Implementing a Loan Calculator in Java with a Graphical User Interface

Ron D. Cox

Johns Hopkins University

Java 605.201.83

Assignment 13 Mini Project #3

Instructor: Dr. Sidney I. Rubey

Date: 4/29/2024

Abstract

This paper presents a detailed description of a simple loan calculator application developed using Java, demonstrating key principles of object-oriented programming and graphical user interface design using the Swing framework. The program allows users to enter loan-related information and computes monthly and total payments based on standard loan amortization formulas. The design and implementation of the application are discussed, followed by a UML diagram illustrating the structure of the application.

Introduction

The loan calculator program is designed to provide users with a tool to calculate monthly and total payments for a given loan amount, interest rate, and loan duration. This functionality is especially useful in financial planning and loan management. The Java programming language, combined with its Swing library, is utilized to create the application's graphical user interface, offering a platform-independent solution.

Program Design

The application is structured around several key classes, each responsible for a specific aspect of the program's functionality:

- **LoanCalculator**: The main class that initializes the application's GUI components and handles event-driven interactions.
- **JTextField**: Used for input fields where users can enter the loan amount, interest rate, and duration.
- **JButton**: A button that users click to calculate the payments based on the input values.
- **JLabel**: Labels that provide guidance on what information the user should enter in each field.

Methods and Application Flow

The application flow is controlled through event-driven programming. Upon launching, the user is presented with a window containing labeled text fields for entering the interest rate, number of years, and loan amount. A 'Calculate' button is provided to trigger the computation. Here's how the application handles operations:

- **Input Reading**: The user inputs are read from the text fields when the 'Calculate' button is clicked.
- **Payment Calculation**: The monthly payment is calculated using the formula $P=i\times A1-(1+i)-nP=1-(1+i)-ni\times A$, where ii is the monthly interest rate, AA is the loan amount, and nn is the total number of payments.
- **Output Display**: The calculated monthly and total payments are then displayed back to the user in the GUI.

UML Diagram

The UML diagram below conceptualizes the relationships between the classes:

```
+-------------------+
|   LoanCalculator  |
+-------------------+
| - annualRateField |
| - numberOfYearsField |
| - loanAmountField |
| - monthlyPaymentField |
| - totalPaymentField |
| - calculateButton |
+-------------------+
| + calculateLoanPayment() |
+-------------------+
```

Conclusion

The loan calculator developed in Java with Swing demonstrates a practical application of object-oriented programming concepts to create a user-friendly graphical interface for calculating loans. The application effectively separates the GUI from the computation logic, enhancing maintainability and scalability. Future enhancements could include more complex financial calculations and improved error handling.

References

1. Schildt, Herbert (2017). *Java: The Complete Reference 10/E, 10th Edition*: McGraw-Hill Education.
2. Eck, D. J. (2019). *Introduction to Programming Using Java*. Version 8.1. Online Textbook.