

## Náhradný test, úloha č. 3

Napíšte generickú triedu `ModifiedIterator<E>` (v nepomenovanom balíku), ktorá bude implementovať rozhranie `Iterator<E>` a poskytovať konštruktor, ktorého argumentmi sú inštancia `iterator` typu `Iterator<E>` a inštancia `repeat` typu `Predicate<E>` (generické rozhranie z balíka `java.util.function`).

Inštancia vytvorená týmto konštruktorom bude reprezentovať iterátor, ktorý sa správa podobne ako iterátor `iterator` z argumentu konšuktora, avšak prvky prechádzané iterátorom `iterator` môže niekedy vrátiť aj dvakrát za sebou. *Presnejšie*: pre každý prvok `e` vrátený iterátorom `iterator` závisí počet vrátení tohto prvku vašim iterátorom od výstupu metódy `repeat.test(e)`. V prípade, že je výstupom booleovská hodnota `true`, vráti sa prvok `e` vašim iterátorom vo dvoch po sebe idúcich volaniach metódy `next`; v opačnom prípade sa prvok `e` vráti iba v jednom volaní metódy `next`.

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`.

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 *neriešate* vygenerovaním všetkých budúcich výstupov metódy `next` už v konštruktoze.

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