

Test č. 1, úloha č. 3

Naprogramujte triedu `BinomialTheorem` v nepomenovanom balíku, ktorá bude reprezentovať a vyhodnocovať binomickú vetu:

$$(a + b)^n = \sum_{k=0}^n \binom{n}{k} a^k b^{n-k},$$

kde $\binom{n}{k}$ je binomický koeficient, pre ktorý platí

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{\prod_{j=0}^{k-1} (n-j)}{k!}.$$

Trieda `BinomialTheorem` má poskytovať:

- Konštruktor

```
public BinomialTheorem(int a, int b, int n),
```

ktorý vytvorí objekt reprezentujúci $(a + b)^n$. Môžete predpokladať, že a , b a n sú nezáporné celé čísla.

- Metódu `public long evaluate(int from, int to)`, ktorá vypočíta sumu

$$\sum_{k=\text{from}}^{\text{to}} \binom{n}{k} a^k b^{n-k}.$$

Môžete predpokladať, že $\text{from} \leq \text{to}$, $n!$ sa zmestí do `long` a všetky kladné celé čísla menšie alebo rovné $(a + b)^n$ sa zmestia do `double` s plnou presnosťou. Hodnoty from a to môžu byť mimo rozsahu 0 až n (Pre k mimo rozsahu 0 až n je $\binom{n}{k}$ definované 0).

- Metódu `public String toString()`, ktorá vráti rozpísanú sumu z vety podľa nasledujúceho formátu:

$$\binom{n}{0} * a^0 * b^n + \binom{n}{1} * a^1 * b^{n-1} + \dots + \binom{n}{n} * a^n * b^0$$

Okolo znakov `+` je jedna medzera, okolo znakov `*` a `^` nie je.

Testovač bude vytvárať inštancie vašej triedy `BinomialTheorem` a volať ich metódy. Príklad takejto postupnosti volaní aj s očakávanými výstupmi možno nájsť nižšie.

Okrem správnosti sa budú hodnotiť aj niektoré ďalšie aspekty odovzdaného riešenia – predovšetkým dodržiavanie elementárnych zásad objektovo orientovaného programovania (hlavne princípu zapuzdrenia) a rešpektovanie konvencií jazyka Java.

Na testovač odovzdávajte súbor `BinomialTheorem.java` obsahujúci zdrojový kód vašej triedy.

Príklad. Predpokladajme, že inštancia triedy `BinomialTheorem` bola vytvorená pomocou volania

```
BinomialTheorem bt = new BinomialTheorem(2, 3, 4);
```

Zavolanie `System.out.print(bt)` vypíše

```
1*2^0*3^4 + 4*2^1*3^3 + 6*2^2*3^2 + 4*2^3*3^1 + 1*2^4*3^0
```

Volania `bt.evaluate(0, 4)` a `bt.evaluate(-1, 3)` potom postupne vrátia čísla 625 a 609.