CSC230 FA2014

Project #3

Inherited Sequences

The purpose of this program is to apply the computing principles of

- inheritance
- class design
- polymorphism

Write a program called SequenceGenerator.java. In this program, you will write several classes that model different types of sequences. All of these classes will be written in the file SequenceGenerator.java. If you need help writing multiple classes in the same file, refer to the video in Blackboard.

Recall that

- An arithmetic sequence is a sequence that has a constant difference between terms. If you know the first term in the sequence and the constant difference, you can find any term in the sequence. The following arithmetic sequence has a first term of 0 and a common difference of 0.5:

  0, 0.5, 1.0, 1.5, 2.0, …

- A geometric sequence is a sequence that has a constant ratio between terms. If you know the first term in the sequence and the constant ratio, you can find any term in the sequence. The following geometric sequence has a first term of 1 and a common ratio of -2:

  1, -2, 4, -8, 16, …

- The Fibonacci sequence is a sequence where each term in the sequence is the sum of the two terms that precede it. The first two terms in the Fibonacci sequence are 0 and 1. For this exercise, we will allow the user to define the first 2 terms, and subsequent terms will be determined by adding the 2 preceding terms. The following “Fibonacci-like” sequence as a first term of 1 and a second term of 0.5:

  1, 0.5, 1.5, 2.0, 3.5, 5.5, 9.0, …

Your assignment:

1. Write an *abstract class* called Sequence. This class defines properties and methods that all sequences will have. This class should include the following:
   a. A PRIVATE field for the firstTerm in the sequence (a double)
   b. A no-arg constructor that sets the firstTerm to 1
   c. A constructor that sets the firstTerm using a parameter value
   d. A method that returns the first term in the sequence
   e. A toString method that you will define later, after all other classes have been written
   f. An abstract method, getNthTerm, that takes an integer argument specifying the desired term
   g. A getNthSum method that takes an integer argument specifying the number of terms to sum. This method calculates and returns the sum of the desired number of terms.

2. Write an ArithemticSequence class that extends the Sequence class. This class should include the following:
   a. A PRIVATE field for the common difference
   b. A no-arg constructor that initializes the first term to 1 and the common difference to 0
   c. A constructor that uses parameters to initialize the first term and common difference
   d. An implementation for the abstract method defined in the Sequence class.

3. Write a GeometricSequence class that extends the Sequence class. This class should include the following:
   a. A PRIVATE field for the ratio
   b. A no-arg constructor that initializes the first term to 1 and the ratio to 1.
   c. A constructor that uses parameters to initialize the ratio and the first term.
   d. An implementation for the abstract method defined in the Sequence class.

4. Write a FibonacciSequence class that extends the Sequence class. This class should include the following:
    a. A PRIVATE field for the second term in the sequence
    b. A no-arg constructor that initializes the first term to 1 and the second term to 1
    c. A constructor that uses parameters to initialize the first and second terms
    d. An implementation for the abstract method defined in the Sequence class.

5. Add a toString method in the Sequence class. This method should display the first 10 terms in a sequence in a comma delimited list.

6. Write a main method that does the following:
    a. Creates a *SEQUENCE reference variable*
    b. Reads from the input file SEQ.in, which is organized as follows:
        i. The first entry on a line is a letter that represents the type of sequence
        ii. The second entry on a line is a numeric value that represents the first term in the sequence
        iii. The third entry on a line is a numeric value that represents either the common difference, ratio or second term, depending upon the type of sequence.

    c. Use toString and other methods to produce output in the following format. This output was produced using a file with this data:
    A 3 5
    G 1 2
    F 0 4

```
The first 10 terms in the sequence are:
3.0, 8.0, 13.0, 18.0, 23.0, 28.0, 33.0, 38.0, 43.0, 48.0
sum of the first 5 terms = 65.0

The first 10 terms in the sequence are:
1.0, 2.0, 4.0, 8.0, 16.0, 32.0, 64.0, 128.0, 256.0, 512.0
sum of the first 5 terms = 31.0

The first 10 terms in the sequence are:
0.0, 4.0, 4.0, 8.0, 12.0, 20.0, 32.0, 52.0, 84.0, 136.0
sum of the first 5 terms = 28.0
```

Caveats

- You must declare a Sequence reference variable in the main method.
  No Arithmetic, Geometric, or Fibonacci reference variables allowed.

- Statements that output the terms and the sum of terms occur *only once* in your main method

- Your toString returns a String that contains the first 10 terms in the sequence and *nothing else*.

- You may not implement the Fibonacci getNthTerm recursively. Doing so will cost you 10 points.