

Test č. 2, úloha č. 1

V tejto úlohe budeme pracovať so stromami výrazov pozostávajúcimi z konštánt určitého typu `E` a rôznych operátorov na `E`. Takéto stromy sú zovšeobecnením aritmetických stromov – tie by sme totiž dostali v prípade, že by sme za `E` zobrali `Integer` a ako operácie by sme uvažovali len bežné aritmetické operácie na celých číslach, ako napríklad sčítovanie, násobenie, alebo unárne mínus.

Priložený ZIP archív obsahuje balík `expressions` a v ňom:

- Hotovú abstraktnú generickú triedu `Node<E>` reprezentujúcu uzol opísaného stromu. V nej je deklarovaná jediná abstraktná metóda `evaluate`, ktorá vyhodnotí výraz reprezentovaný stromom zakoreneným v danom uzle.
- Kostry generických tried `BinaryNode<E>`, `UnaryNode<E>` a `NullaryNode<E>` rozširujúcich triedu `Node<E>` a určených na doprogramovanie. Tieto budú postupne reprezentovať uzly zodpovedajúce binárnym, unárnym a nulárnym operáciám na `E` (kde nulárna operácia na `E` je to isté ako konštanta typu `E`).

V triede `BinaryNode<E>` doprogramujte:

- Konštruktor

```
public BinaryNode(BinaryOperator<E> operator, Node<E> left, Node<E> right),
```

ktorý vytvorí uzol zodpovedajúci binárnemu operátoru `operator` s ľavým synom `left` a pravým synom `right`.

Binárny operátor je tu reprezentovaný ako inštancia typu `BinaryOperator<E>`, čo je rozhranie definované v balíku `java.util.function`. Každá inštancia tohto typu musí poskytovať metódu `apply` s dvoma argumentmi typu `E`, ktorá príslušnú operáciu vyhodnotí na daných vstupoch.

V prípade, že je niektorý z argumentov konštruktora rovný `null`, malo by dôjsť k vyhodneniu výnimky typu `IllegalArgumentException`.

- Metódu `public E evaluate()`, ktorá vráti hodnotu výrazu reprezentovaného podstromom zakoreneným v príslušnom uzle.

V triede `UnaryNode<E>` doprogramujte:

- Konštruktor

```
public UnaryNode(UnaryOperator<E> operator, Node<E> child),
```

ktorý vytvorí uzol zodpovedajúci unárnemu operátoru `operator` s jediným synom `child`.

Unárny operátor je tu reprezentovaný ako inštancia typu `UnaryOperator<E>`, čo je rozhranie definované v balíku `java.util.function`. Každá inštancia tohto typu musí poskytovať metódu `apply` s jediným argumentom typu `E`, ktorá príslušnú operáciu vyhodnotí na danom vstupe.

V prípade, že je niektorý z argumentov konštruktora rovný `null`, malo by dôjsť k vyhodneniu výnimky typu `IllegalArgumentException`.

- Metódu `public E evaluate()`, ktorá vráti hodnotu výrazu reprezentovaného podstromom zakoreneným v príslušnom uzle.

V triede `NullaryNode<E>` doprogramujte:

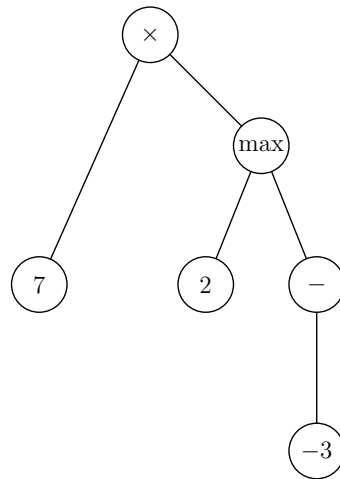
- Konštruktor

```
public NullaryNode(E value),
```

ktorý vytvorí list zodpovedajúci konštante `value`.

- Metódu `public E evaluate()`, ktorá vráti hodnotu výrazu reprezentovaného podstromom zakoreneným v príslušnom uzle.

Príklad. Uvažujme strom uzlov typu `Node<Integer>` schematicky znázornený na nasledujúcom obrázku.



Výstupom metódy `evaluate` pre koreň tohto stromu je potom číslo 21.

V prípade potreby môžete do tried určených na doprogramovanie pridať aj ďalšie premenné alebo pomocné metódy. Abstraktnú triedu `Node<E>` nemeňte. Pri tvorbe kódu rešpektujte konvencie jazyka Java, ako aj základné princípy objektovo orientovaného programovania (predovšetkým zapuzdrenie).

Na testovač odovzdávajte ZIP archív obsahujúci priečinok `expressions` a v ňom zdrojové kódy všetkých tried tohto balíka.