

Test č. 2, úloha č. 2

Stiahnite si kostru generickej triedy `ListComparator<E extends Comparable<E>>`, ktorá má reprezentovať komparátor na *zoznamoch* prvkov typu `E`, pričom na prvkoch typu `E` už predpokladáme existenciu prirodzeného usporiadania.¹ Trieda teda bude implementovať rozhranie `Comparator<List<E>>`.

Doprogramujte metódu `compare` inštancie tejto triedy tak, aby dvojicu inštancií typu `List<E>` porovnávala podobne ako reťazce pri koreňovom usporiadaní. Presnejšie:

- Komparátor by mal byť konzistentný s metódou `equals`, t. j. zoznamy `a`, `b` by si vzhľadom na usporiadanie komparátora mali byť navzájom rovné práve vtedy, keď `a.equals(b)`.
- Dvojica rôznych zoznamov sa najprv porovná podľa dĺžky. Kratší zoznam má byť vždy vyhodnotený ako menší, než dlhší zoznam.
- Dvojica rôznych zoznamov rovnakej dĺžky sa porovná podľa prvej dvojice rôznych prvkov. Ak je teda `i` najmenší index taký, že dané dva zoznamy obsahujú na pozícii `i` rôzne prvky, vyhodnotí sa ako menší ten zo zoznamov, ktorý na tejto pozícii obsahuje menší prvok (vzhľadom na prirodzené usporiadanie na `E`).

Dajte si pozor na to, aby vaša metóda `compare` vstupné zoznamy nemodifikovala; jej vstupmi koniec koncov môžu byť aj nemodifikovateľné zoznamy.

Na testovač odovzdávajte súbor `ListComparator.java` s kódom vašej triedy. Program na testovači bude pomocou vášho komparátora triediť rôzne zoznamy zoznamov a výsledky bude vypisovať na výstup.

¹Prvky typu `E` už teda vieme porovnávať pomocou metódy `compareTo`. Vašou úlohou bude napísať komparátor, ktorý bude určitým špecifickým spôsobom porovnávať *zoznamy* prvkov typu `E`.