

Príprava na skúšku

Úloha č. 1: Iterátor

Napíšte generickú triedu `OutputRepeatingIterator<E>` (v nepomenovanom balíku) implementujúcu rozhranie `Iterator<E>` a poskytujúcu konštruktor, ktorého argumentmi sú (v tomto poradí):

- Inštancia iterator nejakej triedy implementujúcej rozhranie `Iterator<E>` (čiže iterátor cez prvky typu `E`).
- Celé číslo `int initialRepetitionCount`.
- Inštancia `f` nejakej triedy implementujúcej funkcionálne rozhranie `UnaryOperator<Integer>` z balíka `java.util.function`.

Môžete predpokladať, že `iterator != null`, `initialRepetitionCount >= 0` a `f != null`.

Inštancia triedy `OutputRepeatingIterator<E>` bude reprezentovať iterátor, ktorý sa správa podobne ako iterátor `iterator` z prvého argumentu konštruktora, avšak s tým rozdielom, že každý z prvkov vracaných metódou `next` iterátora `iterator` bude váš iterátor vracat niekoľkokrát po sebe. Presnejšie:

- Prvý prvok vrátený iterátorom `iterator` vráti váš iterátor presne `initialRepetitionCount`-krát (ak iterátor `iterator` vráti aspoň jeden prvok).
- Nech $n \geq 2$ a nech bol $(n-1)$ -vý prvok iterátora `iterator` vrátený vašim iterátorom k -krát (kde k môže byť aj nula). Ak v takom prípade existuje n -tý prvok iterátora `iterator`, vráti váš iterátor tento prvok presne `f.apply(k)`-krát.

To znamená, že počet po sebe idúcich vrátení každého nasledujúceho prvku pôvodného iterátora vašim iterátorom získame z počtu vrátení predchádzajúceho prvku pôvodného iterátora vašim iterátorom aplikáciou unárneho operátora `f` z tretieho argumentu konštruktora.

V prípade, že všetky prvky prechádzané iterátorom `iterator` už boli metódou `next` vášho iterátora vrátené toľkokrát, koľko je potrebné, mali by prípadné ďalšie volania metódy `next` vyústiť vo vyhodenie výnimky typu `NoSuchElementException`. Metóda `hasNext` by sa mala správať konzistentne s metódou `next`.

Príklad. Predpokladajme, že:

- Iterátor `iterator` z prvého argumentu konštruktora prechádza cez zoznam `[1, 2, 3, 4, 5, 6]`.
- Druhý argument konštruktora `initialRepetitionCount` má hodnotu `1`.
- Unárny operátor `f` z tretieho argumentu konštruktora je daný lambda výrazom

```
i -> {    if (i == 1) {  
            return 2;  
        } else {  
            return 1;  
        }  
    }
```

Volania metódy `next` vášho iterátora potom budú postupne vracat prvky `1, 2, 2, 3, 4, 4, 5, 6, 6`. Pri pokuse o ďalšie volanie metódy `next` by mala vzniknúť výnimka typu `NoSuchElementException`.

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