Cvičenie 11 - Java 8, Lambda výrazy

Credits: Askar Gafurov

Všeobecné poznámky

- Je odporúčané si každú časť písať do samostatného súboru (aby sa predišlo kolíziám v názvoch metód).
- Pri každom stretnutí symbolu # je odporúčané vytvoriť samostatnú metódu (aby sa predišlo kolíziám v názvoch metód a premenných).
- Vzorove riesenie je tu: https://drive.google.com/file/d/1hIF2Ozpr52h89Rv7faJeSslRbP8TGjBL/view?usp=sharing

Ako zneprehľadniť svoj kód, časť 1

U: Vytvorte v rámci metódy main() triedu Burger s metódami double mass() a String type(). Vyskúšajte ju.

U: Vytvorte rozhranie Drink v rámci triedy Main s metódami double volume() a double alcoholContent() (objem nápoja, respektíve proporcia alkoholu v nápoji).

U: Vytvorte v rámci metódy main() triedu Beer, implementujúcu rozhranie Drink. Vyskúšajte ju.

U: Skúste vytvoriť anonymnú triedu Steak v rámci metódy main(). Podarilo sa?

U: Vytvorte anonymnú triedu Absinth implementujúcu rozhranie Drink v rámci metódy main().

U: Doplňte do rozhrania Drink defaultnú metódu double alcoholVolume(), ktorá vypočíta objem etanólu v danom nápoji (jednoduchym sucinom celkoveho objemu a proporcie alkoholu).

U: Spravte statický import metód triedy java.lang.Math, skuste pouzit nejaku z jeho metod bez odvolavania sa na objekt Math (napriklad cos).

U: Doplňte do rozhrania Drink statickú metódu String warning(), ktorá vracia štandardné upozornenie o škode alkoholu.

Ako zneprehľadniť svoj kód, časť 2

U*: Vyvtorte ArrayList A s prvkami od 1 do 9.

U: Vytvorte funkčné rozhranie UnaryFunction<InputType, OutputType> s abstraktnou metódou OutputType call(InputType arg) a nejakou defaultnou bezvýznamnou metódou. Skuste pridat dalsiu abstraktnu metodu. Co na to povie kompilator?

U: Vytvorte funkčné rozhranie BinaryFunction<InputType1, InputType2, OutputType> s abstraktnou metódou OutputType call(InputType1 arg1, InputType2 arg2).

U*: Vytvorte statickú generickú metódu apply, ktorá na vstupe dostane ArrayList a lambda funkciu, implementujúcu funkčné rozhranie UnaryFunction kompatibilného typu, ktorá bude vracať ArrayList, kde každý prvok je výsledkom aplikácie vstupnej lambda funkcie na prvok vstupného poľa.

U: Pomocou metódy apply vytvorte pole, obsahujúce dvojnásobky čísel z poľa A. Lambda funkciu implementujte pomocou anonymnej triedy.

U: Pomocou metódy apply vytvorte pole, obsahujúce štvorce čísel z poľa A. Lambda funkciu definujte spolu s typmi vstupných premenných.

U: Pomocou metódy apply vytvorte pole, obsahujúce odmocniny čísel z poľa A. Lambda funkciu definujte bez explicitných typov vstupných premenných.

U*: Vytvorte statickú generickú metódu select, ktorá na vstupe dostane ArrayList a lambda funkciu s kompatibilnými typmi, ktorá vráti pole, obsahujúce prvky vstupného poľa, pre ktoré vstupná funkcia vráti hodnotu true.

U: Pomocou metódy select vytvorte pole, obsahujúce prvky poľa A, ktoré sú väčšie ako 2.

U: Pomocou metódy select vytvorte pole, obsahujúce párne prvky poľa A.

U*: Vytvorte statickú generickú metódu acc, ktorá dostane na vstupe ArrayList a binárnu lambda funkciu s kompatibilnými typmi, ktorá postupne aplikuje vstupnú funkciu na prvé dva prvky, ďalej aplikuje ju na výsledok a tretí prvok poľa, následne aplikuje funkciu na predošlý výsledok a ďalší prvok, a tak ďalej, kým neprejde tak celé vstupné pole. Výsledok poslednej operácie vráti ako výstup.

U: Pomocou metódy acc spočítajte súčet všetkých prvkov poľa A.

U: Pomocou metódy acc spočítajte maximum spomedzi prvkov poľa A.

U*: Vytvorte statickú generickú metódu forEach, ktorá dostane na vstupe ArrayList a unárnu lambda funkciu s kompatibilnými typmi, ktorá na každom prvku vstupného poľa zavolá vstupnú funkciu.

U: Pomocou metódy forEach vypíšte na štandardný output na každý riadok samostatne hviezdičky, pričom počet hviezdičiek na k-tom riadku má zodpovedať k-tému prvku poľa A.

U: Pomocou metódy acc nájdite maximum spomedzi prvkov poľa A, pričom použite ako lambda funkciu referenciu na štandardnú metódu java.lang.Math.max().

U: Pomocou metódy apply vytvorte pole, obsahujúce textové reprezentácie prvkov z poľa A, pričom ako lambda funkciu použite referenciu na štandardnú funkciu Object.toString().

U*: Pomocou metódy apply vytvorte pole, obsahujúce odmocniny z prvkov poľa A, pričom ako lambda funkciu použite referenciu na štandardnú metódu java.lang.Math.sqrt (pozor, treba ju pretypovať).

U: Vytvorte premennú inc a priraďte do nej nejakú hodnotu. Pomocou metódy apply vytvorte pole, obsahujúce prvky z poľa A zvýšené o hodnotu v premennej inc.

Ako zneprehľadniť svoj kód, časť 3

U: Vytvorte pole A s reťazcami nad abecedou {a, b}.

U*: Vytvorte druhú implementáciu statickej metódy select, ktorá bude využívať štandardné funkčné rozhranie Predicate.

U: Pomocou novej metódy select vytvorte pole, obsahujúce iba prázdne reťazce z poľa A, pričom použite štandardnú metódu String.isEmpty().

U: Pomocou novej metódy select vytvorte pole, obsahujúce iba neprázdne reťazce z poľa A, príčom použite štandardnú metódu String.isEmpty() a defaultné metódy z funkčného rozhrania Predicate.

U: Vytvorte predikát "reťazec začína na písmeno a" a predikát "reťazec končí na písmeno b", pričom vytvorte ich ako lokálne triedy.

U: Pomocou metódy select a novovytvorených predikátov vytvorte pole, obsahujúce iba tie prvky z poľa A, ktoré začínajú na písmeno a.

U: Pomocou metódy select a novovytvorených predikátov vytvorte pole, obsahujúce iba tie prvky z poľa A, ktoré začínajú na písmeno a a končia na písmeno b, príčom lambda funkciu vytvorte pomocou defaultných metód funkčného rozhrania Predicate.

U*: Vytvorte novú statickú generickú metódu forEach, ktorá bude využívať štandardné funkčné rozhranie Consumer.

U: Vytvorte prázdne pole reťazcov B. Pomocou novovytvorenej metódy forEach naplňte ho prvkami z poľa A, pričom všetky výskyty písmena a nahradte písmenom x (použite štandardnú metódu String.replaceAll).

U#: Vytvorte pole čísel s desatinnou čiarkou A.

U: Vytvorte statickú metódu mySqrt(double x), ktorá vracia druhú odmocninu x, alebo null, ak je x záporné.

U*: Vytvorte novú statickú generickú metódu apply, ktorá bude používať štandardné funkčné rozhranie Function.

U: Pomocou novej metódy apply vytvorte pole B, obsahujúce výsledky aplikácie metódy mySqrt na prvky poľa A, zabalené do Optional.

U: Pomocou metódy foreach vypíšte všetky non-null prvky poľa B.