

Criterion C: Development

Introduction:

The Java program is a database of trips and expenses that allows my client to input and manage any trip expenses.

Summary of Programming Techniques:

- for loop
- method returning a value
- array list
- user defined objects made from an OOP "template" class
- making an array of objects
- simple and compound selection (if/else)
- sorting
- GUI tabs
- GUI popup menus
- use of a flag value (such as -999, or "not set yet")
- overloaded constructors, which work differently depending on the parameters sent
- inheritance between a super class and a sub class
- encapsulation

Structure of the Program:

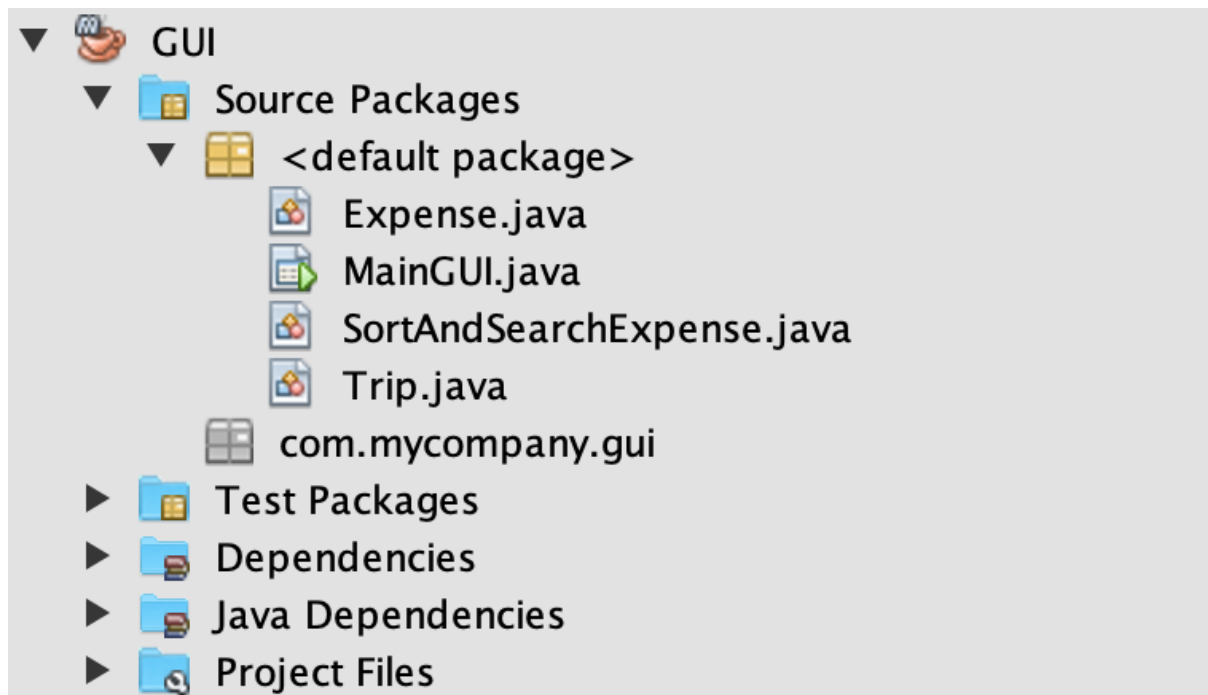


Figure 1: Structure of Program

The program is a database that stores and organizes trips and their expenses. Each expense has its own attributes from the ArrayList of expenses, which stores the different types of information.

I used Object-Oriented Programming to develop my program, as it allows me to use OOP features, such as Inheritance and Encapsulation, as well as its modelling of real-world relationships.

Our usage of Inheritance by having a superclass of Trips and a subclass of Expenses reduces redundancy, which would then allow for faster development and more efficient testing. The usage of Encapsulation allows data to be handled in a private and secure way using the get and set methods.

Data Structures Used: ArrayLists

ArrayLists are a key part of this program. The ArrayList of Expenses were used over LinkedLists as given the scenario, the faster random read access is preferable over the faster and simpler adding and removing of elements from the list. It was also chosen over arrays, as it is a dynamic data structure, which saves the user memory.

Algorithms:

Inputting Information: Inputting information is one of the key features of the program. Below is the code that inputs the trip details, which accounts for any user errors in inputting data in the first half, and actually inputs the data in the second half.

```
if(nameOfTripTF.equals("")){
    errorTF.setText("Please enter a Trip Name");
} else if(numberOfPeopleTF.equals("")) {
    errorTF.setText("Please enter the Number of People");
} else if(numberOfPeopleTF.equals("^\\d+(\\.\\d+)?")){
    errorTF.setText("Please enter the Number of People");
} else if (budgetTF.equals("")) {
    errorTF.setText("Please enter a Budget");
} else if(budgetTF.equals("^\\d+(\\.\\d+)?")) {
    errorTF.setText("Please enter a Budget");
} else {
    tripsQueue.add(new Trip(nameOfTripTF.getText(), Integer.parseInt(numberOfPeopleTF.getText()),
        Double.parseDouble(budgetTF.getText())));
    tripNameTF.setText(nameOfTripTF.getText());
    nameOfTripTF.setText("");
    numberOfPeopleTF.setText("");
    budgetTF.setText("");
}
```

Sorting Expenses: One of the key features of the program is the sorting of the different attributes of the expenses. Below is an example of one of the sorting algorithms, sorting the expenses based on the trip names, which would group the various expenses by the trip name.

```

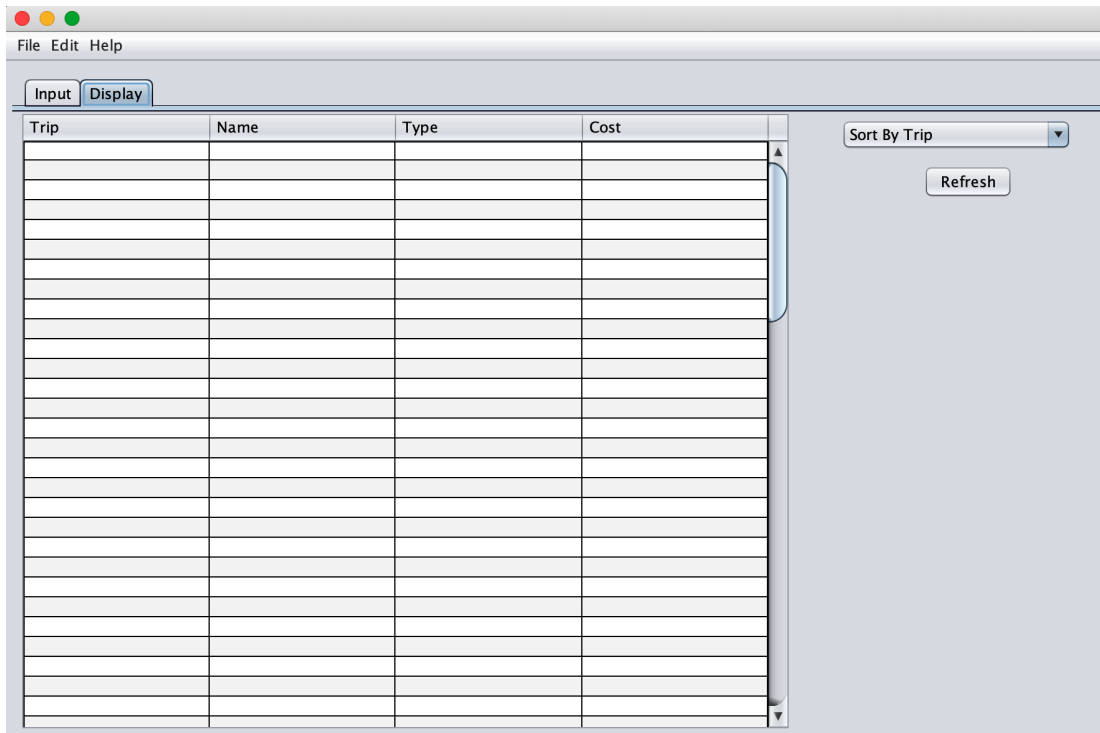
public void selectionSortOfExpensesByTrip(ArrayList<Expense> expenses) {
    for (int i = 0; i < expenses.size() - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < expenses.size(); j++) {
            if (expenses.get(j).getTripName().compareTo(expenses.get(minIndex).getTripName()) < 0) {
                minIndex = j;
            }
        }
        if (minIndex != i) {
            Expense temp = expenses.get(i);
            expenses.set(i, expenses.get(minIndex));
            expenses.set(minIndex, temp);
        }
    }
}
}

```

User Interface/GUI Work

The GUI was created such that it would be simple to use for the client. Each textfield and button has a corresponding label that the client could understand. It was a similar interface to what is used by the client at work, so there is already the level of familiarity.

The screenshot shows a Java Swing window with a menu bar (File, Edit, Help) and two buttons (Input, Display). The main area is divided into two columns: 'New Trip' and 'Expenses'. The 'New Trip' column contains three text input fields labeled 'Name of Trip', 'Number of People', and 'Budget', with an 'Enter' button below them. The 'Expenses' column contains four text input fields labeled 'Trip', 'Name of Expense', 'Cost of Expense', and 'Other', and a radio button group for 'Type of Expense' with options: 'Travel', 'Food', 'Accommodation', 'Activity', and 'Other'. There is also an 'Enter' button below the 'Cost of Expense' field.



Software Tools Used

NetBeans, an integrated development environment for Java was employed as it is a user-friendly but powerful GUI Builder, with access to code libraries, supports Java swing components, and its object-oriented programming support.

Total Word Count: 457