Workshop #5: Polymorphism

Upon successful completion of this workshop, you will have demonstrated the abilities to:

- Practice polymorphism.
- Understand the principles and the use of abstract classes and interfaces in Java
- Describe to your instructor what you have learned in completing this workshop.

To complete this task you should read and study the lecture Polymorphism

1) Write a class named MyPolymorphism, which implements all methods in the interface IPolymorphism as below:

<<interface>> IPolymorphism
+ f1(str:String):int + f2(str:String):String

Where:

- f1(str:String):int - count and return number of words containing at least 1 odd digit.
- f2(str:String):String - return the string s, obtained by replacing words containing at least 2 digits in str with the string "XYZ".

The program output might look something like:

1. Test f1() 2. Test f2() Enter TC (1 or 2): 1 Enter a string: a a1 b2 c34 d6 OUTPUT: 2	1. Test f1() 2. Test f2() Enter TC (1 or 2): 2 Enter a string: a a1 b2 c34 d6 OUTPUT: a a1 b2 XYZ d6

2) Write a class named MyPolymorphism, which implements all methods in the interface IPolymorphism as below:

<<interface>> IPolymorphism
+ f1(str:String):int + f2(str:String):double

Where:

- f1(str:String):int - find and return the number of characters of the longest word in str.
- f2(str:String):double - return the average value of even digits in the string.

The program output might look something like:

1. Test f1() 2. Test f2() Enter TC (1 or 2): 1 Enter a string: ab a4c abcba 14368 uv	1. Test f1() 2. Test f2() Enter TC (1 or 2): 2 Enter a string: ab a4c abcba 14368 uv

OUTPUT: 5	OUTPUT: 5.5

3) Write a class named MyPolymorphism, which implements all methods in the interface IPolymorphism as below:

<<interface>> IPolymorphism
+ f1(str:String):int + f2(str:String):void

Where:

- f1(str:String):int - count and return the number of words containing at least 2 digits.
- f2(str:String):void - return the string s, obtained by replacing words containing at least one even digit in str with the string "ABC".

The program output might look something like:

1. Test f1() 2. Test f2() Enter TC (1 or 2): 1 Enter a string: o p23 2q4 c5 6d 123b OUTPUT: 3	1. Test f1() 2. Test f2() Enter TC (1 or 2): 2 Enter a string: o p23 2q4 c5 6d 123b OUTPUT: o ABC ABC c5 ABC ABC

4) Write a class named MyPolymorphism, which implements all methods in the interface IPolymorphism as below:

<<interface>> IPolymorphism
+ f1(str:String):void + f2(str:String):int

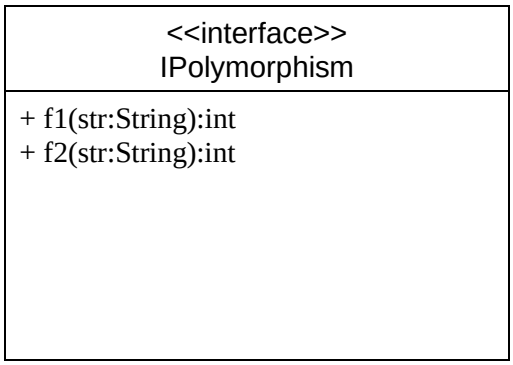
Where:

- f1(str:String):void - find and return words containing at least one even digit.
- f2(str:String):int - return the sum of all odd digits in str.

The program output might look something like:

1. Test f1() 2. Test f2() Enter TC (1 or 2): 1 Enter a string: ob 2c 34d 1e b3 f8 OUTPUT: 2c 34d f8	1. Test f1() 2. Test f2() Enter TC (1 or 2): 2 Enter a string: ob 2c 34d 1e b3 f8 OUTPUT: 7

5) Write a class named MyPolymorphism, which implements all methods in the interface IPolymorphism as below:



Where:

- f1(str:String):int - count and return the number of characters that are letters in str.
- f1(str:String):int
  - if the input string is a number
    - convert the string to an integer
    - return the square of the integer
  - else
    - return the length of the string

The program output might look something like:

```
1. Test f1()
2. Test f2()
Enter TC (1 or 2): 1
Enter a string:
12@ 45a? 67bc 8deg
OUTPUT:
6
```

```
1. Test f1()
2. Test f2()
Enter TC (1 or 2): 2
Enter a string:
16
OUTPUT:
256
```

```
1. Test f1()
2. Test f2()
Enter TC (1 or 2): 1
Enter a string:
12@
OUTPUT:
0
```

```
1. Test f1()
2. Test f2()
Enter TC (1 or 2): 2
Enter a string:
12@ 45a? 67bc 8deg
OUTPUT:
18
```