```

79     } else if (kind.equals("Entertainment")) {
80         return new HealthReceipt(receiptID, date, amount, publisher);
81     } else if (kind.equals("Other")) {
82         return new OtherReceipt(receiptID, date, amount, publisher);
83     }
84     return null;
85 }
86 /* after we submit a receipt it resets the fields */
87 private void resetFields(JTextField... fields) {
88     for (int i = 0; i < fields.length; i++) {
89         fields[i].setText("");
90     }
91 }
92 /* creates an Enterprise that publishes the receipt */
93 private Enterprise createEnterprisePublisher(String... values) {
94     String companyName = values[0];
95     String country = values[1];
96     String city = values[2];
97     String street = values[3];
98     int number = Integer.parseInt(values[4].trim());
99
100     return new Enterprise(companyName, country, city, street, number);
101 }
102
103 public boolean estimateValue(int minVal, int maxVal) {
104     return income >= minVal && income < maxVal;
105 }
106
107 public double basicTax;           // holds the basic tax
108 public double totalTax;           // holds the total tax
109 public double taxIncrease;        // hold the increase tax value
110
111 public void incrementId() { receiptID++; } // Returns taxpayer's id
112 public String getName() { return fullName; } // Returns taxpayer's name
113 public int getAfm() { return afm; } // Returns taxpayer's am
114 public int getReceiptID() { return receiptID; } // Returns taxpayer's receipt id
115 public void setReceiptID(int receiptID) { this.receiptID = receiptID; }
116 public ArrayList<Receipt> getCollectedReceipts() { return collectedReceipts; } //
    Returns a list with taxpayer's receipts
117 public double getBasicTax() { return basicTax; } // returns taxpayer's basic tax
118 public double getTotalTax() { return totalTax; } // returns taxpayer's total tax
119 public String toString() { return taxpayerInfo; }
120 public int getIncome() { return income; }
121 public Receipt getReceiptByFromList(int place) { return
    this.getCollectedReceipts().get(place); }
122
123 public abstract void calculateTax(int income); // calculates the tax for a taxpayer
124 public abstract boolean estimateTaxRaise(double minVal, double maxVal); // returns true
    if the tax raise is between the min and max value
125 }
126

```