CFL CW4

1. FIB

.class public fib.fib  
.super java/lang/Object  
  
.method public <init>()V  
 aload\_0  
 invokenonvirtual java/lang/Object/<init>()V  
 return  
.end method  
  
.method public static write(I)V  
 .limit locals 1  
 .limit stack 2  
 getstatic java/lang/System/out Ljava/io/PrintStream;  
 iload 0  
 invokevirtual java/io/PrintStream/println(I)V  
 return  
.end method  
  
.method public static read()I  
 .limit locals 10  
 .limit stack 10  
  
 ldc 0  
 istore 1 ; this will hold our final integer  
Label1:  
 getstatic java/lang/System/in Ljava/io/InputStream;  
 invokevirtual java/io/InputStream/read()I  
 istore 2  
 iload 2  
 ldc 10 ; the newline delimiter  
 isub  
 ifeq Label2  
 iload 2  
 ldc 32 ; the space delimiter  
 isub  
 ifeq Label2  
  
 iload 2  
 ldc 48 ; we have our digit in ASCII, have to subtract it from 48  
 isub  
 ldc 10  
 iload 1  
 imul  
 iadd  
 istore 1  
 goto Label1  
Label2:  
 ;when we come here we have our integer computed in local variable 1  
 iload 1  
 ireturn  
.end method  
  
.method public static main([Ljava/lang/String;)V  
 .limit locals 200  
 .limit stack 200  
  
 invokestatic fib/fib/read()I  
istore 0  
ldc 0  
istore 1  
ldc 1  
istore 2  
ldc 0  
istore 3  
  
Loop\_begin\_0:  
  
iload 0  
ldc 0  
if\_icmple Loop\_end\_1  
iload 2  
istore 3  
iload 1  
iload 2  
iadd  
istore 2  
iload 3  
istore 1  
iload 0  
ldc 1  
isub  
istore 0  
goto Loop\_begin\_0  
  
Loop\_end\_1:  
  
iload 1  
invokestatic fib/fib/write(I)V  
  
  
 return  
  
.end method

FACT

.class public fact.fact  
.super java/lang/Object  
  
.method public <init>()V  
 aload\_0  
 invokenonvirtual java/lang/Object/<init>()V  
 return  
.end method  
  
.method public static write(I)V  
 .limit locals 1  
 .limit stack 2  
 getstatic java/lang/System/out Ljava/io/PrintStream;  
 iload 0  
 invokevirtual java/io/PrintStream/println(I)V  
 return  
.end method  
  
.method public static read()I  
 .limit locals 10  
 .limit stack 10  
  
 ldc 0  
 istore 1 ; this will hold our final integer  
Label1:  
 getstatic java/lang/System/in Ljava/io/InputStream;  
 invokevirtual java/io/InputStream/read()I  
 istore 2  
 iload 2  
 ldc 10 ; the newline delimiter  
 isub  
 ifeq Label2  
 iload 2  
 ldc 32 ; the space delimiter  
 isub  
 ifeq Label2  
  
 iload 2  
 ldc 48 ; we have our digit in ASCII, have to subtract it from 48  
 isub  
 ldc 10  
 iload 1  
 imul  
 iadd  
 istore 1  
 goto Label1  
Label2:  
 ;when we come here we have our integer computed in local variable 1  
 iload 1  
 ireturn  
.end method  
  
.method public static main([Ljava/lang/String;)V  
 .limit locals 200  
 .limit stack 200  
  
 invokestatic fact/fact/read()I  
istore 0  
ldc 1  
istore 1  
iload 0  
istore 2  
  
Loop\_begin\_0:  
  
iload 1  
iload 0  
if\_icmpge Loop\_end\_1  
iload 2  
iload 1  
imul  
istore 2  
ldc 1  
iload 1  
iadd  
istore 1  
goto Loop\_begin\_0  
  
Loop\_end\_1:  
  
iload 2  
invokestatic fact/fact/write(I)V  
  
  
 return  
  
.end method

1. Please see code

.class public prog.prog  
.super java/lang/Object  
  
.method public <init>()V  
 aload\_0  
 invokenonvirtual java/lang/Object/<init>()V  
 return  
.end method  
  
.method public static write(I)V  
 .limit locals 1  
 .limit stack 2  
 getstatic java/lang/System/out Ljava/io/PrintStream;  
 iload 0  
 invokevirtual java/io/PrintStream/println(I)V  
 return  
.end method  
  
.method public static read()I  
 .limit locals 10  
 .limit stack 10  
  
 ldc 0  
 istore 1 ; this will hold our final integer  
Label1:  
 getstatic java/lang/System/in Ljava/io/InputStream;  
 invokevirtual java/io/InputStream/read()I  
 istore 2  
 iload 2  
 ldc 10 ; the newline delimiter  
 isub  
 ifeq Label2  
 iload 2  
 ldc 32 ; the space delimiter  
 isub  
 ifeq Label2  
  
 iload 2  
 ldc 48 ; we have our digit in ASCII, have to subtract it from 48  
 isub  
 ldc 10  
 iload 1  
 imul  
 iadd  
 istore 1  
 goto Label1  
Label2:  
 ;when we come here we have our integer computed in local variable 1  
 iload 1  
 ireturn  
.end method  
  
.method public static main([Ljava/lang/String;)V  
 .limit locals 200  
 .limit stack 200  
  
 ldc 1  
istore 0  
  
Loop\_begin\_0:  
  
iload 0  
ldc 10  
if\_icmpgt Loop\_end\_1  
ldc 1  
istore 0  
  
Loop\_begin\_2:  
  
iload 0  
ldc 10  
if\_icmpgt Loop\_end\_3  
iload 0  
invokestatic prog/prog/write(I)V  
iload 0  
ldc 1  
iadd  
istore 0  
goto Loop\_begin\_2  
  
Loop\_end\_3:  
  
iload 0  
ldc 1  
iadd  
istore 0  
goto Loop\_begin\_0  
  
Loop\_end\_1:  
  
  
  
 return  
  
.end method

The output of the program will be 1,2,3,4,5,6,7,8,9,10.

The reason why I have chosen to do it this way is that it only seems logical if both of the ‘i’ are referring to the same memory location as they are the same variable. If the programmer wishes to make a to have a different program instead, he would just change one of the variables for example:

for i := 1 upto 10 do {

for j := 1 upto 10 do {

write j

}

write i

}

The following example will result in the following combination being repeated 10 times: 1,2,3,4,5,6,7,8,9,10,1