

Formål:

At oprette en **Metode** og forstå konceptet og kunne forklare det.

Hvordan Objekter tilgås på den rigtige måde. Vi undersøger begreberne og sikrer at vi forstår dem og kan forklare dem for hinanden!

Emner:

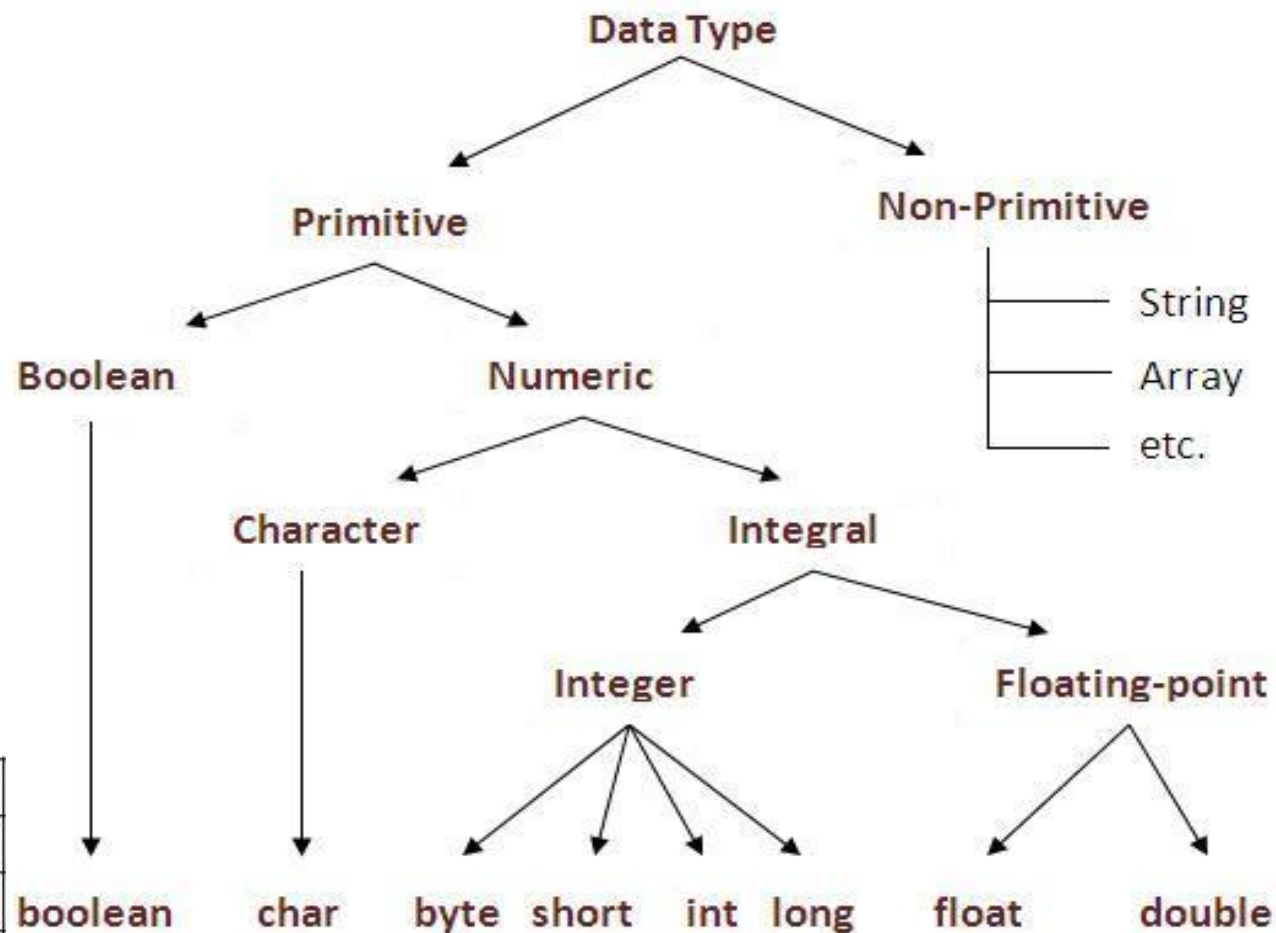
Metoder, Klasser, Attributter og Objekter

Litteratur:

Reges: Kap.: 8.1 - 8.2

Repetition

- Hvad er **datatyper**?
- Hvad bruges de til?
- Hvad er en '**Identifier**'?



	Size	Default Value	Type of Value Stored
byte	1 byte	0	Integral
short	2 byte	0	Integral
int	4 byte	0	Integral
long	8 byte	0L	Integral
char	2 byte	'\u0000'	Character
float	4 byte	0.0f	Decimal
double	8 byte	0.0d	Decimal
boolean	1 bit (till JDK 1.3 it uses 1 byte)	false	True or False

Klasse med metoder:

```
public class Hund{  
  
    String navn = "noname";  
    String farve = "nocolor";  
    int alder = 0;  
    double vaegt = 10.0;  
    String hundelyd = "vov";  
  
    //metode til udskrift af hunde  
    public void hundelInfo(){  
        System.out.println("Navn:" + navn);  
        System.out.println("Farve:" + farve);  
        System.out.println("Alder:" + alder + " år");  
        System.out.println("Vaegt:" + vaegt + " kg");  
    }  
    //metode til hundelyd  
    public void hundelyd(){  
        System.out.println("Hunden siger:" + hundelyd);  
    }  
}
```

Dataelementer som vi skal bruge når vi opretter et objekt af denne klasse (datafelterne er altså IKKE oprettet på dette tidspunkt! Det sker først når noget koden 'bestiller' et objekt med kommandoen 'NEW')

Metoder som vi kan anvende på vores objekt. Det er gennem metoderne vi styrer data og processer vi ønsker der skal udføres på vores objekt. (BEMÆRK: metoderne selv oprettes også først sammen med objektet!)

Klasse - Hovedprogrammet

```
public class Hundekennel{  
  
    public static void main(String[] args){  
  
        //lav et par hunde  
        Hund hund1 = new Hund();  
        Hund hund2 = new Hund();  
  
        //sæt attributer for hunde  
        hund1.navn = "Pluto"  
        hund2.navn = "Jerry";  
  
        //udskriv hund1  
        hund1.hundInfo();  
        //Hvad siger hund1?  
        hund1.hundelyd();  
  
        .....  
    }  
}
```

Her oprettes to **objekter** af klassen hund. I samme øjeblik oprettes metoderne og gennem objektet kan de kaldes.

Her kaldes **metoder** tilhørende objektet.

LiveKodning – step by step 😊

- Vi opretter en klasse – **Kitchen**
- Og gør denne til hovedprogrammet!
- Vi opretter nu en klasse til, komfur – **Stove**
- Vi opretter nu en klasse mere, blus – **Flame**
- Flame har en variabel der angiver temperaturen
- Stove justeres så der optræder Flame1..Flame4 – fire objekter der instantieres.
- I Kitchen vil vi nu skrive kode så vi får stegt på blus 2

Lidt om god programmering

- Når vi arbejder med objekter i kodningen skal vi være opmærksomme på de tilstande og de situationer vores objekt kan optræde i.
- Når vi deler objekter med hinanden er det vigtigt at vi kun bruger andres kode som den er tiltænkt, for at undgå fejl.
- Disciplinen er derfor enkel: Objekter oprettes efter en skabelon og kaldes gennem dens metoder.
- Det betyder: Opret objekter ensartet og undlad at tilgå objektets data uden om metoderne.

Hvorfor er disse regler nødvendige?

Øvelse: "School"

- Opret en mappe i dit bibilotek der hedder school.
- Opret en klasse der hedder 'Student' og tilføj en række identifiere (navn, alder, klasse og faget swdev= en integer) tilføj metoder. En af metoderne skal hedde addLesson. Swdev starter med nul og hvergang addLesson er udført øges med en. Efter fem lektioner udskrives – studerende + navn + har gennemført undervisningen + antal gange.
- Opret en klasse der hedder 'Lecturer' og tilføj (navn, alder, titel, uddannelse) tilføj metoder der kan udskrive navn og titel.
- I hovedprogrammet opretter du nu et par undervisere og en række studerende. Juster på identifiere så du kan skelne dem når de udskrives i konsollen. I School kalder du nogle af metoderne.

1. Forklar sidemanden, og/eller klassen, hvad dit program gør 😊
2. Er der noget ved koden der ikke er så godt i forhold til god programmering?

1 Semester

SWK

Modul 02

Formål:
Introduction to **Parameters**
Methods(parameters, passing a parameter)
Overloading og Constructors

**NÆSTE GANG
i Modul 3**

Med parametre i metoderne undgår vi at skulle tilgå vores datastrukturer direkte. Vi ønsker kun at arbejde gennem metoder når vi håndterer objekter! Med parameterbegrebets introduktion bliver konstruktøren pludselig interessant og begrebet Overloading indføres.

Litteratur: Reges: Kap.: 3 Fokus på 3.1
 Regés: Kap.: 8 Fokus på 8.3, 8.4