IN1010

UKE 2 - GRUPPE 3

MIAU@IFI.UIO.NO

Plan for dagen

- Arv og subklasser
- Implisitt og eksplitt casting
- Abstrakte klasser
- Array og ArrayList
- HashMap

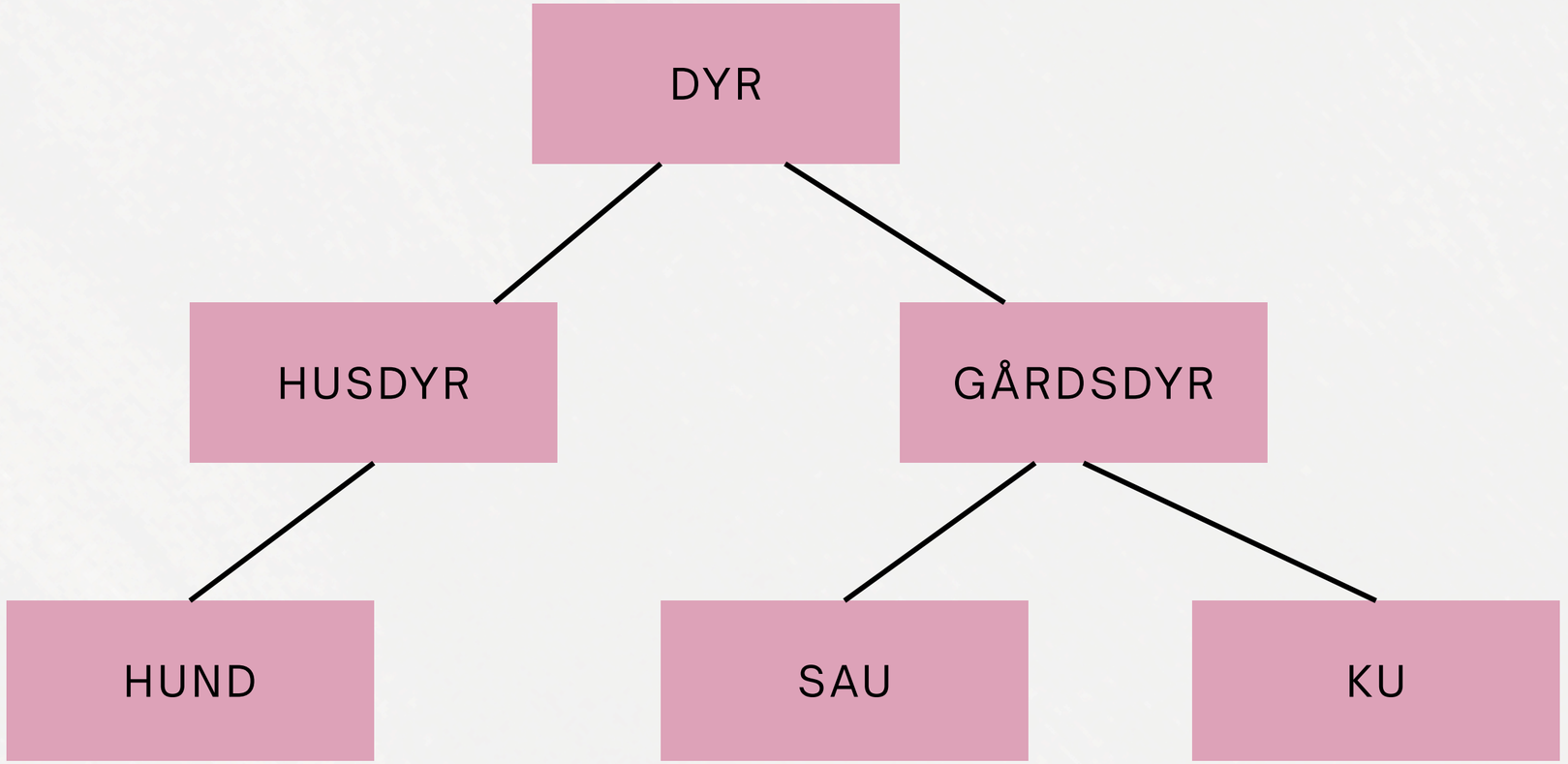
Subklasser og arv

- Klasser som arver egenskaper fra andre klasser
- Arver instansvariabeler og metoder fra superklassen

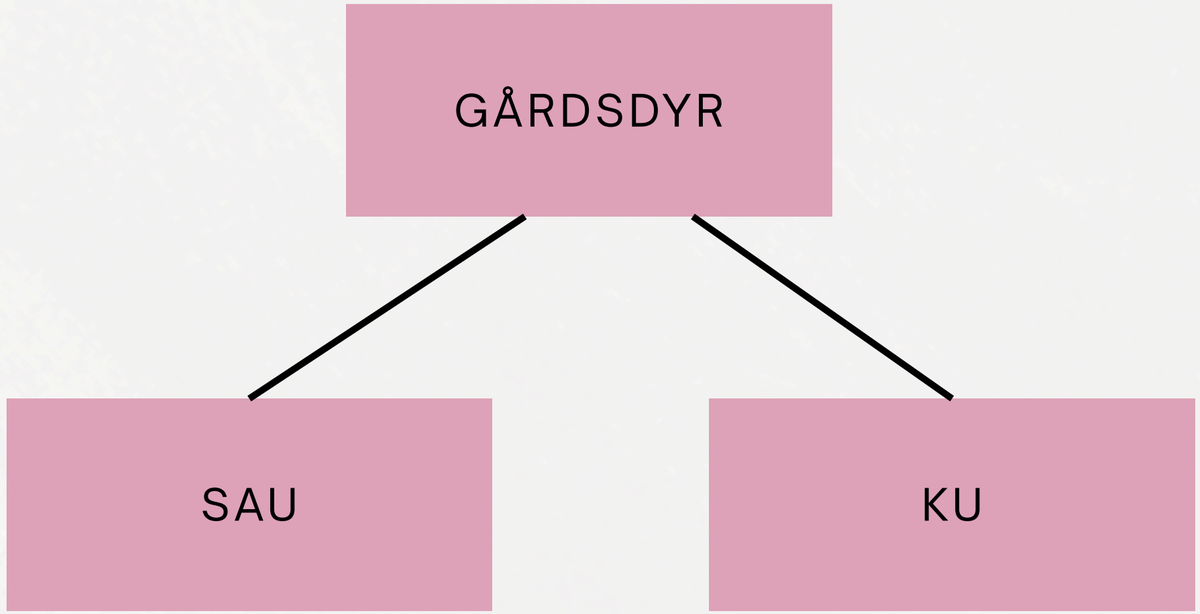
```
class Klasse {/**/}

class Subklasse extends Klasse {/**/}
```

Klassehierarki



Klassehierarki



```
class Gårdsdyr {}
class Ku extends Gårdsdyr {}
class Sau extends Gårdsdyr {}
```

Super

- I konstruktøren til en subklasse, må man arve fra superklassen
  - super(<argumenter>)
- Viktig selvom superklassen ikke tar inn argumenter

```
class Gårdsdyr {
    String type;

    public Gårdsdyr(String type) {
        this.type = type;
    }
}

class Ku extends Gårdsdyr {
    String navn;

    public Ku(String type, String navn) {
        super(type);
        this.navn = navn;
    }
}
```

Implisitt casting

Når man lagrer et objekt i en referanse med typen til en superklasse, så vil objektet konverteres automatisk

```
Run | Debug
public static void main(String[] args) {
    Gårdsdyr gårdsdyr = new Ku();
    Gårdsdyr gårdsdyr2 = new Sau();
}
```

Eksplisitt casting

Om man vil konvertere et objekt tilbake til dens opprinnlige subklasse, må man eksplisitt konvertere objektet.

```
if (gårdsdyr instanceof Ku) {
    Ku ku = (Ku) gårdsdyr;
}
```

Abstrakte klasser

- Deklareres med ordet ‘abstract’
- Ikke lov å lage objekter av abstrakte klasser
    - Egner seg godt for arv, da man kan lage subklasser som ikke er abstrakte

```
abstract class Abstrakt {}
```

Object

- En abstrakt klasse som alle objekter i Java arver fra
  - Inkluderer klasser vi lager selv, selvom vi ikke definerer det
- Derfor vil alle objekter ha noen felles metoder

Array

```
String[] array = new String[10];
array[5] = "Mia";

for (String i : array) {
    System.out.println(i);
}
```

ArrayList

```
import java.util.ArrayList;
```

```
ArrayList<String> arrayList = new ArrayList<>();
arrayList.add(e: "Mia");
arrayList.get(index: 0);
```

HashMap

```java
import java.util.HashMap;
```

```java
HashMap<String, Integer> ordbok = new HashMap<>();
ordbok.put(key: "Mia", value: 5);
ordbok.get(key: "Mia");
```

Oppgave

Lag et program som administrerer en samling (HashMap) av kjøretøy. Programmet skal ha følgende klasser:

1. Kjøretøy-klasse:
   - Egenskaper: id (String), merke (String), modell (String), år (int)
   - Konstruktør: for å initialisere egenskapene
   - Hent-metoder og sett-metoder for egenskapene (kun hent-metode for id)
2. Bil-klasse (arver fra Kjøretøy):
   - Ekstra egenskap: seter (int)
   - Konstruktør: for å initialisere egenskapene (inkludert de arvede egenskapene fra Kjøretøy)
   - Hent-metode og sett-metode for egenskapen seter
3. Motorsykkel-klasse (arver fra Kjøretøy):
   - Ekstra egenskap: motortype (String)
   - Konstruktør: for å initialisere egenskapene (inkludert de arvede egenskapene fra Kjøretøy)
   - Hent-metoder og sett-metoder for egenskapene
4. Program-klasse:
   - Main-metode: Opprett en samling (HashMap) for å lagre kjøretøy. Legg inn to instanser (objekter) av Bil og to instanser (objekter) av Motorsykkel. Skriv ut detaljene for kjøretøyene i samlingen.