

Test č. 2, úloha č. 3

Napíšte generickú triedu `RepetitiveIterator<E>` (v nepomenovanom balíku) implementujúcu rozhranie `Iterator<E>` a poskytujúcu konštruktor, ktorého argumentmi sú inštancia `iterator` typu `Iterator<E>` a inštancia `repetitions` typu `List<Integer>` (v tomto poradí).

Inštancia vytvorená týmto konštruktorom bude reprezentovať iterátor, ktorý sa správa podobne ako iterátor `iterator` z argumentu konšuktora, avšak prvky prechádzané iterátorom `iterator` zakaždým vracia aj niekoľkokrát za sebou. *Presnejšie*: predpokladajme, že vstupný zoznam `repetitions` obsahuje nezáporné celé čísla a_0, \dots, a_{n-1} . Prvok, ktorý je iterátorom `iterator` vrátený ako k -ty v poradí (počítané od 1), potom váš iterátor vráti v presne a_r po sebe idúcich volaniach metódy `next`, kde r je zvyšok po delení čísla $k - 1$ číslom n . To znamená, že prvý prvok iterátora `iterator` vráti váš iterátor presne a_0 -krát, druhý prvok iterátora `iterator` vráti váš iterátor presne a_1 -krát, atď. Niektoré (aj všetky) spomedzi čísel a_0, \dots, a_{n-1} môžu byť aj rovné nule – príslušné prvky vracané pôvodným iterátorom potom bude váš iterátor preskakovať.

V prípade, že metóda `next` už podľa horeuvedenej špecifikácie nemôže vrátiť žiaden ďalší prvok, malo by jej volanie mať za následok vyhodenie výnimky typu `NoSuchElementException`. Metóda `hasNext` by sa mala správať konzistentne s metódou `next`.

Môžete predpokladať, že argumentmi konšuktora sú korektné vytvorený iterátor `iterator` a korektný zoznam `repetitions`, ktorý je dĺžky aspoň 1 a ktorý obsahuje iba nezáporné čísla.

Príklad. Predpokladajme, že argumentmi konšuktora sú iterátor `iterator` cez zoznam obsahujúci prvky

7, 1, 5, 2, 4, -3, -3, 0

a zoznam `repetitions` obsahujúci prvky

1, 2, 2.

Volania metódy `next` vášho iterátora by potom mali postupne vracieť prvky

7, 1, 1, 5, 5, 2, 4, 4, -3, -3, -3, 0, 0.

Pri pokuse o ďalšie volanie metódy `next` by mala vzniknúť výnimka typu `NoSuchElementException`.

Maximálny počet prvkov typu `E`, ktorý si budete pamätať v rámci jednej inštancie vašej triedy, by nemal závisieť od počtu prvkov prechádzaných iterátorom `iterator`. Špeciálne teda úlohu *neriešite* vygenerovaním všetkých budúcich výstupov metódy `next` už v konšuktore.

Na testovač odovzdávajte súbor `RepetitiveIterator.java` s kódom vašej triedy.