

## Test č. 2, úloha č. 2

Nech  $f: K \rightarrow V$  je zobrazenie. Pripomeňme si, že *obrazom* množiny  $S \subseteq K$  nazývame množinu

$$f(S) = \{f(x) \mid x \in S\}$$

a *uzorom* množiny  $T \subseteq V$  nazývame množinu

$$f^{-1}(T) = \{x \in K \mid f(x) \in T\}.$$

Zobrazenie  $f: K \rightarrow V$  je *injektívne*, ak pre všetky  $x_1, x_2 \in K$  také, že  $f(x_1) = f(x_2)$  je  $x_1 = x_2$ .

Zobrazenia v Jave – čiže inštancie rozhrania `Map<K,V>` – sa od bežných zobrazení v matematike líšia predovšetkým tým, že sú *čiasťochne*, t. j. pre niektoré prvky typu `K` nemusí byť definovaný žiaden výstup. Každé zobrazenie má navyše vždy iba konečne veľa kľúčov. Vyššie zavedené pojmy ale prirodzene možno uvažovať aj v kontexte takýchto zobrazení.

Napíšte generickú triedu `ExtendedHashMap<K,V>` v nepomenovanom balíku, ktorá bude rozširovať štandardnú triedu `HashMap<K,V>` z balíka `java.util`. Zobrazenia reprezentované inštanciami vašej triedy budú navyše vedieť počítať obrazy množín prvkov typu `K`, vzory množín prvkov typu `V` a takisto budú obsahovať metódu umožňujúcu zistiť, či je dané zobrazenie injektívne.

Vaša trieda `ExtendedHashMap<K,V>` by mala poskytovať dva konštruktory, ktoré sa budú správať *rovnako* ako príslušné konštruktory triedy `HashMap<K,V>`:

- Konštruktor `public ExtendedHashMap()`, ktorý vytvorí prázdne zobrazenie.
- Konštruktor `public ExtendedHashMap(Map<? extends K, ? extends V> m)`, ktorý vytvorí zobrazenie na základe iného zobrazenia `m`.

Okrem verejných metód zdedených z triedy `HashMap<K,V>` by vaša trieda mala poskytovať nasledujúce tri metódy:

- Metódu `public Set<V> image(Set<K> set)`, ktorá nájde *obraz* množiny `set` prvkov typu `K` pri vašom zobrazení. Prvky množiny `set` môžu, ale nemusia byť kľúčmi vášho zobrazenia. Výstupom metódy `image` je tak množina všetkých prvkov `value` typu `V`, pre ktoré existuje prvok `key` množiny `set`, ktorý je kľúčom vášho zobrazenia a súčasne je obraz prvku `key` vo vašom zobrazení rovný `value`.
- Metódu `public Set<K> preimage(Set<V> set)`, ktorá nájde *vzor* množiny `set` prvkov typu `V` pri vašom zobrazení. Výstupom je teda množina všetkých prvkov `key` typu `K`, ktoré sú kľúčmi vášho zobrazenia a súčasne je obraz prvku `key` vo vašom zobrazení prvkom množiny `set`.
- Metódu `public boolean isInjective()`, ktorá vráti `true` práve vtedy, keď je zobrazenie reprezentované inštanciou vašej triedy injektívne, t. j. neexistujú dva rôzne kľúče tohto zobrazenia, ktoré sa v ňom zobrazia na ten istý prvok typu `V`.

Na testovač odovzdávajte súbor `ExtendedHashMap.java` obsahujúci zdrojový kód vašej triedy. Rešpektujte konvencie jazyka Java a zásady objektovo orientovaného programovania.

**Príklad.** Nech `m` je inštancia triedy `ExtendedHashMap<Integer,Integer>` reprezentujúca zobrazenie  $1 \mapsto 1, 2 \mapsto 1, 3 \mapsto 2, 4 \mapsto 4$ . Výstupom volania metódy `m.isInjective()` je v takom prípade `false`; výstupom volania `m.image(set)` pre množinu `set` obsahujúcu prvky `2,3` je množina obsahujúca prvky `1,2`; výstupom volania `m.preimage(set)` pre množinu `set` obsahujúcu prvky `1,2` je množina obsahujúca prvky `1,2,3`.