

Objektai su rolėmis

Kas tai yra?

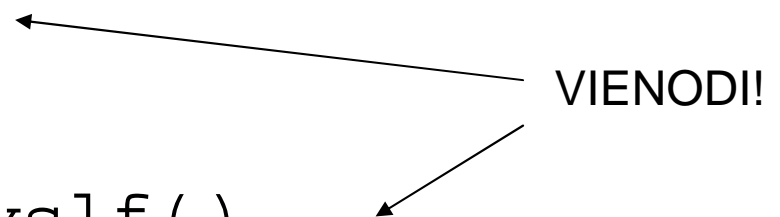
```
john := new Person( "John" , "Smith" , 1987 )
```

```
johnAsAthlete := in Athlete ( john ,  
                               { sport : "tennis" , code : 123 } )
```

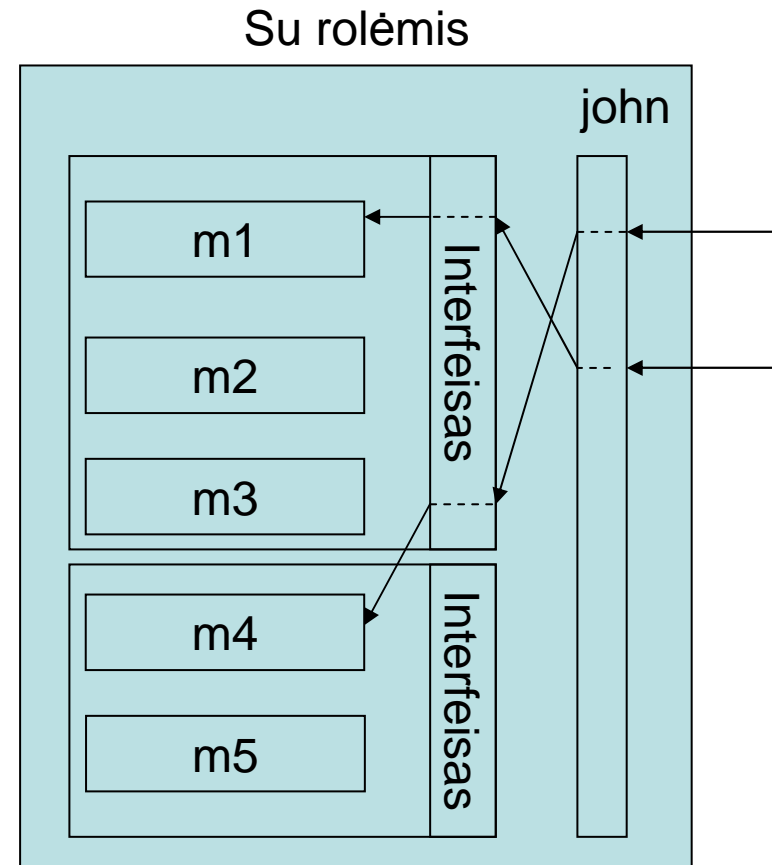
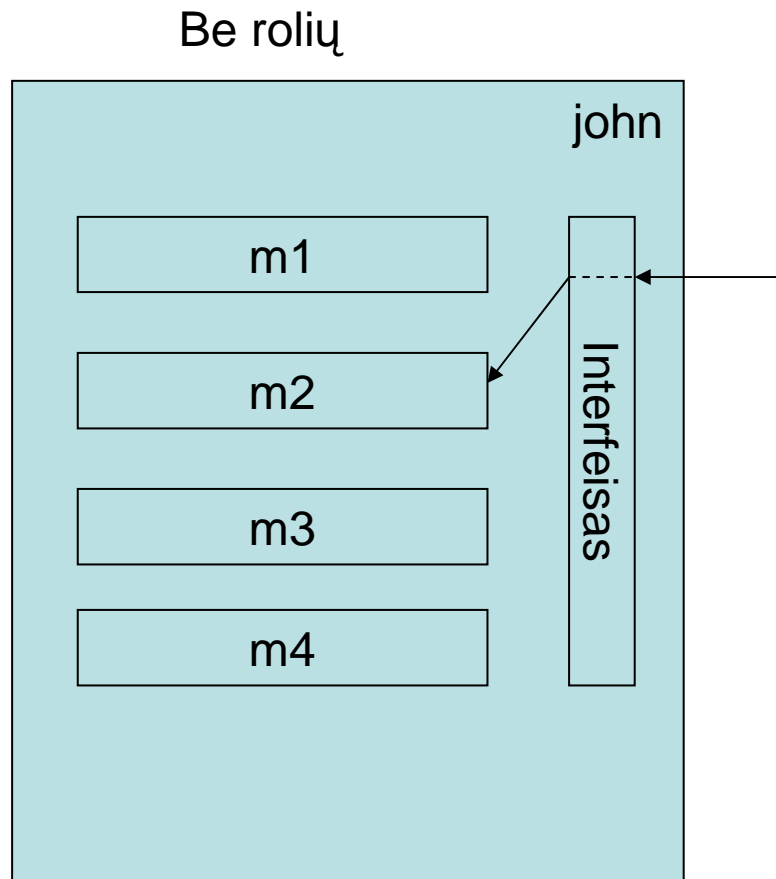
dabar:

```
john.introduceMyself( )
```

```
johnAsAthlete.introduceMyself( )
```



Kaip tai daroma?



Kas kaip atsako?

```
johnAsStudent := in Student (john,  
    {faculty:"Science", code:"333"})
```

```
johnAsStudent.code    - string
```

```
johnAsAthlete.code    - int
```

```
john.introduceMyself() - "I'm Student"
```

Kaip pasiekti norimą rolę?

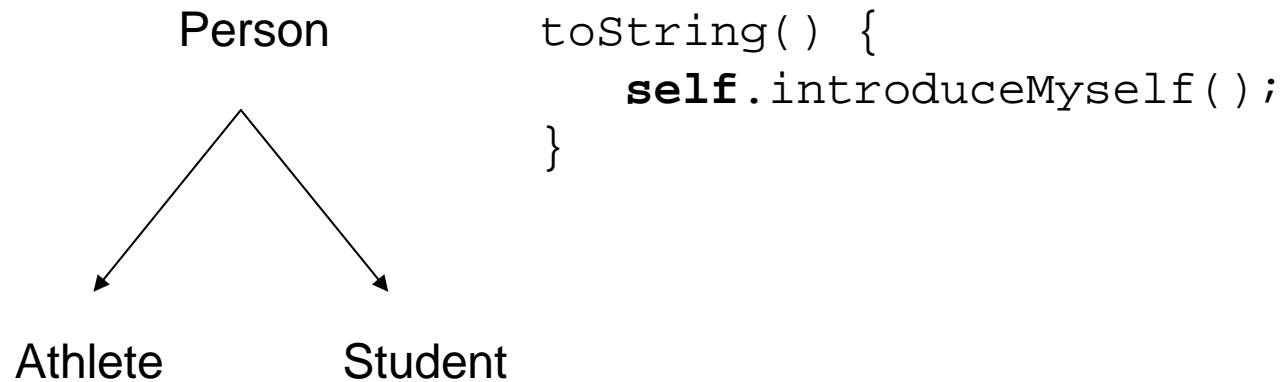
```
john!introduceMyself()
```

Downward lookup – ieškoma nuo objekto klasės **žemyn** iki tinkamos rolės

Upward lookup – ieškoma aukštyn klasių hierarchijoje, kol randama pirmoji implementacija

- yra *double lookup* (*downward*, po to *upward*) operatorius
- ! yra *upward lookup* operatorius

Kas esu AŠ?



<code>john.toString()</code>	<code>"Studentas"</code>
<code>john!toString()</code>	<code>"Asmuo"</code>
<code>johnAsAthlete.toString()</code>	<code>"Atletas"</code>

Ištriname rolę

```
drop Student (john)
```

```
myCar := new Car (johnAsStudent, "Ferrari");
```

```
drop Student (john);
```

```
myCar.owner      ??????
```

Rezultatas priklausys nuo realizacijos

Virtualūs objektai

Kas tai yra?

Tai objektai sukurti ne naudojant klasę, o iš kito objekto, padedant 4 operatoriams:

- **project**
- **rename**
- **extend**
- **times**

Kas tai yra?

```
johnView := john project (name, address, introduceMyself);
```

```
johnView.surname  
johnView.name = john.name
```

error
jie ir keičiasi kartu!

Kas tai yra?

```
johnView := john rename (introduceMyself()=>whoAreYou());
```

```
johnView.whoAreYou() = john.introduceMyself(); // true
```

Kas tai yra?

```
johnView := john extend (spouse: Person);
```

Gali būti naudojamas ir perrašyti savybes:

```
johnItaliano := john extend (  
    introduceMyself := meth():string {  
        return "Me chiamo " + me.Name;  
    }  
)
```

Kas tai yra?

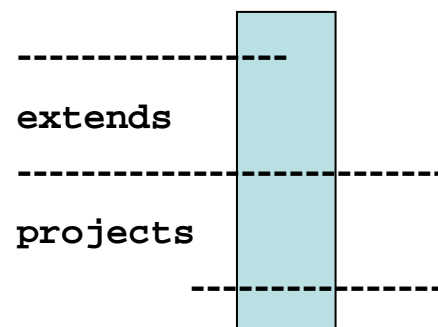
```
johnAndCompany := john times company;
```

Gausime objektą, turintį ir Asmens, ir Darbovietės savybių.

Jeigu yra atributų/metodų vienodu vardu – vieną kurį prieš tai reikia pervadinti (**rename**)

Kas gaunasi?

Gauto objekto klasē NEBŪTINAI yra pirminio objekto klasės poklasis:



Bendra dalis tik tokia

Kas esu AŠ?

me ir **self**

self yra rolė, kuri PRIIMA pranešimą

me yra rolė, kurioje jis parašytas

Kas esu AŠ?

```
johnView := john extend (  
    newMethod := meth:void {  
        return me.age;  
    }  
) project (introduceMyself(), newMethod())
```

Naudoti **self** čia būtų klaida

O dabar dar viską suderinkit su metodo iškvietimo operatoriumi ! ☺

Virtualios klasės

Kas tai yra?

```
classview Adults in Persons
  where ( CurrentDate.Year() - this.BirthYear > 17 )
  store ( char* Phone := "" )
  compute ( newMethod := meth:void {
              return this.Phone;
            } )
  import ( Name )
```

where – uždeda tam tikrą aprobojimą.

store – sukuria naują atributą, nesantį klasėje **Persons**.

compute – sukuria naują metodą, nesantį klasėje **Persons**

import – iš klasės **Persons** “importuojami” atributai/metodai

Sąsajos su virtualiais obj.

Visa tai galime padaryti prieš tai matytais “virtualių objektų operatoriais” **extend.**
ir **project**

Kas tai yra?

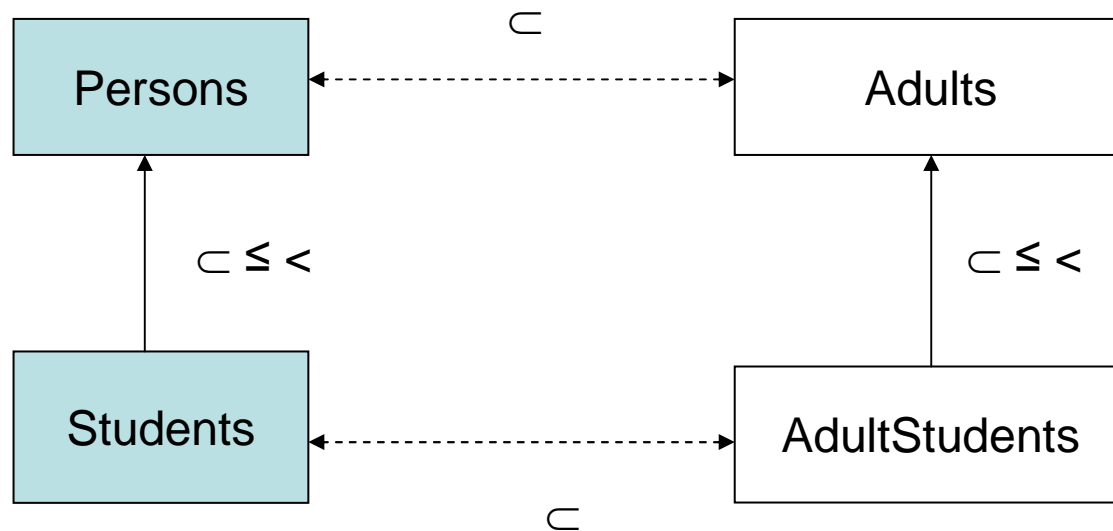
```
classview AdultStudents:Adults in Persons  
    where ...  
    store ...
```

Virtuali klasė `AdultStudents` yra virtualios klasės `Adults` poklasis ir yra padaryta iš (*based on*) realios klasės `Students`.

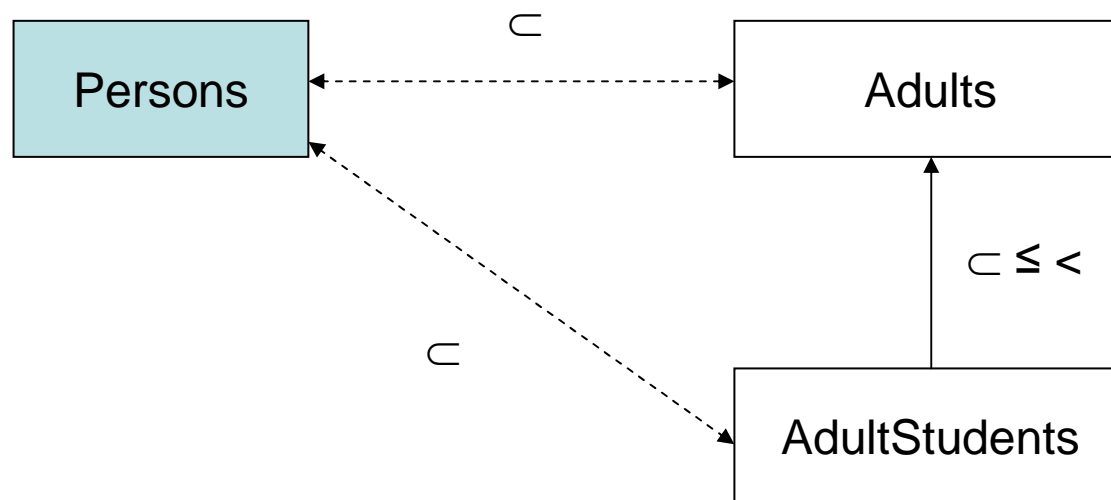
`AdultStudents` paveldi visus **where**, **store**, **compute** ir **import** veiksmus iš virtualios klasės `Adults`.

Kuo skiriasi ryšiai

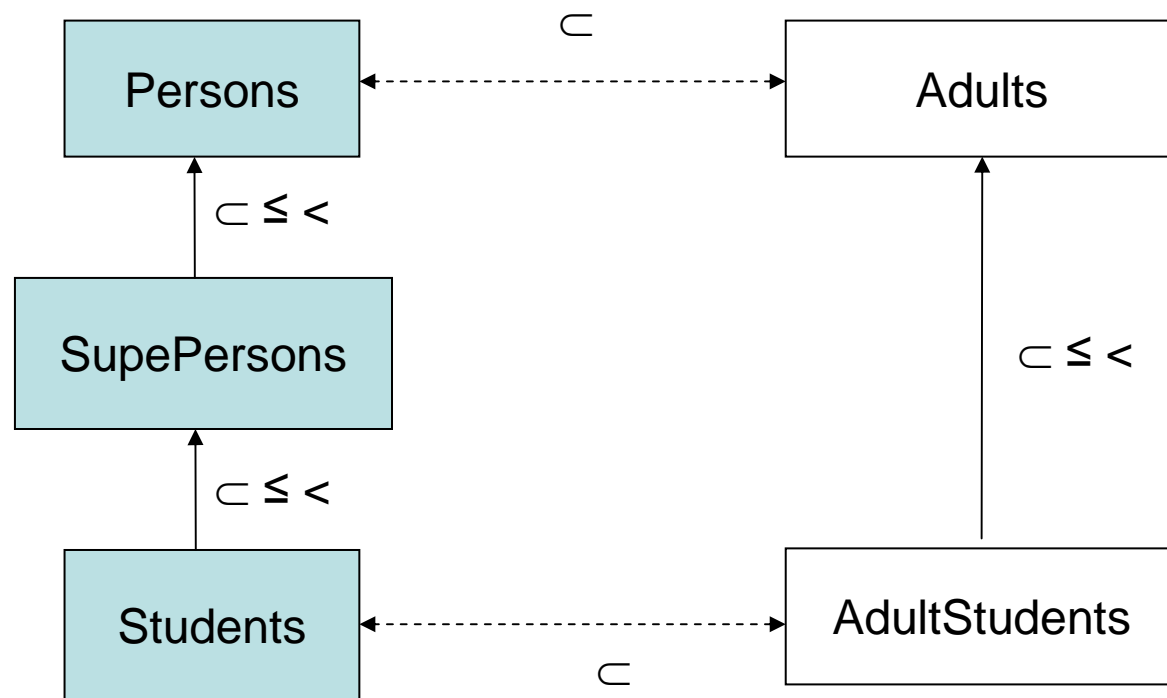
Ryšys *based on* yra tik “aibės/poaibio” ryšys. O virtualių klasių/poklasių ryšys yra ir aibės/poaibio, ir tipo/potipio ir klases/poklasio ryšys.



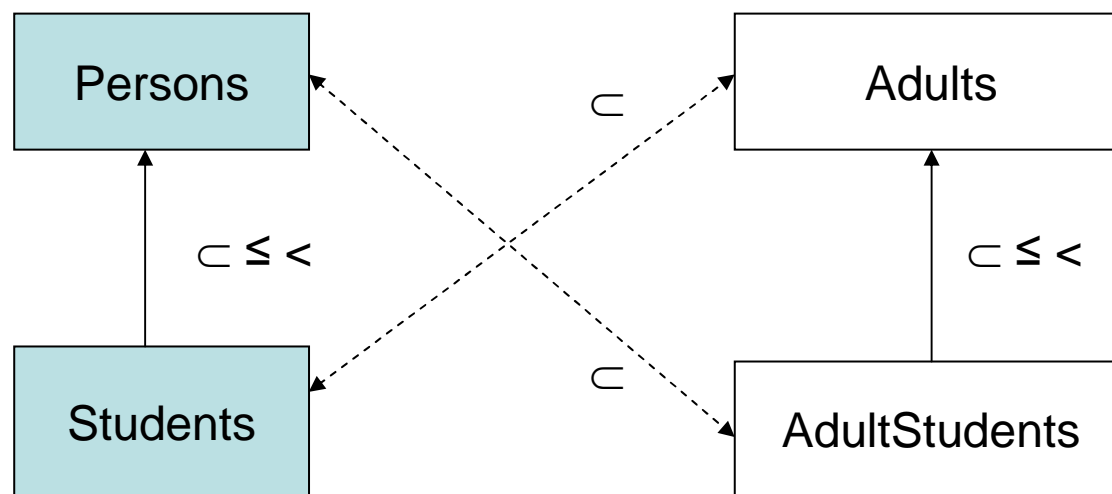
Kaip dar gali būt?



Kaip dar gali būt?



Kaip dar gali būt?



Kaip dar gali būt?

