OBJECT-ORIENTED SYSTEMS DESIGN (Lab5-1)

Heejin Park

Hanyang University

5-1(Display 5.1)

Create classes *RoundStuff* and *RoundStuffDemo* defined as follows.

[*RoundStuff*]

1. Create a variable **private static final double** *PI* and store 3.14159 in it.

2. Create a static method **public static double area(double radius):**

   It returns the area of a circle of the given radius.

3. Create a static method **public static double volume(double radius):**

   It returns the volume of a sphere of the given radius.

5–1

[*RoundStuffDemo*]

Write a class *RoundStuffDemo* that prints the input and output below using the class *RoundStuff*.

<input and output>

```
Enter radius:
2
A circle of radius 2.0 inches
has an area of 12.56636 square inches.
A sphere of radius 2.0 inches
has a volume of 33.51029333333333 cubic inches.
```

5-2 (display 5.3)

Create a class *Temperature*.

(For some methods, **public, private** and/or **return types** are omitted.)

[*Temperature*]

1. Create a instance variable **private double** *degrees.*

2. Create two overloaded constructors *Temperature()'s* whose parameters are as follows.

() : Initialize such that the degrees is 0.

(double degrees) : Initialize such that the instance variable *degrees* is the parameter *degrees*.

5-2

3. Create a method **setDegrees(double degrees):** set the instance variable *degrees* with the parameter *degrees*.

4. Create a method **getDegrees():** It returns *degrees* .

5. Create a method **toString():** It returns *degrees* + “C”.

6. Create a method **equals**(Temperature **otherTemperature):** It returns true if the instance variable *degrees* is the same as *otherTemperature.degrees* and false otherwise.

7. Create a static method **toCelsius(double degreesF):**

   It returns 5*(*degreesF* – 32)/9.

5-2

8. Write a **main** method run it with the input and output.

<input and output>

```
Enter degrees Fahrenheit:
212
Equivalent Celsius temperature is 100.0C
```

5-3 (display 5.4)

Create a class *TurnTaker* and *StaticDemo.*

[*TurnTaker*]

1. Create a static variable and 2 instance variables as follows.

   private static int *turn* : Initialize it 0.

   private int *myTurn.*

   private String *name.*

5-3

2. Create two overloaded constructors *TurnTaker()'s* whose parameters are as follows.

(String **name,** int **turn**) : Initialize such that the instance variable **name** is the same as the parameter **name**. If *turn* is greater than or equal to 0, stores *turn* into *myTurn.* Otherwise, print out "Fatal Error" and exit.

() : Initialize such that the *name* is "No name yet" and *myTurn* is 0.

5-3

3. Create a method **getName()**: It returns *name.*

4. Create a static method **getTurn()**: It increases *turn* by 1 and returns *turn.*

5. Create a method **public boolean isMyTurn()**: It returns true if the instance variable *myTurn* is the same as the static variable *turn* and false otherwise.

5–3

[*StaticDemo*]

Write a class *StaticDemo* that outputs below using the class *TurnTaker*.

<output>

```
Turn = 1
Love from Romeo
Turn = 2
Turn = 3
Love from Juliet
Turn = 4
```

5–4 (display 5.5)

Create a class *InvocationCounter.*

[*InvocationCounter*]

1. Create a **private static int** *numberOfInvocations = 0*.

2. Create a method **public void demoMethod():** It increases *numberOfInvocations* by 1.

3. Create a method **public void outPutCount()** : It increases *numberOfInvocations* by 1and prints out “Number of invocations so far = ” + *numberOfInvocations* .

4. Create a method **public static int numberSoFar()** : It increases *numberOfInvocations* by 1 and returns *numberOfInvocations.*

5-4

5. Write a **main** method that outputs below

– Create *InvocationCounter* object1, then run **demoMethod()** 5 times and run **outputCount()**.

– Create *InvocationCounter* object2, then run **demoMethod()** and **outputCount()** 5 times.

<output>

```
Number of invocations so far = 6
Number of invocations so far = 8
Number of invocations so far = 10
Number of invocations so far = 12
Number of invocations so far = 14
Number of invocations so far = 16
Total number of invocations = 17
```

5–5 (Display 5.7)

Create a class *RoundStuff 2* that combined the class *RoundStuff* and the class *RoundStuffDemo.*

The **main** method should be placed inside the class *RoundStuff 2*.

You should use the constant *PI* defined in the class *Math* instead of including your own definition of *PI*.

The input and output are as follows.

<input and output>

```
Enter radius:
2
A circle of radius 2.0 inches
has an area of 12.566370614359172 square inches.
A sphere of radius 2.0 inches
has a volume of 33.510321638291124 cubic inches.
```

5-6 (Display 5.9)

Create a class *StringProcessor* that prints the input and output below.

You should use the static method *toUpperCase()* of the class *Character*.

<input and output>

```
Enter a one line sentence:
is you is OR is you ain't my BABY?
The revised sentence is:
Is you is or is you ain't my baby?
```