

Test č. 2, úloha č. 1

Napíšte generickú triedu `ComparableHashSet<E extends Comparable<E>>` v nepomenovanom balíku, ktorá bude rozširovať štandardnú triedu `HashSet<E>` z balíka `java.util` a súčasne implementovať rozhranie `Comparable<Set<E>>`. Pôjde o množiny navzájom porovnateľných prvkov, ktoré navyše bude možné porovnávať s ľubovoľnou množinou prvkov rovnakého typu. Na porovnávanie množín sa bude používať variant koreňového usporiadania vysvetlený nižšie.

Vaša trieda by mala poskytovať dva konštruktory, ktoré sa budú správať *rovnako* ako príslušné konštruktory triedy `HashSet<E>`:

- Konštruktor `public ComparableHashSet()`, ktorý vytvorí prázdnu množinu.
- Konštruktor `public ComparableHashSet(Collection<? extends E> c)`, ktorý vytvorí množinu obsahujúcu práve všetky prvky zoskupenia `c`.

Okrem verejných metód zdedených z triedy `HashSet<E>` by vaša trieda mala poskytovať metódu

```
public int compareTo(Set<E> o)
```

realizujúcu porovnanie danej inštancie vašej triedy s množinou `o`. Toto porovnanie by sa malo diať vzhľadom na usporiadanie, pri ktorom sa dvojica inštancií `a, b` rozhrania `Set<E>` porovná nasledovne:

- (i) V prípade, že množiny `a, b` neobsahujú rovnaký počet prvkov, vyhlási sa za menšiu v usporiadaní tá z nich, ktorá obsahuje menej prvkov.
- (ii) V prípade, že množiny `a, b` obsahujú rovnaký počet prvkov a súčasne sa tieto množiny nerovnejú, vyhlási sa menšiu v usporiadaní tá z nich, ktorá obsahuje najmenší prvok ich symetrickej diferencie.

Ak je argumentom metódy `compareTo` referencia `null`, vyhodí sa výnimka `NullPointerException`. V opačnom prípade metóda realizuje porovnanie danej inštancie triedy `ComparableHashSet` s množinou `o` z jej argumentu – a to vzhľadom na vyššie opísané usporiadanie. Výstupom metódy je teda záporné číslo, nula, resp. kladné číslo podľa toho, či je množina, pre ktorú sa metóda volá, menšia, rovnaká, alebo väčšia ako množina `o`. Volanie metódy `compareTo` by nemalo spôsobiť žiadnu zmenu v porovnávaných množinách.

Na testovač odovzdávajte súbor `ComparableHashSet.java` obsahujúci zdrojový kód vašej triedy. Rešpektujte konvencie jazyka Java a zásady objektovo orientovaného programovania. Tam, kde je to vhodné, použite anotáciu `@Override`.

Príklad 1. Nech `a` je inštancia typu `ComparableHashSet<Integer>` reprezentujúca množinu $\{1, 2, 3, 5, 6\}$. Nech je ďalej `o` inštancia typu `TreeSet<Integer>` reprezentujúca množinu $\{1, 2, 3, 4, 7\}$. Množina `a` je vzhľadom na uvažované usporiadanie väčšia ako množina `o`; výstupom metódy `a.compareTo(o)` teda bude kladné celé číslo.

Príklad 2. Nech `a` je inštancia typu `ComparableHashSet<Integer>` reprezentujúca množinu $\{1, 2, 3, 5, 6\}$. Nech je ďalej `o` inštancia typu `TreeSet<Integer>` reprezentujúca množinu $\{0, 1, 2, 3, 4, 7\}$. Množina `a` je vzhľadom na uvažované usporiadanie menšia ako množina `o`; výstupom metódy `a.compareTo(o)` teda bude záporné celé číslo.