

Formål:

Vi undersøger betingede afvikling af kode med begrebet
IF..THEN..ELSE

Emner: Conditional Execution

- IF..Then

Litteratur: Reges: Kap.: 4 fokus: 4.1, 4.2

Betinget kodeafvikling – If..Then

I vores arbejde med at kode, støder vi hurtigt på betingelser. Steder hvor vi kun ønsker en stump kode afviklet i visse tilfælde.

If – konstruktionen giver os denne logiske mulighed: **HVIS** en given betingelse er tilstede **SÅ** vil vi gerne have efterfølgende udført. Vi kan endda tilføje en **ELLERS** vil vi gerne have denne kode udført.

BEMÆRK: IF-sætninger er strengt logiske udtryk der ender med en boolsk værdi: **Sand** eller **Falsk**

Kodeeksempel med simpel If..Then

```
dog1.setDogData("Pluto","sort",25,5,"vruff");  
dog2.setDogData("Jerry","hvid",35,2,"Miav");  
dog3.setDogData("King","Graa",40,6,"vov-vov");
```

```
//udskriv hunde
```

```
if (dog1.getAge()<=50){  
    System.out.println("Her er en ung hund:");  
    System.out.println( dog1.getDogData() );  
}  
System.out.println( dog2.getDogData() );  
System.out.println( dog3.getDogData() );  
dog1.dogBarking();  
dog2.dogBarking();  
dog3.dogBarking();
```

Her benyttes IF-konstruktionen
Betingelsen for at koden køres er at
metoden der returnerer alderen på
hunden giver en værdi under 50

If..Then..Else

```
//udskriv hund
```

```
if (dog2.getAge() <= 1) {  
    System.out.println("Her er en spritny hund:");  
    System.out.println( dog2.getDogData() );  
} else {  
    System.out.println("hunden her er ikke en hvalp:");  
    System.out.println( dog2.getDogData() );  
}
```

Betingelsen er at metoden returnerer en alder mindre end eller lig et år og efterfølgende er tilføjet Else. Else udføres i **alle andre** tilfælde hvor IF ikke er udført.

Når vi ser på objekter...

```
1 public class IfThenObject {
2     public static void main(String args[]) {
3         String a="hund";
4         String b="kat";
5         String c="kat";
6
7         if (a==c)
8             System.out.println("B=C er TRUE");
9         else
10            System.out.println("B=C er False????");
11
12        if (c.equals(b))
13            System.out.println("B og C er ens :-)");
14        else
15            System.out.println("Java compileren virker ikke!!!");
16    }
17 }
```

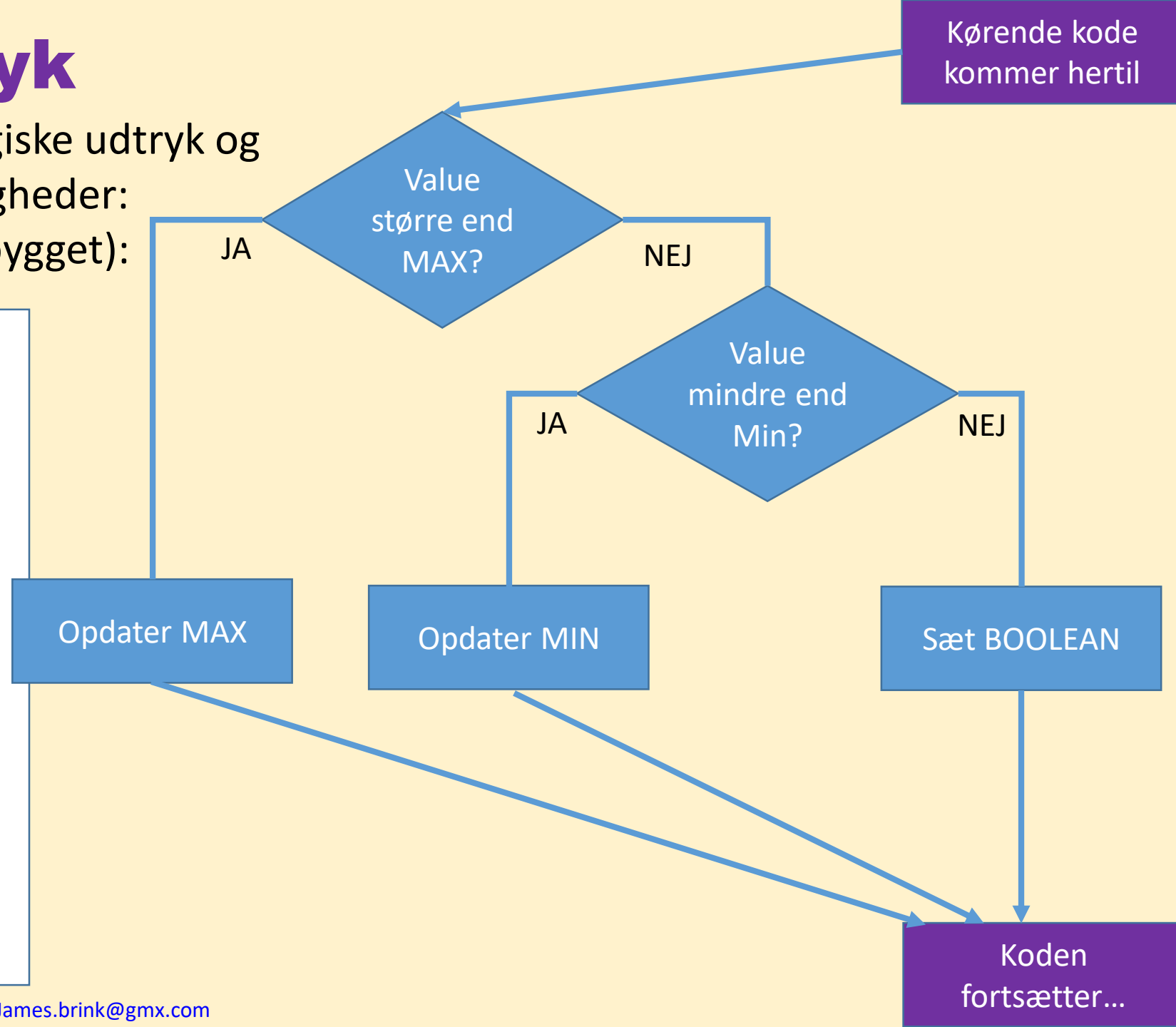
Vær opmærksom på:

- Objekt = Objekt ☠☠☠
- Objekt == Objekt ☠☠☠
- Vi kan ikke bare sætte et objekt til at være lig med et andet eller logisk sammestille dem.
- HVORFOR IKKE?

Komplekse udtryk

IF THEN skal ses som strengt logiske udtryk og kan bruges i et netværk af muligheder: I bogen er et eksempel (lidt udbygget):


```
If (value > max) {  
    max = value;  
} else {  
    if (value < min) {  
        min = value;  
    } else {  
        averageValue = true;  
    }  
}
```



Logik!

Når vi arbejder med logik, som jo er indbegrebet i **If..Then** universet er det vigtigt af få styr på terminologien.

If (a1==b1 || b1>c1 && c1!=a1)



Hvordan
nedbrydes
ovenstående
syntaks af
compileren?

The relational operators are shown here:

Operator	Meaning
==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

The logical operators are shown next:

Operator	Meaning
&	AND
	OR
^	XOR (exclusive OR)
	Short-circuit OR
&&	Short-circuit AND
!	NOT

Lær syntaksen for operatorene – skriv Java-kode til det sidder fast 😊

Øvelse: Betingelser i koden

Kod et Java projekt med en klasse der rummer retur-metoder som kan bruges i hovedprogrammets afvikling.

I hovedprogrammet skal der bruges **if..Then** med og uden **Else**.

Brug også **||** og **&&**.

Repetition

Vi har nu gennemgået de indledende og mest centrale begreber i OOP vedrørende Klasser, objekter og metoder.

- Hvad er en **Klasse**?
 - En skabelon der definerer identifiere og methods som objekter skabes ud fra
- Hvad er et **Objekt**? (uddyb forskellen på Class og Object)
 - Når en identifikator skabes med **new** kommandoen skabes et object efter den givne classes anvisning. Objektet er den forekomst der kan arbejdes med, hvor klassen bare er en model for objektet. Fx:
`Dice dice1 = new Dice();`
- Hvad er en **Metode**?
 - Metoder er en række handlinger der kan kaldes, hvorefter metoderne så foretager noget med objektet.
- Hvad er en **Konstruktør**? (hvad adskiller Constructor fra Method)
 - En konstruktør kan erklæres og vil så blive udført i forbindelse med 'NEW'kommandoen. Typisk bruges konstruktøren til at initialisere data med en given startværdi.

Note: På nuværende tidspunkt skal man have en **viden** om, hvad begreberne dækker over, og have en **grundlæggende forståelse** af konceptet med at **oprette og arbejde med klasser og objekter**.

Diskutér emnerne, læs litteraturen og tal med din underviser hvis der er begrebsmæssig forvirring.

Repetition

Vi har nu gennemgået de indledende og mest centrale begreber i OOP vedrørende Klasser, objekter og metoder.

- Hvad er en **private variabel**?
 - En variabel der erklæres private er beskyttet mod direkte tilgang. Kun objektets metoder kan se den
- Hvad menes med **Getter og Setter** ?
 - Metoder der oprettes for at hente (en getter) en variabel, henholdsvis sætte variabelen kaldes gettere og settere, men er blot metoder som vi kender allerede.
- Hvad gør **this**?
 - This kommandoen sikrer at vores objekts variabel kan skelnes fra variabelen i metoden, da de beejligt hedder det samme. Med **this.var1** henvises til objektes egen variabel, **var1** mens var1 er den der optræder i metodens parameter;
- Hvad betyder **Encapsulation**?
 - Encapsulation, eller indkapsling, er når vi isolerer objektets variable så de ikke kan tilgås direkte. Kommandoen private er en måde at sikre dette.

Note:

På nuværende tidspunkt skal man have en **viden** om, hvad begreberne dækker over, og have en **grundlæggende forståelse** af konceptet med at **oprette - og arbejde med - klasser og objekter**.

Diskutér emnerne, læs litteraturen og tal med din underviser hvis der er begrebsmæssig forvirring.

1 Semester

SWK

Modul 05

Formål: Conditional Execution

Emner: For.Next

**NÆSTE GANG
i Modul 6**

Litteratur:Reges: Regés kap.4

Vi arbejder med klasser, objekter og metoder. Vi bruger parametre og returværdier og husker private-variable og this-begrebet så vores metoder kan bruge samme navne i parameterlisterne.

Vores kode rummer nu If..then og For..Next.