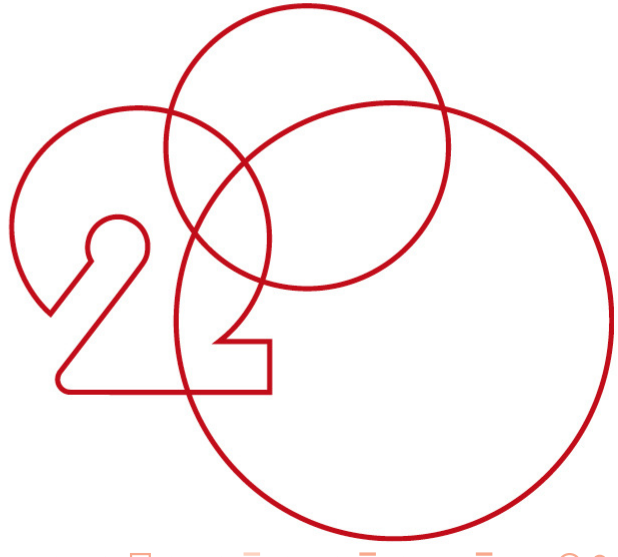
UiO : Universitetet i Oslo

Velkommen!



INF1010 - Fellesøvelser

Praktisk gjennomgang av programmeringsteknikker

Tor Ivar Johansen Espen A. Kristiansen

Institutt for informatikk

February 16, 2011

Idag

Idag

- Oppgaver

Idag

- Oppgaver
- Generics

Idag

- Oppgaver
- Generics
- Package

Idag

- Oppgaver
- Generics
- Package
- Polymorfi

La oss gyve løs på oppgavene!

Oppgave 1

```
class X {}
class Y extends X {}
class Z extends Y {}
```

1. X xy = new Y();
2. Y yy = new Y();
3. Y yx = new X();
4. Y yz = new Z();
5. Z zz = new Z();
6. Z zx = (Z) new X();

Oppgave 1

```
class X {}
class Y extends X {}
class Z extends Y {}
```

1. X xy = new Y();
2. Y yy = new Y();
3. Y yx = new X();
4. Y yz = new Z();
5. Z zz = new Z();
6. Z zx = (Z) new X();

Oppgave 1

```
class X {}
class Y extends X {}
class Z extends Y {}
```

1. X xy = new Y();
2. Y yy = new Y();
3. Y yx = new X();
4. Y yz = new Z();
5. Z zz = new Z();
6. Z zx = (Z) new X();

Oppgave 1

```
class X {}
class Y extends X {}
class Z extends Y {}
```

1. X xy = new Y();
2. Y yy = new Y();
3. Y yx = new X();
4. Y yz = new Z();
5. Z zz = new Z();
6. Z zx = (Z) new X();

Oppgave 1

```
class X {}
class Y extends X {}
class Z extends Y {}
```

1. X xy = new Y();
2. Y yy = new Y();
3. Y yx = new X();
4. Y yz = new Z();
5. Z zz = new Z();
6. Z zx = (Z) new X();

Oppgave 1

```
class X {}
class Y extends X {}
class Z extends Y {}
```

1. X xy = new Y();
2. Y yy = new Y();
3. Y yx = new X();
4. Y yz = new Z();
5. Z zz = new Z();
6. Z zx = (Z) new X();

Oppgave 1

```
class X {}
class Y extends X {}
class Z extends Y {}
```

1. X xy = new Y();
2. Y yy = new Y();
3. Y yx = new X();
4. Y yz = new Z();
5. Z zz = new Z();
6. Z zx = (Z) new X();

Oppgave 1

```
X x = new X();
Y y = new Y();
Z z = new Z();
```

1. y instanceof X
2. y instanceof Y
3. x instanceof Y
4. z instanceof Y
5. z instanceof Z
6. x instanceof Z

Oppgave 1

```
X x = new X();
Y y = new Y();
Z z = new Z();
```

1. y instanceof X
2. y instanceof Y
3. x instanceof Y
4. z instanceof Y
5. z instanceof Z
6. x instanceof Z

Oppgave 1

```
X x = new X();
Y y = new Y();
Z z = new Z();
```

1. y instanceof X
2. y instanceof Y
3. x instanceof Y
4. z instanceof Y
5. z instanceof Z
6. x instanceof Z

Oppgave 1

```
X x = new X();
Y y = new Y();
Z z = new Z();
```

1. y instanceof X
2. y instanceof Y
3. x instanceof Y
4. z instanceof Y
5. z instanceof Z
6. x instanceof Z

Oppgave 1

```
X x = new X();
Y y = new Y();
Z z = new Z();
```

1. y instanceof X
2. y instanceof Y
3. x instanceof Y
4. z instanceof Y
5. z instanceof Z
6. x instanceof Z

Oppgave 1

```
X x = new X();
Y y = new Y();
Z z = new Z();
```

1. y instanceof X
2. y instanceof Y
3. x instanceof Y
4. z instanceof Y
5. z instanceof Z
6. x instanceof Z

Oppgave 1

```
X x = new X();
Y y = new Y();
Z z = new Z();
```

1. y instanceof X
2. y instanceof Y
3. x instanceof Y
4. z instanceof Y
5. z instanceof Z
6. x instanceof Z

Oppgave 1

Svarene var like, hvorfor det?

Oppgave 2

Diskuter bruk av instanceof. I hvilke tilfeller kan man bruke **instanceof** og når bør man unngå det?

Oppgave 3

1. Kan du alltid konvertere fra en subklasse- til en superklasse-peker? Begrunn svaret.
2. Kan du alltid konvertere fra en superklasse- til en subklasse-peker? Begrunn svaret.
3. Når må og når kan man bruke typekonvertering (casting): *EnKlasse peker = (EnAnnenKlasse) enAnnenPeker;*

Generics

Generics

- Hva og hvorfor?

Generics

- Hva og hvorfor?
- Stikkord: Abstraksjon

Generics

- Hva og hvorfor?
- Stikkord: Abstraksjon
- Compile-time typesikkerhet for typer som spesifiseres senere.

Generics

- Hva og hvorfor?
- Stikkord: Abstraksjon
- Compile-time typesikkerhet for typer som spesifiseres senere.
- Fjerner unødig typekonvertering.

Generics

- Hva og hvorfor?
- Stikkord: Abstraksjon
- Compile-time typesikkerhet for typer som spesifiseres senere.
- Fjerner unødig typekonvertering.
- Ypperlig for **Collections**

Generics

- Hva og hvorfor?
- Stikkord: Abstraksjon
- Compile-time typesikkerhet for typer som spesifiseres senere.
- Fjerner unødig typekonvertering.
- Ypperlig for **Collections**
- Suns, nå Oracles tutorial er suveren.

Package

Package

- Organisering av Java-klasser

Package

- Organisering av Java-klasser
- Spesifiserer egne navnerom

Package

- Organisering av Java-klasser
- Spesifiserer egne navnerom
- Etterhvert som programmene blir større er god organisering viktig!

Package

- Organisering av Java-klasser
- Spesifiserer egne navnerom
- Etterhvert som programmene blir større er god organisering viktig!
- **protected** - skjuler tilgang innenfor pakken og av subklasser.

Polymorfi

Polymorfi

- Arver ikke bare attributter, men også metoder.

Polymorfi

- Arver ikke bare attributter, men også metoder.
- Virtuelle metoder redefineres i subklassen.

Polymorfi

- Arver ikke bare attributter, men også metoder.
- Virtuelle metoder redefineres i subklassen.
- Dynamisk binding av metoder.

Polymorfi

- Arver ikke bare attributter, men også metoder.
- Virtuelle metoder redefineres i subklassen.
- Dynamisk binding av metoder.
- Overloading av metoder.

Og nå endelig programmering!