

## Cvičenia č. 5, úloha č. 6

Napíšte generickú triedu `PalilalicIterator<E>` (v nepomenovanom balíku) implementujúcu rozhranie `Iterator<E>` a poskytujúcu konštruktor, ktorého jediným argumentom je inštancia `iterator` nejakej triedy implementujúcej rozhranie `Iterator<E>`; argumentom konštruktora je teda iterátor cez prvky typu `E`. Inštancia triedy `PalilalicIterator<E>` bude reprezentovať iterátor, ktorý sa správa podobne ako iterátor `iterator` z argumentu konštruktora, avšak s tým rozdielom, že každý nasledujúci prvok vrátený iterátorom `iterator` vráti *v práve dvoch* po sebe idúcich volaniach metódy `next` a až následne sa posunie na ďalší prvok. Pôjde teda o iterátor, ktorý „palilalicky“ opakuje dvakrát všetky prvky, cez ktoré prechádza iterátor `iterator`.

V prípade, že už boli všetky prvky prechádzané iterátorom `iterator` vrátené metódou `next` vášho iterátora dvakrát, malo by ďalšie volanie metódy `next` vyústiť vo vyhodenie výnimky typu `NoSuchElementException`. Metóda `hasNext` by sa mala správať konzistentne s metódou `next`.

**Príklad.** Ak je argumentom konštruktora iterátor cez zoznam obsahujúci prvky `1, 2, 3, 4`, majú volania metódy `next` vášho iterátora postupne vracaať prvky `1, 1, 2, 2, 3, 3, 4, 4`. Pri pokuse o prípadné ďalšie volanie metódy `next` by mala vzniknúť výnimka typu `NoSuchElementException`.

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