

Užduotys: *Generics*

Nr. 1

Užduotis

Turime du masyvus - viename skaičiaus tipo elementai `{ 1, 2, 3 }`, kitame eilutės tipo elementai `{"Hello", "world"}`.

Užduotis: parašyti **vieną** metodą `printArray`, kuris gali atspausdinti visus abiejų turimų masyvų elementus. Metodas turi priimti ir masyvą su skaičiais, ir masyvą su eilutėmis.

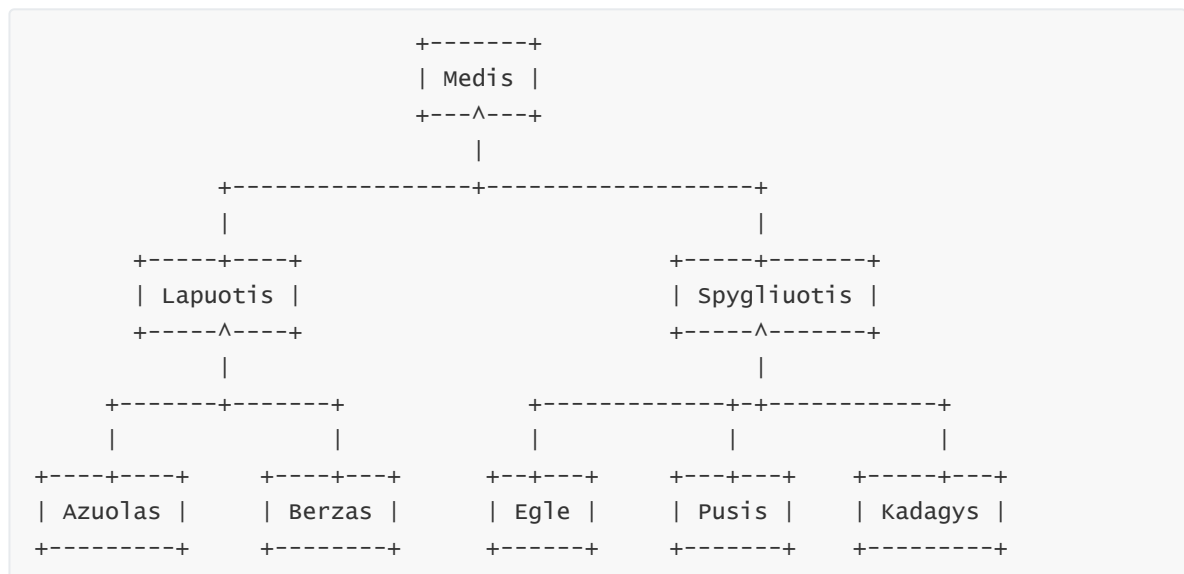
Tikėtinas rezultatas:

```
1
2
3
Hello
World
```

Nr. 2

Užduotis

Sukurti tokią klasių/interfeisų hierarchiją:



Kiekvienam objektui turime turėti galimybę iškviesti metodą `turi()`, kuris atspausdina informaciją apie objektą. Pavyzdžiui, jei sukurtume ąžuolo objektą `Azuolas azuolas = new Azuolas();` ir jam iškviestume metodą `azuolas.turi();` tada turi būti atspausdinta `Azuolas turi` lapus. Nekurti kiekvienam to paties `turi()` metodo kiekvienoje klasėje, o panaudoti paveldėjimo savybes.

Panaudojant *generics* atskiroje klasėje sukuri metodus:

- `ivairusMiskas(...)` - priima bet kokių medžių sąrašą

- `spygliuociuMiskas(...)` - priima bet kokių spgliuočių sąrašą
- `berzuMiskas(...)` - priima tik beržų sąrašą

Visų metodų logika panaši: atspausdina koks tai miškas ir kiekvienam medžiui iškviečia metodą `turi()`

Sukurti sąrašą, į kurį įdėti ivairių mežių ir iškvieisti metodą `ivairusMiskas(...)`.

Sukurti sąrašą, į kurį įdėti ivairių mežių ir iškvieisti metodą `spygliuociuMiskas(...)`.

Sukurti sąrašą, į kurį įdėti ivairių mežių ir iškvieisti metodą `berzuMiskas(...)`.

Tikėtinas rezultatas:

```
Ivairus miskas:
Berzas turi lapus
Azuolas turi lapus
Kadagys turi spyglius
Egle turi spyglius
Pusis turi spyglius

Spygliuociu miskas:
Kadagys turi spyglius
Egle turi spyglius
Pusis turi spyglius

Berzu miskas:
Berzas turi lapus
Berzas turi lapus
Berzas turi lapus
```

Nr. 3

Užduotis

Sukurkite klasę `Pora`, su klasės atributais `Raktas` ir `Reiksme`. Abu klasės atributai gali būti bet koks objektas.

Sukurkite klasę `Mapas` su vienu klasės atributu `sarasasPoru`, kuris bus sarasas, skirtas saugoti klasės `Pora` objektams. Klasė `Mapas` turi turėti metodą `idetiti`, kuriam per parametrus perdavus raktą ir reikšmę, iš jų sukonstruoja porą ir ją įdeda į map'ą. Taip pat reikalingas metodas `gauti`, kuriam per parametą perdavus raktą, gausime reikšmę pagal tą raktą iš sąrašo.

Sukurkite klasę `DnsServer`, kuri turi atributus `ip1` ir `ip2`.

Sukurkite enum `DnsProvider`, kuris saugos keletą reikšmių, pvz. `GOOGLE`.

Atskiroje klasėje `main` metode sukurti map'ą, kuris saugo `DnsProvider -> DnsServer` poras. Sukurti kitą map'ą, kuris saugo `String -> String` poras. Map'us užpildyti duomenimis ir pabandyti gauti reikšmes pagal raktus.

Panaudojimo pavyzdys:

```
Mapas<DnsProvider, DnsServer> dnsMapas = new Mapas<>();

dnsMapas.idetiti(GOOGLE, new DnsServer("8.8.8.8", "8.8.4.4"));
dnsMapas.idetiti(CLOUDFLARE, new DnsServer("1.1.1.1", "1.0.0.1"));
```

```
// ...

DnsServer googleDns = dnsMapas.gauti(GOOGLE);
System.out.println(googleDns);

Mapas<String, String> zodynoMapas = new Mapas<>();
zodynoMapas.ideti("Labas", "Hello");
zodynoMapas.ideti("Pasaulis", "world");
// ...

String labas = zodynoMapas.gauti("Labas");
System.out.println(labas);
```

programa atspausdina:

```
DnsServer{ip1='8.8.8.8', ip2='8.8.4.4'}
Hello
```

Nr. 4

Užduotis

Sukurkite klasę `ListOfNumbers` kuri saugo sąrašą `Double` tipo skaičių. Turi metodą `addNumber`, kuris perduotą skaičių įdeda sąrašą ir `getAverage`, kuris grąžina sąrašo skaičių vidurkį.

Kitoje klasėje sukurkite metodą `findwithMaxAverage`, kuris gali priimti iš anksto nežinomą skaičių `ListOfNumbers` tipo parametrų. Metodas turi surasti ir grąžinti objektą, kurio skaičių vidurkis yra didžiausias.

Pavyzdys:

```
ListOfNumbers listObject1 = new ListOfNumbers();
listObject1.addNumber(1.5d);
listObject1.addNumber(10d);
listObject1.addNumber(2.5d);

ListOfNumbers listObject2 = new ListOfNumbers();
listObject2.addNumber(0d);
listObject2.addNumber(1d);

ListOfNumbers listObject3 = new ListOfNumbers();
listObject3.addNumber(100d);

ListOfNumbers result1 = findwithMaxAverage(listObject1, listObject2,
listObject3);
ListOfNumbers result2 = findwithMaxAverage(listObject1, listObject2);

System.out.println(result1);
System.out.println(result2);
```

atspausdina rezultata

```
ListOfNumbers{numbers=[100.0]}
ListOfNumbers{numbers=[1.5, 10.0, 2.5]}
```

Nr. 5

Užduotis

Sukurkite klasę `SuperPair` su trimis *generic* tipais taip, kad trečiasis tipas galėtų būti tik koks nors `Number` skaičius. Pirmieji du tipai gali būti bet kokie. Klasė `SuperPair` turi turėti konstruktorį su trimis parametrais. Sukurkite kelis tokių porų objektus. Pabandykite trečią parametą paduoti ne skaičių.