**LAB4\_assignment**

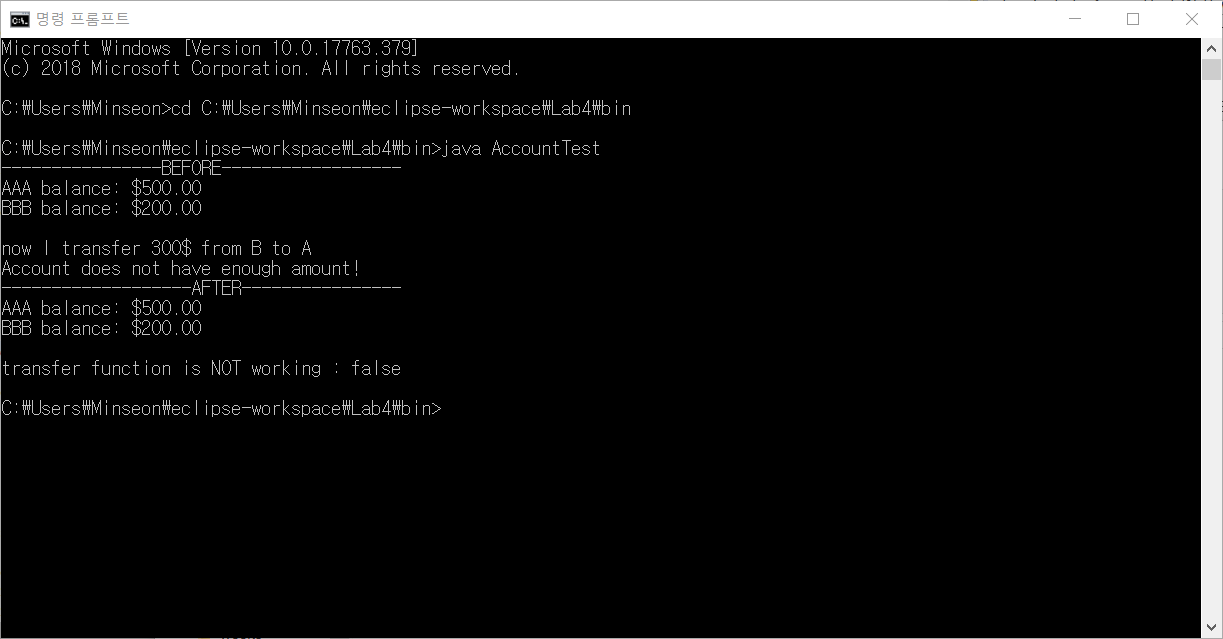
|  |  |  |  |
| --- | --- | --- | --- |
| Name: | 신민선 | Department: | 미디어학과 |
| Student ID: | 201723307 | Room Number: | 팔달관 333호 |
| Due Date: | (March 30, 11:30 ) | | |

Exercise

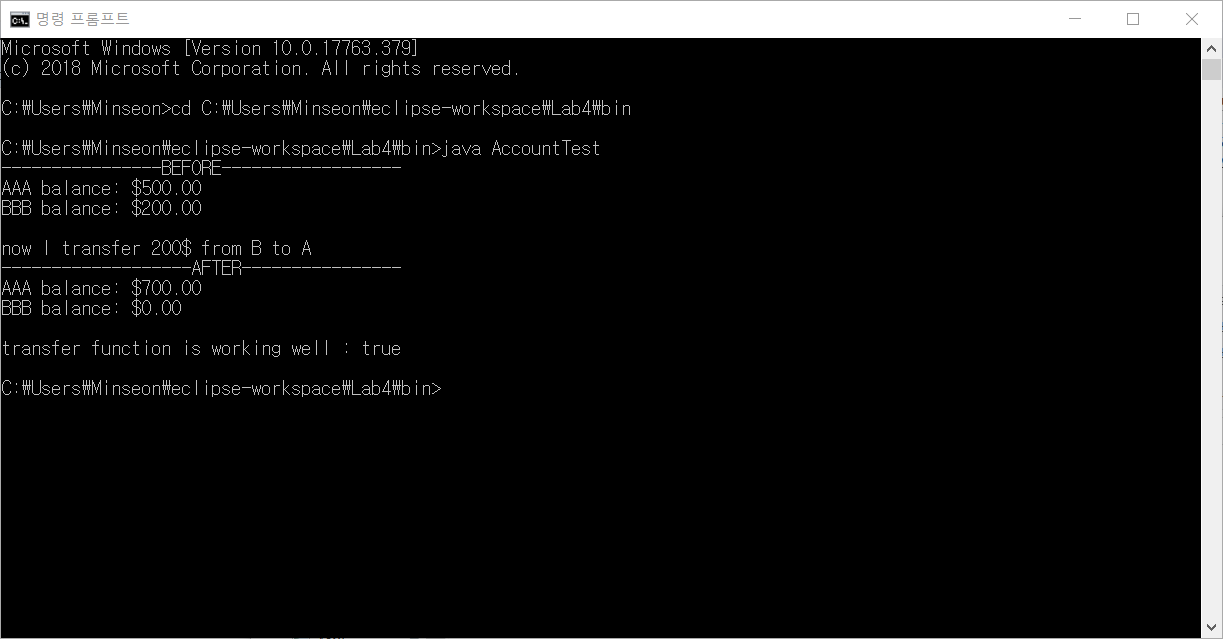
1. Answer the questions about **Fig 3.8** (**Account.java**).

I will answer both A and B together.

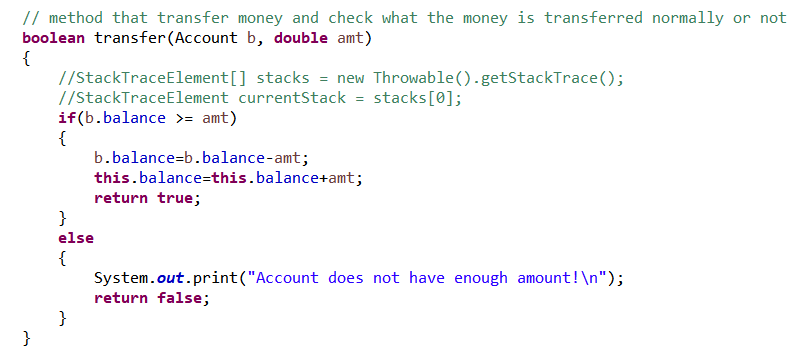
case 1) false



case 2) true



code)



The program has logical error.

Because a variable named ‘no’ about account number is defined only private as access modifier, we have to correct from private to **static** private.

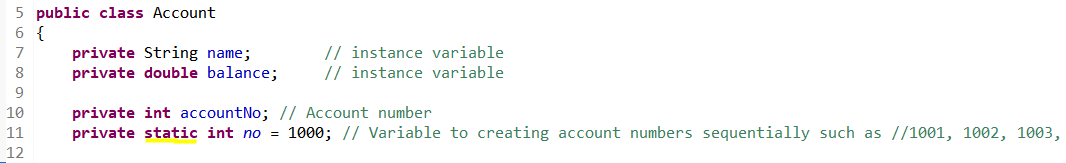
As I put **static modifier**, the account number gradually save it started from 1000.

Static modifier is related with Dynamic memory allocation.

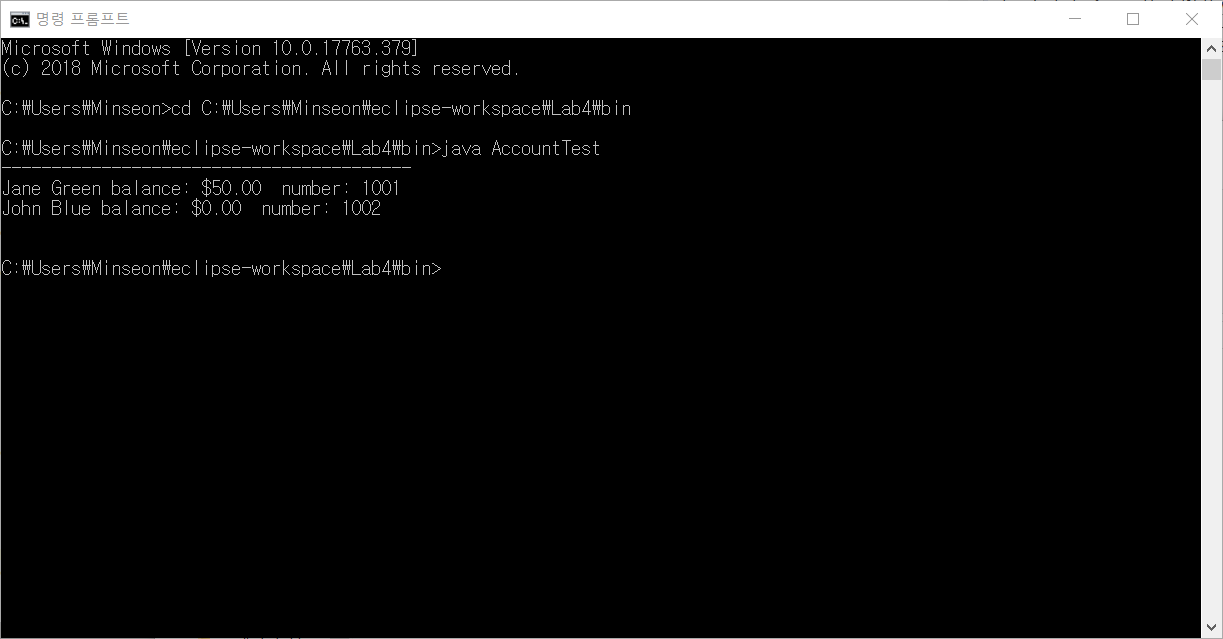
Normal variable(not static) saves heap area, so they automatically reset and reassign addresses. But static variables save method area, which is assign address and maintain it until the program ended.

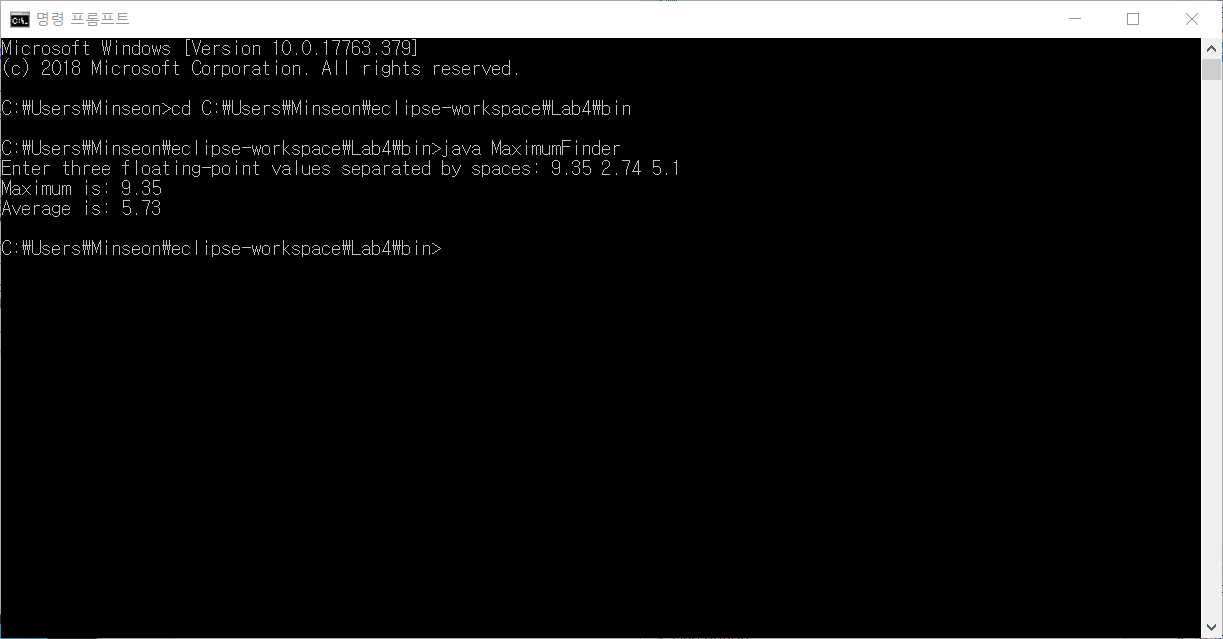
Therefore, in order to maintain the numbers, we have to add **static modifier** in code.

Attach 1) code\_screenshot



Attach2) program running\_screenshot



 Attach) program running

A.

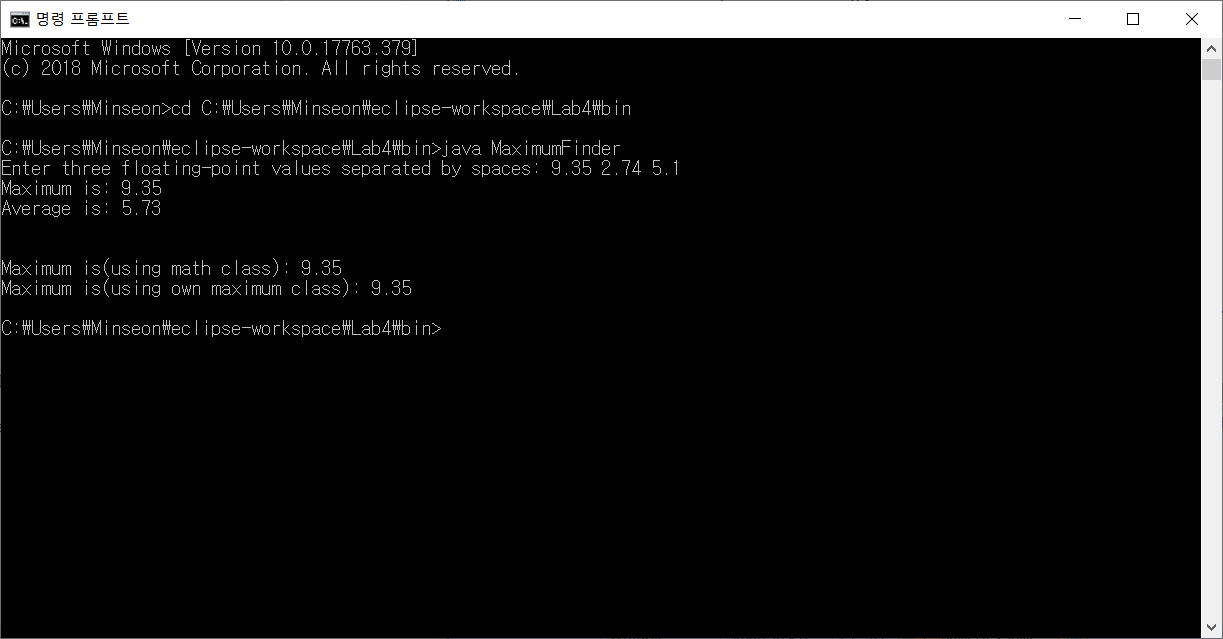
If I remove static modifier in maximum() method, the programming is not working and has a error. The error is because of static modifier.

In main() method, this method already set static modifier and was declared double variable that store maximum() result.

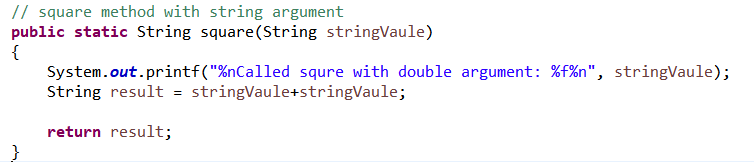
The double variable in static method is automatically regarded to static variable.

In order to use non-static method in static main method, change non-static method to static method.

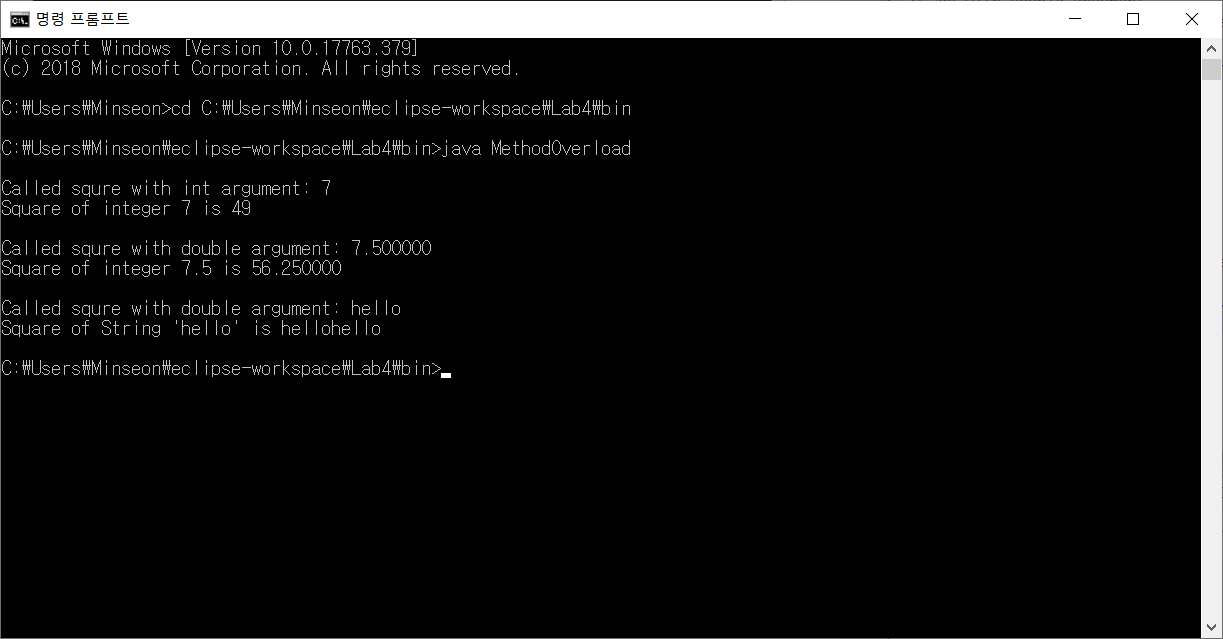
B.



5. A.

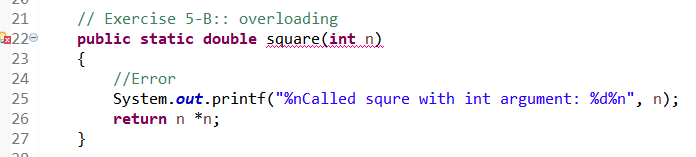
 Attach1) code

Attach2) command program running



B.

First, I add square() method (return type : **double** / parameter type : **int**) in code(MethodOverload.java).



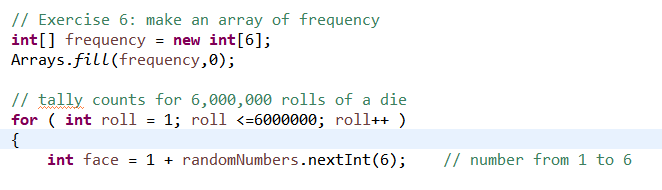
But, in this class, they already same method named square().

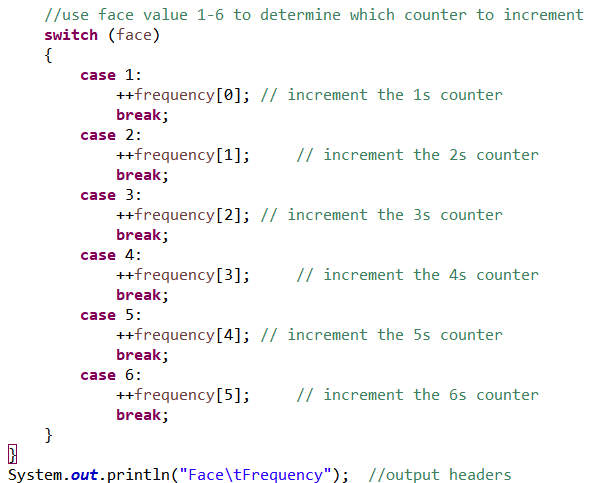
This method has double parameter and return double type.

In order to use overloading system, we have to keep some rules such as same method name, same amount of parameter, same type of parameter. But return type of method is not related to method overloading.

In our code, they have two same named methods and that methods are different return type values(one is int, another is double). But they have same type of parameter(int value). So they can’t be overload method.

6.

 Attach1) revised code



Attach2) command program running

