CSE465 HW1
Ling Xiang
2/11/2019

1.

For arbitrary declaration, I think it is much easier for programmers to write, since programmers can throw a variable declaration whenever they need a new variable. From the example of the questions, when the variable is needed, it is also declared. Like int temp, it is then created in the loop where it is needed. In this case, it is much easier for programmers to write and know the type of the variable. However, there are trade off for being easy to write. When read the program, declaration is not in a section of the same feature on the top of the program which is the simplicity of readability. Decelerations are separated on in the program. As a result, it is very difficult to read, since readers need to read one line by one line to know what kind of variables are created and what the type of the variables.

For declaration on the top of the program, it is the opposite of arbitrary declaration. It is easy to read than write. For simplicity, which means a section of one feature. In this case, declarations are all stack on the top of the program. When readers are trying to figure out programs, they can just skip over the program and know what variables are being used in it. But for programmers, they need to go to the top of the program every time they want to add a new variable which will takes a lot of time and energy.

For example, a==b?b:c which means if a = b, then return b, otherwise c. This is easy to write, as we can only one line of code to finished it. But this is not easy to read for non-programmers.

2.

```
[xiangl2@ceclnx01:~/cse465$ vim Sample.java
[xiangl2@ceclnx01:~/cse465$ javac Sample.java
[xiangl2@ceclnx01:~/cse465$ last | java Sample
xiangl2
xiangl2  pts/13       172.25.232.154   Mon Feb 11 17:30   still logged in
xiangl2  pts/14       172.25.232.154   Mon Feb 11 15:37 - 15:42  (00:05)
xiangl2  pts/63       172.25.232.154   Mon Feb 11 15:34 - 15:42  (00:08)
zmudam   pts/8        10.32.1.80       Mon Feb 11 13:11 - 13:13  (00:02)
xiangl2  pts/15       172.25.163.77    Mon Feb 11 12:27 - 14:39  (02:11)
zmudam   pts/19       10.32.1.80       Mon Feb 11 10:04 - 13:13  (03:09)
zmudam   pts/17       10.32.1.80       Mon Feb 11 10:03 - 13:13  (03:10)
zmudam   pts/11       10.32.1.80       Mon Feb 11 09:27 - 09:32  (00:04)
xiangl2  pts/0        69.135.79.131    Sat Feb  9 23:18 - 23:43  (00:25)
xiangl2  pts/16       69.135.79.131    Sat Feb  9 22:06 - 23:16  (01:10)
xiangl2@ceclnx01:~/cse465$
```

3. Report of Interpreter
1. This interpreter allows java to execute Z+- programs.

2. To Do:

variable can have int or string type

Allow for loops inside for loops

Allow +=, -=, *=, = expression

3. Do not allow programs to call a program.

4. For example program prog11.zpm, interpreter can not solve the program calling, as a result, it returns run time error whenever encountered the word “PROC” and “ENDPROC”

5. Compare the runtime speed of z+- program - Prog2.zpm

I used

```
final long startTime = System.currentTimeMillis();
final long endTime = System.currentTimeMillis();
System.out.println(endTime - startTime);
```

z+- program: 66

Java: 0

As the number shows, z+- program is much slower than Java compiler