



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

IDK0051 Objektorienteeritud programmeerimine Javas

Martin Rebane (martin.rebane@ttu.ee)

Mis tüüpi objektiga on tegu?

- Liides
- Object
- Klass
- Alamklass

Iga klassi ülemklass

Objekt

See objekt pärib Object,
Liides, Klass ja
Alamklass meetodid
ja väljad



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Meetodi ülekirjutamine (*override*)

- Alamklass defineerib sama nime ja samade argumentidega meetodi
- Tagastustüüp peab olema sama või originaaltüübi alamtüüp:

Ülemklass:

```
public Student getTwo() {
```

Alamklass:

```
public UndergradStudent getTwo() {
```

Meetodi ülelaadimine (*overload*)

- Mitu sama nime, kuid erinevate argumentidega meetodit
- Üle laadida saab nii ülemklassi kui ka sama klassi meetodit



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Automaatne argumendita konstruktor

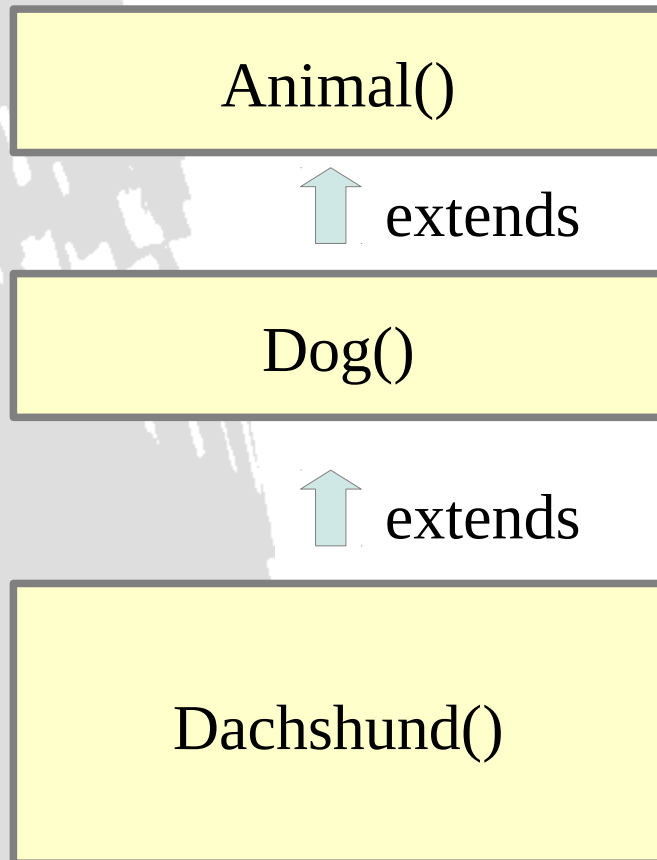
- Chaininguga kutsutakse Object klassi konstruktor
- Kui teie klassis puudub konstruktor, luuakse vaikimisi argumendita konstruktor



1918

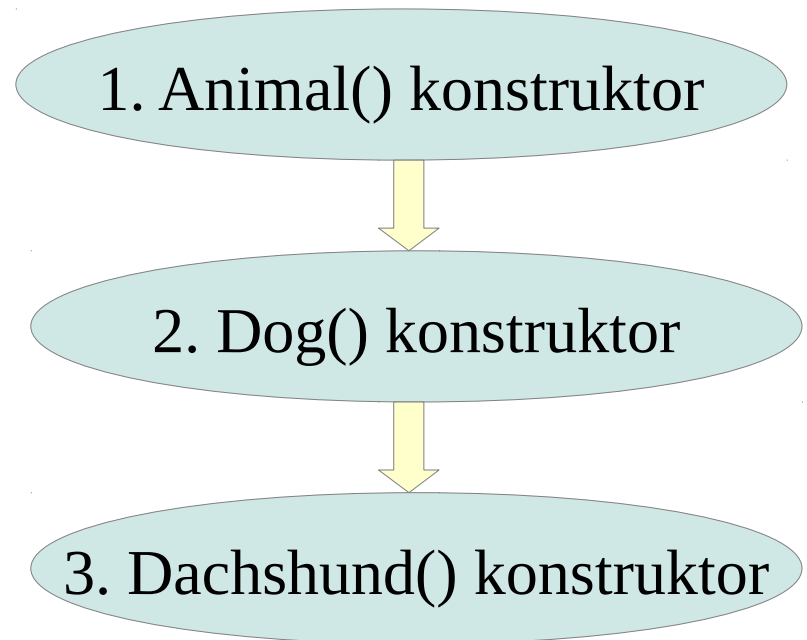
TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Konstruktorite käivitusahel



new Dachshund() ;

Käivitamise järjekord



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Käivitusahela tagajärg:

- Alamklassi konstruktori käivituses on ülemklassi väljad algväärtustatud
- St alamklassi konstruktoris saame kasutada ülemklassi välju ja meetodeid

Polümorfism OOPis

- Erinevat tüüpi objektide kasutamine ühise tüübi abil, näiteks:
 - Car-objekt (aga samas ka Vehicle)
 - Bus-objekt (aga samas ka Vehicle)
- Realiseerivad meetodit `noOfPassengers()` erinevalt
 - Kood mis käsitleb kumbagi Vehicle-tüüpi objektina, ei pea teadma tegelikku tüüpi, millest objekt loodi ja saab `noOfPassengers()` meetodit üheselt välja kutsuda



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Täna

- *Static factory method*
- Mälukasutus Javas: pinu ja kuhi
- Null-tüüp ja null-viide
- Optional



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Static factory method



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Static factory method

- *Static factory method* on alternatiiv konstruktorile – loote objekti staatilise meetodi sees ja tagastate selle

```
public class MyFactory {  
    public static MyFactory getInstance() {  
        return new MyFactory();  
    }  
}
```

Tagastatav tüüp

NB! Ei tagasta mitte klassi, vaid objekti!



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY



**Miks küll peaks sellise asjaga
tegelema kui meil on konstruktor
olemas?**



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Static factory method – MIKS?

- Kontrollite väljastatud objektide arvu:
 - Võimaldab „kallite” objektide taaskasutust, nt andmebaasiühendus
- Kontrollite väljastatud objekti tüüpi:
 - Väljastate lubatud tüübi asemel mõne optimiseeritud alamtüübi

Kontseptsioon oleks justkui tuttav?

Polümorfism!



TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY



Miks tagastada alamtüüp?

- Optimeerimine – sobiv alamtüüp vastavalt andmete mahule või töökeskkonnale
- Äriloogika – sobiv alamtüüp vastavalt sisendargumendile (*static factory method* võib võtta ka argumendi)



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Static factory method – MIKS KÜLL?

- Konstruktorite konflikt:

```
public Product(int weight) {  
  
}
```

```
public Product(int length) {  
  
}
```

Parameetrid *weight* ja *length* on erinevad, kuid sama tüüpi – kahte sellist konstruktorit luua ei saa



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Workaround: mängida tüüpidega

Something smells here!

```
public Product(int weight) {  
}
```



```
public Product(long length) {  
}
```

Kood kompileerub, aga kas on hea kasutada?
Kumb versioon tahtis int ja kumb long?



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Lahendus 1

Initsialiseerimine privaatse konstruktoriga

```
private Product(int weight) {  
}
```

```
private Product(long length) {  
}
```

Sama seis, kuid
bad code ei pääse
klassist välja

```
public static Product getInstanceByWeight(int weight){  
    return new Product(weight);  
}
```

```
public static Product getInstanceByLength(int length){  
    return new Product((long)length);  
}
```

Teisendus

Kasutajale nime järgi selge, kumba kasutada!



19

T

KAÜLIKOOL
TECHNOLOGY

Lahendus ühe konstruktoriga

```
private Product(int weight, int length) {  
}
```

```
public static Product getInstanceByWeight(int weight){  
    return new Product(weight, 0);  
}
```

```
public static Product getInstanceByLength(int length){  
    return new Product(0, length);  
}
```



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Lahendus setteritega

```
private Product() {
```

```
public static Product getInstanceByWeight(int weight){  
    Product product = new Product();  
    product.setWeight(weight);  
    return product;  
}
```

```
public static Product getInstanceByLength(int length){  
    Product product = new Product();  
    product.setLength(length);  
    return product;  
}
```



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY



Miinused: *static factory method*

Konstruktorite käivitusahel!

- Kui konstruktor on privaatne, siis ei saa luua alamklasse. **Miks?**
- Objektitüüp vajab tundmaõppimist: kuidas luua uut objekti, ei ole esmapilgul sugugi selge
- Rohkem infot: Bloch: *Effective Java (2nd ed)*



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Kokkulepped nimetuste osas

- Iseloomustav nimi:
`getInstanceByWeight`
- `valueOf(args)`, `of(args)` –
argumendist tehakse vastav objekt
- `getType(args)` – tagastab mingit teist
tüüpi objekti (*factory method* teises klassis)



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Kokkulepped nimetuste osas

- `getInstance(args)` – sarnane `of()`'ile, kuid tagastav objekt ei ole üheselt iseloomustatav meetodi argumentidega

Objekti loogiline sisu tuleneb argumendist `[args]`

```
University.of("Rektor Keevallik");  
University.getInstance(4, 455, true);
```

Objekti loogiline sisu ei ole üheselt `[args]`'ist tuletatav (või `[args]` puudub)



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Kokkulepped nimetuste osas

- `newType()` ja `newInstance()` - sarnased `getType` ja `getInstance`'le, kuid tagastavad iga kord kindlasti uue objekti



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Java 9 factory methods

- Java 9 lisab factory methods kolleksioonidele.

List.of()
Set.of() ...

- Lisainfo: <http://openjdk.java.net/jeps/269>



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Muutujate tüübid ja mälukasutus

Stack (pinu) ja *heap* (kuhi)



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Muutujate tüübid Javas

- Primitiivsed tüübid (*primitive types*): int, long, short, byte, float, double, boolean
 - Konkreeted väärtused – väärtus on seotud muutujaga
- Objektitüübid (*reference types*): kõik ülejäänud, sh massiivid (*arrays*)
 - Väärtus on muutujast lahutatav

Muutujate tüübid Javas

Tervik

```
int numOfStudents = 1;  
(nimi ja väärtus koos)
```

Primitiivset tüüpi muutuja

Viide objektitüübile

Objekt ise

S

Student myStudent

Student objekt

Objektitüüpi
muutuja



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Mälukasutus JVM's

- *Stack* (pinu) on mälupiirkond, kus:
 - **Kohene mälueraldus** muutujatele nende deklareerimisel
 - Siin hoitakse **primitiivset tüüpi** väärtusi
 - Üks *stack* iga lõime (*threadi*) ehk käivitatava ühiku kohta



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Mälukasutus JVM's

- *Heap* (kuhi) on mälupiirkond, kus:
 - Mälueraldus **vajaduse tekkimisel**
 - Siin hoitakse **objektitüüpe**
 - Üks heap terve JVM kohta



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Muutujatüübid mälus

Stack e pinu

```
int numOfStudents = 1;  
(nimi ja väärtus koos)
```

Student myStudent

Heap e kuhi

Student objekt



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Seostamine

Meetodi seostamine

- Probleem: klassist võime objekti luua nii otse kui läbi alamklassi. Kumba tegime?
- Meie teame, kuidas Java käitub. Aga kuidas Java teab millist meetodit välja kutsuda (nt polümorfse koodi korral)?



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Seostamine

- Lahendus: tuleb luua seos meetodi väljakutse ja meetodi keha vahel
- Teie teete meetodi väljakutse, Java otsib sobiva meetodi keha



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Seostamine

- Alamklass – ülemklass näide: ülemklassi meetod asendatakse alamklassi omaga
- Javas on enamik meetodeist virtuaalsed (v.a static, final, private), st meetodid ei ole staatiliselt seostatud klassiga, vaid õige konfiguratsioon paigaldatakse töö ajal



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Meetodi seostamine

- *Binding* – meetodi nime ja kehade seostamine tüübi põhjal
- *Dispatch* – meetodi nime ja kehade seostamine objekti põhjal



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Meetodite seostamine

- Meetodi väljakutsel õige meetodi keha leidmine

Private, static, final meetodid

- *Static binding* – meetodi kehad ja nimi seostatakse kompileerimisel

Ülejäänud

- *Dynamic dispatch* – meetodi kehad ja nimi seostatakse töö ajal (*runtime*)



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Meetodi seostamine (*dispatch*)

- Probleem: klassist võime luua otse objekti või läbi alamklassi?

```
Car a = new Car();  
Car b = new SportsCar();  
a.drive();  
b.drive();
```

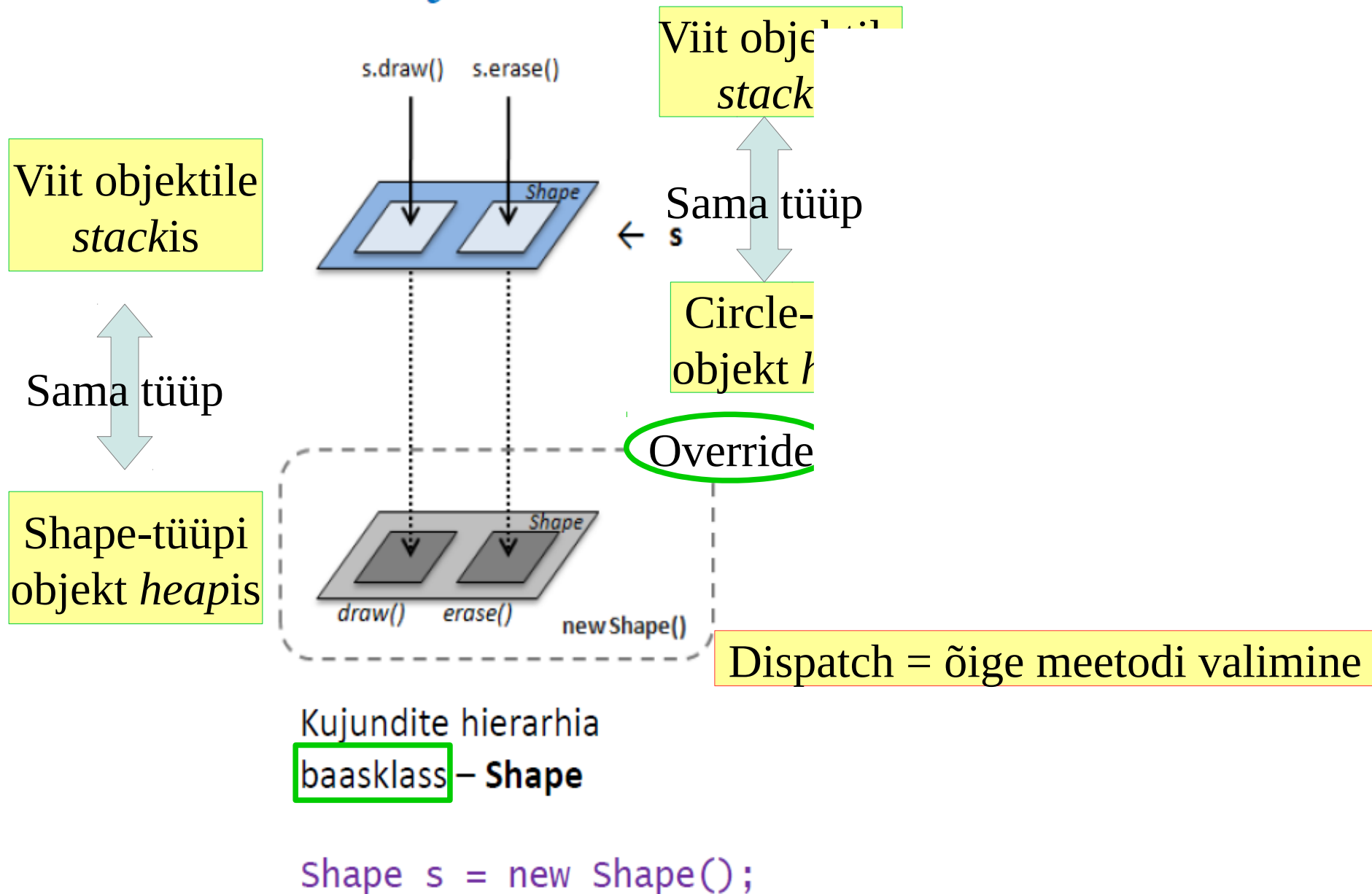
- Kuidas Java teab millist meetodit polümorfse koodi korral välja kutsuda?



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Mis juhtub klassi laiendamisel?

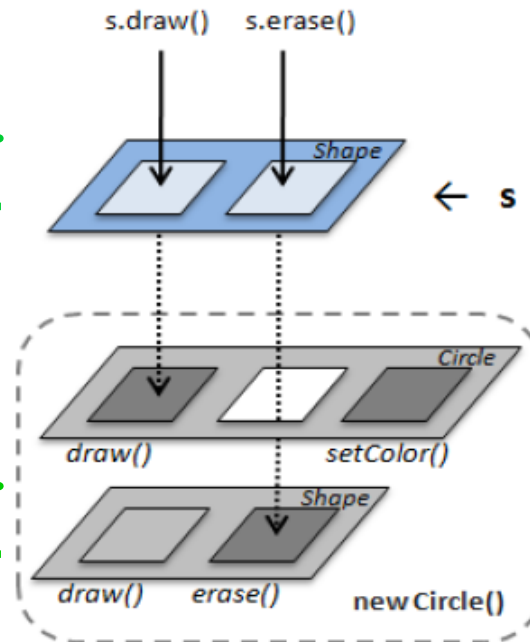


Meetodite seostamine teisendusel

Üks objekt, kaks erinevat tüüpi viita

Viit objektile

Objekt



Ülesteisendus (upcasting)

```
Shape s = new Circle();
```

Ülemtüüp

Alamtüüp



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY



NULL

Kui objekt ei olegi...



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Null ehk tühi viide

- *null* – praktilise käsitluse jaoks lihtsalt puuduv väärtus

Student s = null;

„Student-tüüpi objekti ei ole”

- Iga objektitüübi vaikeväärtus:

- Student s;

Kuniks viide
Student-tüüpi objektile
puudub, asendab seda
null-viide (tühi viide)



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Nulli sisikond

- Sõnega „null” on seotud spetsiaalne null-tüüp
- Null-tüübi väärtus on alati null-viide
- Nulltüübil puudub nimi, ei saa deklareerida:

```
Null myNull = null;
```



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Null-tüüp

- Objekti ei saa teisendada null-tüübiks:

~~Product item;
(null) item;~~

- Null-tüüpi väärtuse (null-viida) võib omistada ükskõik millisele **objektitüübile**

Mis juhtub objekti deklareerimisel?

1. Deklareerime Student-tüüpi objekti, kuid pole seda veel loonud

Viitab kuni Student-objekti loomiseni null-tüüpi väärtusele, milleks on null-viide

`Student myStudent;`




1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY


Olulised omadused

- null-viidaga muutujat saab kasutada

```
Product item = null;  
System.out.println(item);
```



```
List<Product> products =  
    new ArrayList<>();  
products.add(item);
```



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Olulised omadused

- Null-viitega muutujal ei saa välja kutsuda defineeritud tüübi meetodeid ega kasutada välju:

```
NullPlayer notHere = null;  
notHere.operateOnObject();
```



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Nulli kasutuskohad

- Objekt võib puududa:
 - Andmebaasipäringu vastus
 - Veebiteenuse vastus
 - Konfiguratsiooni osa
 - Süsteemi komponent



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Nulli miinused

- Iga objekt võib olla *null*, seega:
 - kui on ootamatult null, siis programm *crashib* `NullPointerException`iga
- Ootuspärast nulli ei ole võimalik mõistlikult kommunikeerida

JAMA!



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Ümbernurgalahendus

```
NullPlayer n = new NullPlayer();  
Product product = n.getProduct();  
  
if (product != null) {  
    product.operate();  
}
```

Mida see kood ütleb?



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Nulli kontroll

- Kas me **ei usalda sisendit** või on programmi arhitektuuris **väärtuse puudumine lubatud**?
- Kood ei räägi, miks null-kontroll sisse on kirjutatud



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Optional – Java 8 lahendus



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Optional

- Võimaldab selgelt väljendada, kui väärtuse puudumine on planeeritud stsenaarium

`Optional<Product> product;`



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Optional li loomine

Objekt kindlasti olemas

```
Product prod = new Product();  
Optional<Product> optProd = Optional.of(prod);
```

Mis see on?

Static factory method

Optionali loomine

Me ei ole kindlad, kas objekt on olemas või *null*

```
Optional<Product> optProd;
```

```
Product prod = service.getProduct();  
optProd = Optional.ofNullable(prod);
```

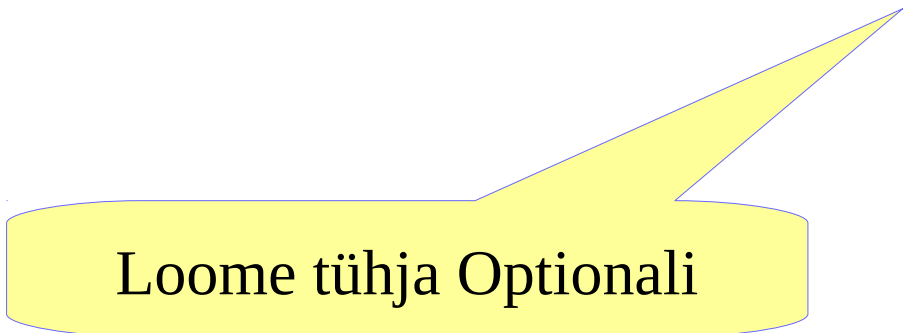


1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Optional liomine

```
Optional<Product> optProd = Optional.empty();
```



Loome tühja Optionali



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Optional kasutamine

```
if (opt.isPresent()) {  
    Product product = opt.get();  
}
```

Sarnaselt *if (opt != null)*. Eelis
nulli ees: selgus, et objekti
disain **lubab** objekti puudumist



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Optional kasutamine: advanced

- `orElse`, `orElseGet`, `orElseThrow`
- `filter`, `flatMap`, `map`, `ifPresent`



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Oluline kasutuskoht: API

- Kasutage tagastatava tüübina Optionali, kui väärtus võib puududa



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Korralduslikud teadaanded

- ained.ttu.ee



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY