Compiler Design

Java Code Generation

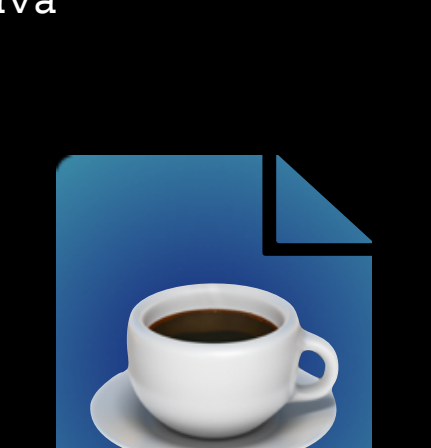
First Step

```
→ ~ vim QuadraticEquationSolver.java
```

```java
public class QuadraticEquationSolver {

    public static void main(String[] args) {
        double a = 1.0;
        double b = -3.0;
        double c = 2.0;

        double[] roots = solveQuadraticEquation(a, b, c);

        if (roots != null) {
            System.out.println("Root 1: " + roots[0]);
            System.out.println("Root 2: " + roots[1]);
        } else {
            System.out.println("The equation has no real roots.");
        }
    }

    /**
     * Solves the quadratic equation ax^2 + bx + c = 0.
     *
     * @param a Coefficient of x^2
     * @param b Coefficient of x
     * @param c Constant term
     * @return An array containing the two roots, or null if there are no real roots.
     */
    public static double[] solveQuadraticEquation(double a, double b, double c) {
        // Calculate the discriminant (b^2 - 4ac)
        double discriminant = b * b - 4 * a * c;

        // Check if the discriminant is non-negative (real roots exist)
        if (discriminant < 0) {
            return null; // No real roots
        }

        // Calculate the two roots using the quadratic formula
        double sqrtDiscriminant = Math.sqrt(discriminant);
        double root1 = (-b + sqrtDiscriminant) / (2 * a);
        double root2 = (-b - sqrtDiscriminant) / (2 * a);

        return new double[] { root1, root2 };
    }
}
```

Second Step

```
→ ~ javac -g QuadraticEquationSolver.java
```



QuadraticEquation
Solver.class

Third Step

```
→ ~ javac -c -l QuadraticEquationSolver.class > QuadraticEquationSolver.txt
```

```
Compiled from "QuadraticEquationSolver.java"
public class QuadraticEquationSolver {
  public QuadraticEquationSolver();
    Code:
       0: aload_0
       1: invokespecial #1                  // Method java/lang/Object."<init>":()V
       4: return
    LineNumberTable:
      line 1: 0
    LocalVariableTable:
      Start  Length  Slot  Name   Signature
          0       5     0  this   LQuadraticEquationSolver;

  public static void main(java.lang.String[]);
    Code:
       0: dconst_1
       1: dstore_1
       2: ldc2_w        #7                  // double -3.0d
       5: dstore_3
       6: ldc2_w        #9                  // double 2.0d
       9: dstore        5
      11: dload_1
      12: dload_3
      13: dload         5
      15: invokestatic  #11                 // Method solveQuadraticEquation:(DDD)[D
      18: astore        7
      20: aload         7
      22: ifnull        58
      25: getstatic     #17                 // Field java/lang/System.out:Ljava/io/PrintStream;
      28: aload         7
      30: iconst_0
      31: daload
      32: invokedynamic #23,  0             // InvokeDynamic #0:makeConcatWithConstants:(D)Ljava/lang/String;
      37: invokevirtual #27                 // Method java/io/PrintStream.println:(Ljava/lang/String;)V
      40: getstatic     #17                 // Field java/lang/System.out:Ljava/io/PrintStream;
      43: aload         7
      45: iconst_1
      46: daload
      47: invokedynamic #33,  0             // InvokeDynamic #1:makeConcatWithConstants:(D)Ljava/lang/String;
      52: invokevirtual #27                 // Method java/io/PrintStream.println:(Ljava/lang/String;)V
      55: goto          66
      58: getstatic     #17                 // Field java/lang/System.out:Ljava/io/PrintStream;
      61: ldc           #34                 // String The equation has no real roots.
      63: invokevirtual #27                 // Method java/io/PrintStream.println:(Ljava/lang/String;)V
      66: return
    LineNumberTable:
      line 4: 0
      line 5: 2
      line 6: 6
      line 8: 11
      line 10: 20
      line 11: 25
      line 12: 40
      line 14: 58
      line 16: 66
    LocalVariableTable:
      Start  Length  Slot  Name   Signature
          0      67     0  args   [Ljava/lang/String;
          2      65     1     a   D
          6      61     3     b   D
         11      56     5     c   D
         20      47     7 roots   [D

  public static double[] solveQuadraticEquation(double, double, double);
    Code:
       0: dload_2
       1: dload_2
       2: dmul
       3: ldc2_w        #36                 // double 4.0d
       6: dload_0
       7: dmul
       8: dload         4
      10: dmul
      11: dsub
      12: dstore        6
      14: dload         6
      16: dconst_0
      17: dcmpg
      18: ifge          23
      21: aconst_null
      22: areturn
      23: dload         6
      25: invokestatic  #38                 // Method java/lang/Math.sqrt:(D)D
      28: dstore        8
      30: dload_2
      31: dneg
      32: dload         8
      34: dadd
      35: ldc2_w        #9                  // double 2.0d
      38: dload_0
      39: dmul
      40: ddiv
      41: dstore        10
      43: dload_2
      44: dneg
      45: dload         8
      47: dsub
      48: ldc2_w        #9                  // double 2.0d
      51: dload_0
      52: dmul
      53: ddiv
      54: dstore        12
      56: iconst_2
      57: newarray       double
      59: dup
      60: iconst_0
      61: dload         10
      63: dastore
      64: dup
      65: iconst_1
      66: dload         12
      68: dastore
      69: areturn
    LineNumberTable:
      line 28: 0
      line 31: 14
      line 32: 21
      line 36: 23
      line 37: 30
      line 38: 43
      line 40: 56
    LocalVariableTable:
      Start  Length  Slot  Name   Signature
          0      70     0     a   D
          0      70     2     b   D
          0      70     4     c   D
         14      56     6 discriminant   D
         30      40     8 sqrtDiscriminant   D
         43      27    10 root1   D
         56      14    12 root2   D
}
```

Fourth Step

Bytecode Explanation

Constructor: public QuadraticEquationSolver()

```
public QuadraticEquationSolver();
  Code:
     0: aload_0
     1: invokespecial #1                  // Method java/lang/Object."<init>":()V
     4: return
```

1. aload_0: Load this onto the stack.
2. invokespecial #1: Call the constructor of the superclass (Object).
3. return: Return from the constructor.

Method: public static void main(java.lang.String[])

```
public static void main(java.lang.String[]);
  Code:
     0: dconst_1
     1: dstore_1
     2: ldc2_w        #7                  // double -3.0d
     5: dstore_3
     6: ldc2_w        #9                  // double 2.0d
     9: dstore        5
    11: dload_1
    12: dload_3
    13: dload         5
    15: invokestatic  #11                 // Method solveQuadraticEquation:(DDD)[D
    18: astore        7
    20: aload         7
    22: ifnull        58
    25: getstatic     #17                 // Field java/lang/System.out:Ljava/io/PrintStream;
    28: aload         7
    30: iconst_0
    31: daload
    32: invokedynamic #23,  0             // InvokeDynamic #0:makeConcatWithConstants:(D)Ljava/lang/String;
    37: invokevirtual #27                 // Method java/io/PrintStream.println:(Ljava/lang/String;)V
    40: getstatic     #17                 // Field java/lang/System.out:Ljava/io/PrintStream;
    43: aload         7
    45: iconst_1
    46: daload
    47: invokedynamic #33,  0             // InvokeDynamic #1:makeConcatWithConstants:(D)Ljava/lang/String;
    52: invokevirtual #27                 // Method java/io/PrintStream.println:(Ljava/lang/String;)V
    55: goto          66
    58: getstatic     #17                 // Field java/lang/System.out:Ljava/io/PrintStream;
    61: ldc           #34                 // String The equation has no real roots.
    63: invokevirtual #27                 // Method java/io/PrintStream.println:(Ljava/lang/String;)V
    66: return
```

1. dconst_1: Push the constant 1.0 onto the stack.
2. dstore_1: Store the double value at local variable index 1 (a).
3. ldc2_w #7: Load the double constant -3.0 onto the stack.
4. dstore_3: Store the double value at local variable index 3 (b).
5. ldc2_w #9: Load the double constant 2.0 onto the stack.
6. dstore 5: Store the double value at local variable index 5 (c).
7. dload_1: Load the double value from local variable index 1 onto the stack.
8. dload_3: Load the double value from local variable index 3 onto the stack.
9. dload 5: Load the double value from local variable index 5 onto the stack.
10. invokestatic #11: Call the static method solveQuadraticEquation with 3 double arguments.
11. astore 7: Store the reference to the result array in local variable index 7.
12. aload 7: Load the array reference from local variable index 7 onto the stack.
13. ifnull 58: Jump to instruction at byte 58 if the reference is null.
14. getstatic #17: Get the System.out field.
15. aload 7: Load the array reference from local variable index 7 onto the stack.
16. iconst_0: Push integer constant 0 onto the stack.
17. daload: Load the double value from the array at index 0.
18. invokedynamic #23: Invoke the dynamic method to concatenate the double value to a string.
19. invokevirtual #27: Call println with the concatenated string.
20. getstatic #17: Get the System.out field.
21. aload 7: Load the array reference from local variable index 7 onto the stack.
22. iconst_1: Push integer constant 1 onto the stack.
23. daload: Load the double value from the array at index 1.
24. invokedynamic #33: Invoke the dynamic method to concatenate the double value to a string.
25. invokevirtual #27: Call println with the concatenated string.
26. goto 66: Jump to instruction at byte 66.
27. getstatic #17: Get the System.out field.
28. ldc #34: Push the constant string “The equation has no real roots.” onto the stack.
29. invokevirtual #27: Call println with the string.
30. return: Return from the main method.

Method: public static double[] solveQuadraticEquation(double, double, double)

```
public static double[] solveQuadraticEquation(double, double, double);
  Code:
     0: dload_2
     1: dload_2
     2: dmul
     3: ldc2_w        #36                 // double 4.0d
     6: dload_0
     7: dmul
     8: dload         4
    10: dmul
    11: dsub
    12: dstore        6
    14: dload         6
    16: dconst_0
    17: dcmpg
    18: ifge          23
    21: aconst_null
    22: areturn
    23: dload         6
    25: invokestatic  #38                 // Method java/lang/Math.sqrt:(D)D
    28: dstore        8
    30: dload_2
    31: dneg
    32: dload         8
    34: dadd
    35: ldc2_w        #9                  // double 2.0d
    38: dload_0
    39: dmul
    40: ddiv
    41: dstore        10
    43: dload_2
    44: dneg
    45: dload         8
    47: dsub
    48: ldc2_w        #9                  // double 2.0d
    51: dload_0
    52: dmul
    53: ddiv
    54: dstore        12
    56: iconst_2
    57: newarray       double
    59: dup
    60: iconst_0
    61: dload         10
    63: dastore
    64: dup
    65: iconst_1
    66: dload         12
    68: dastore
    69: areturn
```

1. dload_2: Load the double value from local variable index 2 (b) onto the stack.
2. dload_2: Load the double value from local variable index 2 (b) onto the stack.
3. dmul: Multiply the two top values on the stack.
4. ldc2_w #36: Load the double constant 4.0 onto the stack.
5. dload_0: Load the double value from local variable index 0 (a) onto the stack.
6. dmul: Multiply the two top values on the stack.
7. dload 4: Load the double value from local variable index 4 (c) onto the stack.
8. dmul: Multiply the two top values on the stack.
9. dsub: Subtract the top value on the stack from the second-to-top value.
10. dstore 6: Store the result in local variable index 6 (discriminant).
11. dload 6: Load the double value from local variable index 6 (discriminant) onto the stack.
12. dconst_0: Push the constant 0.0 onto the stack.
13. dcmpg: Compare the two top values on the stack (if discriminant > 0).
14. ifge 23: Jump to instruction at byte 23 if the comparison result is greater or equal.
15. aconst_null: Push null onto the stack.
16. areturn: Return null from the method.
17. dload 6: Load the double value from local variable index 6 (discriminant) onto the stack.
18. `invokestatic