Group Name

Project Title

- **Budget Planner**

Group Members Matric Numbers and Names

1.	**Praise Ikenna Onyeaghala**	**BHU/22/04/05/0092 (Group Leader).**
2.	**Daniel Sase Jr**	**BHU/22/04/05/0061**
3.	**Samson Praise Chidera**	**BHU/22/04/09/0028**
4.	**Shettima IJASINI DANIEL**	**BHU/22/04/09/0004**
5.	**Kashim samaila**	**BHU/22/04/05/0068**

Project Description and Screenshots

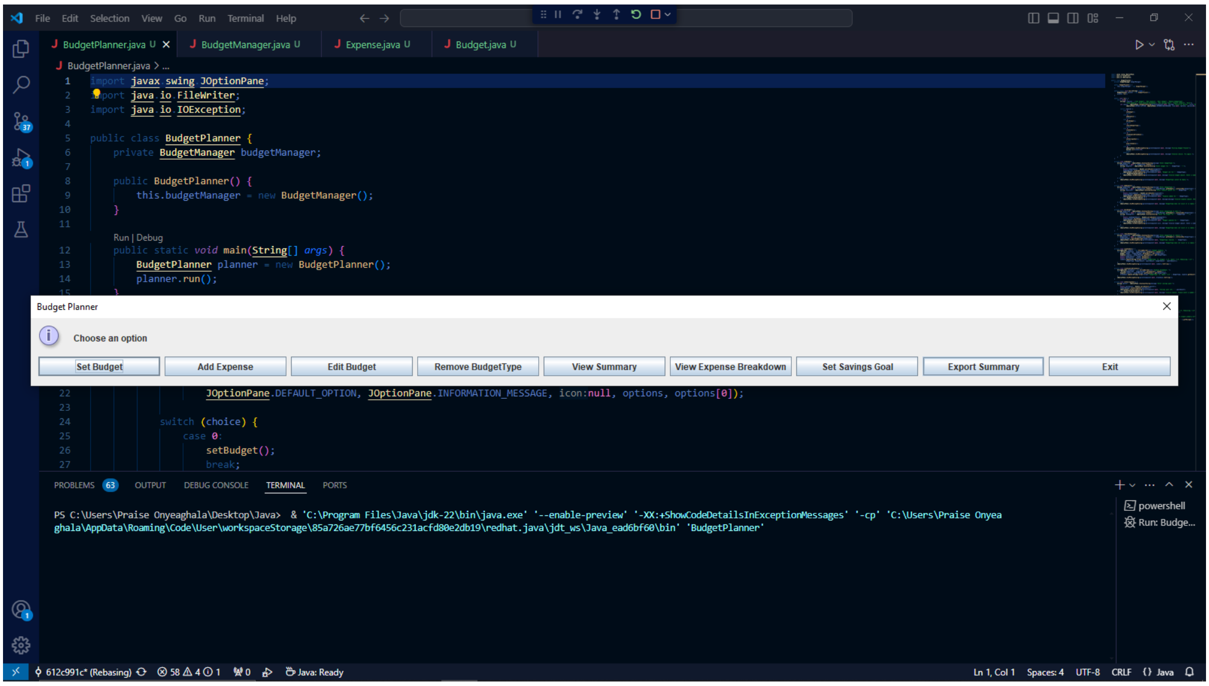
The Budget Planner app is a basic Java program that helps you manage your money. It allows you to set budgets, track expenses, and view financial summaries. The app uses a simple graphical interface with pop-up windows for interacting with the user. Its main parts include:

1. **BudgetPlanner**: The main part of the app that provides the user interface. It lets you do things like set budgets, add expenses, edit budgets, remove categories, view summaries, set savings goals, view expense breakdowns, and export summaries.

2. **BudgetManager**: This part handles the budgets and expenses. It stores data in collections and lets you do things like set budgets, add expenses, edit budgets, remove categories, and set savings goals.

3. **Budget**: This represents a budget for a specific category. It has details about the category and the budgeted amount.

4. **Expense**: This represents expenses for a specific category. It has details about the category and the total amount spent. The app is created using Object-Oriented Programming (OOP) principles. This makes it modular, easy to maintain, and simple to add new features.



```java
import javax.swing.JOptionPane;
import java.io.FileWriter;
import java.io.IOException;

public class BudgetPlanner {
    private BudgetManager budgetManager;

    public BudgetPlanner() {
        this.budgetManager = new BudgetManager();
    }

    Run | Debug
    public static void main(String[] args) {
        BudgetPlanner planner = new BudgetPlanner();
        planner.run();
    }
```

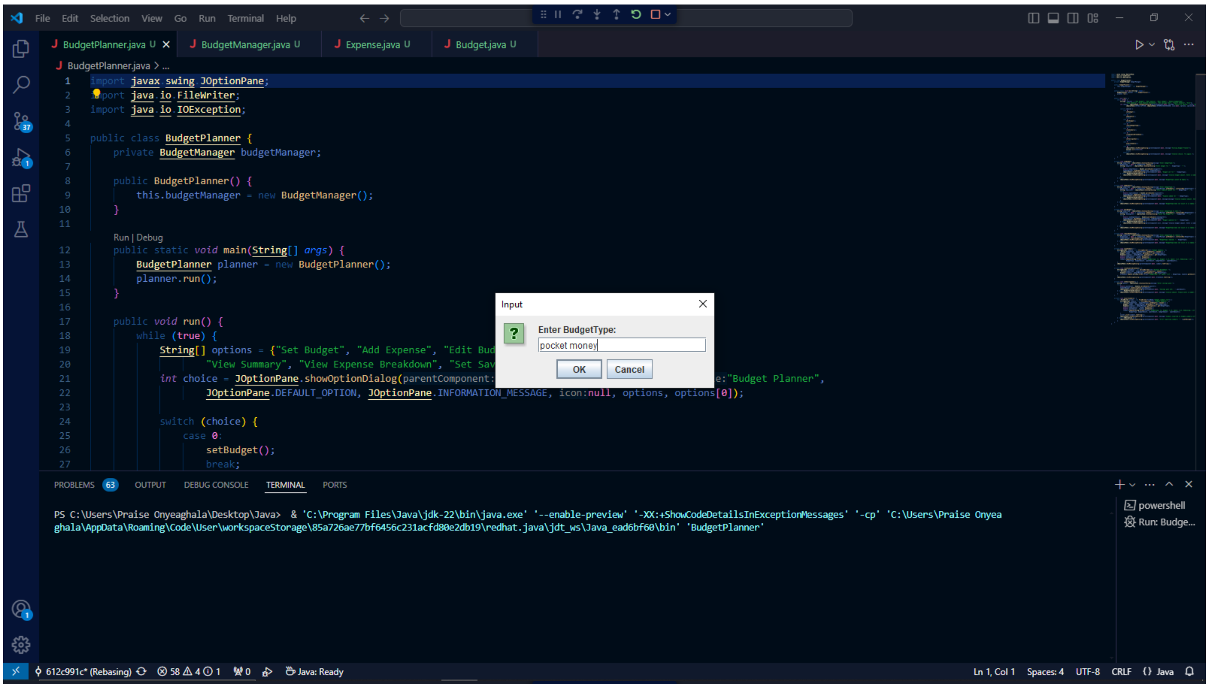
Budget Planner

ⓘ Choose an option

Set Budget | Add Expense | Edit Budget | Remove BudgetType | View Summary | View Expense Breakdown | Set Savings Goal | Export Summary | Exit

```java
                JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, icon:null, options, options[0]);

            switch (choice) {
                case 0:
                    setBudget();
                    break;
```

```
PS C:\Users\Praise Onyeaghala\Desktop\Java>  & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Praise Onyea
ghala\AppData\Roaming\Code\User\workspaceStorage\85a726ae77bf6456c231acfd80e2db19\redhat.java\jdt_ws\Java_ead6bf60\bin' 'BudgetPlanner'
```



```java
import javax.swing.JOptionPane;
import java.io.FileWriter;
import java.io.IOException;

public class BudgetPlanner {
    private BudgetManager budgetManager;

    public BudgetPlanner() {
        this.budgetManager = new BudgetManager();
    }

    public static void main(String[] args) {
        BudgetPlanner planner = new BudgetPlanner();
        planner.run();
    }

    public void run() {
        while (true) {
            String[] options = {"Set Budget", "Add Expense", "Edit Bud
                    "View Summary", "View Expense Breakdown", "Set Sav
            int choice = JOptionPane.showOptionDialog(parentComponent:                e:"Budget Planner",
                    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, icon:null, options, options[0]);

            switch (choice) {
                case 0:
                    setBudget();
                    break;
```

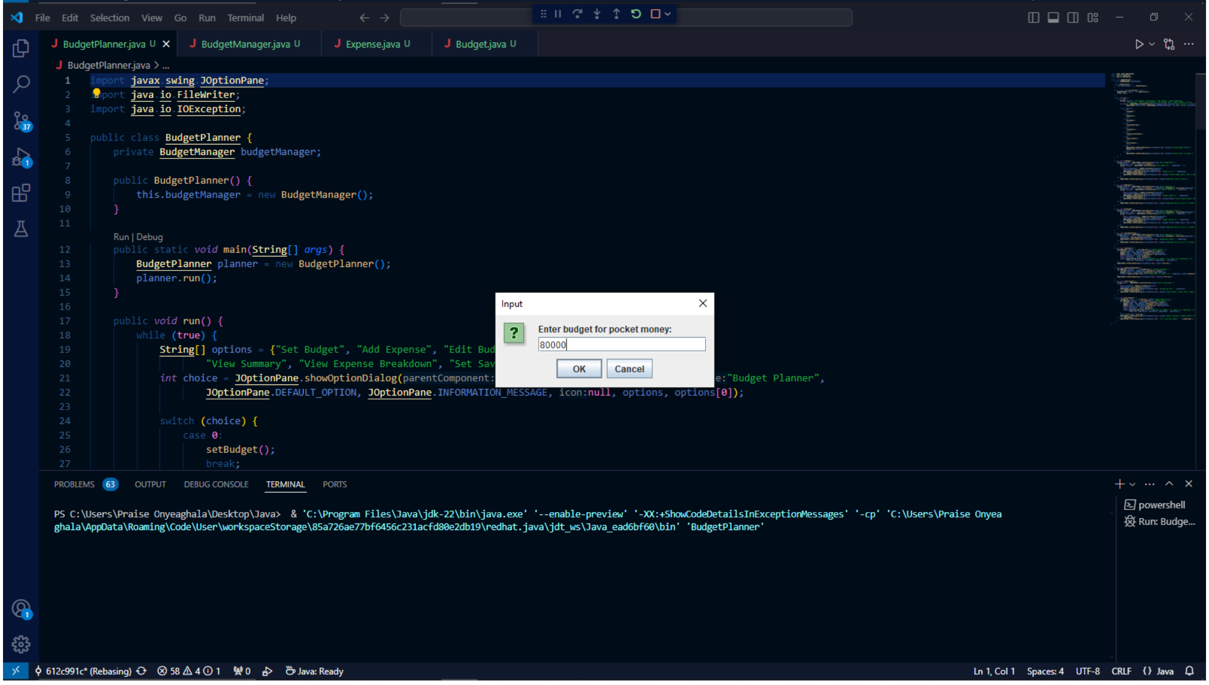
Input

Enter BudgetType:

pocket money

OK Cancel

```
PS C:\Users\Praise Onyeaghala\Desktop\Java>  & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Praise Onyea
ghala\AppData\Roaming\Code\User\workspaceStorage\85a726ae77bf6456c231acfd80e2db19\redhat.java\jdt_ws\Java_ead6bf60\bin' 'BudgetPlanner'
```



```java
import javax.swing.JOptionPane;
import java.io.FileWriter;
import java.io.IOException;

public class BudgetPlanner {
    private BudgetManager budgetManager;

    public BudgetPlanner() {
        this.budgetManager = new BudgetManager();
    }

    public static void main(String[] args) {
        BudgetPlanner planner = new BudgetPlanner();
        planner.run();
    }

    public void run() {
        while (true) {
            String[] options = {"Set Budget", "Add Expense", "Edit Bud
                    "View Summary", "View Expense Breakdown", "Set Sav
            int choice = JOptionPane.showOptionDialog(parentComponent:                e:"Budget Planner",
                    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, icon:null, options, options[0]);

            switch (choice) {
                case 0:
                    setBudget();
                    break;
```

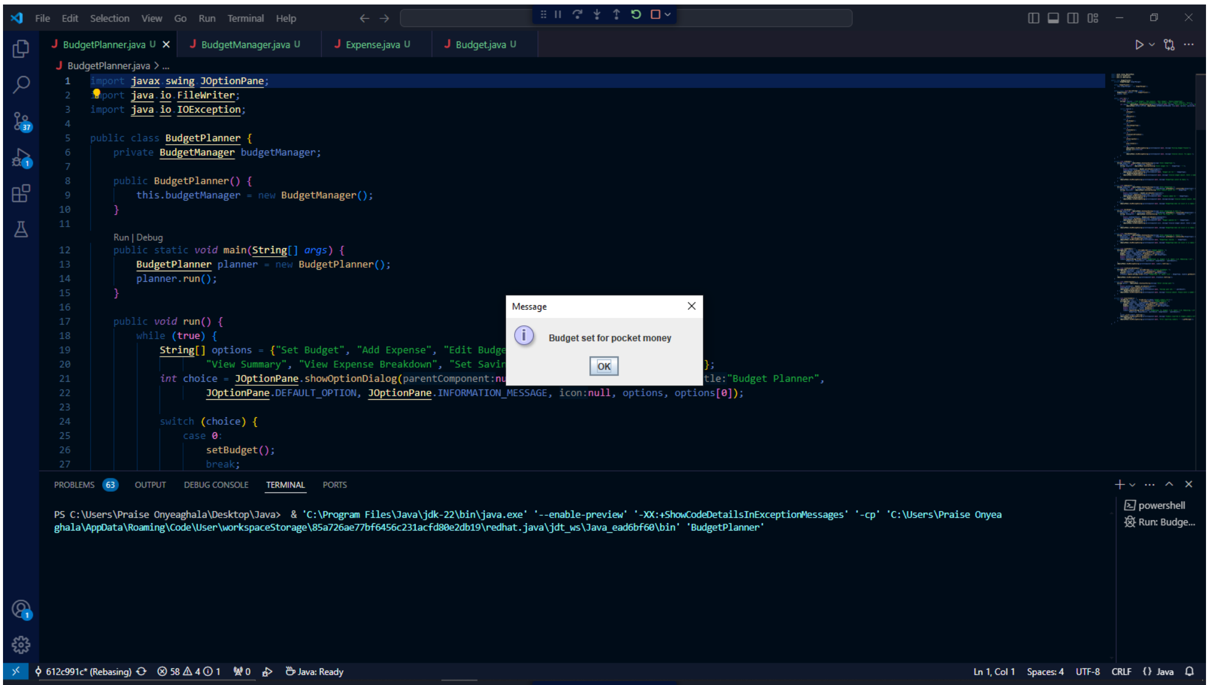
Input

Enter budget for pocket money:

80000

OK Cancel

```
PS C:\Users\Praise Onyeaghala\Desktop\Java>  & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Praise Onyea
ghala\AppData\Roaming\Code\User\workspaceStorage\85a726ae77bf6456c231acfd80e2db19\redhat.java\jdt_ws\Java_ead6bf60\bin' 'BudgetPlanner'
```



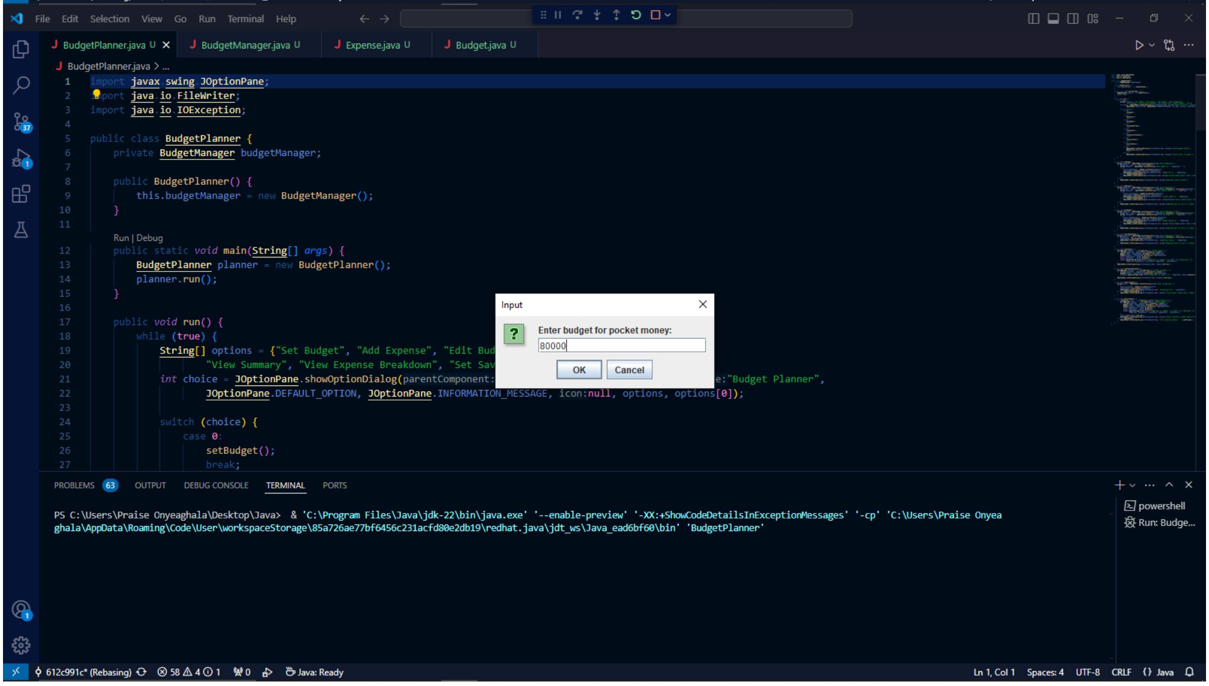
```java
import javax.swing.JOptionPane;
import java.io.FileWriter;
import java.io.IOException;

public class BudgetPlanner {
    private BudgetManager budgetManager;

    public BudgetPlanner() {
        this.budgetManager = new BudgetManager();
    }

    Run | Debug
    public static void main(String[] args) {
        BudgetPlanner planner = new BudgetPlanner();
        planner.run();
    }

    public void run() {
        while (true) {
            String[] options = {"Set Budget", "Add Expense", "Edit Budge
                    "View Summary", "View Expense Breakdown", "Set Savir                          };
            int choice = JOptionPane.showOptionDialog(parentComponent:nu                          tle:"Budget Planner",
                    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, icon:null, options, options[0]);

            switch (choice) {
                case 0:
                    setBudget();
                    break;
```

Message

Budget set for pocket money

OK

```
PS C:\Users\Praise Onyeaghala\Desktop\Java>  & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Praise Onyea
ghala\AppData\Roaming\Code\User\workspaceStorage\85a726ae77bf6456c231acfd80e2db19\redhat.java\jdt_ws\Java_ead6bf60\bin' 'BudgetPlanner'
```



```java
import javax.swing.JOptionPane;
import java.io.FileWriter;
import java.io.IOException;

public class BudgetPlanner {
    private BudgetManager budgetManager;

    public BudgetPlanner() {
        this.budgetManager = new BudgetManager();
    }

    Run | Debug
    public static void main(String[] args) {
        BudgetPlanner planner = new BudgetPlanner();
        planner.run();
    }

    public void run() {
        while (true) {
            String[] options = {"Set Budget", "Add Expense", "Edit Bud
                    "View Summary", "View Expense Breakdown", "Set Sav
            int choice = JOptionPane.showOptionDialog(parentComponent:                                    e:"Budget Planner",
                    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, icon:null, options, options[0]);

            switch (choice) {
                case 0:
                    setBudget();
                    break;
```

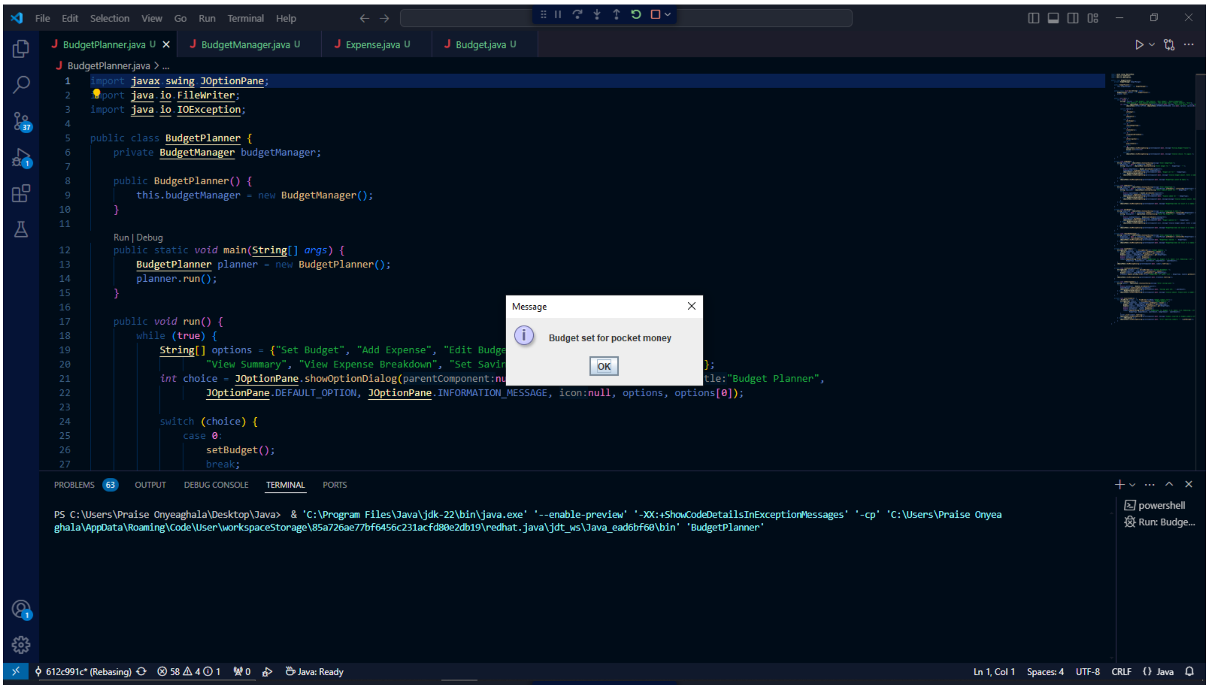
Input

Enter budget for pocket money:

80000

OK Cancel

```
PS C:\Users\Praise Onyeaghala\Desktop\Java>  & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Praise Onyea
ghala\AppData\Roaming\Code\User\workspaceStorage\85a726ae77bf6456c231acfd80e2db19\redhat.java\jdt_ws\Java_ead6bf60\bin' 'BudgetPlanner'
```



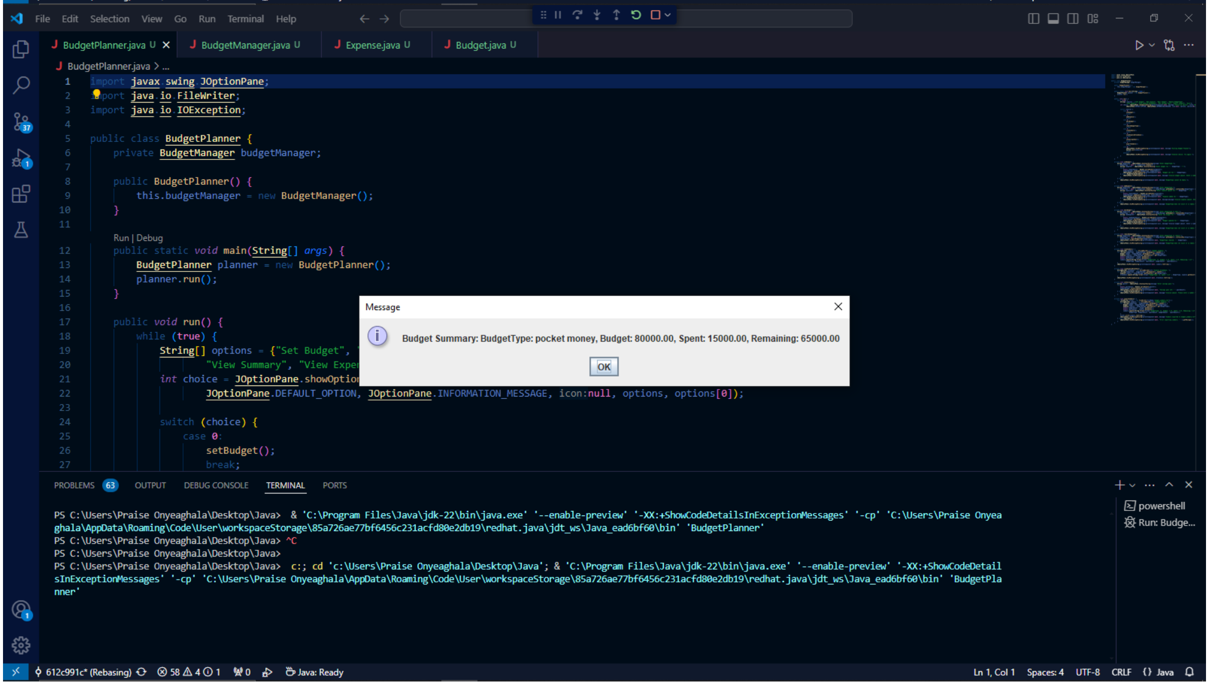
```
import javax.swing.JOptionPane;
import java.io.FileWriter;
import java.io.IOException;

public class BudgetPlanner {
    private BudgetManager budgetManager;

    public BudgetPlanner() {
        this.budgetManager = new BudgetManager();
    }

    public static void main(String[] args) {
        BudgetPlanner planner = new BudgetPlanner();
        planner.run();
    }

    public void run() {
        while (true) {
            String[] options = {"Set Budget", "Add Expense", "Edit Budge
                    "View Summary", "View Expense Breakdown", "Set Savir                                  };
            int choice = JOptionPane.showOptionDialog(parentComponent:nu                                  tle:"Budget Planner",
                    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, icon:null, options, options[0]);

            switch (choice) {
                case 0:
                    setBudget();
                    break;
```

Message

Budget set for pocket money

OK

```
PS C:\Users\Praise Onyeaghala\Desktop\Java>  & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Praise Onyea
ghala\AppData\Roaming\Code\User\workspaceStorage\85a726ae77bf6456c231acfd80e2db19\redhat.java\jdt_ws\Java_ead6bf60\bin' 'BudgetPlanner'
```



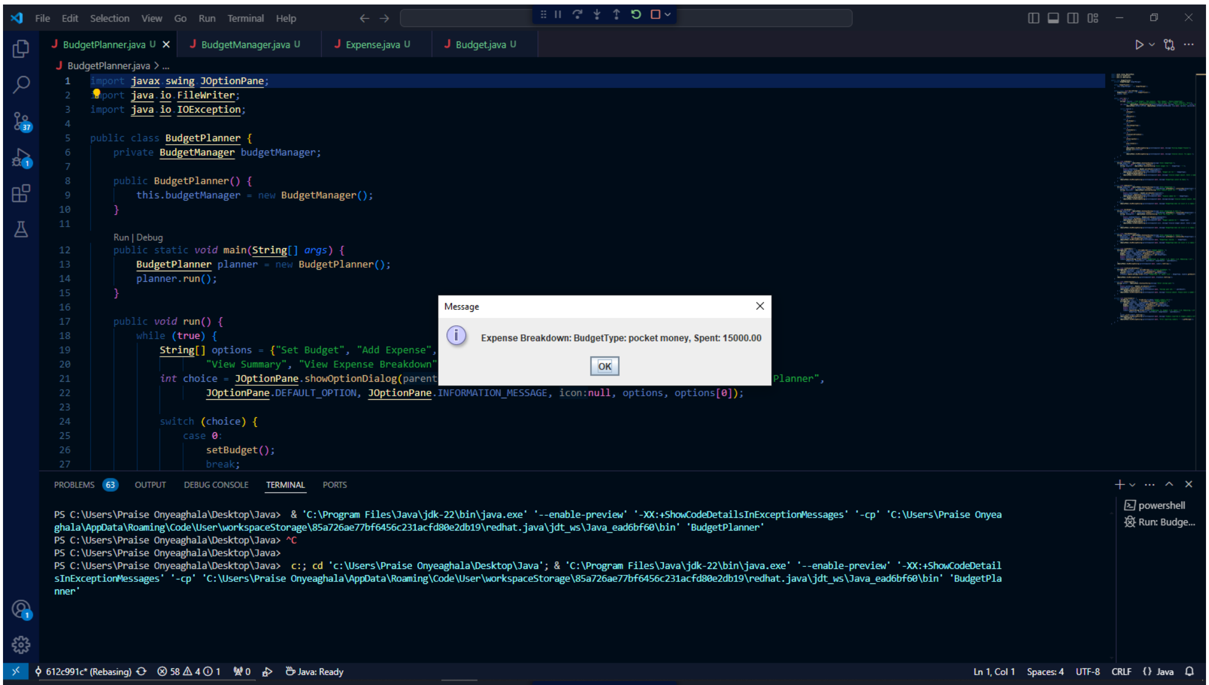
```
import javax.swing.JOptionPane;
import java.io.FileWriter;
import java.io.IOException;

public class BudgetPlanner {
    private BudgetManager budgetManager;

    public BudgetPlanner() {
        this.budgetManager = new BudgetManager();
    }

    public static void main(String[] args) {
        BudgetPlanner planner = new BudgetPlanner();
        planner.run();
    }

    public void run() {
        while (true) {
            String[] options = {"Set Budget",
                    "View Summary", "View Exper
            int choice = JOptionPane.showOptior
                    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, icon:null, options, options[0]);

            switch (choice) {
                case 0:
                    setBudget();
                    break;
```

Message

Budget Summary: BudgetType: pocket money, Budget: 80000.00, Spent: 15000.00, Remaining: 65000.00

OK

```
PS C:\Users\Praise Onyeaghala\Desktop\Java>  & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Praise Onyea
ghala\AppData\Roaming\Code\User\workspaceStorage\85a726ae77bf6456c231acfd80e2db19\redhat.java\jdt_ws\Java_ead6bf60\bin' 'BudgetPlanner'
PS C:\Users\Praise Onyeaghala\Desktop\Java> ^C
PS C:\Users\Praise Onyeaghala\Desktop\Java>
PS C:\Users\Praise Onyeaghala\Desktop\Java>  c:; cd 'c:\Users\Praise Onyeaghala\Desktop\Java'; & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetail
sInExceptionMessages' '-cp' 'C:\Users\Praise Onyeaghala\AppData\Roaming\Code\User\workspaceStorage\85a726ae77bf6456c231acfd80e2db19\redhat.java\jdt_ws\Java_ead6bf60\bin' 'BudgetPla
nner'
```



```java
import javax.swing.JOptionPane;
import java.io.FileWriter;
import java.io.IOException;

public class BudgetPlanner {
    private BudgetManager budgetManager;

    public BudgetPlanner() {
        this.budgetManager = new BudgetManager();
    }

    Run | Debug
    public static void main(String[] args) {
        BudgetPlanner planner = new BudgetPlanner();
        planner.run();
    }

    public void run() {
        while (true) {
            String[] options = {"Set Budget", "Add Expense",
                    "View Summary", "View Expense Breakdown"
            int choice = JOptionPane.showOptionDialog(parent                                          Planner",
                    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, icon:null, options, options[0]);

            switch (choice) {
                case 0:
                    setBudget();
                    break;
```

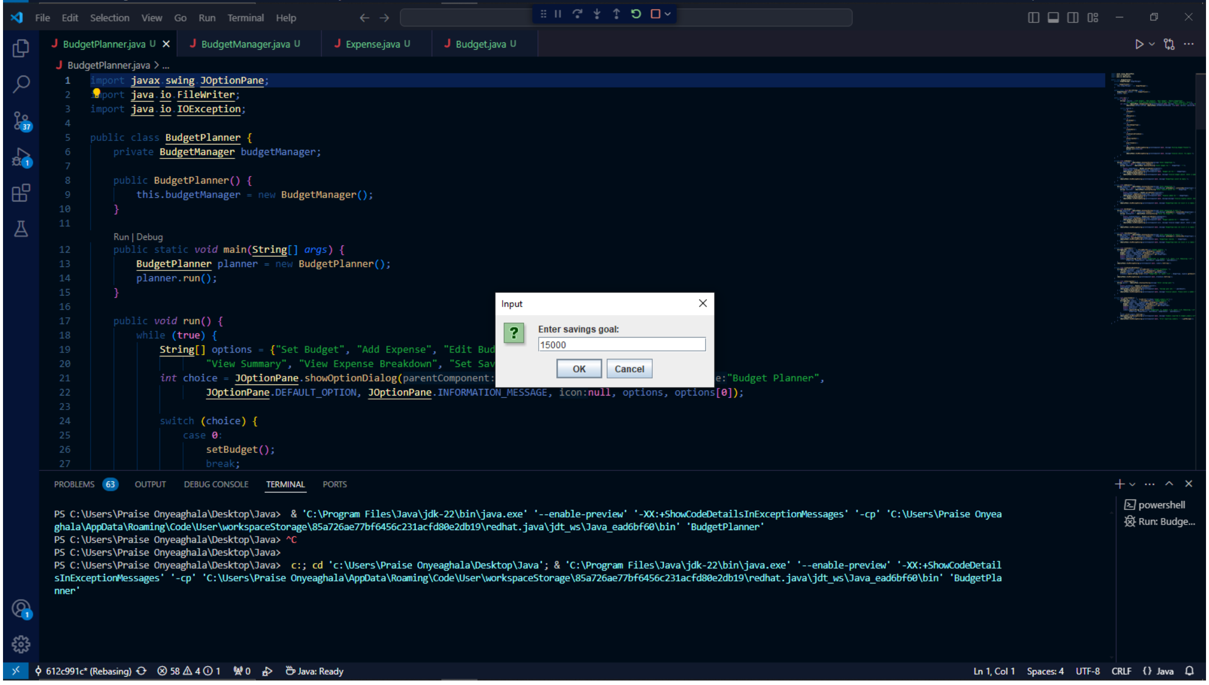
Message

Expense Breakdown: BudgetType: pocket money, Spent: 15000.00

OK

```
PS C:\Users\Praise Onyeaghala\Desktop\Java>  & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Praise Onyea
ghala\AppData\Roaming\Code\User\workspaceStorage\85a726ae77bf6456c231acfd80e2db19\redhat.java\jdt_ws\Java_ead6bf60\bin' 'BudgetPlanner'
PS C:\Users\Praise Onyeaghala\Desktop\Java> ^C
PS C:\Users\Praise Onyeaghala\Desktop\Java>
PS C:\Users\Praise Onyeaghala\Desktop\Java>  c:; cd 'c:\Users\Praise Onyeaghala\Desktop\Java'; & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetail
sInExceptionMessages' '-cp' 'C:\Users\Praise Onyeaghala\AppData\Roaming\Code\User\workspaceStorage\85a726ae77bf6456c231acfd80e2db19\redhat.java\jdt_ws\Java_ead6bf60\bin' 'BudgetPla
nner'
```

612c991c* (Rebasing) ⊗ 58 ⚠ 4 ⓘ 1 0 Java: Ready Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Java



```java
import javax.swing.JOptionPane;
import java.io.FileWriter;
import java.io.IOException;

public class BudgetPlanner {
    private BudgetManager budgetManager;

    public BudgetPlanner() {
        this.budgetManager = new BudgetManager();
    }

    Run | Debug
    public static void main(String[] args) {
        BudgetPlanner planner = new BudgetPlanner();
        planner.run();
    }

    public void run() {
        while (true) {
            String[] options = {"Set Budget", "Add Expense", "Edit Bud
                    "View Summary", "View Expense Breakdown", "Set Sav
            int choice = JOptionPane.showOptionDialog(parentComponent:                  e:"Budget Planner",
                    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, icon:null, options, options[0]);

            switch (choice) {
                case 0:
                    setBudget();
                    break;
```

Input

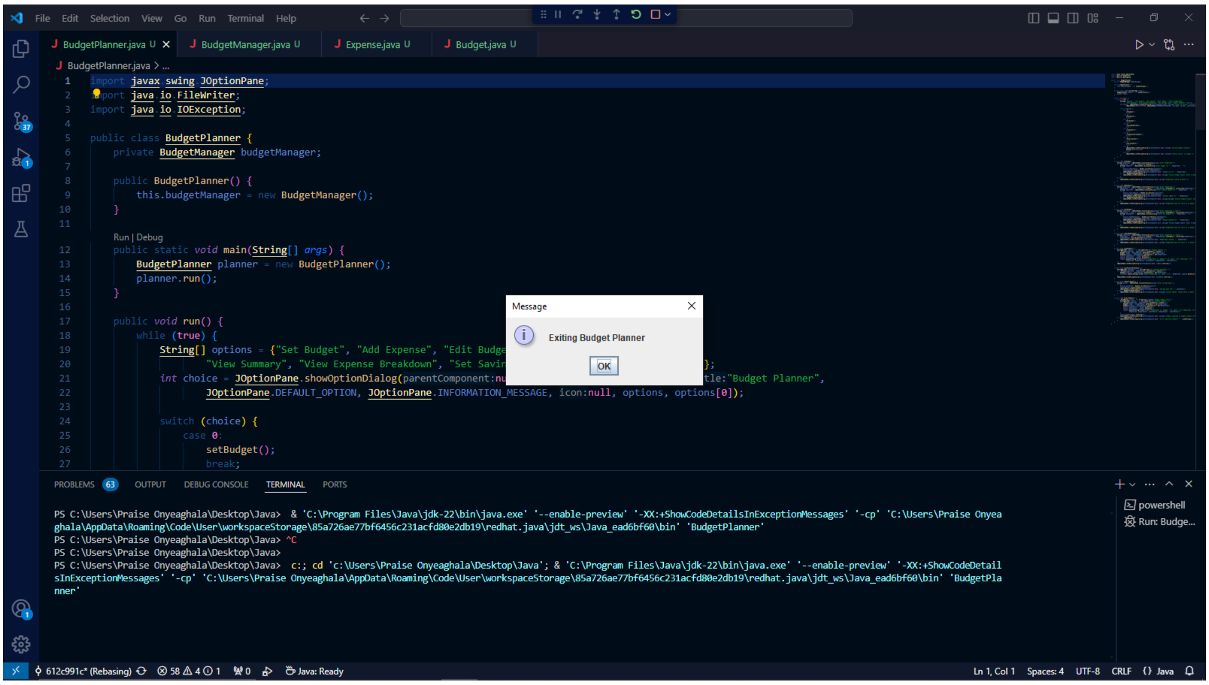
Enter savings goal:

15000

OK Cancel

```
PS C:\Users\Praise Onyeaghala\Desktop\Java>  & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Praise Onyea
ghala\AppData\Roaming\Code\User\workspaceStorage\85a726ae77bf6456c231acfd80e2db19\redhat.java\jdt_ws\Java_ead6bf60\bin' 'BudgetPlanner'
PS C:\Users\Praise Onyeaghala\Desktop\Java> ^C
PS C:\Users\Praise Onyeaghala\Desktop\Java>
PS C:\Users\Praise Onyeaghala\Desktop\Java>  c:; cd 'c:\Users\Praise Onyeaghala\Desktop\Java'; & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetail
sInExceptionMessages' '-cp' 'C:\Users\Praise Onyeaghala\AppData\Roaming\Code\User\workspaceStorage\85a726ae77bf6456c231acfd80e2db19\redhat.java\jdt_ws\Java_ead6bf60\bin' 'BudgetPla
nner'
```

612c991c* (Rebasing) ⊗ 58 ⚠ 4 ⓘ 1 0 Java: Ready Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Java



```
import javax.swing.JOptionPane;
import java.io.FileWriter;
import java.io.IOException;

public class BudgetPlanner {
    private BudgetManager budgetManager;

    public BudgetPlanner() {
        this.budgetManager = new BudgetManager();
    }

    public static void main(String[] args) {
        BudgetPlanner planner = new BudgetPlanner();
        planner.run();
    }

    public void run() {
        while (true) {
            String[] options = {"Set Budget", "Add Expense", "Edit Budge
                    "View Summary", "View Expense Breakdown", "Set Savir                              };
            int choice = JOptionPane.showOptionDialog(parentComponent:nu                              tle:"Budget Planner",
                    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, icon:null, options, options[0]);

            switch (choice) {
                case 0:
                    setBudget();
                    break;
```

Message

Exiting Budget Planner

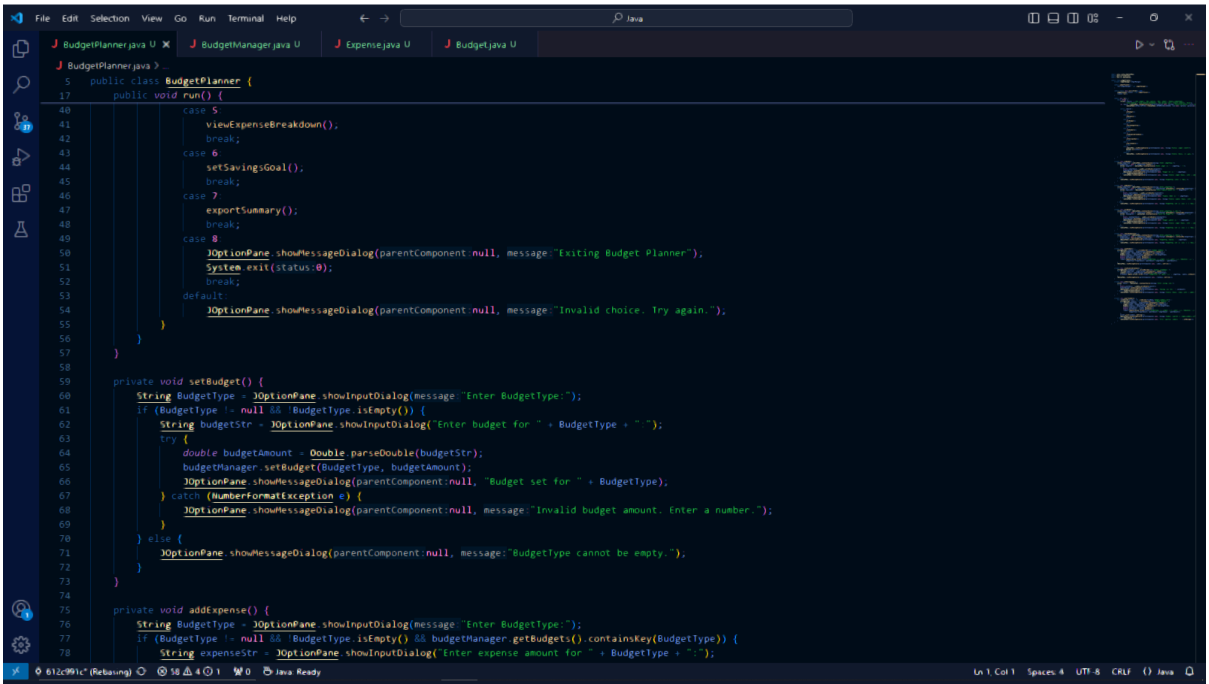
OK

```
PS C:\Users\Praise Onyeaghala\Desktop\Java>  & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Praise Onyea
ghala\AppData\Roaming\Code\User\workspaceStorage\85a726ae77bf6456c231acfd80e2db19\redhat.java\jdt_ws\Java_ead6bf60\bin' 'BudgetPlanner'
PS C:\Users\Praise Onyeaghala\Desktop\Java> ^C
PS C:\Users\Praise Onyeaghala\Desktop\Java>
PS C:\Users\Praise Onyeaghala\Desktop\Java>  c:; cd 'c:\Users\Praise Onyeaghala\Desktop\Java'; & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetail
sInExceptionMessages' '-cp' 'C:\Users\Praise Onyeaghala\AppData\Roaming\Code\User\workspaceStorage\85a726ae77bf6456c231acfd80e2db19\redhat.java\jdt_ws\Java_ead6bf60\bin' 'BudgetPla
nner'
```

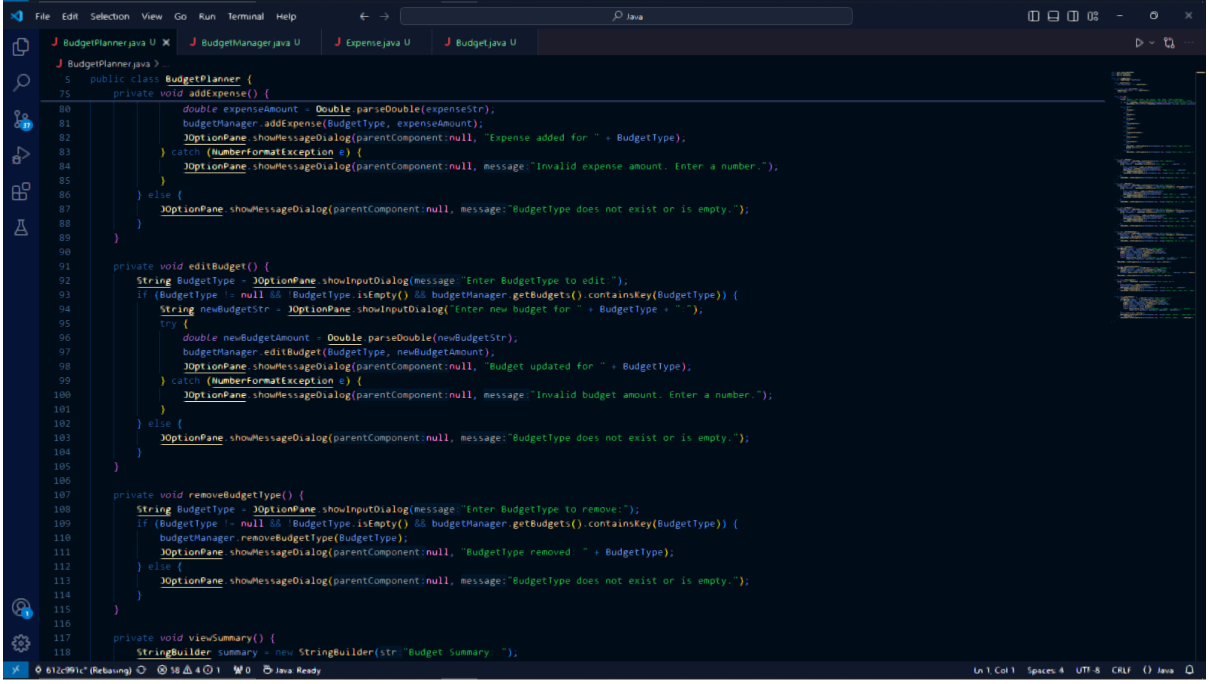
Team Member One Contribution(Praise ikenna Onyeaghala BHU/22/04/05/0092)

The BudgetPlanner class in the Budget Planner application manages user interactions, using dialog boxes for input and information display. It serves as the controller in the Model-View-Controller design pattern, handling data flow between the user interface and the business logic. Responsibilities include:

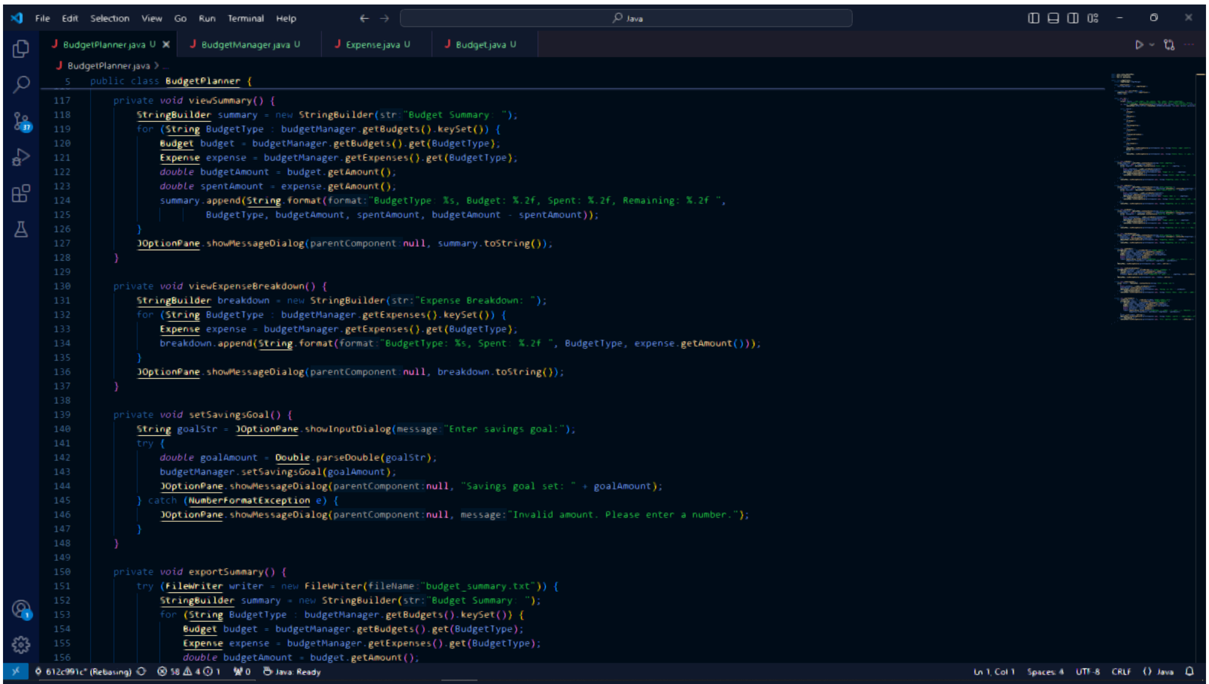
- **managing the user interface,**
- **handling events,**
- **interacting with data,**
- **controlling the application flow**



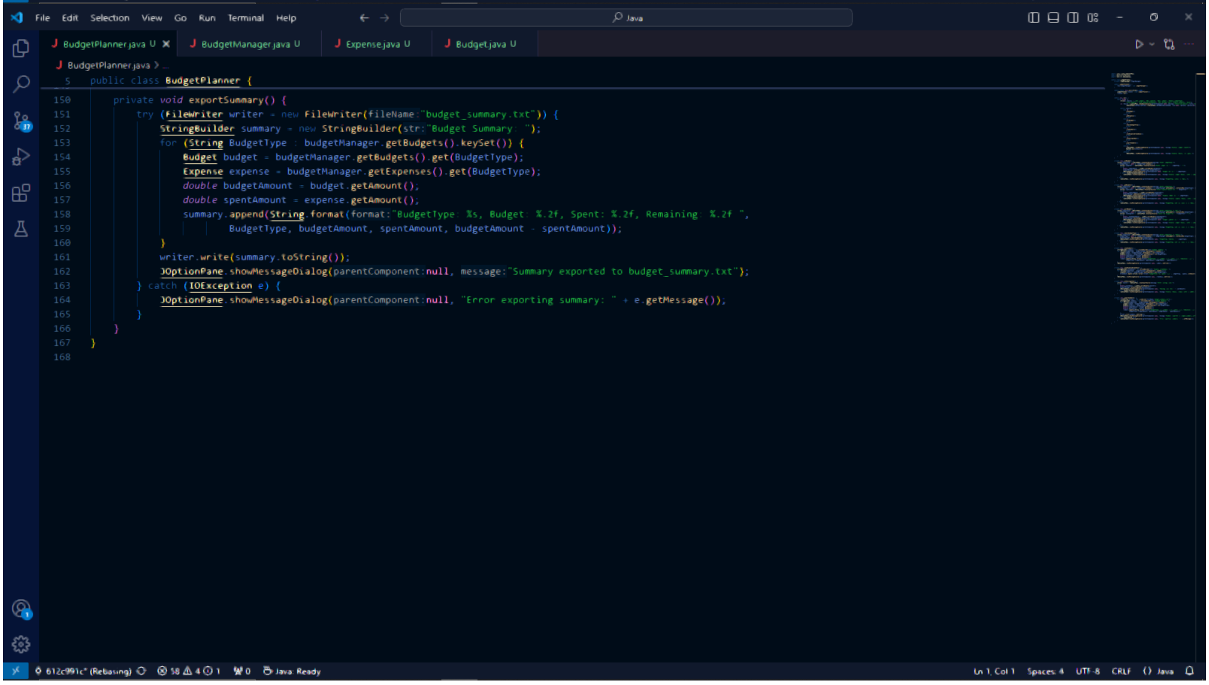
```java
public class BudgetPlanner {
    public void run() {
                case 5:
                    viewExpenseBreakdown();
                    break;
                case 6:
                    setSavingsGoal();
                    break;
                case 7:
                    exportSummary();
                    break;
                case 8:
                    JOptionPane.showMessageDialog(parentComponent:null, message:"Exiting Budget Planner");
                    System.exit(status:0);
                    break;
                default:
                    JOptionPane.showMessageDialog(parentComponent:null, message:"Invalid choice. Try again.");
            }
        }
    }

    private void setBudget() {
        String BudgetType = JOptionPane.showInputDialog(message:"Enter BudgetType:");
        if (BudgetType != null && !BudgetType.isEmpty()) {
            String budgetStr = JOptionPane.showInputDialog("Enter budget for " + BudgetType + ":");
            try {
                double budgetAmount = Double.parseDouble(budgetStr);
                budgetManager.setBudget(BudgetType, budgetAmount);
                JOptionPane.showMessageDialog(parentComponent:null, "Budget set for " + BudgetType);
            } catch (NumberFormatException e) {
                JOptionPane.showMessageDialog(parentComponent:null, message:"Invalid budget amount. Enter a number.");
            }
        } else {
            JOptionPane.showMessageDialog(parentComponent:null, message:"BudgetType cannot be empty.");
        }
    }

    private void addExpense() {
        String BudgetType = JOptionPane.showInputDialog(message:"Enter BudgetType:");
        if (BudgetType != null && !BudgetType.isEmpty() && budgetManager.getBudgets().containsKey(BudgetType)) {
            String expenseStr = JOptionPane.showInputDialog("Enter expense amount for " + BudgetType + ":");
```



```java
public class BudgetPlanner {
    private void addExpense() {
                double expenseAmount = Double.parseDouble(expenseStr);
                budgetManager.addExpense(BudgetType, expenseAmount);
                JOptionPane.showMessageDialog(parentComponent:null, "Expense added for " + BudgetType);
            } catch (NumberFormatException e) {
                JOptionPane.showMessageDialog(parentComponent:null, message:"Invalid expense amount. Enter a number.");
            }
        } else {
            JOptionPane.showMessageDialog(parentComponent:null, message:"BudgetType does not exist or is empty.");
        }
    }

    private void editBudget() {
        String BudgetType = JOptionPane.showInputDialog(message:"Enter BudgetType to edit:");
        if (BudgetType != null && !BudgetType.isEmpty() && budgetManager.getBudgets().containsKey(BudgetType)) {
            String newBudgetStr = JOptionPane.showInputDialog("Enter new budget for " + BudgetType + ":");
            try {
                double newBudgetAmount = Double.parseDouble(newBudgetStr);
                budgetManager.editBudget(BudgetType, newBudgetAmount);
                JOptionPane.showMessageDialog(parentComponent:null, "Budget updated for " + BudgetType);
            } catch (NumberFormatException e) {
                JOptionPane.showMessageDialog(parentComponent:null, message:"Invalid budget amount. Enter a number.");
            }
        } else {
            JOptionPane.showMessageDialog(parentComponent:null, message:"BudgetType does not exist or is empty.");
        }
    }

    private void removeBudgetType() {
        String BudgetType = JOptionPane.showInputDialog(message:"Enter BudgetType to remove:");
        if (BudgetType != null && !BudgetType.isEmpty() && budgetManager.getBudgets().containsKey(BudgetType)) {
            budgetManager.removeBudgetType(BudgetType);
            JOptionPane.showMessageDialog(parentComponent:null, "BudgetType removed: " + BudgetType);
        } else {
            JOptionPane.showMessageDialog(parentComponent:null, message:"BudgetType does not exist or is empty.");
        }
    }

    private void viewSummary() {
        StringBuilder summary = new StringBuilder(str:"Budget Summary: ");
```



```java
public class BudgetPlanner {
    private void viewSummary() {
        StringBuilder summary = new StringBuilder(str:"Budget Summary: ");
        for (String BudgetType : budgetManager.getBudgets().keySet()) {
            Budget budget = budgetManager.getBudgets().get(BudgetType);
            Expense expense = budgetManager.getExpenses().get(BudgetType);
            double budgetAmount = budget.getAmount();
            double spentAmount = expense.getAmount();
            summary.append(String.format(format:"BudgetType: %s, Budget: %.2f, Spent: %.2f, Remaining: %.2f ",
                    BudgetType, budgetAmount, spentAmount, budgetAmount - spentAmount));
        }
        JOptionPane.showMessageDialog(parentComponent:null, summary.toString());
    }

    private void viewExpenseBreakdown() {
        StringBuilder breakdown = new StringBuilder(str:"Expense Breakdown: ");
        for (String BudgetType : budgetManager.getExpenses().keySet()) {
            Expense expense = budgetManager.getExpenses().get(BudgetType);
            breakdown.append(String.format(format:"BudgetType: %s, Spent: %.2f ", BudgetType, expense.getAmount()));
        }
        JOptionPane.showMessageDialog(parentComponent:null, breakdown.toString());
    }

    private void setSavingsGoal() {
        String goalStr = JOptionPane.showInputDialog(message:"Enter savings goal:");
        try {
            double goalAmount = Double.parseDouble(goalStr);
            budgetManager.setSavingsGoal(goalAmount);
            JOptionPane.showMessageDialog(parentComponent:null, "Savings goal set: " + goalAmount);
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(parentComponent:null, message:"Invalid amount. Please enter a number.");
        }
    }

    private void exportSummary() {
        try (FileWriter writer = new FileWriter(fileName:"budget_summary.txt")) {
            StringBuilder summary = new StringBuilder(str:"Budget Summary: ");
            for (String BudgetType : budgetManager.getBudgets().keySet()) {
                Budget budget = budgetManager.getBudgets().get(BudgetType);
                Expense expense = budgetManager.getExpenses().get(BudgetType);
                double budgetAmount = budget.getAmount();
```

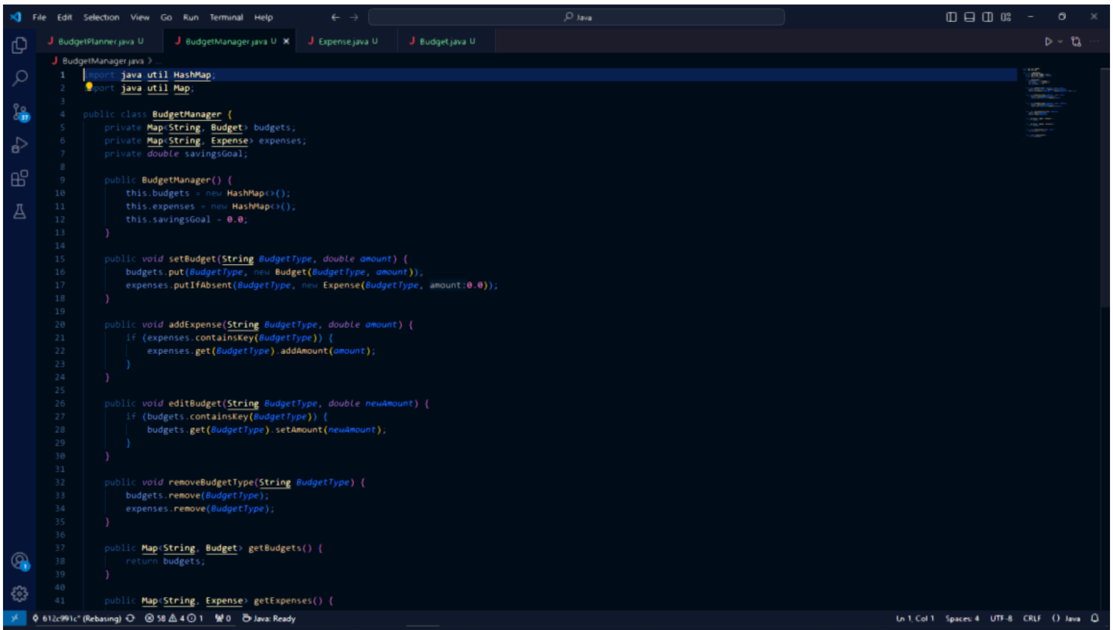


```java
public class BudgetPlanner {
    private void exportSummary() {
        try (FileWriter writer = new FileWriter(fileName:"budget_summary.txt")) {
            StringBuilder summary = new StringBuilder(str:"Budget Summary: ");
            for (String BudgetType : budgetManager.getBudgets().keySet()) {
                Budget budget = budgetManager.getBudgets().get(BudgetType);
                Expense expense = budgetManager.getExpenses().get(BudgetType);
                double budgetAmount = budget.getAmount();
                double spentAmount = expense.getAmount();
                summary.append(String.format(format:"BudgetType: %s, Budget: %.2f, Spent: %.2f, Remaining: %.2f ",
                        BudgetType, budgetAmount, spentAmount, budgetAmount - spentAmount));
            }
            writer.write(summary.toString());
            JOptionPane.showMessageDialog(parentComponent:null, message:"Summary exported to budget_summary.txt");
        } catch (IOException e) {
            JOptionPane.showMessageDialog(parentComponent:null, "Error exporting summary: " + e.getMessage());
        }
    }
}
```

- Team Member Two Cntribution (Daniel Sase Jr   BHU/22/04/05/0061)

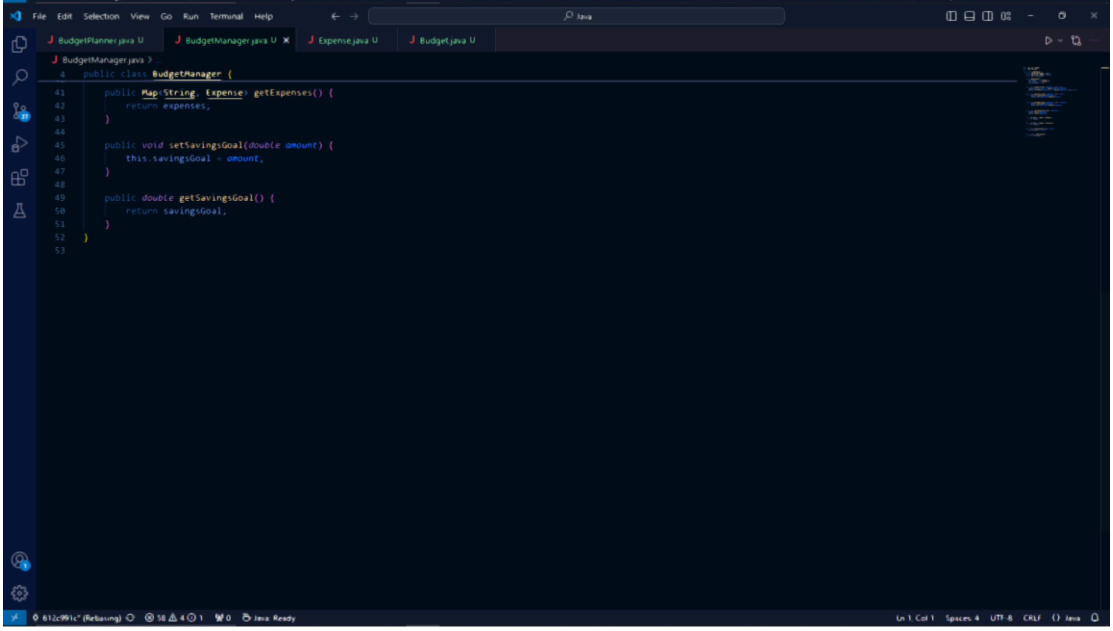
The BudgetManager class in the Budget Planner application is responsible for managing budgets and expenses. It serves as the "Model" in the Model-View-Controller (MVC) pattern, handling data manipulation and business logic. responsibilities include:

- **data storage,**
- **data manipulation,**
- **business logic implementation.**
-



```java
import java.util.HashMap;
import java.util.Map;

public class BudgetManager {
    private Map<String, Budget> budgets;
    private Map<String, Expense> expenses;
    private double savingsGoal;

    public BudgetManager() {
        this.budgets = new HashMap<>();
        this.expenses = new HashMap<>();
        this.savingsGoal = 0.0;
    }

    public void setBudget(String BudgetType, double amount) {
        budgets.put(BudgetType, new Budget(BudgetType, amount));
        expenses.putIfAbsent(BudgetType, new Expense(BudgetType, amount:0.0));
    }

    public void addExpense(String BudgetType, double amount) {
        if (expenses.containsKey(BudgetType)) {
            expenses.get(BudgetType).addAmount(amount);
        }
    }

    public void editBudget(String BudgetType, double newAmount) {
        if (budgets.containsKey(BudgetType)) {
            budgets.get(BudgetType).setAmount(newAmount);
        }
    }

    public void removeBudgetType(String BudgetType) {
        budgets.remove(BudgetType);
        expenses.remove(BudgetType);
    }

    public Map<String, Budget> getBudgets() {
        return budgets;
    }

    public Map<String, Expense> getExpenses() {
```

-



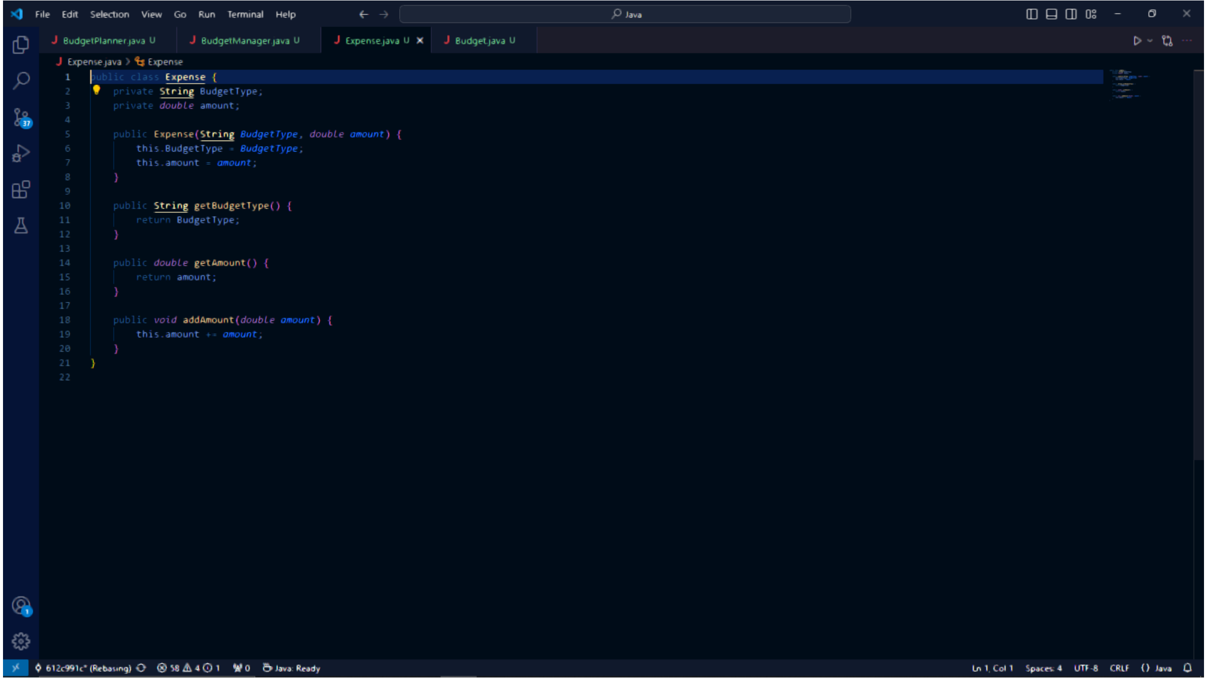
```java
public class BudgetManager {
    public Map<String, Expense> getExpenses() {
        return expenses;
    }

    public void setSavingsGoal(double amount) {
        this.savingsGoal = amount;
    }

    public double getSavingsGoal() {
        return savingsGoal;
    }
}
```

Team Member Three Contribution (**Samson Praise Chidera BHU/22/04/09/0028**)

The Budget class in the Budget Planner application serves as a data model representing the budget allocated for a specific category. It encapsulates all the details associated with a budget, including the category name and the amount allocated. Its responsibilities are

- **data encapsulation,**
- **data manipulation, and**
- **serving as a representation of a specific financial category's budget.**

It provides methods to access and modify the budgeted amount, ensuring structured storage and access of information. This allows for operations such as setting and updating the budget, making it easy to manage and manipulate multiple categories in the application.

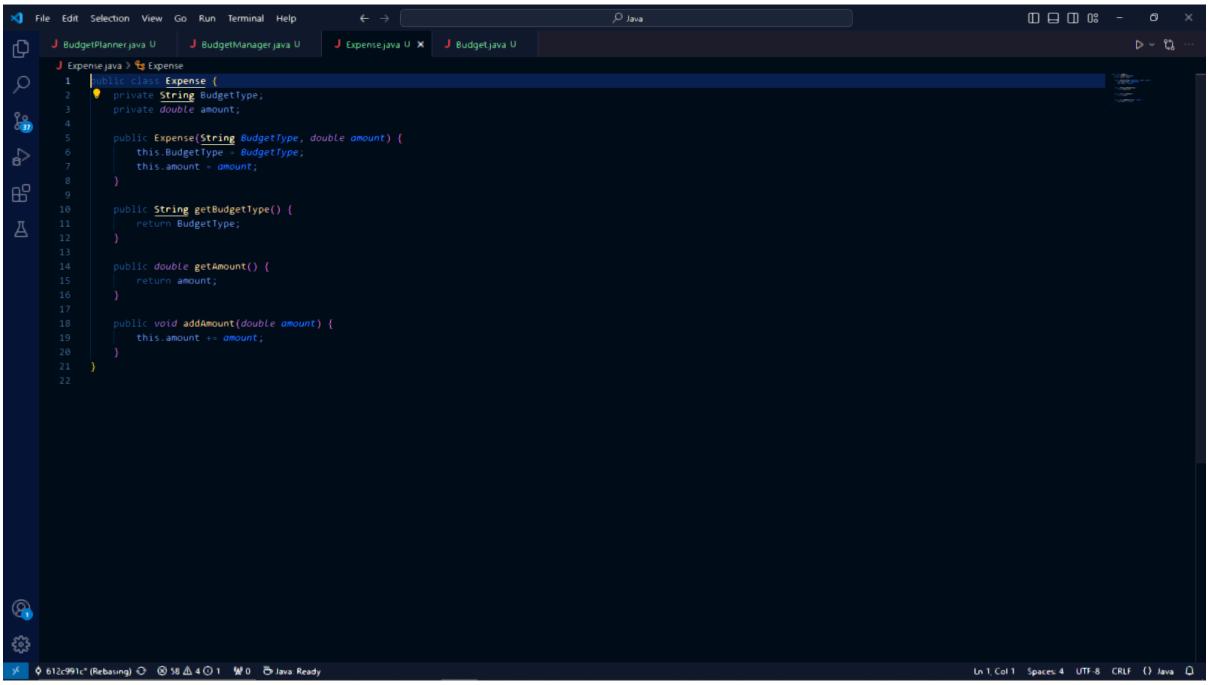


```java
public class Expense {
    private String BudgetType;
    private double amount;

    public Expense(String BudgetType, double amount) {
        this.BudgetType = BudgetType;
        this.amount = amount;
    }

    public String getBudgetType() {
        return BudgetType;
    }

    public double getAmount() {
        return amount;
    }

    public void addAmount(double amount) {
        this.amount += amount;
    }
}
```

Team Member Four Contribution (Shettima IJASINI DANIEL **BHU/22/04/09/0004**
Kashim samaila **BHU/22/04/05/00/68**)

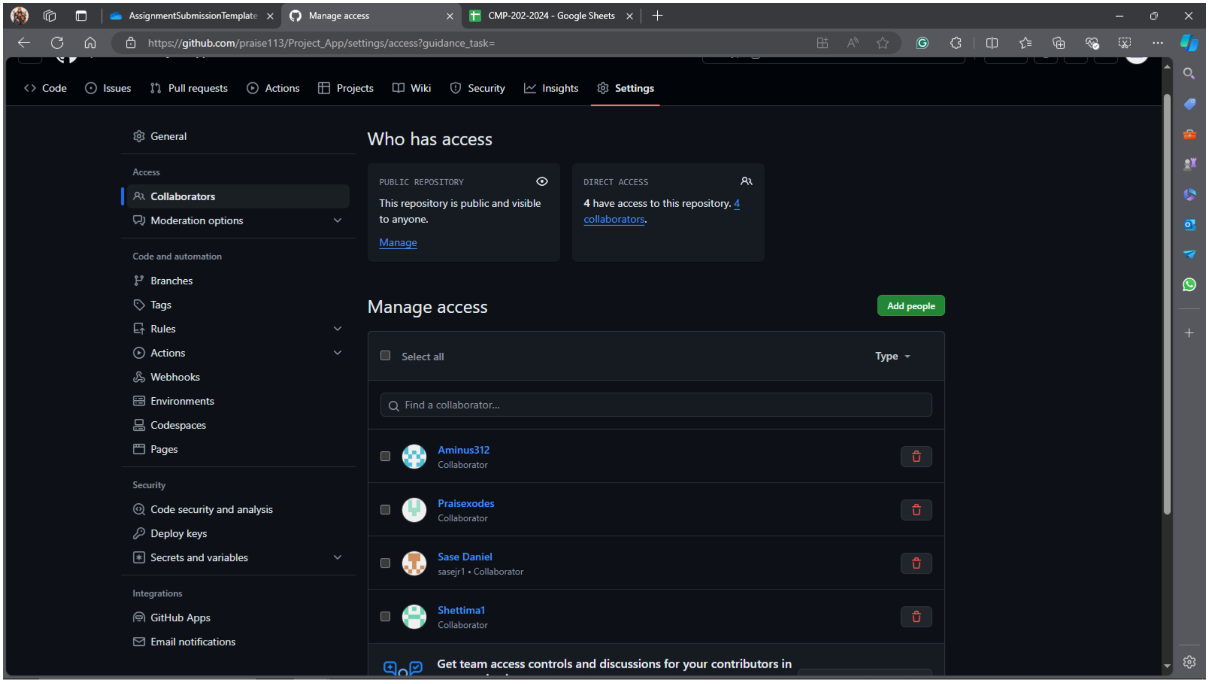
The Expense class within the Budget Planner application serves as a data model for managing and encapsulating spending details related to a specific category. It stores information about the category name and the total amount spent, providing ease in tracking and managing expenses across different categories. The class is responsible for:

- **data encapsulation**
- **Manipulation**
- **representation of expenses for a specific category.**



```java
public class Expense {
    private String BudgetType;
    private double amount;

    public Expense(String BudgetType, double amount) {
        this.BudgetType = BudgetType;
        this.amount = amount;
    }

    public String getBudgetType() {
        return BudgetType;
    }

    public double getAmount() {
        return amount;
    }

    public void addAmount(double amount) {
        this.amount += amount;
    }
}
```

Screenshot of Github Contribution Page for the project



Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

Manage

DIRECT ACCESS

4 have access to this repository. 4 collaborators.

Manage access

□ Aminus312
Collaborator

□ Praisexodes
Collaborator

□ Sase Daniel
sasejr1 • Collaborator

□ Shettima1
Collaborator

Get team access controls and discussions for your contributors in