

JAVA programavimo kalba

Java parametrizuoti tipai (generics)

Parametrizuoti tipai (Generics)

Su parametrizuotomis (arba generic) klasėmis bei interfeisais jau esame susidūrę dirbdami su kolekcijomis:

- `Map<String, Integer> map;`

Jei pažiūrėsime į Map interfeiso aprašymą pamatysime:

- `public interface Map<K,V> { ...`

Toks užrašymas reiškia, kad norint panaudoti Map interfeisą būtina nurodyti dvi klases - vieną kaip raktą `K`, o kitą kaip reikšmę `V`.

Interfeiso apraše mums iš anksto nėra žinoma kokios tai klasės, todėl vietoj konkrečių klasių pavadinimų naudojame pavadinimus `K`, `V` ir pan.

Parametrizuoti tipai (Generics)

Parametrizuota klasė ar interfeisas apsirašo:

- `class Name<T1, T2, ... Tn> { ... };`

Jei ji praplečia kitą parametrizuotą klasę arba realizuoja parametrizuotą interfeisą tai užrašoma atitinkamai:

- `interface List<E> extends Collection<E> { ... }`

Paprastai tokiems parametrams naudojamos raidės:

- E - elementas (plačiai naudojama kolekcijose)
- K - raktas (key)
- V - reikšmė (value)
- T - tipas (type)
- N - skaičius (number)
- S, U, V - kiti parametrai

Parametrizuoti tipai (Generics)

- Aprašant parametrizuotus tipus galima jiems nurodyti ir tam tikrus reikalavimus, t.y. nurodant ką jie turi būti praplėtę ar realizavę
 - <T **extends** ClassA>
 - <T **implements** InterfaceB>

Parametrizuoti metodai (Generic methods)

Kaip kad galima nurodyti parametrizuotas klases, panašiai galima nurodyti parametrizuotus metodus:

- `public <T> int methodA(T a) { ... }`
- `public <T, S extends Number> T methodB(T a, S b) { ... }`

Parametrizuotas paveldimumas (inheritance)

Dirbant su parametrizuotais tipais svarbu suprasti kaip veikia paveldimumas.

Žinome, kad Integer yra subtipas Number, todėl galima **Number** tipo kintamajam priskirti **Integer** tipo reikšmę:

- Integer a = 1;
- Number b = a;

Tačiau **List<Integer>** nėra subtipas **List<Number>**

Nežinomas tipas

Aprašant parametrizuotas klases ar metodus arba nurodant kintamųjų ar parametrų tipus galima, vietoj konkretaus tipo, galima naudoti klaustuką - ?

Tai reiškia, kad mums nesvarbu arba mes nežinome koks tai tipas. Tokiu atveju mes negalėsime naudotis jokiais metodais. Tokį kintamąjį mes galėsime tik priskirti kitam.

- ? galima patikslinti naudojant extends, implements, bet galimas dar vienas patikslinimo būdas:
- <? super T> - reiškia nežinomas tipas, kuris yra supertipas tipui T

PECS principas

Dirbant su kolekcijomis svarbu laikytis PECS principo.

PECS - Producer (gamintojas) extends - consumer(naudotojas) super

Uždaviniai

1. Parašykite parametrizuotą (generic) klasę, kuri realizuoja E tipo objektų saugyklą. Taip pat padarykite kad juos galima iteruoti dviem būdais - ta pačia tvarka kaip ir įdėta ir atvirkštine tvarka.