

## Test č. 2, úloha č. 3

Stiahnite si priloženú kostru generickej triedy `NestedList<E, T extends Iterable<E>>` reprezentujúcej vnorený list (obdoba viacrozmerných polí). Typový parameter `E` je typ dát uložených v najvnútornejšej iterovateľnej štruktúre. Parameter `T` je typ vnútornej štruktúry, ktorá má vedieť iterovať cez prvky typu `E`. Trieda `NestedList` tiež implementuje rozhranie `Iterable<E>`, typ `T` teda môže byť „jednoduchý“ iterovateľný typ ako `ArrayList<E>` alebo aj ďalší `NestedList<E, ...>`.

V kostre sú už implementované dva konštruktory, prvý dostane na vstupe zoznam typu `T` a druhý variabilný počet premenných typu `T` (tento konštruktor je použitý v príklade). Inštancie typu `T` uloží do premennej `list`.

Implementuje telo metódy `iterator`, ktorá má vrátiť iterátor prechádzajúci cez prvky typu `E`. Tento iterátor musí poskytovať metódy `next` a `hasNext`; nie je potrebné, aby ste implementovali zvyšné metódy iterátora. Iterátor má najprv vrátiť všetky prvky nachádzajúce sa v prvej štruktúre v zozname `list`, následne všetky prvky v druhej štruktúre v zozname `list`,... Iterátor prejde všetky štruktúry v zozname `list` podľa štandardného poradia. Prvky typu `E` v štruktúre typu `T` prejde podľa iterátora, ktorý typ `T` poskytuje.

Potom, čo iterátor už prešiel všetky prvky, mali by ďalšie volania metódy `next` vyhodíť výnimku typu `NoSuchElementException`. Metóda `hasNext` by sa mala správať konzistentne s metódou `next`. Môžete predpokladať, že žiadne inštancie typu `T` uložené v zozname `list` nebudú `null`.

Na testovač odovzdávajte súbor `NestedList` obsahujúci vami doplnenú triedu.

**Príklad.** Predpokladajme, že inštancia triedy `NestedList` bola vytvorená pomocou volania

```
NestedList<Integer, NestedList<Integer, List<Integer>>> nestedList =  
new NestedList<>(new NestedList<>(List.of(1, 2, 3), List.of(4, 5, 6)),  
                new NestedList<>(List.of(7, 8, 9), List.of(10, 11, 12))  
                );
```

Volania metódy `next` iterátora tejto inštancie vrátia prvky `1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12`.