



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

IDK2015 Objektorienteeritud programmeerimine

Martin Rebane (martin.rebane@ttu.ee)

Mis on liidese (interface) sagedaseim kasutusjuht?

- Tüübina, konkreetsete klasside asemel defineerime tüübi liidese abil:

```
List<String> i = new ArrayList<>();
```

- Mis siin veel toimub?
 - Ülesteisendus, upcasting: ArrayList tüüpi objekti käsitleme List-tüüpi objektina



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Teisenduse viise

- Explicit casting (konkreetne, selge):

SomeType a = (SomeType) obj;

(võimalik nii üles- kui allateisendus)

- Implicit casting (varjatud teisendus):

GeneralType a = new Type();

(ainult ülesteisendus)



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Täna

- Local class ja anonymous class
- Thread ehk lõim
- Lõimedega programmeerimine: sissejuhatus



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Kuidas veel klasse laiendada?

- Näiteks siis kui te koodi ei taaskasuta?

```
new Class() {  
    public void overriddenMethod()  
    {  
        // ylekirjutatud  
    }  
};
```



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Local class

- Klass meetodi sees – võite defineerida nagu tavalise klassi



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Anonymous class

- Nimetu klass, mille defineerite lausendi (*expression*) sees



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Täna

- Local class ja anonymous class
- Thread ehk lõim
- Lõimedega programmeerimine:
sissejuhatus



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Protsess ja lõim

- Põhimõtteliselt mõlemad tagavad koodi täitmise järjestatuse
- Protsess – eraldi mälueraldusega
- Thread e lõim – jagatud mälu



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Programm

- Kasutaja vaatevinklist on üks programm sageli üks protsess (kuid mitte alati!)
- Üks protsess võib aga käivitada erinevaid lõimesid



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Paralleeltöö

- Paralleeltöö ehk lõimedega programmeerimine on vajalik eelkõige kahel otstarbel:
 - teha mitut asja samal ajal
 - kasutada efektiivselt arvuti ressursi (eelkõige protsessori tuumasid)



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Lõim ehk thread

- Meie jaoks Java objekt
- Oluline meetod run()



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Täna

- Local class ja anonymous class
- Thread ehk lõim
- Lõimedega programmeerimine:
sissejuhatus



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

join()

- Käesolev thread peab ootama kuni viidatud thread on töö lõpetanud:

```
th.join();
```

- Millise nimega threadi objekt ootab?



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Vastus:

- Ei tea nime..
- Pealõim ootas



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Paralleelne vs järjestikune täitmine

- Lõimed töötavad iseseisvalt, mitte nii nagu me neid välja kutsume...
- ... muidu poleks mõtet lõimesid luua



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Sünkroniseerimine

- Ligipääs objektile või meetodile sünkroniseeritakse:
 - Tagame, et vaid üks lõim korraga antud koodi kasutab
 - Tagame koodi järjestikuse täitmise (synchronized keyword abil)



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Täna vaatasime

- Selge ja varjatud teisendus
- Lokaalsed klassid
- Anonüümsed klassid
- Protsess ja lõim
- Paralleeltöö
- Sünkroniseerimine



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY