

Test č. 2, úloha č. 3

Napíšte generickú triedu `RepeatedElementsIterator<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>`; ide teda o iterátor cez prvky typu `E`.
- Zobrazenie `repetitions` priradujúce prvkom typu `E` hodnoty typu `Integer`. Tento argument je teda typu `Map<E, Integer>`.

Môžete predpokladať, že žiaden z týchto dvoch argumentov nie je `null`.

Inštancia triedy `RepeatedElementsIterator<E>` bude reprezentovať iterátor, ktorý sa správa podobne ako iterátor `iterator` z argumentu konštruktora, avšak prvky `e` typu `E`, ktoré sú kľúčmi v zobrazení `repetitions`, bude jeho metóda `next` vracaf v presne `repetitions.get(e)` po sebe idúcich volaniach a až následne sa iterátor posunie na ďalší prvok iterátora `iterator`. Hodnota `repetitions.get(e)` môže byť aj nulová, čo zodpovedá preskočeniu prvku iterátora `iterator` – môžete ale predpokladať, že je vždy nezáporná. Prvky `e` typu `E`, ktoré v zobrazení `repetitions` nie sú kľúčmi, bude iterátor vracaf práve raz. Pokiaľ už neexistuje žiaden ďalší prvok, ktorý by mohla takáto metóda `next` vrátiť, malo by dôjsť k vyhodneniu výnimky typu `NoSuchElementException`.

Metóda `hasNext` inštancie triedy `RepeatedElementsIterator<E>` by sa mala správať konzistentne s metódou `next` – mala by teda vracaf `true` práve vtedy, keď nasledujúce volanie metódy `next` nepovedie k vyhodneniu výnimky.

Prípadná zmena zobrazenia, ktoré bolo použité ako druhý argument konštruktora, by nemala mať vplyv na správanie vytvoreného iterátora ani v prípade, že sa udeje v priebehu iterovania. Môžete ale predpokladať, že počas iterovania nedochádza k modifikácii postupnosti objektov, cez ktoré prechádza iterátor `iterator` (v mnohých triedach, ako napríklad `ArrayList`, sa takáto súčasná zmena ani nepovoľuje a vedie k vyhodneniu výnimky `ConcurrentModificationException`; môžete ale predpokladať, že nenastane ani takáto situácia).

V inštancii vašej triedy si pamätajte najviac nejaký malý konštantný počet prvkov typu `E`. Špeciálne teda úlohu *neriešte* nájdením celej postupnosti vracaných prvkov hneď v konštruktore iterátora.

Na testovač odovzdávajte súbor `RepeatedElementsIterator.java` s kódom vašej triedy. Pri tvorbe kódu rešpektujte konvencie jazyka Java.

Príklad. Predpokladajme, že prvým argumentom konštruktora je iterátor cez `ArrayList` obsahujúci postupnosť prvkov `1, 2, 3, 4, 5, 2, 3, 4, 5` a druhým je inštancia triedy `HashMap` s kľúčmi `0, 2, 3, 4`, pričom obrazmi týchto kľúčov sú postupne hodnoty `2, 3, 0, 1`. Postupnosť výstupov metódy `next` potom bude

`1, 2, 2, 2, 4, 5, 2, 2, 2, 4, 5.`

Pri prípadnom ďalšom pokuse o volanie metódy `next` sa vyhodí výnimka typu `NoSuchElementException`.