Assignment II- Inheritance and Polymorphism

Assignment Submission Deadline: 12-05-2021

1. Create an abstract class **Sequence** with an instance variable **number** of double type. Add a no-argument constructor that set **number** variable to zero and a parameterized constructor that take **number** variable's value as parameter and set the value to it. Design appropriate setter method for the instance variable. Add two abstract method **Term(int n) and tostring (int n)** in this abstract class. Now answer/perform the following tasks:
    a) Can we create an instance of Sequence class? Give proper reason of your answer.
    b) Create **MySequence** class that extends **Sequence** class. The main target of this **MySequence** class is to print a sequence of numbers that uses Fibonacci series. Override **tostring(int n)** method that will return a string that represents the series upto n^{th} term and **Term(int n)** method will return the value of n^{th} term where n=1, 2, 3 …n. You have to calculate values of each term using **Term(int n)** method.
    Example:
    n=4
    Output of tostring(4)= 0 , 1 , 1, 2(calculated by Term(1), Term(2)..Term (4))

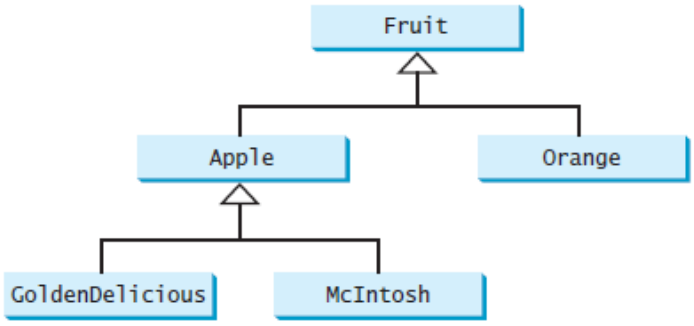
    c) Create another new class **Series** that extends **Sequence**. Design a method **sum(int n)** that finds the sum of the first n terms in the series. Override **toString(int n)** that will return a string that represents the series using the formula **((n - 1) + number)* n** upto n^{th} term and **Term** method that will return the value of n^{th} term. It will have a method **sumString(int n)** that will make a String of the form: t1 + t2 + t3 + ... + tn = Sn if n>4 and make a String of the form: t1 + t2 + t3 + ... + tn if n<=4 .
    What will be the value of **number** attribute that will calculate the first term in the above Sequence be 4. Print the first five terms in that sequence. Find and print the sum of the first five terms in the sequence
2. A very well-known topic of operating system is command. We can use cut, paste, undo, save command for various purpose. To visualize the effect of different commands create a **Command** Interface that has a method Execute (). Also declare two string variables **buffer** and **doc** on which different command will be performed.
    a) Implement a Save command by designing **Save** class such that new Save (n) creates a command whose Execute() method can save any string to the **doc** string variable.
    b) Implement Copy command by designing **Copy** class such that new Copy (n) copies a part of characters from the **doc** string, put them in a string variable called **buffer**. Inside the copy class design a **Cut (n, m)** method that removes the contents of **doc** string variable from n to m-1 length.

c) Implement a Paste command by designing **Paste** class such that new Paste(n) creates a command whose execute method inserts the contents of **buffer** at position n in **doc** variable, and its undo method reverses the action.

Now perform following actions using above class:

doc = ”abcdefgh”
Cut(3, 5);// doc = ”abcfgh” and buffer = ”de”
Paste(0); // doc = ”deabcfgh” and buffer = ”de”
undo(); // doc = ”abcfgh” and buffer = ”de”
undo(); // doc = ”abcdefgh” and buffer = ”de”

3. Suppose that Fruit, Apple, Orange, GoldenDelicious, and McIntosh are defined in the following inheritance hierarchy:



Assume that the following code is given:

```
Fruit fruit = new GoldenDelicious();

Orange orange = new Orange();
```

Answer the following questions with a brief justification:

a) Is *fruit* instanceof *GoldenDelicious*?
b) Suppose the method *makeOrangeJuice* is defined in the Orange class. Can *orange* invoke this method? Can *fruit* invoke this method?
c) Is the statement `GoldenDelicious p = new Apple()` legal?

4. Answer the following questions:
a) Explain the sentence: an interface is also a type.
b) Write a `compareTo` method that compares one `Person` object to another. Assume that a `Person` consists of a first name, last name, and age.
c) Define interface inconsistency?
d) What is the difference between an abstract class and an interface?
e) When would one choose to use an abstract class rather than an interface and vice versa?