

# Klasės narių pasiekiamumas (**protected** panaudojimas)

Mindaugas Karpinskas  
2017



# Trumai

---

- Klasės narių panaudojimas
  - a. **private**,
  - b. (nutylimas),
  - c. **protected**,
  - d. **public**

# Klasės narių panaudojimas

---

Java kalboje galima kontroliuoti klasės narių panaudojimą. Kol kas visi nariai buvo vieši (public). Šis modifikatorius pažymi, kad klasės metodas ar kintamasis yra "demokratiškas" ir juo gali naudotis toje pačioje ir kitose klasėse laisvai.

public alternatyvos: private, (Nutylimas), protected

Modifikatorius	Aprašymas
Private	Narys pasiekiamas tik klasės viduje.
(Nutylimas)	Narys pasiekiamas toms paketo klasėms, kurios nepraplečia šią klasę.
Protected	Narys pasiekiamas paketo klasėms.
Public	Narys pasiekiamas kitoms klasėms.

# Klasės narių pasiekiamumas (panaudojimas)

---

- Panaudojimo modifikatorius leidžia programuotojui apsaugoti savo klasę nuo vartotojo neapgalvotų veiksmų.
- Gerai apgalvoti modifikatorių reikšmės naudinga ir pačiam programuotojui, nes grįžus po ilgesnės pertraukos prie klasės redagavimo užsimiršta detalės ir galima padaryti sunkiai aptinkamą klaidą neatsargiai pakeitus kokio nors kintamojo reikšmę.
- Antra vertus **private** modifikatoriumi neverta piktnaudžiauti, nes jis mažina jūsų klasės panaudojimo lankstumą.
- Tokia metodų ir kintamųjų panaudojimo valdymo koncepcija vadinama "duomenų maskavimu" ( angl. data hiding).

# Klasių paveldėjimas **Inheritance**

---

Kintamų panaudojimo valdymas suvaržo jūsų klasės panaudojimą. Objektiniame programavime plačiai naudojamas paveldimumas inheritance, kuris yra vienas pagrindinių šios programavimo paradigmos privalumų. Pažymėti, kad naujai rašoma klasė turi paveldėti visus anksčiau parašytos klasės kintamuosius ir metodus, rašomas Java kalbos raktinis žodis extends ir praplečiamos klasės pavadinimas.

# Panaudojimo taisyklės

---

- Kalbėdami apie duomenų apsaugos privalumus pristatėme **private** modifikatorių.
  - Šis modifikatorius leidžia naudoti kintamąjį ar klasę tik tos klasės viduje.
  - Yra dar du kiti klasės metodų ir kintamųjų modifikatoriai, tai protected.

# Protected modifikatorius

---

Protected modifikatorius suteikia teisę naudotis kintamuoju arba metodu tik klasėms, kurios yra tame pačiame pakete. Kokiam paketui yra priskiriama klasė yra paskelbiama jos aprašymo pradžioje:

```
package somePackage;
```

# Protected. PVZ

```
package lt.codeacademy.sstudy.protect.one;

public class TestFields {

    public static void main(String[] args)
    {
        Field test = new Field();
        System.out.println(test.age);
    }

}
```

```
package lt.codeacademy.sstudy.protect.one;

public class Field {
    protected int age = 100;
}
```

> 100

TestFields klasė gali pasiekti **age** kintamąjį.



# Protected. PVZ

Skirtingi paketai

```
package lt.codeacademy.one;
```

```
import lt.codeacademy.two.Field;
```

```
public class TestFields {
```

```
    public static void main(String[]  
args) {  
        Field test = new Field();  
        System.out.println(test.age);  
    }  
}
```

```
package lt.codeacademy.two;
```

```
public class Field {  
    protected int age = 100;  
}
```

Exception in thread "main" java.lang.Error: Unresolved compilation  
problem:

The field age is not visible

at ...

TestFields klasė negali pasiekti **age** kintamojo.

Atsirado import eilute nes klase kitame pakete

# Protected. PVZ paveldējimas

```
package lt.codeacademy.sstudy.protect.two;

import lt.codeacademy.two.Field;
public class TestFields extends Field {
    public static void main(String[] args)
    {
        TestFields test = new
TestFields();
        test.spausdink();
    }
    public void spausdink() {
        System.out.println(super.age);
    }
}
```

TestFields klasē gali pasiekti **age** kintamāji.

```
package lt.codeacademy.sstudy.protect.one;

public class Field {
    protected int age = 100;
}
```

> 100

# Panaudojimas

---

Modifikatoriai	Tos pačios klasės nariai	To paties paketo klasės	To paties paketo poklasės nariai	Poklasės nariai	Visos kitos klasės
<b>private</b>	Taip	Ne	Ne	Ne	Ne
	Taip	Taip	Taip	Ne	Ne
<b>protected</b>	Taip	Taip	Taip	Taip	Ne
<b>public</b>	Taip	Taip	Taip	Taip	Taip

Įsiminti!

# Užduotis2

---

Sukurti dvi klases skirtingose paketuose. Antra klasė paveldi pirmąją klasę. Pirmoji klasė turi **protected** metodą `pasisveikinimas()`;

Patikrinti ar antroji klasė gali: pasiekti metodą, jei klasė kitame pakete;

main metode patestuoti

# Klausimai

---

