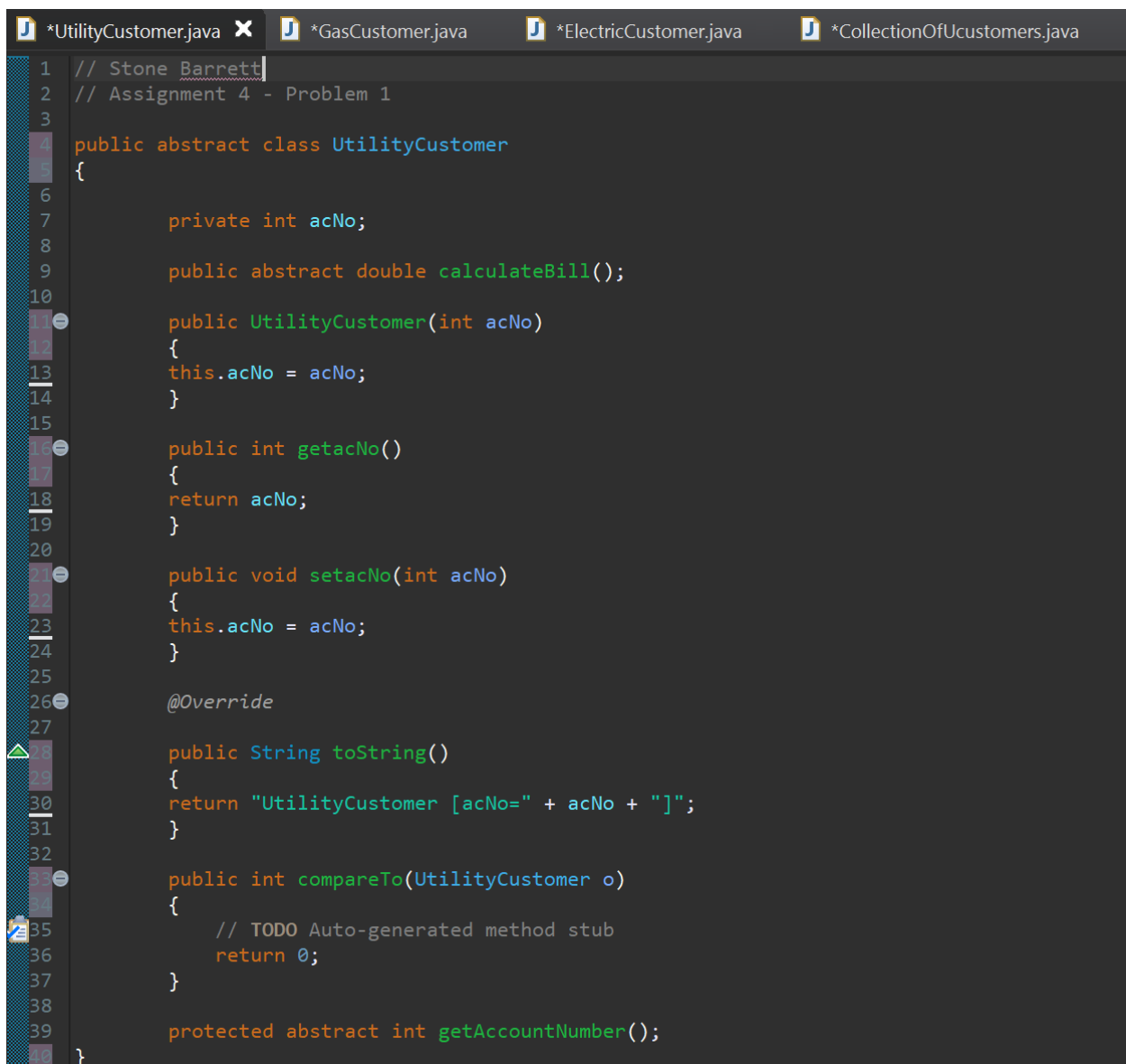


Stone Barrett  
Assignment 04  
7/14/19

## Problem 1: Utilities

Screenshots of the code:

For this first problem, understanding how inheritance in Java works was crucial. I needed to make each of the four classes do exactly what they needed and to work well with each other. Each of the Gas and Electric classes contained the variables required as well as a method to calculate bills. The Utility class is the parent class of those two and contains what they need to inherit. The main function lies in the Collection class and it's there that the array is set up and the information is printed back to a user. Unfortunately, as you can see in the output screenshot below, there was an error that I could not resolve in time. I had a few other students look and try to figure out the problem, and I gathered that my sorting method didn't work with the information I passed to it.



```
1 // Stone Barrett
2 // Assignment 4 - Problem 1
3
4 public abstract class UtilityCustomer
5 {
6     private int acNo;
7
8     public abstract double calculateBill();
9
10
11     public UtilityCustomer(int acNo)
12     {
13         this.acNo = acNo;
14     }
15
16     public int getacNo()
17     {
18         return acNo;
19     }
20
21     public void setacNo(int acNo)
22     {
23         this.acNo = acNo;
24     }
25
26     @Override
27
28     public String toString()
29     {
30         return "UtilityCustomer [acNo=" + acNo + "]";
31     }
32
33     public int compareTo(UtilityCustomer o)
34     {
35         // TODO Auto-generated method stub
36         return 0;
37     }
38
39     protected abstract int getAccountNumber();
40 }
```

```
*UtilityCustomer.java *GasCustomer.java x *ElectricCustomer.java *CollectionOfUcustomers.java
1 // Stone Barrett
2 // Assignment 4 - Problem 1
3
4 public class GasCustomer extends UtilityCustomer
5 {
6     // Initialization
7     private int cubicMetersUsed;
8     private final double PRICE_PER_CUBIC_METER = 2.75;
9
10    public GasCustomer(int acNo, int cubicMetersUsed)
11    {
12        super(acNo);
13        this.setCubicMetersUsed(cubicMetersUsed);
14    }
15
16    @Override
17
18    public int compareTo(UtilityCustomer o)
19    {
20        if (this.getAccountNumber() < o.getAccountNumber())
21        {
22
23            return 1;
24        }
25        else if (this.getAccountNumber() == o.getAccountNumber())
26        {
27            return 0;
28        }
29
30        return -1;
31    }
32
33    protected int getAccountNumber()
34    {
35        // TODO Auto-generated method stub
36        return 0;
37    }
38
39    @Override
40
41    public double calculateBill()
42    {
43
44        return cubicMetersUsed * PRICE_PER_CUBIC_METER;
45    }
46
47
48
```

```
49    public int getCubicMetersUsed()
50    {
51        return cubicMetersUsed;
52    }
53
54    public void setCubicMetersUsed(int cubicMetersUsed)
55    {
56        this.cubicMetersUsed = cubicMetersUsed;
57    }
58
59    public String toString()
60    {
61        return "Gas " + super.toString() + "\nGas Consumption: " + cubicMetersUsed + "\nAmount Charged: " + calculateBill()
62            + "\n";
63    }
64 }
```

```
*UtilityCustomer.java    *GasCustomer.java    *ElectricCustomer.java X    *CollectionOfUcustomers.java

1 // Stone Barrett
2 // Assignment 4 - Problem 1
3
4 public class ElectricCustomer extends UtilityCustomer
5 {
6     // Initialization
7     private int kWattHourUsed;
8     private final double PRICE_PER_KWATT = 1;
9     private final double DELIVERY_FEE = 30;;
10
11     // Constructor
12     public ElectricCustomer(int acNo, int kWattHourUsed)
13     {
14         super(acNo);
15         this.kWattHourUsed = kWattHourUsed;
16     }
17
18     @Override
19
20     public double calculateBill()
21     {
22         return kWattHourUsed * PRICE_PER_KWATT + DELIVERY_FEE;
23     }
24
25     @Override
26
27     public int compareTo(UtilityCustomer o)
28     {
29         if (this.getAccountNumber() < o.getAccountNumber())
30         {
31             return 1;
32         }
33         else if (this.getAccountNumber() == o.getAccountNumber())
34         {
35             return 0;
36         }
37
38         return -1;
39     }
40
41     public int getkWattHourUsed()
42     {
43         return kWattHourUsed;
44     }
45
46     public void setkWattHourUsed(int kWattHourUsed)
47     {
48         this.kWattHourUsed = kWattHourUsed;
49     }
50
51     public String toString()
52     {
53         return "Electricity " + super.toString() + "\nElectricity Consumption: " + kWattHourUsed + "\nAmount Charged: "
54             + calculateBill() + "\n";
55     }
56
57     @Override
58     protected int getAccountNumber()
59     {
60         // TODO Auto-generated method stub
61         return 0;
62     }
63 }
```

```

UtilityCustomer.java GasCustomer.java ElectricCustomer.java CollectionOfUcustomers.java
1 // Stone Barrett
2 // Assignment 4 - Problem 1
3
4 // Import library and scanner
5 import java.util.Arrays;
6 import java.util.Scanner;
7
8 // Class begin
9 public class CollectionOfUcustomers {
10
11     // Main function
12     public static void main(String args[]) {
13
14         // Set up scanner
15         Scanner input = new Scanner(System.in);
16
17         // Initialization
18         UtilityCustomer[] utilityCustomers = new UtilityCustomer[10];
19         int accountNo = 0, cubicMetUsed, kWattUsed;
20
21         // Interface
22         for (int idx = 0; idx < 10; idx++)
23         {
24             System.out.println("User #" + (idx + 1));
25
26             // Electric
27             if (idx % 2 == 0)
28             {
29                 System.out.print("Gas Account Number: ");
30                 accountNo = Integer.parseInt(input.nextLine());
31                 System.out.print("Gas consumption : ");
32                 cubicMetUsed = Integer.parseInt(input.nextLine());
33                 utilityCustomers[idx] = new GasCustomer(accountNo, cubicMetUsed);
34             }
35
36             // Gas
37             else
38             {
39                 System.out.print("Electricity Account Number: ");
40                 accountNo = Integer.parseInt(input.nextLine());
41                 System.out.print("Electricity consumption : ");
42                 kWattUsed = Integer.parseInt(input.nextLine());
43                 utilityCustomers[idx] = new ElectricCustomer(accountNo, kWattUsed);
44             }
45
46         }
47
48         // Sort in descending order
49         Arrays.sort(utilityCustomers);
50
51         System.out.println("Sorted list of Utiliy Customers");
52
53         // print utility customer details
54         for (int idx = 0; idx < 10; idx++)
55         {
56             System.out.println("=====");
57             System.out.println(utilityCustomers[idx]);
58         }
59
60         // Close scanner
61         input.close();
62     }

```

```
User #1
Gas Account Number: 1
Gas consumption : 100
User #2
Electricity Account Number: 2
Electricity consumption : 100
User #3
Gas Account Number: 3
Gas consumption : 100
User #4
Electricity Account Number: 4
Electricity consumption : 100
User #5
Gas Account Number: 5
Gas consumption : 100
User #6
Electricity Account Number: 6
Electricity consumption : 100
User #7
Gas Account Number: 7
Gas consumption : 100
User #8
Electricity Account Number: 8
Electricity consumption : 100
User #9
Gas Account Number: 9
Gas consumption : 100
User #10
Electricity Account Number: 10
Electricity consumption : 100
Exception in thread "main" java.lang.ClassCastException: ElectricCustomer cannot be cast to java.lang.Comparable
    at java.util.ComparableTimSort.countRunAndMakeAscending(Unknown Source)
    at java.util.ComparableTimSort.sort(Unknown Source)
    at java.util.Arrays.sort(Unknown Source)
    at CollectionOfUcustomers.main(CollectionOfUcustomers.java:49)
```

## **Problem 2: Tip**

This problem I considered to be more interesting than the first. Also, I like JavaFX but before doing anything I needed to allow access to the correct libraries within the build path of the project file. I imported all the libraries I needed and set up the grid pane. I decided to use radio buttons instead of a drop-down menu. I added a text field to enter the meal price. The four options for tip percentages are there for one to be selected. After this, the sales tax is set to 8% and all of the price is calculated and reported back to the user.

Screenshots of the code:

```
*UtilityCustomer.java  *GasCustomer.java  *ElectricCustomer.java  *CollectionOfUcustomers.java  Tip.java X

1 // Stone Barnett
2 // Assignment 4 - Problem 2
3
4 // Importing JavaFX Libraries
5 import javafx.application.Application;
6 import javafx.event.ActionEvent;
7 import javafx.event.EventHandler;
8 import javafx.geometry.Pos;
9 import javafx.scene.Scene;
10 import javafx.scene.control.*;
11 import javafx.scene.layout.GridPane;
12 import javafx.stage.Stage;
13
14 // Importing number format
15 import java.text.NumberFormat;
16
17 // Class begin
18 public class Tip extends Application
19 {
20     // GUI
21     GridPane gridPane;
22     Label lblMessage, lblTip, lblResult;
23     TextField txtAmount;
24     RadioButton rdb0, rdb15, rdb18, rdb20;
25     ToggleGroup buttonGroup;
26     Button btnCalculate;
27
28
29     // Value variables
30     double amount = 0, tip = 0, salesTax = 0, billAmount = 0;
31
32     // For currency format
33     NumberFormat numberFormat= NumberFormat.getCurrencyInstance();
34     final double SALES_TAX=0.08;
35
36     @Override
37     public void start(Stage primaryStage)
38     {
39         gridPane = new GridPane();
40         gridPane.setAlignment(Pos.CENTER);
41         gridPane.setHgap(10);
42         gridPane.setVgap(10);
43
44
45         //first row
46         lblMessage = new Label("Enter food Charge: ");
47         txtAmount = new TextField();
48         txtAmount.setMaxWidth(50);
49         gridPane.add(lblMessage,0,0);
50         gridPane.add(txtAmount,1,0);
51
52         //second row
53         lblTip = new Label("Please select tip: ");
54         gridPane.add(lblTip,0,1);
55
56         //Third row
57         rdb0 = new RadioButton("0%");
58         rdb15 = new RadioButton("15%");
59         rdb18 = new RadioButton("18%");
60         rdb20 = new RadioButton("20%");
```

```
*UtilityCustomer.java *GasCustomer.java *ElectricCustomer.java *CollectionOfUcustomers.java Tip.java X
59 rdb18= new RadioButton("18%");
60 rdb20 = new RadioButton("20%");
61
62 //add to button group
63 buttonGroup= new ToggleGroup();
64 rdb0.setToggleGroup(buttonGroup);
65 rdb20.setToggleGroup(buttonGroup);
66 rdb18.setToggleGroup(buttonGroup);
67 rdb15.setToggleGroup(buttonGroup);
68 rdb0.setSelected(true);
69 gridPane.add(rdb0,0,2);
70 gridPane.add(rdb15,1,2);
71 gridPane.add(rdb18,2,2);
72 gridPane.add(rdb20,3,2);
73
74 //Forth row
75 btnCalculate = new Button("Calculate");
76 gridPane.add(btnCalculate,1,3);
77
78 //Fifth row
79 lblResult = new Label();
80 gridPane.add(lblResult,1,4);
81
82 //Create stage
83 primaryStage.setTitle("Bill Calculator");
84 Scene scene = new Scene(gridPane);
85 primaryStage.setScene(scene);
86 primaryStage.setWidth(500);
87 primaryStage.setHeight(300);
88 primaryStage.show();
89
90
91 //Add handler to Button
92 btnCalculate.setOnAction(new EventHandler<ActionEvent>()
93 {
94     @Override
95     public void handle(ActionEvent event)
96     {
97         amount = Double.parseDouble(txtAmount.getText());
98         if(rdb0.isSelected())
99             tip = 0;
100         else if(rdb15.isSelected())
101             tip =0.15;
102         else if(rdb18.isSelected())
103             tip = 0.18;
104         else if(rdb20.isSelected())
105             tip=0.20;
106         tip = amount * tip;
107         salesTax = amount * SALES_TAX;
108         billAmount = amount + tip + salesTax;
109         String result = "Food Change: " + numberFormat.format(amount) + "\nTip Amount: " + numberFormat.format(tip) +
110             "\nSales Tax: " + numberFormat.format(salesTax) +
111             "\nTotal Bill Amount: " + numberFormat.format(billAmount);
112         lblResult.setText(result);
113     }
114 });
115 }
116
117 public static void main(String[] args)
118 {
119     Launch(args);
120 }
```



Enter food Charge:

Please select tip:

☐ 0%

☒ 15%

☐ 18% ☐ 20%

Food Charge: \$15.00

Tip Amount: \$2.25

Sales Tax: \$1.20

Total Bill Amount: \$18.45