

REPORT:

This report was prepared by the JULIA programming group.

Analyzing Student Loan Management CODE WITH JAVA PARSER IN VSCODE (MAVEN)

INTRODUCTION

This report details the analysis of Java code, representing a StudentLoan class, using the Java Parser library within the VSCode IDE. The analysis leverages the Maven build tool for project management.

Analysis Approach

The Java Parser library is a tool for parsing Java source code into an Abstract Syntax Tree (AST). This AST represents the structure of the code, allowing programmatic access to its elements (classes, methods, variables, etc.). This analysis likely involved:

Dependency inclusion: The project has included the javaparser-core dependency in its pom.xml file for Maven to manage the Java Parser library.

Code Parsing: The Java code (PatientVitals.java), is parsed using the Java Parser library. This generates an AST representing the code's structure.

AST Traversal: Code is written to traverse the AST, extracting relevant information like class definitions, methods, and fields.

Tokenization: The code tokenizes the source code, breaking it down into smaller elements like keywords, identifiers, operators, and literals. This allows counting the total number of tokens and categorizing them by type.

Output Generation: The analysis results were then presented and displayed within the VSCode console and written to a separate report file.

Expected Analysis Results

The analysis generated the following outputs:

1. Abstract Syntax Tree (AST):

The AST was generated and the class definitions were made and their relationships within the code.

2. Total Number of Tokens:

This represents the total count of all individual elements (keywords, identifiers, etc.) identified within the source code.

3. Token Types and Counts:

The analysis categorized and counted the different types of tokens found in the code.
This includes:

- Keywords (e.g., public, class, if)
- Identifiers (e.g., variable names, method names)
- Operators (e.g., +, -, *)
- Literals (e.g., string literals, numeric literals)
- Other token types (e.g., separators, comments)

Benefits of this Analysis

We understood code structure through the AST visualization and Tokenization helped us analyze code complexity and identify potential areas for improvement or optimization.

Limitations

The report cannot access the actual content of the code beyond the analysis results.

Conclusion

This analysis demonstrates how the Java Parser library, integrated with VSCode and Maven, can be used to explore and understand Java code structure. By examining the AST and tokenization results, developers gain valuable insights into the code's complexity and organization.

Here is the output;

```
PS C:\Users\USER\Documents\demo> ^C
PS C:\Users\USER\Documents\demo>
PS C:\Users\USER\Documents\demo> c:: cd 'c:\Users\USER\Documents\demo'; & 'C:\Program
Files\Java\jdk-21\bin\java.exe'
 '@C:\Users\USER\AppData\Local\Temp\cp_56txxfoc7g3b59h11e32ai01v.argfile'
 'com.example.Main1'
```

Token Counts:

EOL: 146

OPERATOR: 45

LITERAL: 45

SEPARATOR: 386

KEYWORD: 81

IDENTIFIER: 236

WHITESPACE_NO_EOL: 1294

COMMENT: 13

Total Tokens: 2246

The AST is below:

CompilationUnit: package com.example;

import java.util.ArrayList...

└─ PackageDeclaration: package com.example;

...

| └─ Name: com.example...

| | └─ Name: com...

└─ ImportDeclaration: import java.util.ArrayList;

...

| └─ Name: java.util.ArrayList...

| | └─ Name: java.util...

| | | └─ Name: java...

└─ ImportDeclaration: import java.util.InputMismatchException;

...

| └─ Name: java.util.InputMismatchException...

| | └─ Name: java.util...

| | | └─ Name: java...

└─ ImportDeclaration: import java.util.Scanner;

...

| └─ Name: java.util.Scanner...

| | └─ Name: java.util...

| | | └─ Name: java...

└─ ClassOrInterfaceDeclaration: // Class to represent a student loan

class Studen...

| └─ SimpleName: StudentLoan...

| └─ FieldDeclaration: private String studentName;...

| | └─ Modifier: private ...

| | └─ VariableDeclarator: studentName...

| | | └─ ClassOrInterfaceType: String...

| | | | └─ SimpleName: String...

| | | └─ SimpleName: studentName...

| └─ FieldDeclaration: private double loanAmount;...

| | └─ Modifier: private ...

| | └─ VariableDeclarator: loanAmount...

| | | └─ PrimitiveType: double...

| | | └─ SimpleName: loanAmount...

| └─ FieldDeclaration: private double interestRate;...

| | └─ Modifier: private ...

| | └─ VariableDeclarator: interestRate...

| | | └─ PrimitiveType: double...

| | | └─ SimpleName: interestRate...

| └─ FieldDeclaration: private int weeksToRepay;...

| | └─ Modifier: private ...

| | └─ VariableDeclarator: weeksToRepay...

| | | └─ PrimitiveType: int...

```

| | | └─ SimpleName: weeksToRepay...
| | └─ ConstructorDeclaration: public StudentLoan(String studentName, double loan...
| |   | └─ Modifier: public ...
| |   | └─ SimpleName: StudentLoan...
| |   | └─ Parameter: String studentName...
| |   |   | └─ ClassOrInterfaceType: String...
| |   |   |   | └─ SimpleName: String...
| |   |   |   | └─ SimpleName: studentName...
| |   | └─ Parameter: double loanAmount...
| |   |   | └─ PrimitiveType: double...
| |   |   | └─ SimpleName: loanAmount...
| |   | └─ Parameter: double interestRate...
| |   |   | └─ PrimitiveType: double...
| |   |   | └─ SimpleName: interestRate...
| |   | └─ Parameter: int weeksToRepay...
| |   |   | └─ PrimitiveType: int...
| |   |   | └─ SimpleName: weeksToRepay...
| |   └─ BlockStmt: {
| |     this.studentName = studentName;
| |     this.l...
| |   | └─ ExpressionStmt: this.studentName = studentName;...
| |   |   | └─ AssignExpr: this.studentName = studentName...
| |   |   |   | └─ FieldAccessExpr: this.studentName...
| |   |   |   |   | └─ ThisExpr: this...
| |   |   |   |   | └─ SimpleName: studentName...
| |   |   |   |   | └─ NameExpr: studentName...
| |   |   |   |   | └─ SimpleName: studentName...
| |   |   | └─ ExpressionStmt: this.loanAmount = loanAmount;...
| |   |   |   | └─ AssignExpr: this.loanAmount = loanAmount...
| |   |   |   |   | └─ FieldAccessExpr: this.loanAmount...
| |   |   |   |   |   | └─ ThisExpr: this...
| |   |   |   |   |   | └─ SimpleName: loanAmount...
| |   |   |   |   |   | └─ NameExpr: loanAmount...
| |   |   |   |   |   | └─ SimpleName: loanAmount...
| |   |   | └─ ExpressionStmt: this.interestRate = interestRate;...
| |   |   |   | └─ AssignExpr: this.interestRate = interestRate...
| |   |   |   |   | └─ FieldAccessExpr: this.interestRate...
| |   |   |   |   |   | └─ ThisExpr: this...
| |   |   |   |   |   | └─ SimpleName: interestRate...
| |   |   |   |   |   | └─ NameExpr: interestRate...
| |   |   |   |   |   | └─ SimpleName: interestRate...
| |   |   | └─ ExpressionStmt: this.weeksToRepay = weeksToRepay;...
| |   |   |   | └─ AssignExpr: this.weeksToRepay = weeksToRepay...
| |   |   |   |   | └─ FieldAccessExpr: this.weeksToRepay...
| |   |   |   |   |   | └─ ThisExpr: this...

```

```

| | | | | | └─ SimpleName: weeksToRepay...
| | | | | └─ NameExpr: weeksToRepay...
| | | | | └─ SimpleName: weeksToRepay...
| | └─ MethodDeclaration: public double calculateTotalPayment() {
doubl...
| | | └─ Modifier: public ...
| | | └─ SimpleName: calculateTotalPayment...
| | | └─ PrimitiveType: double...
| | | └─ BlockStmt: {
double totalInterest = 0;
double curre...
| | | | └─ ExpressionStmt: double totalInterest = 0;...
| | | | | └─ VariableDeclarationExpr: double totalInterest = 0...
| | | | | | └─ VariableDeclarator: totalInterest = 0...
| | | | | | | └─ PrimitiveType: double...
| | | | | | | └─ SimpleName: totalInterest...
| | | | | | | └─ IntegerLiteralExpr: 0...
| | | | └─ ExpressionStmt: double currentInterestRate = interestRate;...
| | | | | └─ VariableDeclarationExpr: double currentInterestRate = interestRate...
| | | | | | └─ VariableDeclarator: currentInterestRate = interestRate...
| | | | | | | └─ PrimitiveType: double...
| | | | | | | └─ SimpleName: currentInterestRate...
| | | | | | | └─ NameExpr: interestRate...
| | | | | | | └─ SimpleName: interestRate...
| | | | └─ ForStmt: for (int i = 0; i < weeksToRepay; i++) {
tota...
| | | | | └─ VariableDeclarationExpr: int i = 0...
| | | | | | └─ VariableDeclarator: i = 0...
| | | | | | | └─ PrimitiveType: int...
| | | | | | | └─ SimpleName: i...
| | | | | | | └─ IntegerLiteralExpr: 0...
| | | | | └─ BinaryExpr: i < weeksToRepay...
| | | | | | └─ NameExpr: i...
| | | | | | | └─ SimpleName: i...
| | | | | | | └─ NameExpr: weeksToRepay...
| | | | | | | └─ SimpleName: weeksToRepay...
| | | | | └─ UnaryExpr: i++...
| | | | | | └─ NameExpr: i...
| | | | | | | └─ SimpleName: i...
| | | | | └─ BlockStmt: {
totalInterest += loanAmount * (currentInter...
| | | | | | └─ ExpressionStmt: totalInterest += loanAmount * (currentInterestRate...
| | | | | | | └─ AssignExpr: totalInterest += loanAmount * (currentInterestRate...
| | | | | | | | └─ NameExpr: totalInterest...
| | | | | | | | └─ SimpleName: totalInterest...

```

```
| | | | | | | | └─ BinaryExpr: loanAmount * (currentInterestRate / 100)...
| | | | | | | | └─ NameExpr: loanAmount...
| | | | | | | | └─ SimpleName: loanAmount...
| | | | | | | | └─ EnclosedExpr: (currentInterestRate / 100)...
| | | | | | | | └─ BinaryExpr: currentInterestRate / 100...
| | | | | | | | └─ NameExpr: currentInterestRate...
| | | | | | | | └─ SimpleName: currentInterestRate...
| | | | | | | | └─ IntegerLiteralExpr: 100...
| | | | | └─ IfStmt: if (i < 3) {
// Interest increases by 5% each...
| | | | | | └─ BinaryExpr: i < 3...
| | | | | | └─ NameExpr: i...
| | | | | | └─ SimpleName: i...
| | | | | | └─ IntegerLiteralExpr: 3...
| | | | | └─ BlockStmt: {
// Interest increases by 5% each week for t...
| | | | | | └─ ExpressionStmt: // Interest increases by 5% each week for the firs...
| | | | | | └─ AssignExpr: currentInterestRate += 5...
| | | | | | └─ NameExpr: currentInterestRate...
| | | | | | └─ SimpleName: currentInterestRate...
| | | | | | └─ IntegerLiteralExpr: 5...
| | | | | └─ BlockStmt: {
// Reset interest rate to 10% after the 4th...
| | | | | | └─ ExpressionStmt: // Reset interest rate to 10% after the 4th week
...
| | | | | | | | └─ AssignExpr: currentInterestRate = 10...
| | | | | | | | └─ NameExpr: currentInterestRate...
| | | | | | | | └─ SimpleName: currentInterestRate...
| | | | | | | | └─ IntegerLiteralExpr: 10...
| | | | └─ ReturnStmt: return loanAmount + totalInterest;...
| | | | └─ BinaryExpr: loanAmount + totalInterest...
| | | | └─ NameExpr: loanAmount...
| | | | └─ SimpleName: loanAmount...
| | | | └─ NameExpr: totalInterest...
| | | | └─ SimpleName: totalInterest...
| └─ MethodDeclaration: // Getters and setters
public String getStudentNa...
| | └─ Modifier: public ...
| | └─ SimpleName: getStudentName...
| | └─ ClassOrInterfaceType: String...
| | └─ SimpleName: String...
| | └─ BlockStmt: {
return studentName;
}...
| | | └─ ReturnStmt: return studentName;...
```

```

| | | | | └─ NameExpr: studentName...
| | | | | └─ SimpleName: studentName...
| └─ MethodDeclaration: public double getLoanAmount() {
  return loanAm...
| | └─ Modifier: public ...
| | └─ SimpleName: getLoanAmount...
| | └─ PrimitiveType: double...
| | └─ BlockStmt: {
  return loanAmount;
}...
| | | └─ ReturnStmt: return loanAmount;...
| | | | └─ NameExpr: loanAmount...
| | | | └─ SimpleName: loanAmount...
| └─ MethodDeclaration: public double getInterestRate() {
  return inte...
| | └─ Modifier: public ...
| | └─ SimpleName: getInterestRate...
| | └─ PrimitiveType: double...
| | └─ BlockStmt: {
  return interestRate;
}...
| | | └─ ReturnStmt: return interestRate;...
| | | | └─ NameExpr: interestRate...
| | | | └─ SimpleName: interestRate...
| └─ MethodDeclaration: public int getWeeksToRepay() {
  return weeksTo...
| | └─ Modifier: public ...
| | └─ SimpleName: getWeeksToRepay...
| | └─ PrimitiveType: int...
| | └─ BlockStmt: {
  return weeksToRepay;
}...
| | | └─ ReturnStmt: return weeksToRepay;...
| | | | └─ NameExpr: weeksToRepay...
| | | | └─ SimpleName: weeksToRepay...
└─ ClassOrInterfaceDeclaration: // Class to manage student loans
class LoanManage...
| └─ SimpleName: LoanManager...
| └─ FieldDeclaration: private ArrayList<StudentLoan> loans = new ArrayLi...
| | └─ Modifier: private ...
| | └─ VariableDeclarator: loans = new ArrayList<>()...
| | | └─ ClassOrInterfaceType: ArrayList<StudentLoan>...
| | | | └─ SimpleName: ArrayList...
| | | | └─ ClassOrInterfaceType: StudentLoan...
| | | | └─ SimpleName: StudentLoan...

```

```

| | | | └─ SimpleName: loans...
| | | | └─ ObjectCreationExpr: new ArrayList<>()...
| | | | | └─ ClassOrInterfaceType: ArrayList<>...
| | | | | | └─ SimpleName: ArrayList...
| | | └─ MethodDeclaration: public void addLoan(StudentLoan loan) {
| |   loans...
| |   | └─ Modifier: public ...
| |   | └─ SimpleName: addLoan...
| |   | └─ Parameter: StudentLoan loan...
| |   | | └─ ClassOrInterfaceType: StudentLoan...
| |   | | | └─ SimpleName: StudentLoan...
| |   | | | └─ SimpleName: loan...
| |   | └─ VoidType: void...
| |   └─ BlockStmt: {
| |     loans.add(loan);
| |   }...
| | | | └─ ExpressionStmt: loans.add(loan);...
| | | | | └─ MethodCallExpr: loans.add(loan)...
| | | | | | └─ NameExpr: loans...
| | | | | | | └─ SimpleName: loans...
| | | | | | | └─ SimpleName: add...
| | | | | | | └─ NameExpr: loan...
| | | | | | | └─ SimpleName: loan...
| | | └─ MethodDeclaration: public void displayLoans() {
| |   for (StudentLoan...
| |   | └─ Modifier: public ...
| |   | └─ SimpleName: displayLoans...
| |   | └─ VoidType: void...
| |   | └─ BlockStmt: {
| |   |   for (StudentLoan loan : loans) {
| |   |     S...
| |   |     | └─ ForEachStmt: for (StudentLoan loan : loans) {
| |   |     |   System.out.p...
| |   |     | | └─ VariableDeclarationExpr: StudentLoan loan...
| |   |     | | | └─ VariableDeclarator: loan...
| |   |     | | | | └─ ClassOrInterfaceType: StudentLoan...
| |   |     | | | | | └─ SimpleName: StudentLoan...
| |   |     | | | | | └─ SimpleName: loan...
| |   |     | | | | └─ NameExpr: loans...
| |   |     | | | | └─ SimpleName: loans...
| |   |     | | | └─ BlockStmt: {
| |   |     | |   System.out.println("Student Name: " + loan....
| |   |     | | | └─ ExpressionStmt: System.out.println("Student Name: " + loan.getStud...
| |   |     | | | | └─ MethodCallExpr: System.out.println("Student Name: " + loan.getStud...
| |   |     | | | | | └─ FieldAccessExpr: System.out...

```



```
| | | | | | | | | | NameExpr: System...
| | | | | | | | | | SimpleName: System...
| | | | | | | | | | SimpleName: out...
| | | | | | | | | | SimpleName: println...
| | | | | | | | | | BinaryExpr: "Student Name: " + loan.getStudentName()...
| | | | | | | | | | StringLiteralExpr: "Student Name: "...
| | | | | | | | | | MethodCallExpr: loan.getStudentName()...
| | | | | | | | | | NameExpr: loan...
| | | | | | | | | | SimpleName: loan...
| | | | | | | | | | SimpleName: getStudentName...
| | | | | | | | ExpressionStmt: // Formatting loan amount
System.out.println("Loa...
| | | | | | | | | | MethodCallExpr: System.out.println("Loan Amount: K" + String.forma...
| | | | | | | | | | FieldAccessExpr: System.out...
| | | | | | | | | | NameExpr: System...
| | | | | | | | | | SimpleName: System...
| | | | | | | | | | SimpleName: out...
| | | | | | | | | | SimpleName: println...
| | | | | | | | | | BinaryExpr: "Loan Amount: K" + String.format("%.2f", loan.getL...
| | | | | | | | | | StringLiteralExpr: "Loan Amount: K"...
| | | | | | | | | | MethodCallExpr: String.format("%.2f", loan.getLoanAmount())...
| | | | | | | | | | NameExpr: String...
| | | | | | | | | | SimpleName: String...
| | | | | | | | | | SimpleName: format...
| | | | | | | | | | StringLiteralExpr: "%.2f"...
| | | | | | | | | | MethodCallExpr: loan.getLoanAmount()...
| | | | | | | | | | NameExpr: loan...
| | | | | | | | | | SimpleName: loan...
| | | | | | | | | | SimpleName: getLoanAmount...
| | | | | | | | ExpressionStmt: System.out.println("Interest Rate: " + loan.getInt...
| | | | | | | | | | MethodCallExpr: System.out.println("Interest Rate: " + loan.getInt...
| | | | | | | | | | FieldAccessExpr: System.out...
| | | | | | | | | | NameExpr: System...
| | | | | | | | | | SimpleName: System...
| | | | | | | | | | SimpleName: out...
| | | | | | | | | | SimpleName: println...
| | | | | | | | | | BinaryExpr: "Interest Rate: " + loan.getInterestRate() + "% pe...
| | | | | | | | | | BinaryExpr: "Interest Rate: " + loan.getInterestRate()...
| | | | | | | | | | StringLiteralExpr: "Interest Rate: "...
| | | | | | | | | | MethodCallExpr: loan.getInterestRate()...
| | | | | | | | | | NameExpr: loan...
| | | | | | | | | | SimpleName: loan...
| | | | | | | | | | SimpleName: getInterestRate...
| | | | | | | | | | StringLiteralExpr: "% per week"...
| | | | | | | | ExpressionStmt: System.out.println("Weeks to Repay: " + loan.getWe...
```

```
| | | | | | | └─ MethodCallExpr: System.out.println("Weeks to Repay: " + loan.getWe...
| | | | | | | └─ FieldAccessExpr: System.out...
| | | | | | | └─ NameExpr: System...
| | | | | | | └─ SimpleName: System...
| | | | | | | └─ SimpleName: out...
| | | | | | | └─ SimpleName: println...
| | | | | | | └─ BinaryExpr: "Weeks to Repay: " + loan.getWeeksToRepay()...
| | | | | | | └─ StringLiteralExpr: "Weeks to Repay: "...
| | | | | | | └─ MethodCallExpr: loan.getWeeksToRepay()...
| | | | | | | └─ NameExpr: loan...
| | | | | | | └─ SimpleName: loan...
| | | | | | | └─ SimpleName: getWeeksToRepay...
| | | | | └─ ExpressionStmt: // Formatting total payment
System.out.println("T...
| | | | | | | └─ MethodCallExpr: System.out.println("Total Payment: K" + String.for...
| | | | | | | └─ FieldAccessExpr: System.out...
| | | | | | | └─ NameExpr: System...
| | | | | | | └─ SimpleName: System...
| | | | | | | └─ SimpleName: out...
| | | | | | | └─ SimpleName: println...
| | | | | | | └─ BinaryExpr: "Total Payment: K" + String.format("%.2f", loan.ca...
| | | | | | | └─ StringLiteralExpr: "Total Payment: K"...
| | | | | | | └─ MethodCallExpr: String.format("%.2f",
loan.calculateTotalPayment()...
| | | | | | | └─ NameExpr: String...
| | | | | | | └─ SimpleName: String...
| | | | | | | └─ SimpleName: format...
| | | | | | | └─ StringLiteralExpr: "%.2f"...
| | | | | | | └─ MethodCallExpr: loan.calculateTotalPayment()...
| | | | | | | └─ NameExpr: loan...
| | | | | | | └─ SimpleName: loan...
| | | | | | | └─ SimpleName: calculateTotalPayment...
| | | | | └─ ExpressionStmt: System.out.println();...
| | | | | └─ MethodCallExpr: System.out.println()...
| | | | | └─ FieldAccessExpr: System.out...
| | | | | └─ NameExpr: System...
| | | | | └─ SimpleName: System...
| | | | | └─ SimpleName: out...
| | | | | └─ SimpleName: println...
└─ ClassOrInterfaceDeclaration: public class Main {
/**
 * @param args...
| └─ Modifier: public ...
| └─ SimpleName: Main...
```

```
public static void main(...
| | |─ Modifier: public ...
| | |─ Modifier: static ...
| | |─ SimpleName: main...
| | |─ Parameter: String[] args...
| | | |─ ArrayType: String[]...
| | | | |─ ClassOrInterfaceType: String...
| | | | | |─ SimpleName: String...
| | | |─ SimpleName: args...
| | |─ VoidType: void...
| | |─ BlockStmt: {
Scanner scanner = new Scanner(System.in);
```

```

| | | | └─ ExpressionStmt: Scanner scanner = new Scanner(System.in);...
| | | |   └─ VariableDeclarationExpr: Scanner scanner = new Scanner(System.in)...
| | | |     └─ VariableDeclarator: scanner = new Scanner(System.in)...
| | | |       └─ ClassOrInterfaceType: Scanner...
| | | |         └─ SimpleName: Scanner...
| | | |           └─ SimpleName: scanner...
| | | |             └─ ObjectCreationExpr: new Scanner(System.in)...
| | | |               └─ ClassOrInterfaceType: Scanner...
| | | |                 └─ SimpleName: Scanner...
| | | |                   └─ FieldAccessExpr: System.in...
| | | |                     └─ NameExpr: System...
| | | |                       └─ SimpleName: System...
| | | |                         └─ SimpleName: in...
| | | └─ ExpressionStmt: LoanManager loanManager = new LoanManager();...
| | |   └─ VariableDeclarationExpr: LoanManager loanManager = new LoanManager()...
| | |     └─ VariableDeclarator: loanManager = new LoanManager()...
| | |       └─ ClassOrInterfaceType: LoanManager...
| | |         └─ SimpleName: LoanManager...
| | |           └─ SimpleName: loanManager...
| | |             └─ ObjectCreationExpr: new LoanManager()...
| | |               └─ ClassOrInterfaceType: LoanManager...
| | |                 └─ SimpleName: LoanManager...
| | └─ ExpressionStmt: System.out.println("Enter your name");...
| |   └─ MethodCallExpr: System.out.println("Enter your name")...
| |     └─ FieldAccessExpr: System.out...
| |       └─ NameExpr: System...
| |         └─ SimpleName: System...
| |           └─ SimpleName: out...
| |             └─ SimpleName: println...

```

```
| | | | | └─ StringLiteralExpr: "Enter your name:"...  
| | | | └─ ExpressionStmt: String name = scanner.nextLine();...  
| | | |   │├─ VariableDeclarationExpr: String name = scanner.nextLine()...  
| | | |     │├─ VariableDeclarator: name = scanner.nextLine()...  
| | | |       │├─ ClassOrInterfaceType: String...  
| | | |         │├─ SimpleName: String...  
| | | |           ├── SimpleName: name...  
| | | |             └─ MethodCallExpr: scanner.nextLine()...  
| | | |               ├── NameExpr: scanner...  
| | | |                 └─ SimpleName: scanner...  
| | | |                   └─ SimpleName: nextLine...  
| | | └─ ExpressionStmt: System.out.println("Welcome " + name + " to Julia ..."  
| | |   │├─ MethodCallExpr: System.out.println("Welcome " + name + " to Julia ..."  
| | |     │├─ FieldAccessExpr: System.out...  
| | |       │├─ NameExpr: System...  
| | |         │├─ SimpleName: System...  
| | |           ├── SimpleName: out...  
| | |             └─ SimpleName: println...  
| | |       └─ BinaryExpr: "Welcome " + name + " to Julia Student Loan Manage..."  
| | |         │├─ BinaryExpr: "Welcome " + name...  
| | |           │├─ StringLiteralExpr: "Welcome "...  
| | |             └─ NameExpr: name...  
| | |               └─ SimpleName: name...  
| | |         └─ StringLiteralExpr: " to Julia Student Loan Manager. Do you want to pr..."  
| | └─ ExpressionStmt: String proceed = scanner.nextLine();...  
| |   │├─ VariableDeclarationExpr: String proceed = scanner.nextLine()...  
| |     │├─ VariableDeclarator: proceed = scanner.nextLine()...  
| |       │├─ ClassOrInterfaceType: String...  
| |         │├─ SimpleName: String...  
| |           ├── SimpleName: proceed...  
| |             └─ MethodCallExpr: scanner.nextLine()...  
| |               ├── NameExpr: scanner...  
| |                 └─ SimpleName: scanner...  
| |                   └─ SimpleName: nextLine...  
| └─ IfStmt: if (proceed.equalsIgnoreCase("no")) {  
System....  
| | | | └─ MethodCallExpr: proceed.equalsIgnoreCase("no")...  
| | | |   │├─ NameExpr: proceed...  
| | | |     │├─ SimpleName: proceed...  
| | | |       ├── SimpleName: equalsIgnoreCase...  
| | | |         └─ StringLiteralExpr: "no"...  
| | └─ BlockStmt: {  
System.out.println("Terminating...");  
  
...  
| | | | └─ ExpressionStmt: System.out.println("Terminating...");;
```

```
| | | | | | └─ MethodInvocationExpr: System.out.println("Terminating...")...
| | | | | |   └─ FieldAccessExpr: System.out...
| | | | | |     └─ NameExpr: System...
| | | | | |       └─ SimpleName: System...
| | | | | |         └─ SimpleName: out...
| | | | | |           └─ SimpleName: println...
| | | | | |             └─ StringLiteralExpr: "Terminating..."...
| | | | | └─ ReturnStmt: return;...
| | | | └─ IfStmt: if (!proceed.equalsIgnoreCase("yes")) {
Syste...
| | | | | | └─ UnaryExpr: !proceed.equalsIgnoreCase("yes")...
| | | | | |   └─ MethodInvocationExpr: proceed.equalsIgnoreCase("yes")...
| | | | | |     └─ NameExpr: proceed...
| | | | | |       └─ SimpleName: proceed...
| | | | | |         └─ SimpleName: equalsIgnoreCase...
| | | | | |           └─ StringLiteralExpr: "yes"...
| | | | | └─ BlockStmt: {
System.out.println("Invalid response. Termi...
| | | | | | └─ ExpressionStmt: System.out.println("Invalid response. Terminating....
| | | | | |   └─ MethodInvocationExpr: System.out.println("Invalid response. Terminating....
| | | | | |     └─ FieldAccessExpr: System.out...
| | | | | |       └─ NameExpr: System...
| | | | | |         └─ SimpleName: System...
| | | | | |           └─ SimpleName: out...
| | | | | |             └─ SimpleName: println...
| | | | | |               └─ StringLiteralExpr: "Invalid response. Terminating..."...
| | | | | └─ ReturnStmt: return;...
| | | └─ ExpressionStmt: System.out.println("Do you understand the interest...
| | |   └─ MethodInvocationExpr: System.out.println("Do you understand the interest...
| | |     └─ FieldAccessExpr: System.out...
| | |       └─ NameExpr: System...
| | |         └─ SimpleName: System...
| | |           └─ SimpleName: out...
| | |             └─ SimpleName: println...
| | |               └─ StringLiteralExpr: "Do you understand the interest rates? (yes/no)"...
| | └─ ExpressionStmt: String understand = scanner.nextLine();...
| |   └─ VariableDeclarationExpr: String understand = scanner.nextLine()...
| |     └─ VariableDeclarator: understand = scanner.nextLine()...
| |       └─ ClassOrInterfaceType: String...
| |         └─ SimpleName: String...
| |           └─ SimpleName: understand...
| |             └─ MethodInvocationExpr: scanner.nextLine()...
| |               └─ NameExpr: scanner...
| |                 └─ SimpleName: scanner...
| |                   └─ SimpleName: nextLine...
```

```

| | | | └─ IfStmt: if (!understand.equalsIgnoreCase("yes")) {
Sy...
| | | | | └─ UnaryExpr: !understand.equalsIgnoreCase("yes")...
| | | | | | └─ MethodCallExpr: understand.equalsIgnoreCase("yes")...
| | | | | | | └─ NameExpr: understand...
| | | | | | | | └─ SimpleName: understand...
| | | | | | | | └─ SimpleName: equalsIgnoreCase...
| | | | | | | | └─ StringLiteralExpr: "yes"...
| | | | | └─ BlockStmt: {
System.out.println("Terminating...");
...
| | | | | | └─ ExpressionStmt: System.out.println("Terminating...");...
| | | | | | | └─ MethodCallExpr: System.out.println("Terminating...")...
| | | | | | | | └─ FieldAccessExpr: System.out...
| | | | | | | | | └─ NameExpr: System...
| | | | | | | | | | └─ SimpleName: System...
| | | | | | | | | | └─ SimpleName: out...
| | | | | | | | | | └─ SimpleName: println...
| | | | | | | | | | └─ StringLiteralExpr: "Terminating..."...
| | | | | | └─ ReturnStmt: return;...
| | | | └─ ExpressionStmt: System.out.println("Interest rates are as follows:...
| | | | | └─ MethodCallExpr: System.out.println("Interest rates are as follows:...
| | | | | | └─ FieldAccessExpr: System.out...
| | | | | | | └─ NameExpr: System...
| | | | | | | | └─ SimpleName: System...
| | | | | | | | └─ SimpleName: out...
| | | | | | | | └─ SimpleName: println...
| | | | | | └─ StringLiteralExpr: "Interest rates are as follows:"...
| | | | └─ ExpressionStmt: System.out.println("Week 1: 10% interest");...
| | | | | └─ MethodCallExpr: System.out.println("Week 1: 10% interest")...
| | | | | | └─ FieldAccessExpr: System.out...
| | | | | | | └─ NameExpr: System...
| | | | | | | | └─ SimpleName: System...
| | | | | | | | └─ SimpleName: out...
| | | | | | | | └─ SimpleName: println...
| | | | | | └─ StringLiteralExpr: "Week 1: 10% interest"...
| | | | └─ ExpressionStmt: System.out.println("Week 2: 15% interest");...
| | | | | └─ MethodCallExpr: System.out.println("Week 2: 15% interest")...
| | | | | | └─ FieldAccessExpr: System.out...
| | | | | | | └─ NameExpr: System...
| | | | | | | | └─ SimpleName: System...
| | | | | | | | └─ SimpleName: out...
| | | | | | | | └─ SimpleName: println...
| | | | | | └─ StringLiteralExpr: "Week 2: 15% interest"...
| | | | └─ ExpressionStmt: System.out.println("Week 3: 20% interest");...

```

```
| | | | └─ MethodCallExpr: System.out.println("Week 3: 20% interest")...
| | | | | └─ FieldAccessExpr: System.out...
| | | | | | └─ NameExpr: System...
| | | | | | | └─ SimpleName: System...
| | | | | | | └─ SimpleName: out...
| | | | | | | └─ SimpleName: println...
| | | | | | | └─ StringLiteralExpr: "Week 3: 20% interest"...
| | | | └─ ExpressionStmt: System.out.println("Week 4: 25% interest");...
| | | | | └─ MethodCallExpr: System.out.println("Week 4: 25% interest")...
| | | | | | └─ FieldAccessExpr: System.out...
| | | | | | | └─ NameExpr: System...
| | | | | | | | └─ SimpleName: System...
| | | | | | | | └─ SimpleName: out...
| | | | | | | | └─ SimpleName: println...
| | | | | | | | └─ StringLiteralExpr: "Week 4: 25% interest"...
| | | | └─ ExpressionStmt: System.out.println("If it exceeds 4 weeks, interes...
| | | | | └─ MethodCallExpr: System.out.println("If it exceeds 4 weeks, interes...
| | | | | | └─ FieldAccessExpr: System.out...
| | | | | | | └─ NameExpr: System...
| | | | | | | | └─ SimpleName: System...
| | | | | | | | └─ SimpleName: out...
| | | | | | | | └─ SimpleName: println...
| | | | | | | | └─ StringLiteralExpr: "If it exceeds 4 weeks, interest returns to 10%."...
| | | | └─ ExpressionStmt: double loanAmount;...
| | | | | └─ VariableDeclarationExpr: double loanAmount...
| | | | | | └─ VariableDeclarator: loanAmount...
| | | | | | | └─ PrimitiveType: double...
| | | | | | | └─ SimpleName: loanAmount...
| | | | └─ WhileStmt: while (true) {
try {
    System.out.prin...
| | | | | └─ BooleanLiteralExpr: true...
| | | | | └─ BlockStmt: {
try {
    System.out.println("How much...
| | | | | | └─ TryStmt: try {
System.out.println("How much do you wan...
| | | | | | | └─ BlockStmt: {
System.out.println("How much do you want to...
| | | | | | | | └─ ExpressionStmt: System.out.println("How much do you want to
borrow...
| | | | | | | | | └─ MethodCallExpr: System.out.println("How much do you want to
borrow...
| | | | | | | | | | └─ FieldAccessExpr: System.out...
| | | | | | | | | | | └─ NameExpr: System...
```

```
| | | | | | | | | | └─ SimpleName: System...  
| | | | | | | | | | └─ SimpleName: out...  
| | | | | | | | | | └─ SimpleName: println...  
| | | | | | | | | | └─ StringLiteralExpr: "How much do you want to borrow starting  
from K200...  
| | | | | | | | | | └─ ExpressionStmt: loanAmount = scanner.nextDouble();...  
| | | | | | | | | |   └─ AssignExpr: loanAmount = scanner.nextDouble()...  
| | | | | | | | | |     └─ NameExpr: loanAmount...  
| | | | | | | | | |       └─ SimpleName: loanAmount...  
| | | | | | | | | |         └─ MethodCallExpr: scanner.nextDouble()...  
| | | | | | | | | |           └─ NameExpr: scanner..  
| | | | | | | | | |             └─ SimpleName: scanner..  
| | | | | | | | | |               └─ SimpleName: nextDouble..  
| | | | | | | | | | └─ IfStmt: if (loanAmount < 200 || loanAmount > 1000) {  
...  
| | | | | | | | | | └─ BinaryExpr: loanAmount < 200 || loanAmount > 1000...  
| | | | | | | | | |   └─ BinaryExpr: loanAmount < 200...  
| | | | | | | | | |     └─ NameExpr: loanAmount...  
| | | | | | | | | |       └─ SimpleName: loanAmount...  
| | | | | | | | | |         └─ IntegerLiteralExpr: 200...  
| | | | | | | | | |   └─ BinaryExpr: loanAmount > 1000...  
| | | | | | | | | |     └─ NameExpr: loanAmount...  
| | | | | | | | | |       └─ SimpleName: loanAmount...  
| | | | | | | | | |         └─ IntegerLiteralExpr: 1000...  
| | | | | | | | | | └─ BlockStmt: {  
    throw new IllegalArgumentException("Invalid...  
| | | | | | | | | | └─ ThrowStmt: throw new IllegalArgumentException("Invalid loan  
a...  
| | | | | | | | | | └─ ObjectCreationExpr: new IllegalArgumentException("Invalid  
loan amount....  
| | | | | | | | | |   └─ ClassOrInterfaceType: IllegalArgumentException...  
| | | | | | | | | |     └─ SimpleName: IllegalArgumentException...  
| | | | | | | | | |       └─ StringLiteralExpr: "Invalid loan amount."...  
| | | | | | | | | | └─ BreakStmt: // Break the loop if input is valid  
break;...  
| | | | | | | | | | └─ CatchClause: catch (InputMismatchException e) {  
    System.out...  
| | | | | | | | | |   └─ Parameter: InputMismatchException e...  
| | | | | | | | | |     └─ ClassOrInterfaceType: InputMismatchException...  
| | | | | | | | | |       └─ SimpleName: InputMismatchException...  
| | | | | | | | | |         └─ SimpleName: e...  
| | | | | | | | | | └─ BlockStmt: {  
    System.out.println("Invalid input. Please e...  
| | | | | | | | | | └─ ExpressionStmt: System.out.println("Invalid input. Please enter a ...
```



```

| | | | | | | | | | MethodCallExpr: System.out.println("Invalid input. Please enter
a ...
| | | | | | | | | | FieldAccessExpr: System.out...
| | | | | | | | | | NameExpr: System...
| | | | | | | | | | SimpleName: System...
| | | | | | | | | | SimpleName: out...
| | | | | | | | | | SimpleName: println...
| | | | | | | | | | StringLiteralExpr: "Invalid input. Please enter a number."...
| | | | | | | | | | ExpressionStmt: // Clear the invalid input
scanner.next();...
| | | | | | | | | | MethodCallExpr: scanner.next()...
| | | | | | | | | | NameExpr: scanner...
| | | | | | | | | | SimpleName: scanner...
| | | | | | | | | | SimpleName: next...
| | | | | | | | CatchClause: catch (IllegalArgumentException e) {
System....
| | | | | | | | Parameter: IllegalArgumentException e...
| | | | | | | | ClassOrInterfaceType: IllegalArgumentException...
| | | | | | | | SimpleName: IllegalArgumentException...
| | | | | | | | SimpleName: e...
| | | | | | | | BlockStmt: {
System.out.println(e.getMessage());
}...
| | | | | | | | ExpressionStmt: System.out.println(e.getMessage());...
| | | | | | | | MethodCallExpr: System.out.println(e.getMessage())...
| | | | | | | | FieldAccessExpr: System.out...
| | | | | | | | NameExpr: System...
| | | | | | | | SimpleName: System...
| | | | | | | | SimpleName: out...
| | | | | | | | SimpleName: println...
| | | | | | | | MethodCallExpr: e.getMessage()...
| | | | | | | | NameExpr: e...
| | | | | | | | SimpleName: e...
| | | | | | | | SimpleName: getMessage...
| | | | | ExpressionStmt: int weeksToRepay;...
| | | | | VariableDeclarationExpr: int weeksToRepay...
| | | | | VariableDeclarator: weeksToRepay...
| | | | | PrimitiveType: int...
| | | | | SimpleName: weeksToRepay...
| | | | WhileStmt: while (true) {
try {
System.out.prin...
| | | | BooleanLiteralExpr: true...
| | | | BlockStmt: {
try {

```

```

    System.out.println("Paying b...
| | | | | └─ TryStmt: try {
    System.out.println("Paying back after h...
| | | | | └─ BlockStmt: {
    System.out.println("Paying back after how m...
| | | | | └─ ExpressionStmt: System.out.println("Paying back after how many
wee...
| | | | | └─ MethodCallExpr: System.out.println("Paying back after how many
wee...
| | | | | └─ FieldAccessExpr: System.out...
| | | | | └─ NameExpr: System...
| | | | | └─ SimpleName: System...
| | | | | └─ SimpleName: out...
| | | | | └─ SimpleName: println...
| | | | | └─ StringLiteralExpr: "Paying back after how many weeks?"...
| | | | | └─ ExpressionStmt: weeksToRepay = scanner.nextInt();...
| | | | | └─ AssignExpr: weeksToRepay = scanner.nextInt()...
| | | | | └─ NameExpr: weeksToRepay...
| | | | | └─ SimpleName: weeksToRepay...
| | | | | └─ MethodCallExpr: scanner.nextInt()...
| | | | | └─ NameExpr: scanner...
| | | | | └─ SimpleName: scanner...
| | | | | └─ SimpleName: nextInt...
| | | | | └─ IfStmt: if (weeksToRepay <= 0) {
    throw new IllegalArg...
| | | | | └─ BinaryExpr: weeksToRepay <= 0...
| | | | | └─ NameExpr: weeksToRepay...
| | | | | └─ SimpleName: weeksToRepay...
| | | | | └─ IntegerLiteralExpr: 0...
| | | | | └─ BlockStmt: {
    throw new IllegalArgumentException("Invalid...
| | | | | └─ ThrowStmt: throw new IllegalArgumentException("Invalid
number...
| | | | | └─ ObjectCreationExpr: new IllegalArgumentException("Invalid
number of we...
| | | | | └─ ClassOrInterfaceType: IllegalArgumentException...
| | | | | └─ SimpleName: IllegalArgumentException...
| | | | | └─ StringLiteralExpr: "Invalid number of weeks."...
| | | | | └─ BreakStmt: // Break the loop if input is valid
break;...
| | | | | └─ CatchClause: catch (InputMismatchException e) {
    System.ou...
| | | | | └─ Parameter: InputMismatchException e...
| | | | | └─ ClassOrInterfaceType: InputMismatchException...
| | | | | └─ SimpleName: InputMismatchException...

```

```
| | | | | | | | └─ SimpleName: e...  
| | | | | | | | └─ BlockStmt: {  
    System.out.println("Invalid input. Please e...  
| | | | | | | | └─ ExpressionStmt: System.out.println("Invalid input. Please enter a ...  
| | | | | | | | └─ MethodCallExpr: System.out.println("Invalid input. Please enter  
a ...  
| | | | | | | | └─ FieldAccessExpr: System.out...  
| | | | | | | | └─ NameExpr: System...  
| | | | | | | | └─ SimpleName: System...  
| | | | | | | | └─ SimpleName: out...  
| | | | | | | | └─ SimpleName: println...  
| | | | | | | | └─ StringLiteralExpr: "Invalid input. Please enter a number."...  
| | | | | | | | └─ ExpressionStmt: // Clear the invalid input  
scanner.next();...  
| | | | | | | | └─ MethodCallExpr: scanner.next()...  
| | | | | | | | └─ NameExpr: scanner...  
| | | | | | | | └─ SimpleName: scanner...  
| | | | | | | | └─ SimpleName: next...  
| | | | | | | └─ CatchClause: catch (IllegalArgumentException e) {  
    System....  
| | | | | | | └─ Parameter: IllegalArgumentException e...  
| | | | | | | └─ ClassOrInterfaceType: IllegalArgumentException...  
| | | | | | | └─ SimpleName: IllegalArgumentException...  
| | | | | | | └─ SimpleName: e...  
| | | | | | | └─ BlockStmt: {  
    System.out.println(e.getMessage());  
}...  
| | | | | | | | └─ ExpressionStmt: System.out.println(e.getMessage());...  
| | | | | | | | └─ MethodCallExpr: System.out.println(e.getMessage())...  
| | | | | | | | └─ FieldAccessExpr: System.out...  
| | | | | | | | └─ NameExpr: System...  
| | | | | | | | └─ SimpleName: System...  
| | | | | | | | └─ SimpleName: out...  
| | | | | | | | └─ SimpleName: println...  
| | | | | | | | └─ MethodCallExpr: e.getMessage()...  
| | | | | | | | └─ NameExpr: e...  
| | | | | | | | └─ SimpleName: e...  
| | | | | | | | └─ SimpleName: getMessage...  
| | | | └─ ExpressionStmt: StudentLoan loan = new StudentLoan(name, loanAmoun...  
| | | | └─ VariableDeclarationExpr: StudentLoan loan = new StudentLoan(name,  
loanAmoun...  
| | | | └─ VariableDeclarator: loan = new StudentLoan(name, loanAmount, 10,  
weeks...  
| | | | | | └─ ClassOrInterfaceType: StudentLoan...  
| | | | | | └─ SimpleName: StudentLoan...
```

```
| | | | | | | | └─ SimpleName: loan...  
| | | | | | | | └─ ObjectCreationExpr: new StudentLoan(name, loanAmount, 10,  
weeksToRepay...  
| | | | | | | | └─ ClassOrInterfaceType: StudentLoan...  
| | | | | | | | └─ SimpleName: StudentLoan...  
| | | | | | | | └─ NameExpr: name...  
| | | | | | | | └─ SimpleName: name...  
| | | | | | | | └─ NameExpr: loanAmount...  
| | | | | | | | └─ SimpleName: loanAmount...  
| | | | | | | | └─ IntegerLiteralExpr: 10...  
| | | | | | | | └─ NameExpr: weeksToRepay...  
| | | | | | | | └─ SimpleName: weeksToRepay...  
| | | | └─ ExpressionStmt: loanManager.addLoan(loan);...  
| | | | └─ MethodCallExpr: loanManager.addLoan(loan)...  
| | | | └─ NameExpr: loanManager...  
| | | | └─ SimpleName: loanManager...  
| | | | └─ SimpleName: addLoan...  
| | | | └─ NameExpr: loan...  
| | | | └─ SimpleName: loan...  
| | | └─ ExpressionStmt: System.out.println("Calculating total payment...")...  
| | | └─ MethodCallExpr: System.out.println("Calculating total payment...")...  
| | | └─ FieldAccessExpr: System.out...  
| | | └─ NameExpr: System...  
| | | └─ SimpleName: System...  
| | | └─ SimpleName: out...  
| | | └─ SimpleName: println...  
| | | └─ StringLiteralExpr: "Calculating total payment..."...  
| | └─ ExpressionStmt: // Formatting total payment  
System.out.println("T...  
| | | | └─ MethodCallExpr: System.out.println("Total payment: K" + String.for...  
| | | | └─ FieldAccessExpr: System.out...  
| | | | └─ NameExpr: System...  
| | | | └─ SimpleName: System...  
| | | | └─ SimpleName: out...  
| | | | └─ SimpleName: println...  
| | | | └─ BinaryExpr: "Total payment: K" + String.format("%.2f", loan.ca...  
| | | | └─ StringLiteralExpr: "Total payment: K"...  
| | | | └─ MethodCallExpr: String.format("%.2f", loan.calculateTotalPayment())...  
| | | | └─ NameExpr: String...  
| | | | └─ SimpleName: String...  
| | | | └─ SimpleName: format...  
| | | | └─ StringLiteralExpr: "%.2f"...  
| | | | └─ MethodCallExpr: loan.calculateTotalPayment()...  
| | | | └─ NameExpr: loan...  
| | | | └─ SimpleName: loan...
```

```
| | | | | | | | | └─ SimpleName: calculateTotalPayment...  
PS C:\Users\USER\Documents\demo>
```