

Pass1.java

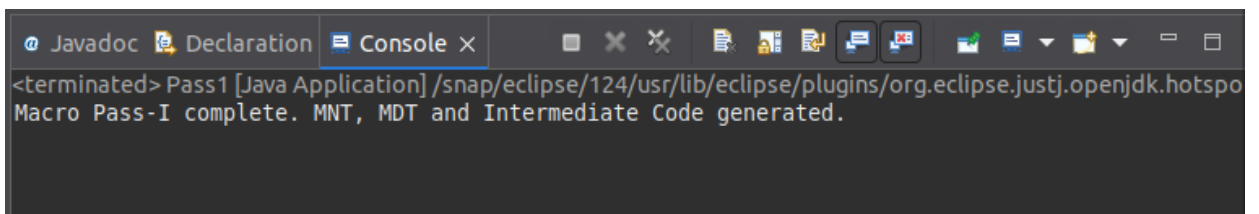
```
1  package Macroprocessor;
2
3  import java.io.*;
4  import java.util.*;
5
6  public class Pass1 {
7      public static void main(String[] args) throws IOException {
8          // Sample macro code (can be replaced with file input)
9          String[] code = {
10              "MACRO",
11              "INCR &ARG1",
12              "A 1,&ARG1",
13              "MEND",
14              "START 101",
15              "INCR TERM",
16              "MOVEM AREG,TERM",
17              "END"
18          };
19
20          List<String[]> MNT = new ArrayList<>();
21          List<String> MDT = new ArrayList<>();
22          BufferedWriter inter = new BufferedWriter(new
23              FileWriter("intermediate.txt"));
24          BufferedWriter mntFile = new BufferedWriter(new FileWriter("mnt.txt"));
25          BufferedWriter mdtFile = new BufferedWriter(new FileWriter("mdt.txt"));
26
27          boolean isMacro = false;
28          int mdtIndex = 0;
29
30          for (int i = 0; i < code.length; i++) {
31              String line = code[i];
32              String[] parts = line.trim().split("\\s+", 2);
33
34              if (parts[0].equals("MACRO")) {
35                  isMacro = true;
36                  String[] defParts = code[i + 1].trim().split("\\s+", 2);
37                  String macroName = defParts[0];
38                  MNT.add(new String[]{macroName, String.valueOf(mdtIndex)});
39                  i++; // skip macro name line
40                  continue;
41              }
42
43              if (parts[0].equals("MEND")) {
44                  MDT.add("MEND");
45                  mdtIndex++;
46                  isMacro = false;
47                  continue;
48              }
49          }
50      }
51  }
```

```

49         if (isMacro) {
50             MDT.add(line);
51             mdtIndex++;
52         } else {
53             inter.write(line + "\n");
54         }
55     }
56
57     // Writing MNT
58     for (String[] entry : MNT) {
59         mntFile.write(entry[0] + "\t" + entry[1] + "\n");
60     }
61
62     // Writing MDT
63     for (int i = 0; i < MDT.size(); i++) {
64         mdtFile.write(i + "\t" + MDT.get(i) + "\n");
65     }
66
67     inter.close();
68     mntFile.close();
69     mdtFile.close();
70
71     System.out.println("Macro Pass-I complete. MNT, MDT and Intermediate Code
generated.");
72     }
73 }

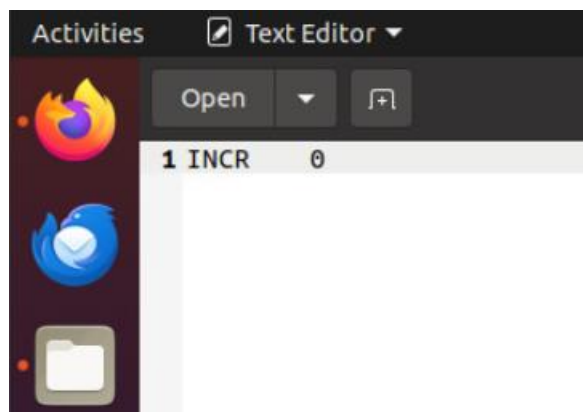
```

Output :

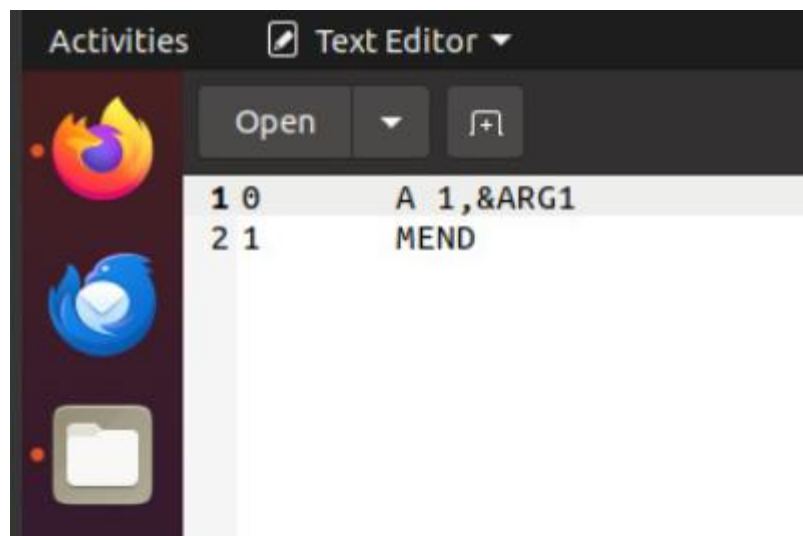


The screenshot shows the Eclipse IDE's Console window. The title bar includes tabs for 'Javadoc', 'Declaration', and 'Console'. The console output displays the message: '<terminated> Pass1 [Java Application] /snap/eclipse/124/usr/lib/eclipse/plugins/org.eclipse.justj.openjdk.hotspot... Macro Pass-I complete. MNT, MDT and Intermediate Code generated.'

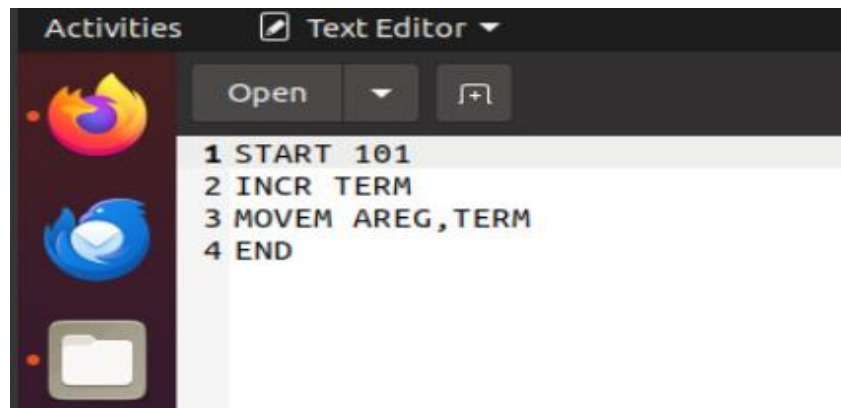
MNT.txt



MDT.txt



Intermediate.txt



Pass2.java

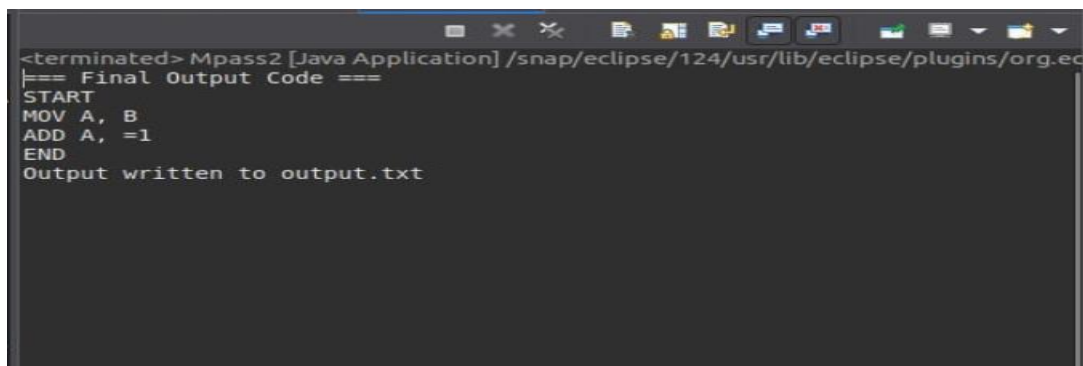
```
1  package Macroprocessor;
2
3  import java.io.*;
4  import java.util.*;
5
6  public class Pass2 {
7      private final Map<String, Integer> MNT = new HashMap<>();
8      private final List<String> MDT = new ArrayList<>();
9      private final List<String> intermediateCode = new ArrayList<>();
10     private final List<String> outputCode = new ArrayList<>();
11
12     public Pass2() {
13         // Simulating already available MNT, MDT, and intermediate code from Pass-I
14
15         // Macro Name Table: Macro name → MDT index
16         MNT.put("INCR", 0);
17
18         // Macro Definition Table
19         MDT.add("INCR &ARG");
20         MDT.add("ADD &ARG, =1");
21         MDT.add("MEND");
22
23         // Intermediate Code (no macro definitions)
24         intermediateCode.add("START");
25         intermediateCode.add("MOV A, B");
26         intermediateCode.add("INCR A");
27         intermediateCode.add("END");
28     }
29
30     public void expandMacros() {
31         for (String line : intermediateCode) {
32             String[] tokens = line.trim().split("\\s+");
33
34             if (tokens.length == 0) continue;
35
36             String macroName = tokens[0];
37
38             if (MNT.containsKey(macroName)) {
39                 int index = MNT.get(macroName);
40                 String actualArg = tokens.length > 1 ? tokens[1] : "";
41
42                 // Skip the macro header in MDT
43                 index++;
44
45                 while (!MDT.get(index).equalsIgnoreCase("MEND")) {
46                     String defLine = MDT.get(index);
47
48                     // Replace formal argument with actual argument
49                     String expandedLine = defLine.replace("&ARG", actualArg);
```

```

50         outputCode.add(expandedLine);
51         index++;
52     }
53     } else {
54         outputCode.add(line);
55     }
56 }
57 }
58
59 public void displayOutput() {
60     System.out.println("=== Final Output Code ===");
61
62     try (BufferedWriter writer = new BufferedWriter(new
63 FileWriter("output.txt"))) {
64         for (String line : outputCode) {
65             System.out.println(line);           // Console output
66             writer.write(line + "\n");          // Write to output.txt
67         }
68         System.out.println("Output written to output.txt");
69     } catch (IOException e) {
70         System.out.println("Error writing to file: " + e.getMessage());
71     }
72 }
73
74 public static void main(String[] args) {
75     Pass2 pass2 = new Pass2();
76     pass2.expandMacros();
77     pass2.displayOutput();
78 }

```

Output

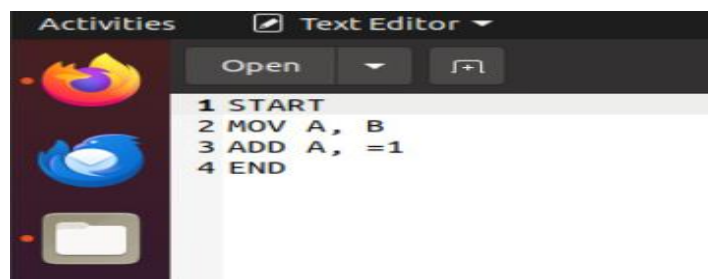


```

<terminated> Mpass2 [Java Application] /snap/eclipse/124/usr/lib/eclipse/plugins/org.ec
=== Final Output Code ===
START
MOV A, B
ADD A, =1
END
Output written to output.txt

```

Output.txt



```

1 START
2 MOV A, B
3 ADD A, =1
4 END

```