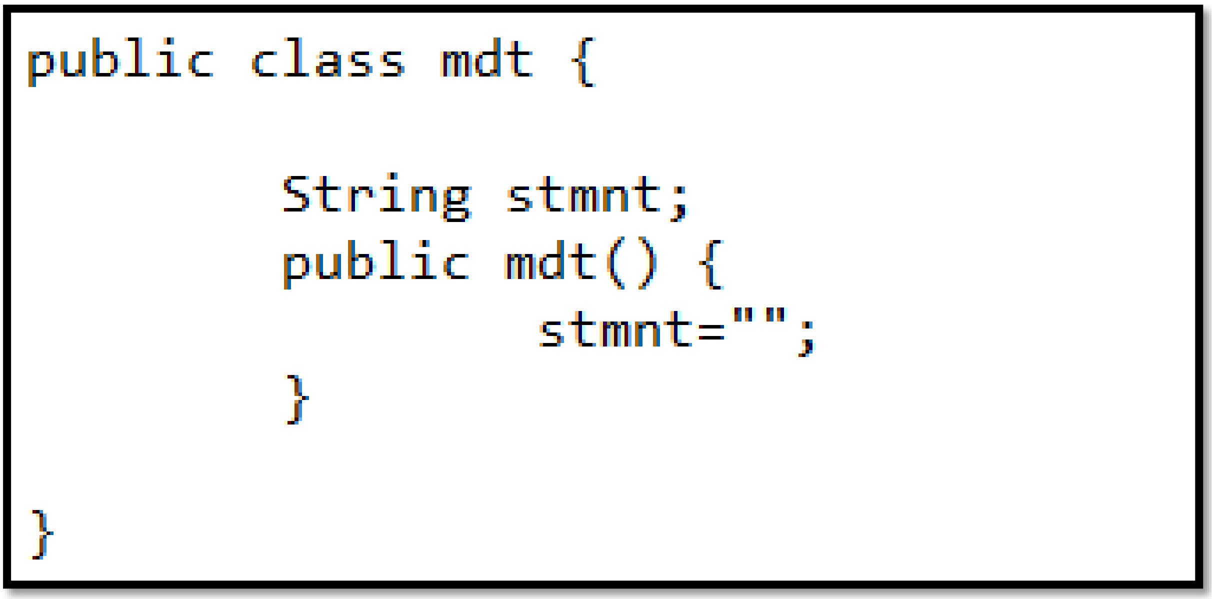
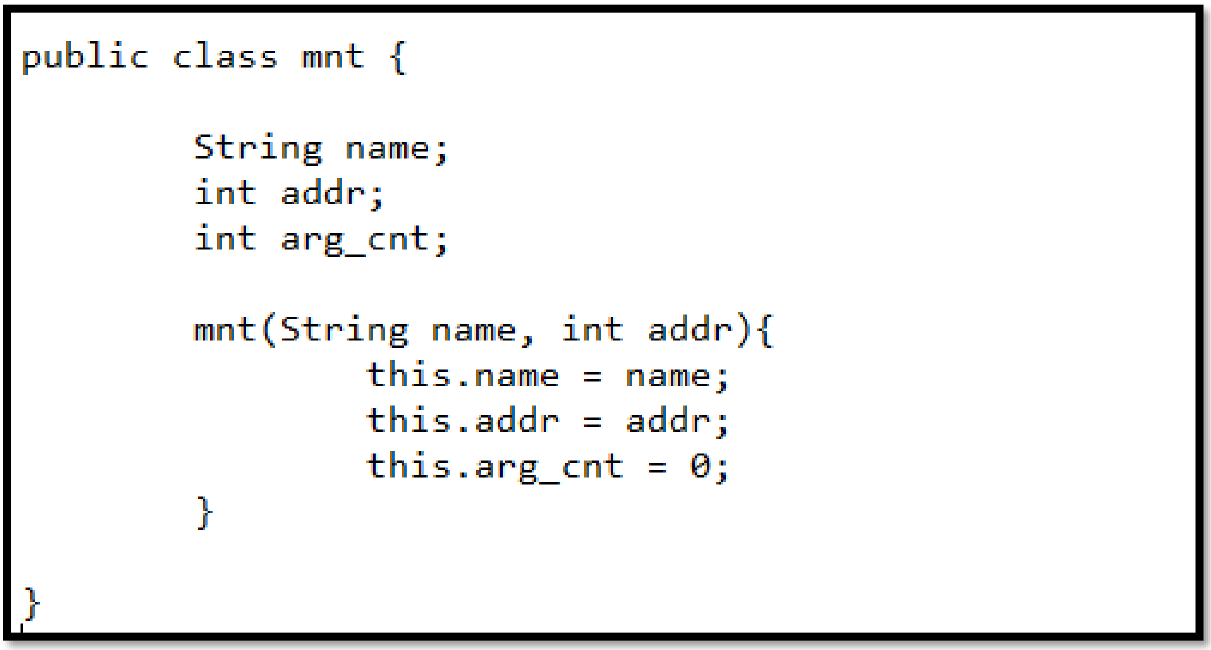
mdt.java File Code :



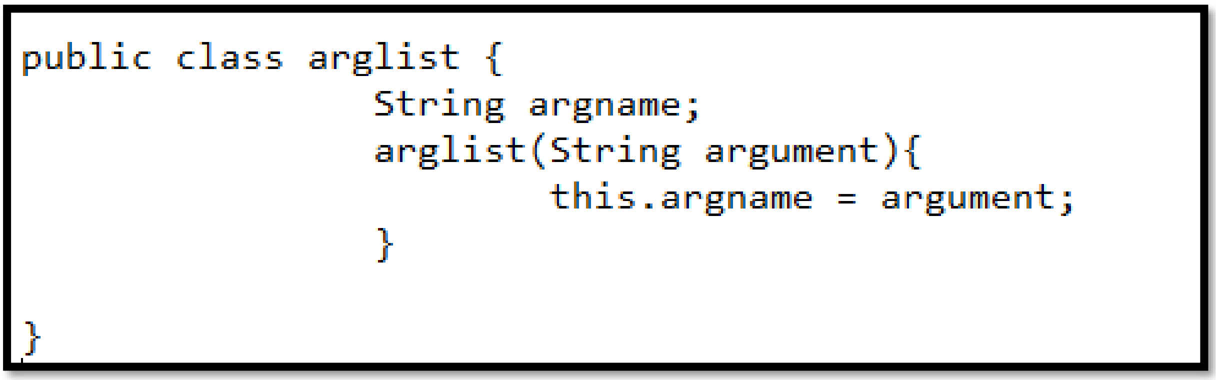
```java
public class mdt {

        String stmnt;
        public mdt() {
                stmnt="";
        }

}
```

mnt.java File Code :



```java
public class mnt {

        String name;
        int addr;
        int arg_cnt;

        mnt(String name, int addr){
                this.name = name;
                this.addr = addr;
                this.arg_cnt = 0;
        }

}
```

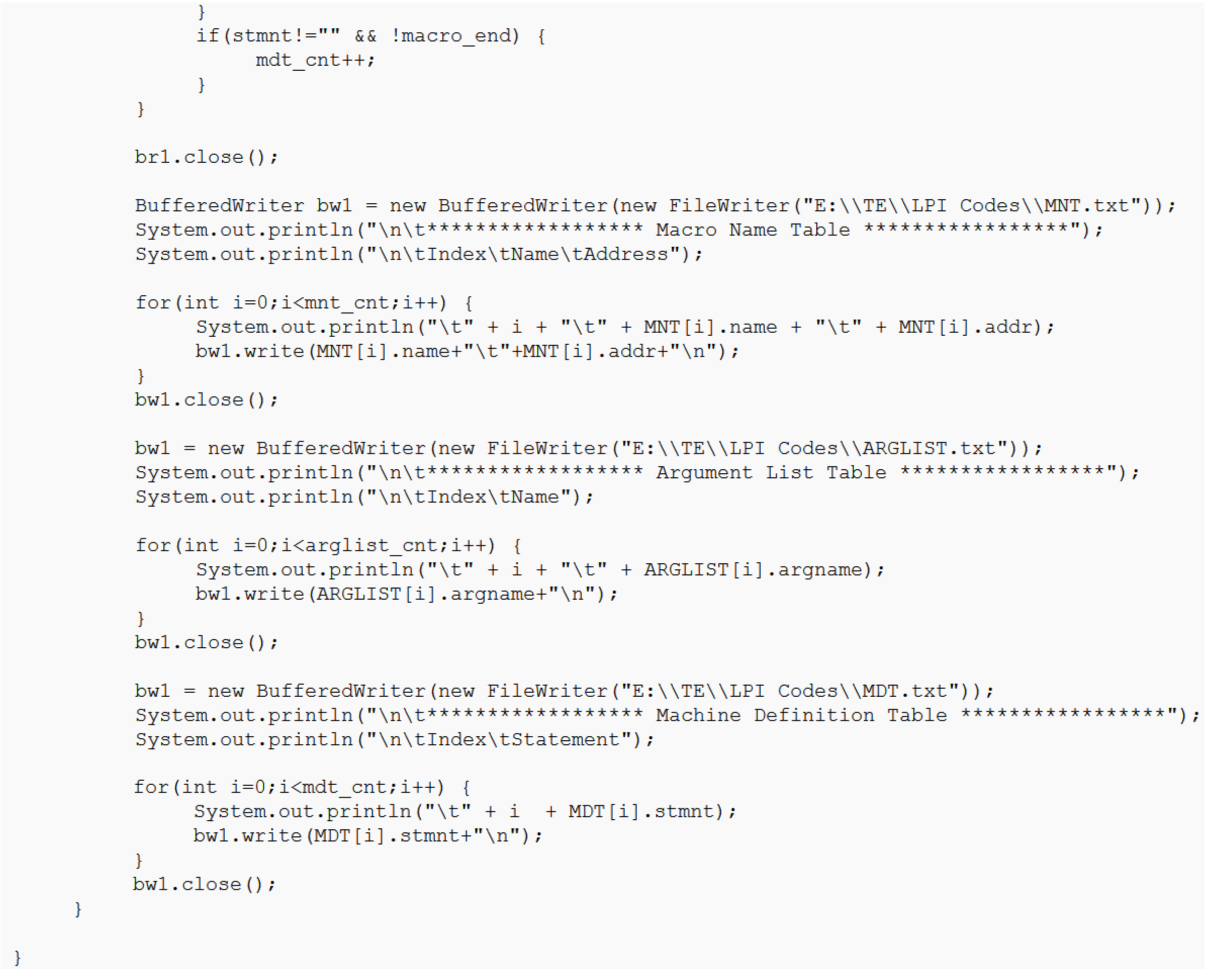
arglist.java File Code :



```java
public class arglist {
                String argname;
                arglist(String argument){
                        this.argname = argument;
                }

}
```

Mpass1.java

```
import java.io.*;

public class Mpass1 {

     public static void main(String[] args)throws IOException {

          BufferedReader br1 = new BufferedReader(new FileReader("E:\\TE\\LPI Codes
\\MacroProcessor\\src\\input.txt"));
          String line;
          mdt[] MDT = new mdt[20];
          mnt[] MNT = new mnt[4];
          arglist[] ARGLIST = new arglist[10];

          boolean macro_start = false;
          boolean macro_end = false;
          boolean fill_arglist = false;
          int mdt_cnt = 0;
          int mnt_cnt = 0;
          int arglist_cnt = 0;

          while((line = br1.readLine()) != null) {
               line  = line.replaceAll(","," ");
               String[] tokens = line.split("\\s+");
               MDT[mdt_cnt] = new mdt();

               String stmnt="";

               for(int i=0;i<tokens.length;i++) {
                    if(tokens[i].equalsIgnoreCase("mend")) {
                         MDT[mdt_cnt++].stmnt = "\t"+tokens[i];
                          macro_end = true;
                    }
                    if(tokens[i].equalsIgnoreCase("macro")) {
                         macro_start = true;
                         macro_end = false;
                    }
                    else if(!macro_end) {
                         if(macro_start) {
                              MNT[mnt_cnt++] = new mnt(tokens[i],mdt_cnt);
                              macro_start = false;
                              fill_arglist = true;
                         }
                         if(fill_arglist) {
                              while(i<tokens.length) {
                                   MDT[mdt_cnt].stmnt = MDT[mdt_cnt].stmnt + "\t" + tokens[i];
                                   stmnt = stmnt + "\t" + tokens[i];
                                   if(tokens[i].matches("&[a-zA-Z]+") || tokens[i].matches("&[a-
zA-Z]+[0-9]+")) {

                                        ARGLIST[arglist_cnt++] = new arglist(tokens[i]);
                                   }
                                   i++;
                              }
                              fill_arglist = false;
                         }
                         else {
                              if(tokens[i].matches("[a-zA-Z]+") || tokens[i].matches("[a-zA-
Z]+[0-9]+") || tokens[i].matches("[0-9]")) {
                                   MDT[mdt_cnt].stmnt = MDT[mdt_cnt].stmnt + "\t" + tokens[i];
                                   stmnt = stmnt + "\t" + tokens[i];
                              }
                              if(tokens[i].matches("&[a-zA-Z]+") || tokens[i].matches("&[a-zA-
Z]+[0-9]+")) {
                                   for(int j=0;j<arglist_cnt;j++) {
                                        if(tokens[i].equals(ARGLIST[j].argname)) {
                                             MDT[mdt_cnt].stmnt = MDT[mdt_cnt].stmnt + "\t#" + (j+
1);

                                             stmnt = stmnt + "\t+" + (j+1);
                                        }
                                   }
                              }
                         }
                    }
```

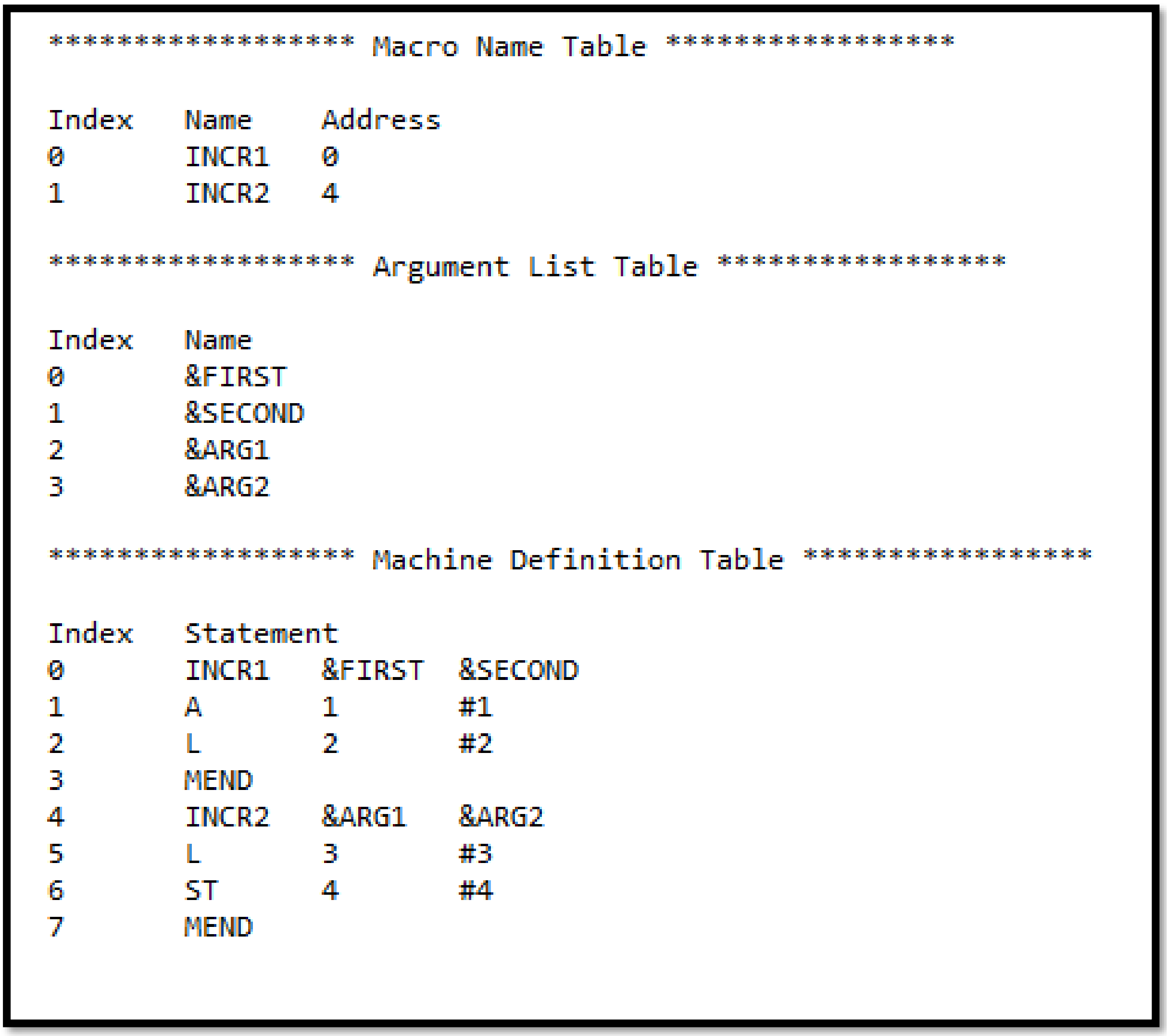


```java
               }
               if(stmnt!="" && !macro_end) {
                    mdt_cnt++;
               }
          }

          br1.close();

          BufferedWriter bw1 = new BufferedWriter(new FileWriter("E:\\TE\\LPI Codes\\MNT.txt"));
          System.out.println("\n\t****************** Macro Name Table *****************");
          System.out.println("\n\tIndex\tName\tAddress");

          for(int i=0;i<mnt_cnt;i++) {
               System.out.println("\t" + i + "\t" + MNT[i].name + "\t" + MNT[i].addr);
               bw1.write(MNT[i].name+"\t"+MNT[i].addr+"\n");
          }
          bw1.close();

          bw1 = new BufferedWriter(new FileWriter("E:\\TE\\LPI Codes\\ARGLIST.txt"));
          System.out.println("\n\t****************** Argument List Table *****************");
          System.out.println("\n\tIndex\tName");

          for(int i=0;i<arglist_cnt;i++) {
               System.out.println("\t" + i + "\t" + ARGLIST[i].argname);
               bw1.write(ARGLIST[i].argname+"\n");
          }
          bw1.close();

          bw1 = new BufferedWriter(new FileWriter("E:\\TE\\LPI Codes\\MDT.txt"));
          System.out.println("\n\t****************** Machine Definition Table *****************");
          System.out.println("\n\tIndex\tStatement");

          for(int i=0;i<mdt_cnt;i++) {
               System.out.println("\t" + i  + MDT[i].stmnt);
               bw1.write(MDT[i].stmnt+"\n");
          }
          bw1.close();
     }

}
```

Input.txt file:

```
MACRO
INCR1 &FIRST, &SECOND
A 1, &FIRST
L 2, &SECOND
MEND
MACRO
INCR2 &ARG1, &ARG2
L 3, &ARG1
ST 4, &ARG2
MEND
PRG2 START
USING *, BASE
INCR1 DATA1, DATA2
INCR2 DATA3, DATA4
FOUR DC F'4'
FIVE DC F'5'
BASE EQU 8
TEMP DS 1F
DROP 8
END
```

Output:



```
****************** Macro Name Table *****************

Index   Name    Address
0       INCR1   0
1       INCR2   4

****************** Argument List Table *****************

Index   Name
0       &FIRST
1       &SECOND
2       &ARG1
3       &ARG2

****************** Machine Definition Table *****************

Index   Statement
0       INCR1   &FIRST  &SECOND
1       A       1       #1
2       L       2       #2
3       MEND
4       INCR2   &ARG1   &ARG2
5       L       3       #3
6       ST      4       #4
7       MEND
```