Cvičenia č. 3, úloha č. 3

Napíšte triedu `ExtendedBigInteger` v nepomenovanom balíku rozširujúcu triedu `BigInteger` z balíka `java.math`. Táto trieda bude reprezentovať rovnaké neobmedzene veľké celé čísla ako trieda `BigInteger`, ale navyše bude vedieť počítať k-te klesajúce faktoriály. V triede `ExtendedBigInteger` implementujte:

- Konštruktor, ktorý vytvorí inštanciu triedy `ExtendedBigInteger` z reťazca typu `String`, ktorý dostane ako argument (pri implementácii využite rovnaký konštruktor triedy `BigInteger`).
- Konštruktor, ktorý vytvorí inštanciu triedy `ExtendedBigInteger` z inštancie triedy `BigInteger`.
- Metódu `public ExtendedBigInteger fallingFactorial(int k)`, ktorá na výstupe vráti *k-ty klesajúci faktoriál* čísla reprezentovaného inštanciou triedy `ExtendedBigInteger`, pre ktorú bola metóda volaná. Môžete predpokladať, že k je prirodzené číslo (môže to byť aj nula). Pod k-tym klesajúcim faktoriálom čísla x pritom pre $x \in \mathbb{R}$ a $k \in \mathbb{N}$ rozumieme hodnotu

  $$x^{\underline{k}} := x(x-1)\dots(x-k+1) = \prod_{j=0}^{k-1}(x-j).$$

  Špeciálne pre $k = 0$ dostávame prázdny súčin, ktorého hodnota je 1.

Nepôjde tu o plnohodnotné rozšírenie funkcionality triedy `BigInteger`: trieda `ExtendedBigInteger` bude poskytovať podstatne menšiu sadu konštruktorov, než trieda `BigInteger` a mnohé z jej zdedených metód budú namiesto inštancií triedy `ExtendedBigInteger` vracať na výstupe inštancie triedy `BigInteger`. Úlohu teda treba chápať iba ako cvičenie zamerané na mechanizmus dedenia.
Táto úloha sa dá vyriešiť bez toho, aby ste v triede `ExtendedBigInteger` deklarovali akékoľvek premenné. Špeciálne tam nedeklarujte žiadne premenné typu `BigInteger` – všetko podstatné získate dedením.

Na testovač odovzdávajte súbor `ExtendedBigInteger.java` obsahujúci zdrojový kód vašej triedy. Testovač bude načítavať rôzne dvojice čísel $n \in \mathbb{Z}$ a $k \in \mathbb{N}$, pre ktoré bude pomocou vašej triedy `ExtendedBigInteger` počítať hodnotu $n^{\underline{k}}$.

Príklad vstupu:

```
4 2
5 3
10000000000000000000 2
```

Príklad výstupu:

```
12
60
99999999999999999990000000000000000000
```