

Developing a Java Business Application With NetBeans GUI

Louie Venegas
CEIS355



INTRODUCTI ON

Java is one of the most widely used languages for business applications. It provides tools for building GUIs, reading and writing files, and connecting to databases. For this project I used all of those to design and build a landscape calculator: a GUI application that reads and writes files and interacts with a database.

```
mirror_mod = modifier_0
#set mirror object to mirror_0
mirror_mod.mirror_object =
```

```
operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True
```

```
#selection at the end -add
mirror_ob.select= 1
modifier_ob.select=1
context.scene.objects.active
= bpy.context.selected_objects[0]
data.objects[one.name].select
```

```
print("please select exactly
```

```
-- OPERATOR CLASSES --
```

```
types.Operator):
    X mirror to the selected
    object.mirror_mirror_x"
    mirror X"
```

```
context):
    context.active_object is not
```

Designing The Application

Leveraging Visio

*Visio wireframe
diagram*

*Visio UML Class
diagram*

UML Class Diagram

Customer

-customerID: int
-name: String
-address: String
-yardType: String
-length: double
-width: double
-totalCost: double
+toString(): String
+getDetails(): String

DataIO

-DATABASE_NAME: String
-CONNECTION_STRING: String
-USER_NAME: String
-PASSWORD: String
+add(cust: Customer): void
+delete(customerID: int): void
+getList(): ArrayList<customer>

Visio Wire Frame Diagram

Louie Venegas Landscaping

File Order

Welcome Information Customer List

Louie Venegas Landscapes
Would you like A Grass or Gravel Yard?

☐ Grass
\$5 per sqft

☐ Gravel
\$2 per sqft



Next

Reset



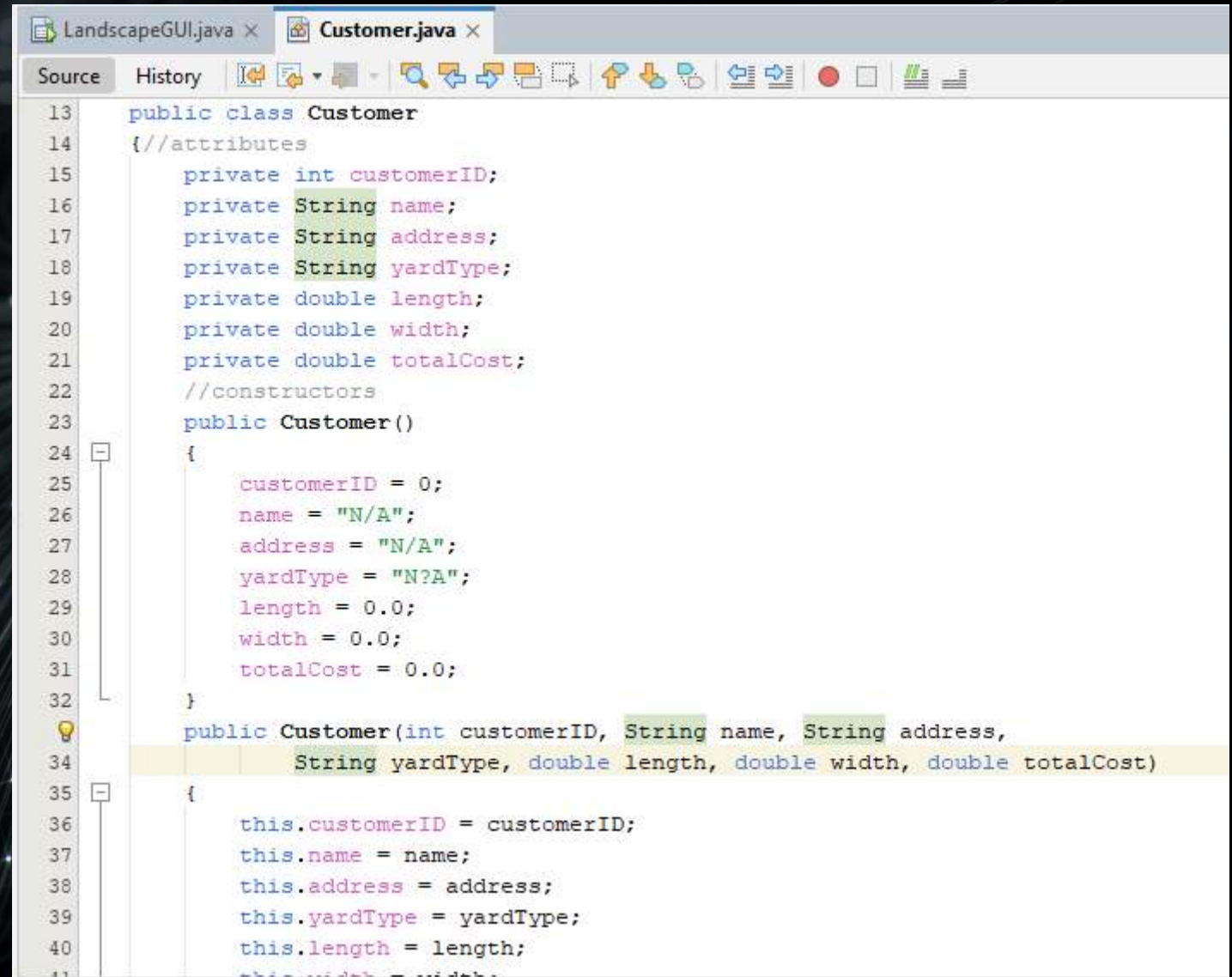
Starting the GUI for the Application

-
- *Take the UML Class diagram to code.*
 - *Create the GUI application with labels, textfields, and buttons.*
 - *Add events to the command buttons.*

public class

Customer:

This is the class used throughout the Application to construct the customer objects. This had to be the first thing taken to code because it would be called upon time and time again as the application is designed.



```
13 public class Customer
14 { //attributes
15     private int customerID;
16     private String name;
17     private String address;
18     private String yardType;
19     private double length;
20     private double width;
21     private double totalCost;
22     //constructors
23     public Customer()
24     {
25         customerID = 0;
26         name = "N/A";
27         address = "N/A";
28         yardType = "N/A";
29         length = 0.0;
30         width = 0.0;
31         totalCost = 0.0;
32     }
33     public Customer(int customerID, String name, String address,
34                     String yardType, double length, double width, double totalCost)
35     {
36         this.customerID = customerID;
37         this.name = name;
38         this.address = address;
39         this.yardType = yardType;
40         this.length = length;
41         this.width = width;
```

GUI Application with Labels, Textfields, and Buttons.

```
private boolean validateInputs()
{
    String sName = txtName.getText();
    String sAddress = txtAddress.getText();
    String sWidth = txtWidth.getText();
    String sLength = txtLength.getText();

    if (sName.equals(""))
    {
        JOptionPane.showMessageDialog(this, "Enter Your Information", "Name Error", JOptionPane.ERROR_MESSAGE);
        txtName.requestFocusInWindow();
        return false;
    }

    if (sAddress.equals(""))
    {
        JOptionPane.showMessageDialog(this, "Enter Your Information", "Address Error", JOptionPane.ERROR_MESSAGE);
        txtAddress.requestFocusInWindow();
        return false;
    }

    if (sWidth.length() <= 5)
    {
        JOptionPane.showMessageDialog(this, "Enter Your Information", "Width Error", JOptionPane.ERROR_MESSAGE);
        txtWidth.requestFocusInWindow();
        return false;
    }

    if (sLength.length() <= 5)
    {
        JOptionPane.showMessageDialog(this, "Enter Your Information", "Length Error", JOptionPane.ERROR_MESSAGE);
        txtLength.requestFocusInWindow();
        return false;
    }

    return true;
}
```

Venegas Landscape

Please Enter Your Information

Name:

Address:

Width(ft): Length(ft):

Order Summary

Type: Grass Width: 5.0 Length: 5.0 Total Cost: \$125.00

Width Error

Width must be a number

Click the icon in the Navigator to access various useful settings of the form (in Properties).

Venegas Landscape

Please Enter Your Information

Order Summary

Name:

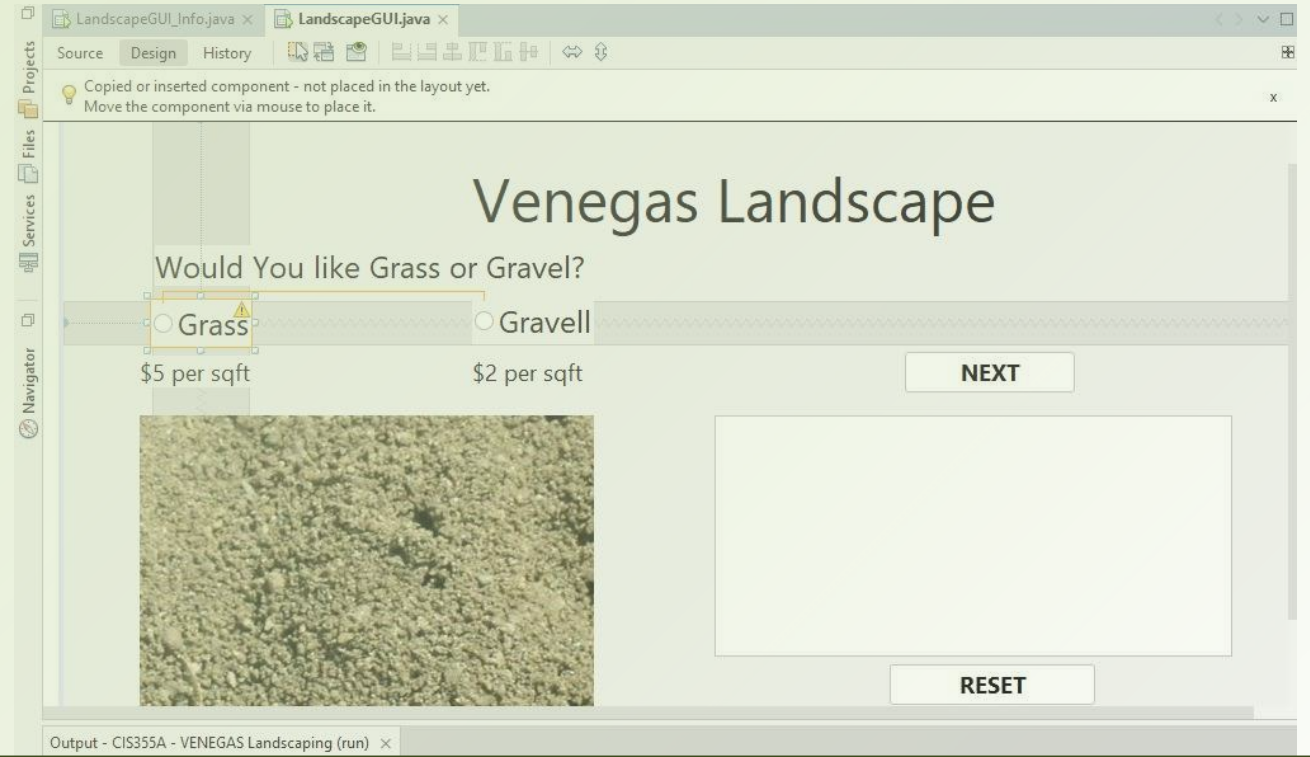
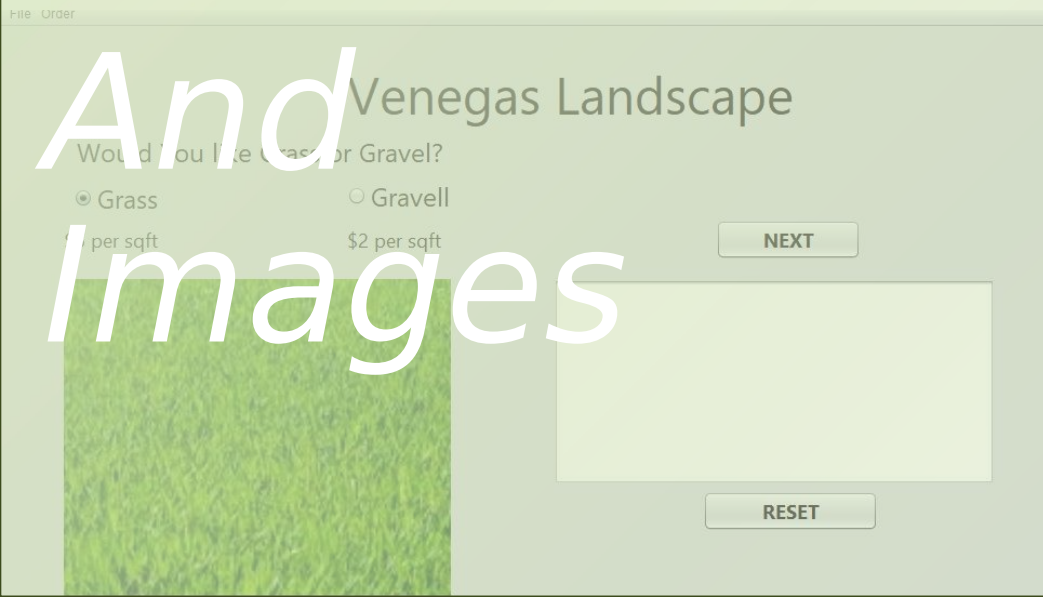
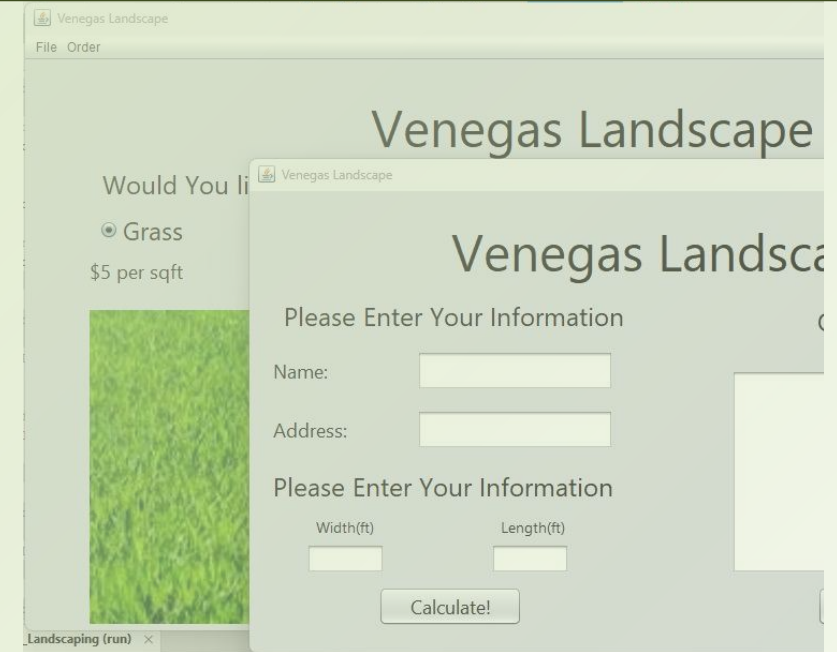
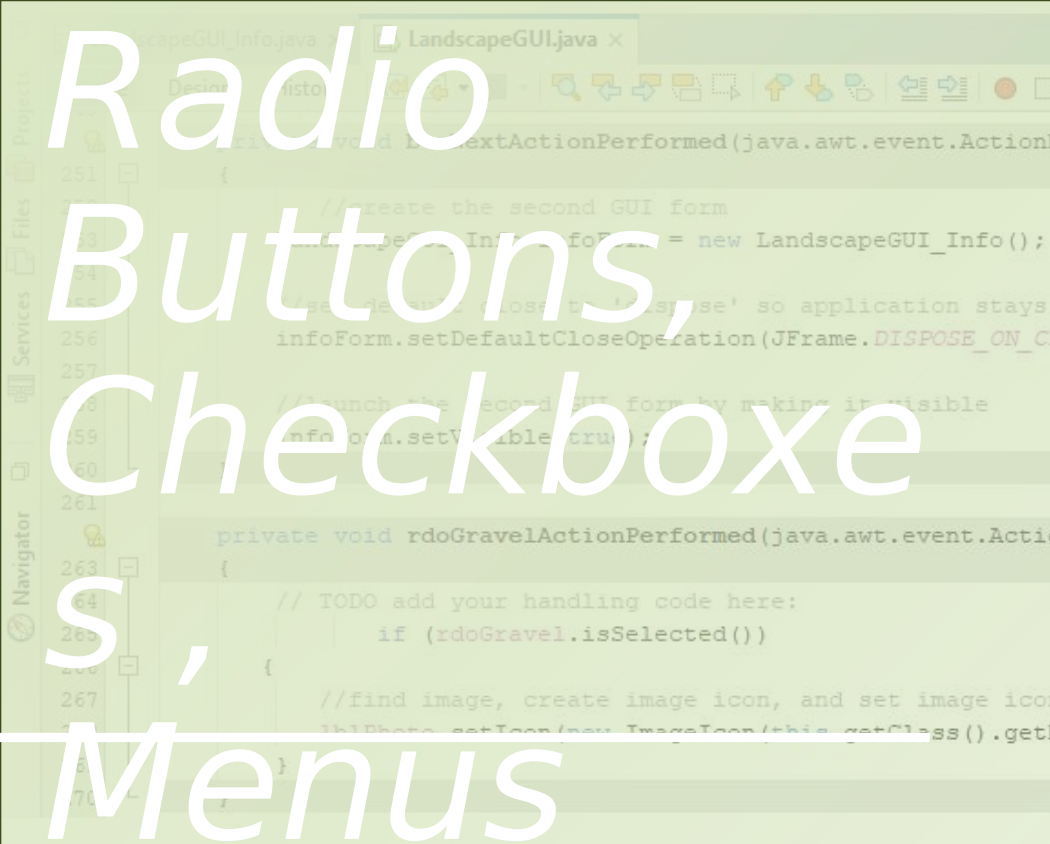
Address:

Please Enter Your Information

Advanced Controls in the GUI Application

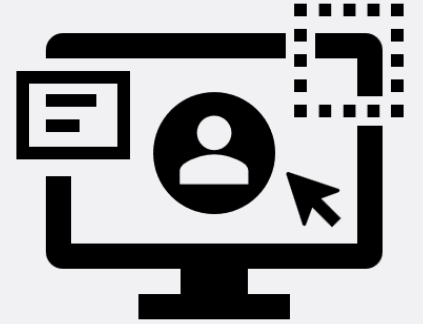
- *Adding Radio Buttons and Checkboxes*
 - *Adding a Menu*
 - *Adding images*



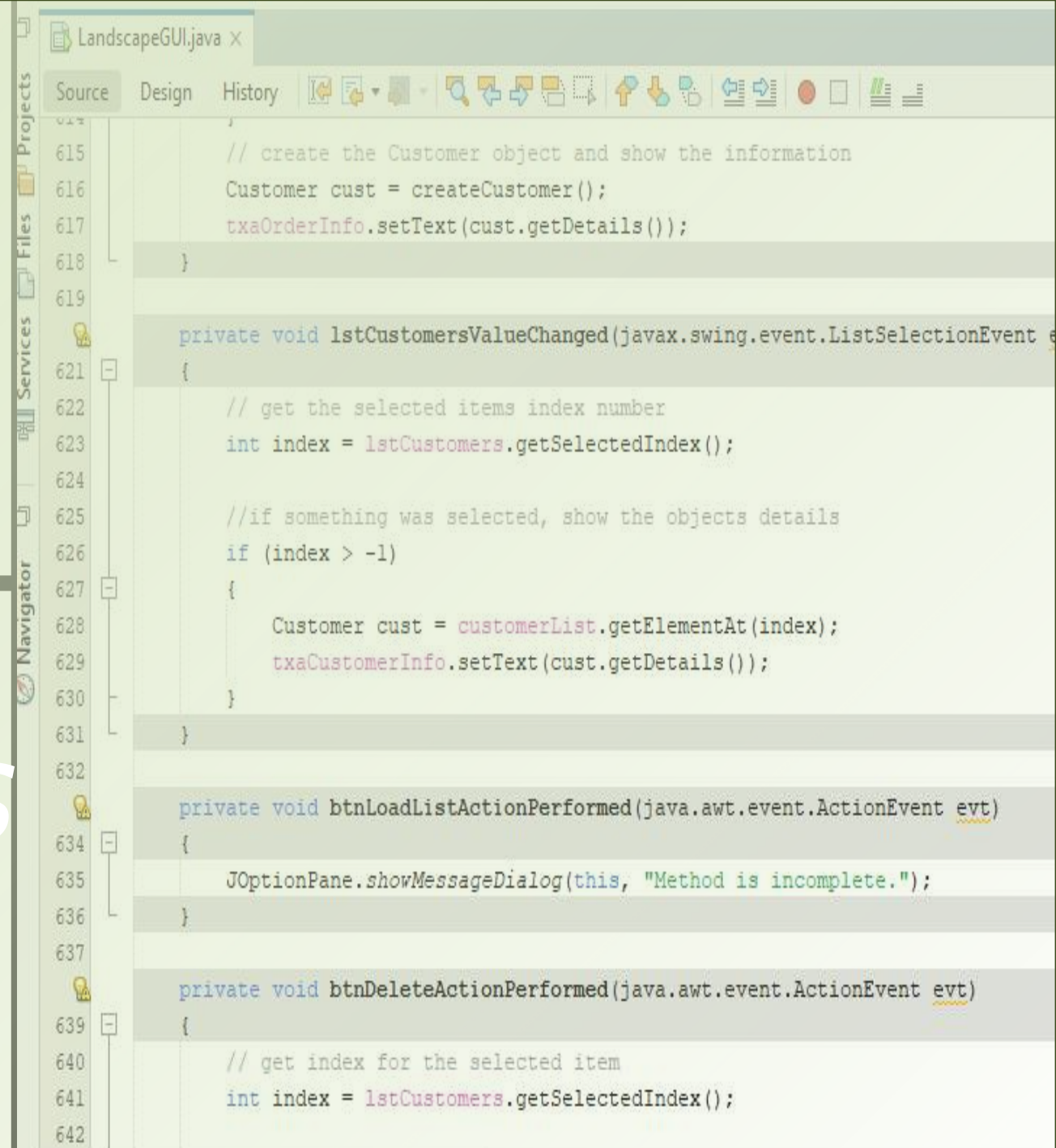
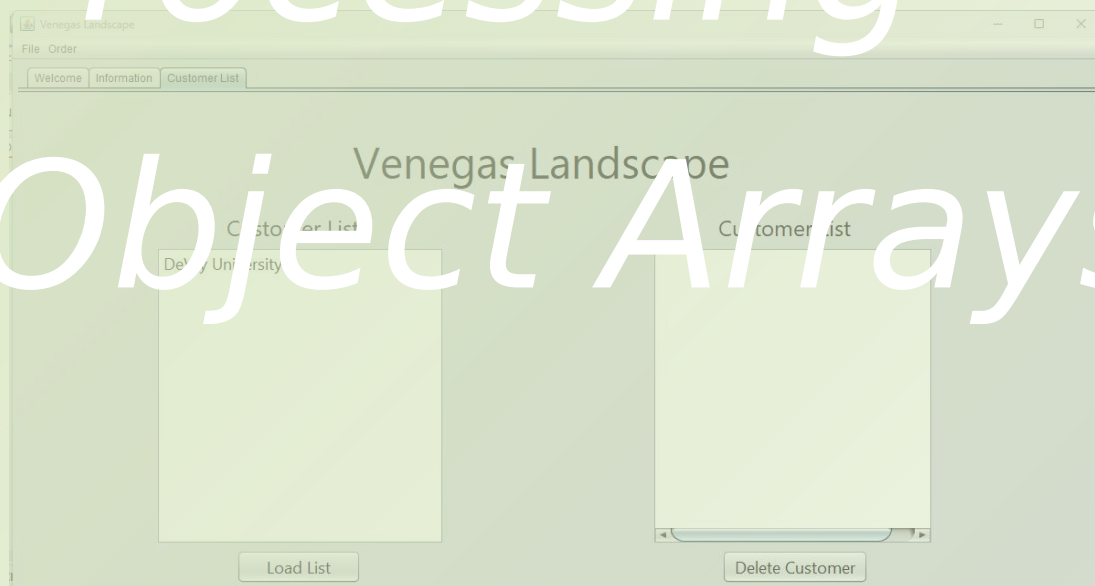
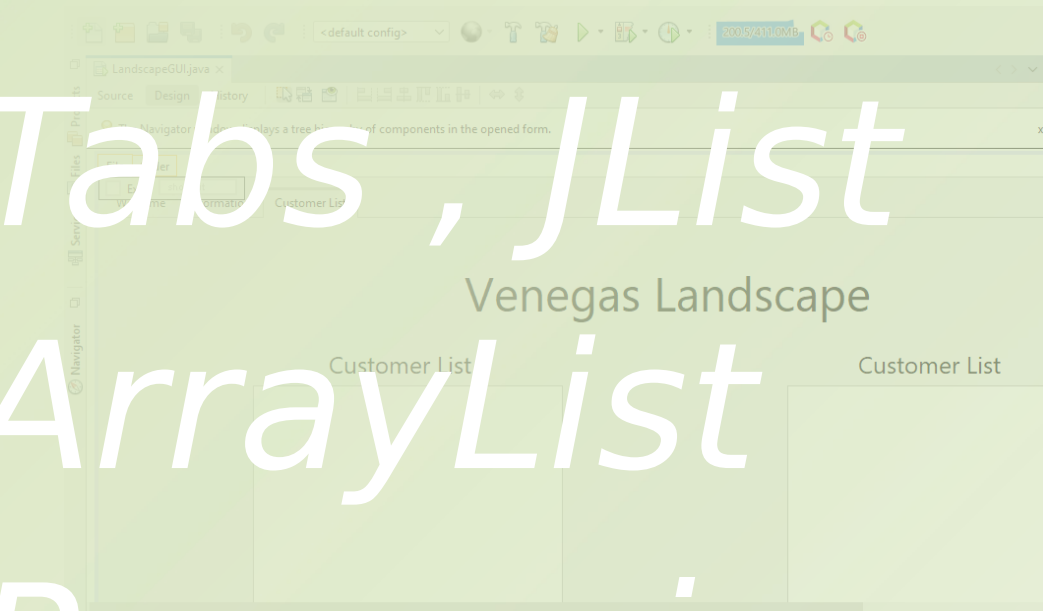


Inserting ArrayList, Tabs & JList Controls

- *Add Tabs*
- *Add JList*
- *Use ArrayList to
Process Object
Arrays*



Tabs, JList ArrayList Processing Object Arrays



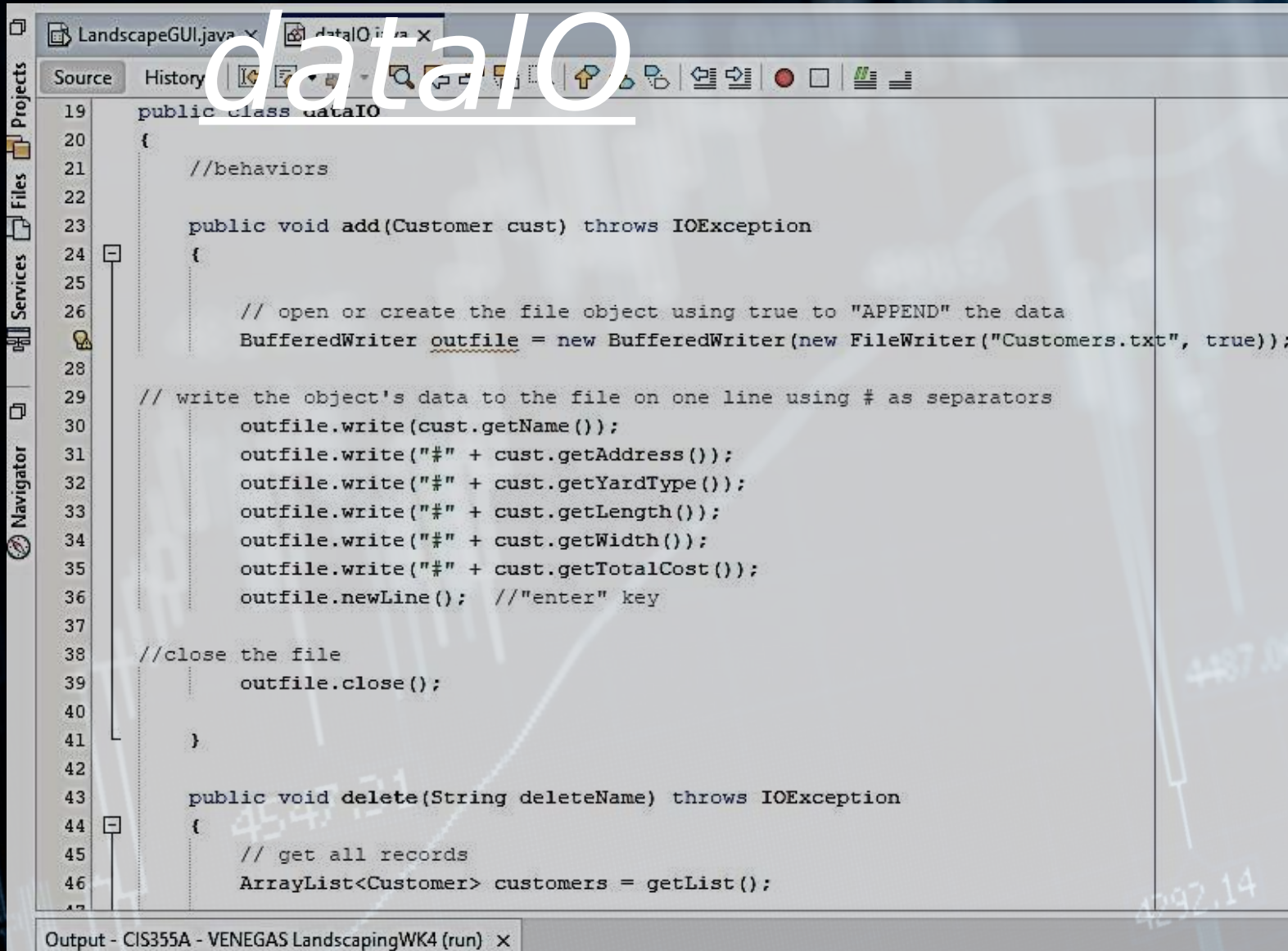
Data IO With Files

- Write data from the GUI to a file.
- Read data from the file and display the data on the GUI application.



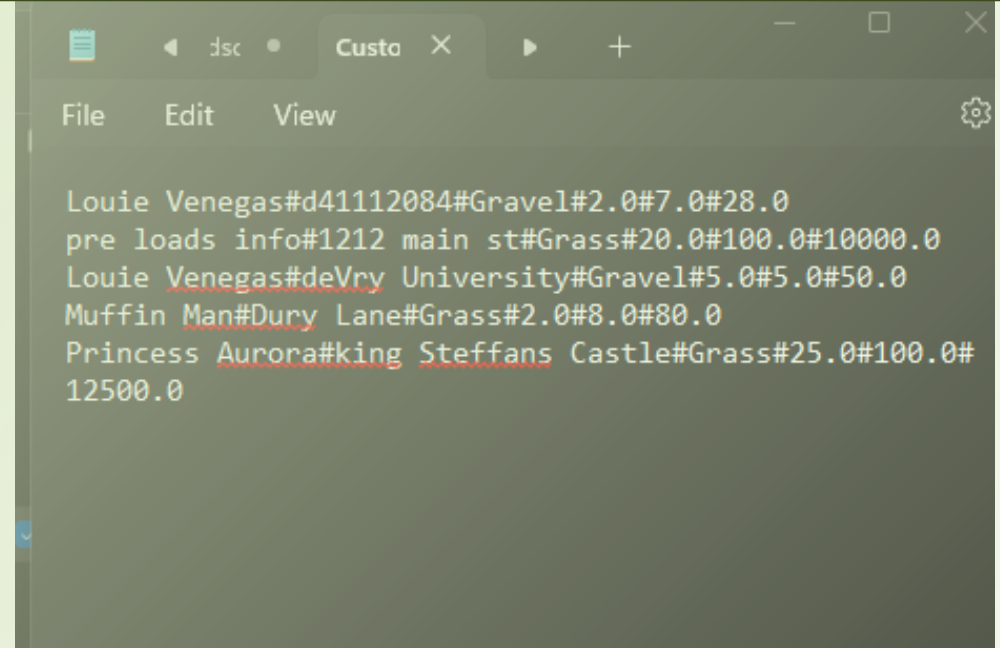
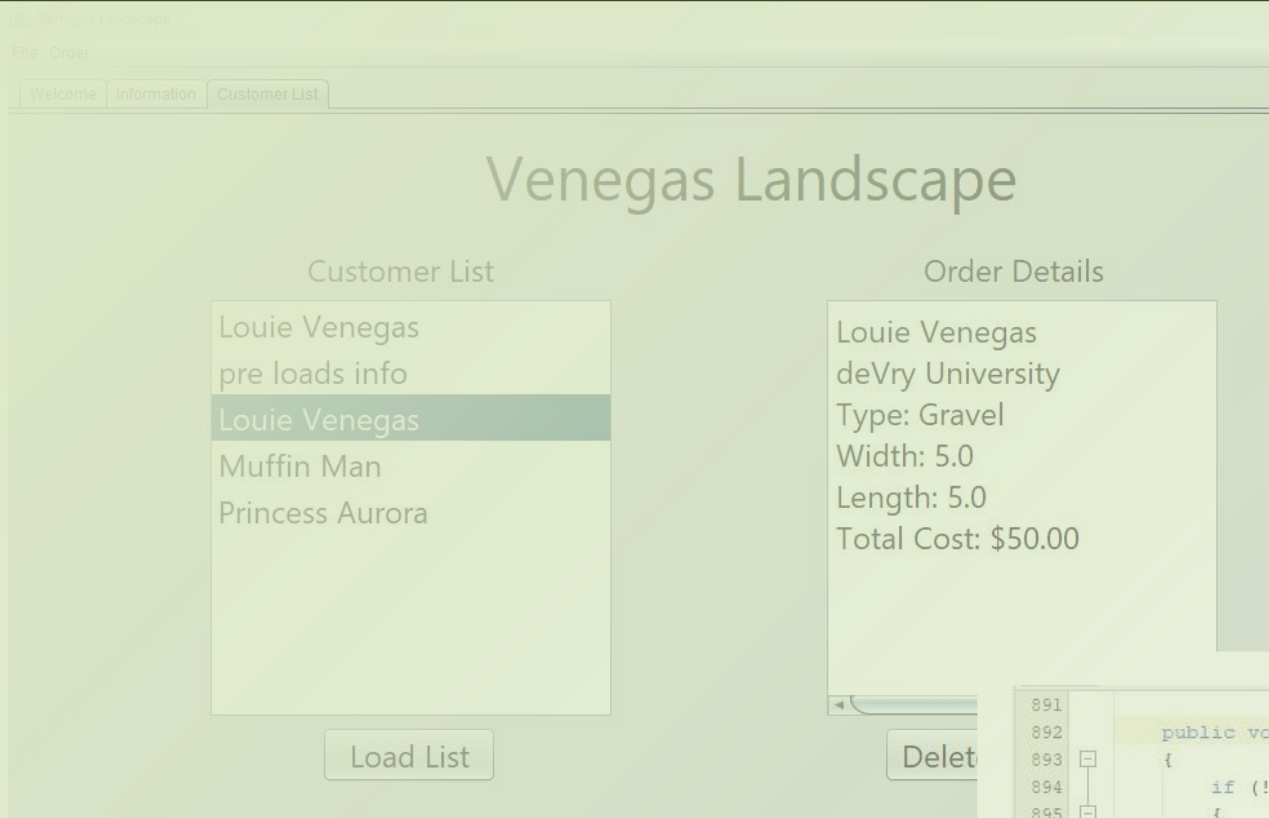
Public class

data/IO

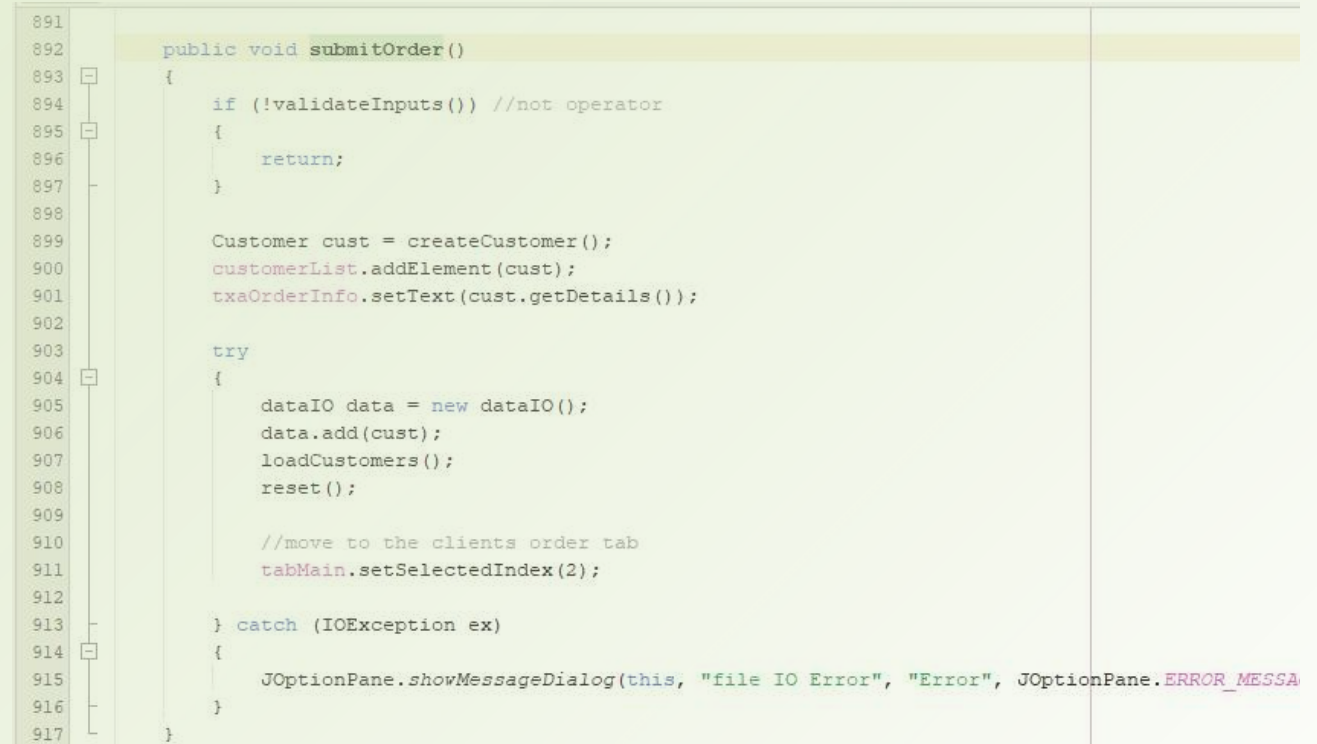


```
19 public class dataIO
20 {
21     //behaviors
22
23     public void add(Customer cust) throws IOException
24     {
25
26         // open or create the file object using true to "APPEND" the data
27         BufferedWriter outfile = new BufferedWriter(new FileWriter("Customers.txt", true));
28
29         // write the object's data to the file on one line using # as separators
30         outfile.write(cust.getName());
31         outfile.write("#" + cust.getAddress());
32         outfile.write("#" + cust.getYardType());
33         outfile.write("#" + cust.getLength());
34         outfile.write("#" + cust.getWidth());
35         outfile.write("#" + cust.getTotalCost());
36         outfile.newLine(); // "enter" key
37
38         //close the file
39         outfile.close();
40     }
41
42     public void delete(String deleteName) throws IOException
43     {
44         // get all records
45         ArrayList<Customer> customers = getList();
46     }
```

Output - CIS355A - VENEGAS LandscapingWK4 (run) x



*Writing data from the GUI to a file.
Reading data from the file and display the data on the GUI application.*





Data IO With Databases

- Write data from the GUI to a database.
- Read data from the database and display the data on the GUI application.
- Use a three-tier architecture to handle large development projects.

Public class dataIO

```
public class dataIO
{
    //constants
    private final String DATABASE_NAME = "cis355a";
    private final String CONNECTION_STRING = "jdbc:mysql://localhost:3306/" + DATABASE_NAME;
    private final String USER_NAME = "root";
    private final String PASSWORD = "Santino2.";

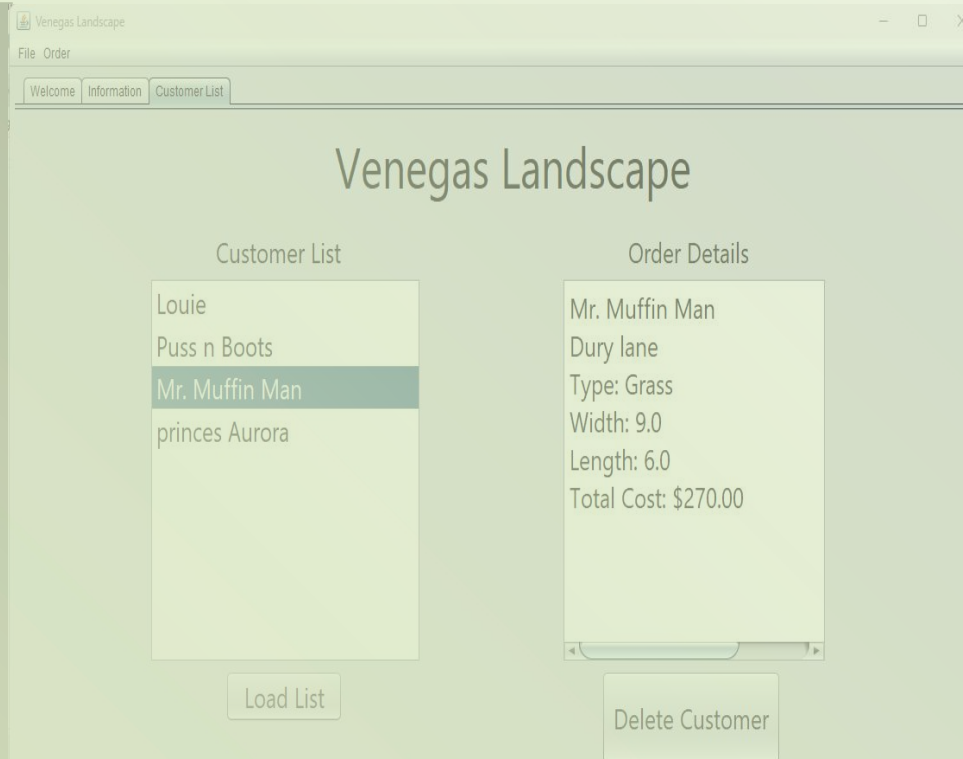
    //behaviors
    public void add(Customer cust) throws ClassNotFoundException, SQLException
    {
        //check for driver
        Class.forName("com.mysql.cj.jdbc.Driver");

        //connect to database
        Connection conn = DriverManager.getConnection(CONNECTION_STRING, USER_NAME, PASSWORD);

        //add record
        String strSQL = "Insert Into landscape (CustomerName, CustomerAddress,"
            + "LandscapeType,YardLength, YardWidth,LandscapeCost) "
            + "VALUES (?, ?, ?, ?, ?, ?) ";
        PreparedStatement pstmt = conn.prepareStatement(strSQL);
        pstmt.setString(1, cust.getName());
        pstmt.setString(2, cust.getAddress());
        pstmt.setString(3, cust.getYardType());
        pstmt.setDouble(4, cust.getLength());
    }
}
```

```
private void btnLoadListActionPerformed(java.awt.event.ActionEvent evt)
{
    loadCustomers();
}
```

Writing data
from the GUI
to a
database.
Reading data
from the
database and
display the
data on the



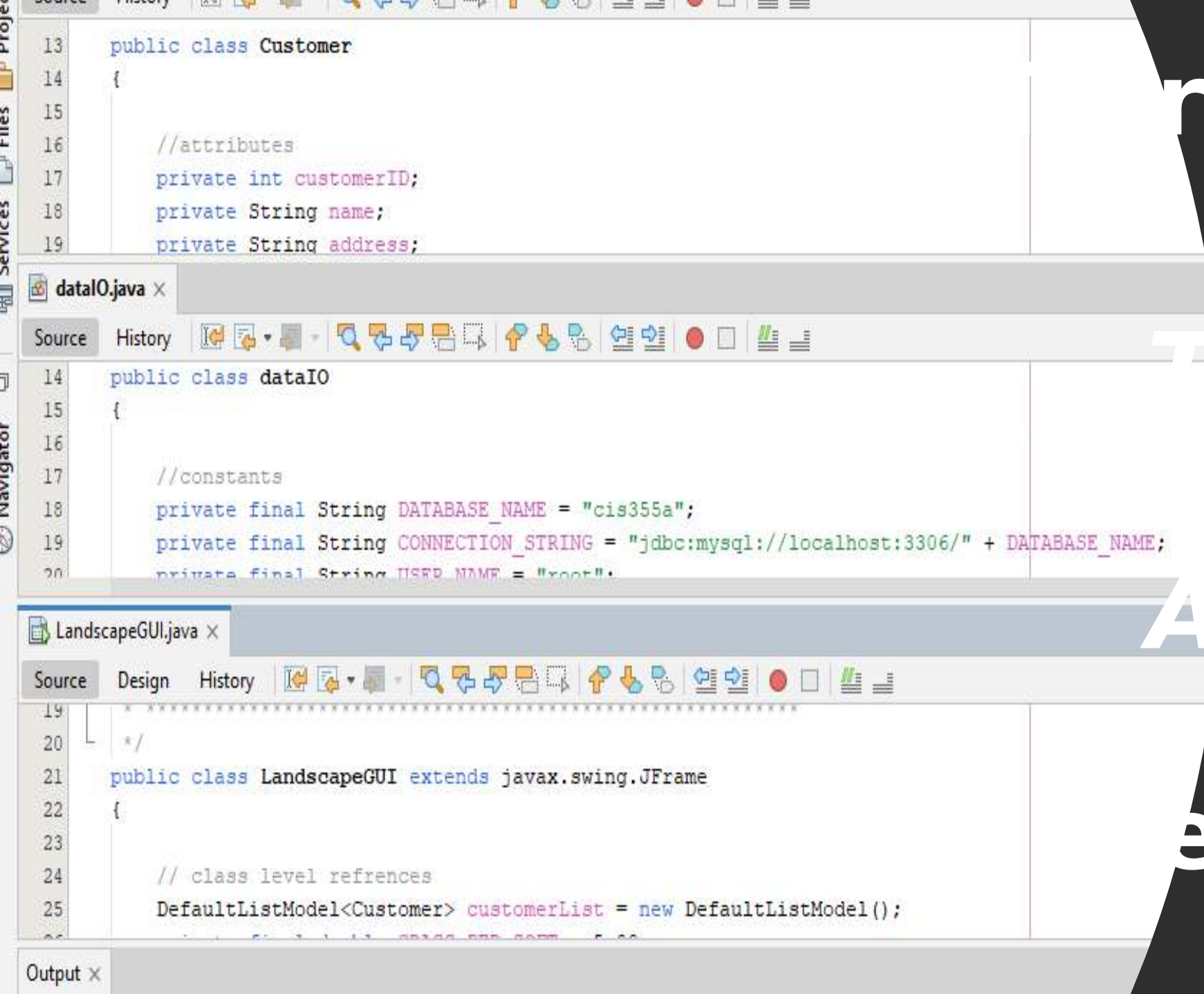
```
private void loadCustomers()
{
    try
    {
        dataIO data = new dataIO(); // create DataIO object
        ArrayList<Customer> customers = data.getList();

        // clear out the DefaultListModel and textarea
        customerList.clear();
        txtaOrderInfo.setText("");

        // copy each object from the ArrayList over to the DefaultListModel
        for (int i = 0; i < customers.size(); i++)
        {
            customerList.addElement(customers.get(i));
        }
    } catch (SQLException ex)
    {
        JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage(),
            "File IO Error", JOptionPane.ERROR_MESSAGE);
    }
}
```

```
5 • select * from landscape
```

[illegible]



Three Tier Architecture

er

Challenge

Developing a GUI application with NetBeans in Java presents several challenges. Managing the layout can be complex. Integrating event handling requires a solid understanding of Java's event-driven model. Additionally, connecting the program to a database introduced difficulties such as configuring the database connection, handling SQL queries, and ensuring data integrity. Debugging and testing the application was also very time-consuming, especially when unexpected behavior arises. Working through those problems is where most of the learning happened.



Career

Developing this GUI application with NetBeans in Java has significantly enhanced my career skills. This process involved taking on various challenges that contributed to my growth as a developer.

- Design and manage responsive user interfaces: I gained expertise in creating layouts that work seamlessly across different screen sizes.**
- Integrate event handling: I deepened my understanding of Java's event-driven programming model to create interactive applications.**
- Connect to a database: I learned valuable skills in database configuration, SQL query handling, and ensuring data integrity.**
- Debug and test applications: I improved my problem-solving abilities and attention to detail by addressing unexpected behavior and performance issues.**
- Enhance user experience: I developed a keen eye for creating user-friendly and intuitive applications.**

Together these gave me a working foundation in building, connecting, and

Peer Review



[Trent Tanney](#)

Aug 22 7:40am



Good morning Louie,

Your name: Trent Tanney

Student you are reviewing: Louie Venegas

Functionality (1-10) with comments: 10 - The information is formatted well, it's easy to read, and I think you have a great start!

Design (1-10) with comments: 10 - The design is very clean and professional. I like the dark/muted colors, it brings the focus to the information presented.

Communication (1-10) with comments: 10 The information is communicated very clearly. When I did mine, I struggled with what information to put on my section introduction slides and what to include with the screenshots of my application/code. It looks like you should have a good balance there.

Trent



Reply



Like



Mark as Unread