Object-Oriented Programming

Exercise series 2

Exercise 1

A *binomial coefficient* $\binom{n}{k}$ is a positive integer, depending on two parameters n and k such that $0 \leq k \leq n$, defined by the following formula :

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

A simple way of calculating binomial coefficients consists in using *Pascal's triangle*, which is a triangular array of binomial coefficients defined recursively as

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \quad \text{if } n > 0 \text{ and } k > 0$$
$$\binom{n}{0} = \binom{n}{n} = 1,$$

where n is the index of a line and k is the index of a column. The first ten lines of Pascal's triangle are shown below :

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
```

Your task consists in solving in an object-oriented way the problem of computing efficiently binomial coefficients, using Pascal's triangle. To do so, answer the following questions :

— Do you need to manage several Pascal's triangles represented as separate objects ?

— Do you need to use instance variables, class variables, or both ? Why (not) ?

— What kind of data structure can you use to manage the coefficients ?

Exercise 2

Sketch a `Pascal` Java class complying with the design that you have obtained for the previous exercise. In particular, define its variables and its interface. Additionally, answer the following questions :

— Assume that a user of your `Pascal` class can request the computation of arbitrary binomial coefficients, possibly for large values of n. How do you compute a binomial coefficient for given values of n and k ? Can you take advantage of previous computations in order to speed up future requests ?

— What kind of variable are you going to use to store a coefficient ?

— How do you handle the computation of a coefficient for invalid values of n and k ?

— What is the appropriate visibility for each component of your `Pascal` class ? (Justify your answer.)

Suggestion : In order to be able to visualize your computations, consider adding a method for displaying the first lines of Pascal's triangle in text format.

Exercise 3

Implement your `Pascal` class, in line with your answers to exercises 1 and 2. Then, write a `main()` method in a side class (named `BinomialCoefficients` for instance) and test your solution.