Basale Programmeringsteknikker: Booleans, Branches og Loops

- **Teori**
  - **Boolean typen**
  - **Boolske operator:**

Navn	Eksempel	Sand når
NOT	`!value`	Et udtryk ikke er sandt
AND	`value1 && value2`	Når begge udtryk er sande
OR	`value1 || value2`	Når mindst et udtryk er sandt
XOR	`value1 ^ value2`	Når netop et udtryk er sandt

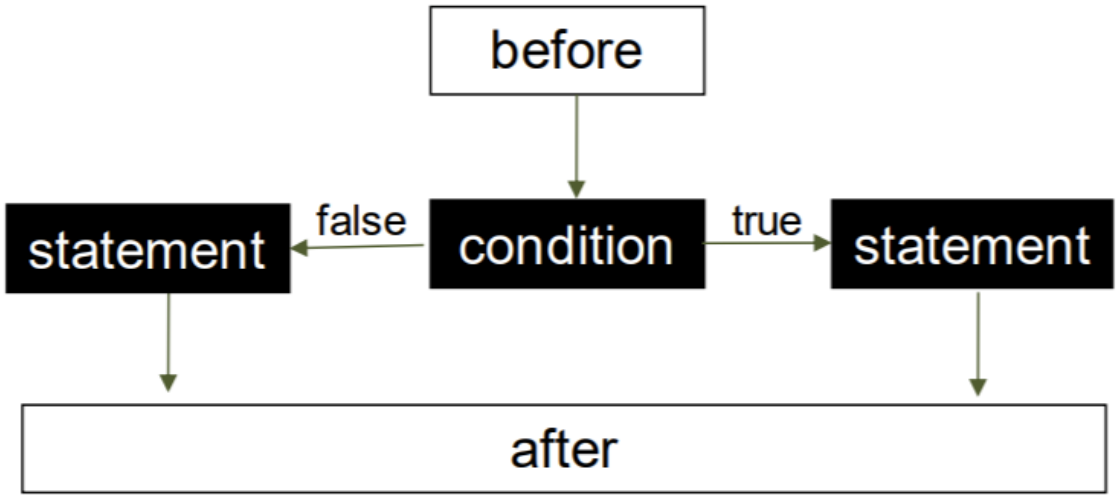
```
boolean goalFailed = false;
boolean timeLeft = false;
boolean succes = !goalFailed && timeLeft;
//Succes kriteriet er, hvis målet ikke er fejlet, og tiden ikke er udløbet
```

  - **Ternary operator (conditional operator)**

```
//syntaks:
(<condition>?<true-expr>:<false-expr>)
//eksempel
int y = (x>0 ? 1 : -1);
// x = 12, y = 1
// x = 0, y = -1, fordi x da ikke er større end 0
```

  - **Valg med branching**

```
if (<condition>)
    <statement>;
else
    <statement>;
```




```
System.out.println("Input is "+i);
if (i%2==1)
    i++;
System.out.println("Output is "+i);
//der bliver kun udprintet lige tal
```

▼ **Branching med blocks**

```
double radius = 12.0

if (radius < 0.0) {
    System.out.println("Incorrect input");
} else {
    area = 3.14159 * radius * radius;
    System.out.println("Area is " + area);
}
```

```
int i=1, j=2, k=3;
if (i<j){
    if (i>k)
      System.out.println("A");
    } else {
      System.out.println("B");
}
```

▼ Switch statements

```
switch (<expr>) {
case <value1>:
    <statements>;
    break;
case <value2>:
    <statements>;
    break;
case <value3>:
    <statements>;
    break;
default:
    <statements>;
}
```

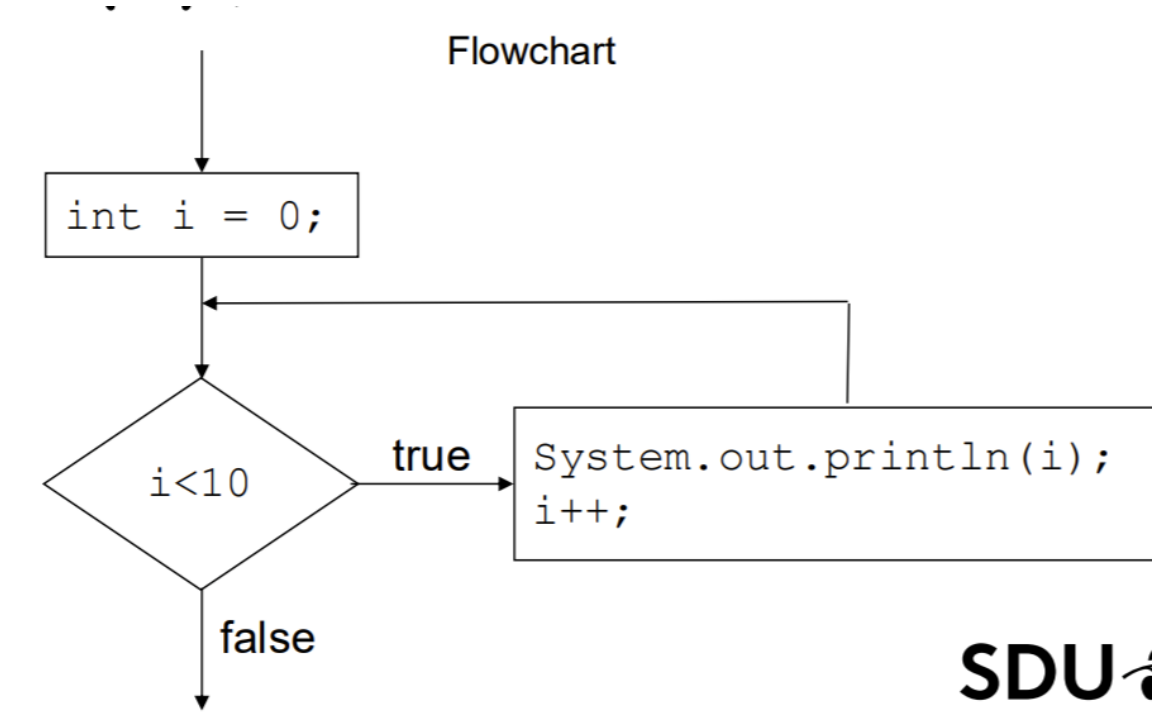
```
char direction = 'S';

switch (direction) {
case 'N':
    System.out.println("Going north ");
    break;
case 'E':
    System.out.println("Going east ");
    break;
case 'S':
    System.out.println("Going south ");
    break;
case 'W':
    System.out.println("Going west ");
    break;
default:
    System.out.println("I don't understand?!?");
```

▼ While loop

Afvikler et **statement**, så længe en *condition* er sand

Syntaks: while (<condition>) <statement>

```
int i = 0;
while (i<10) {
      System.out.println(i);
      i++; //der incrementes efter print, så i stopper på 9
}
```



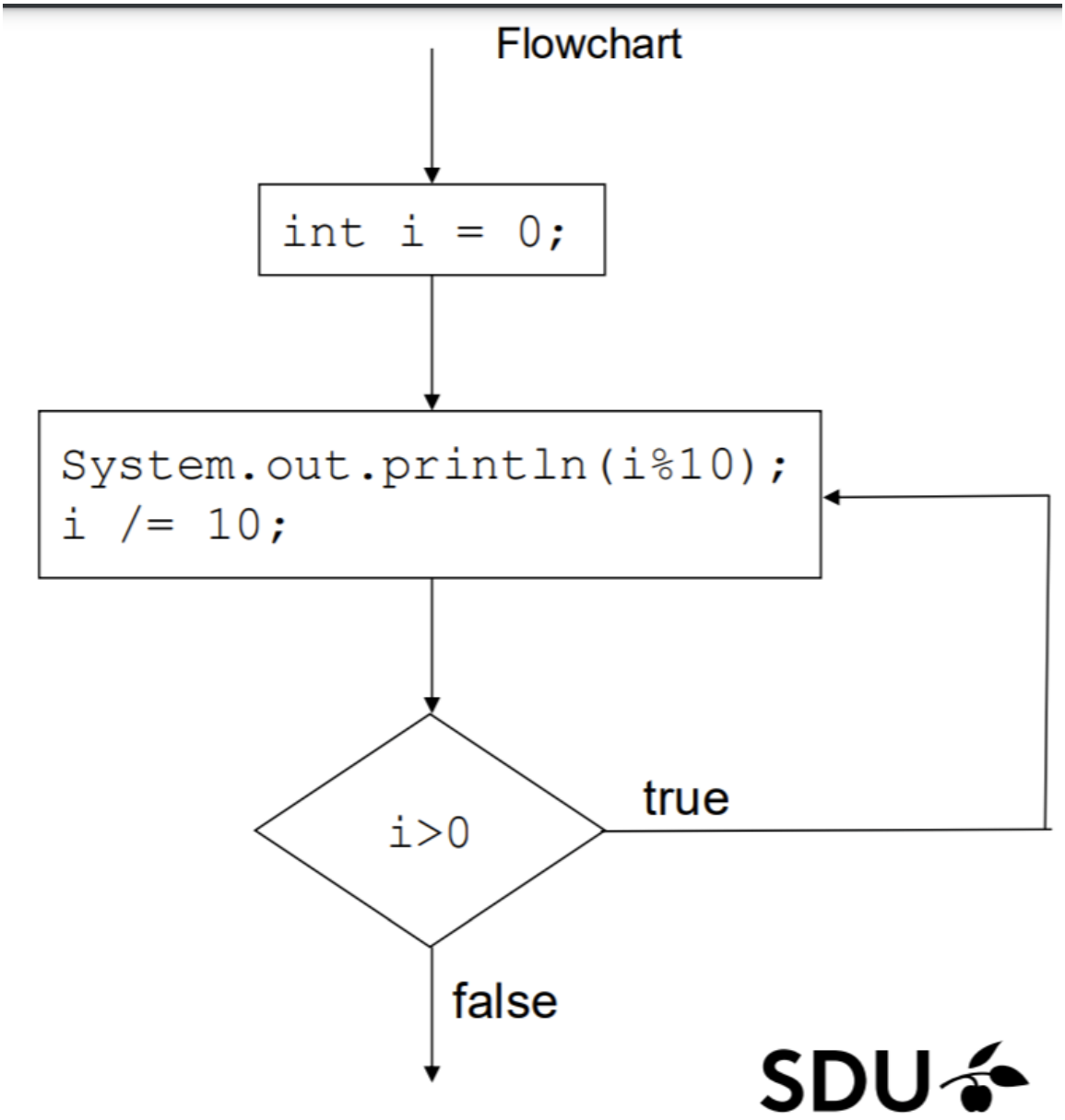
▼ **Do-while**

som en et while-loop, men den evaluerer *continuation condition* **efter** statement.

syntaks: do <statement> while (<condition>);

```
do {
    <doSomething>
} while ( <boolean-expr>);
```

```
int i = 12345;
do {
    System.out.println(i%10);
    i /= 10;
} while (i>0);
```



▼ **For loop**

Syntaks: for (<statements> ; <condition> ; <statement>) { <statement> }

initialization condition update body

```
for (int i=0 ; i<10 ; i++) {
    System.out.println(i);
}
```

VS

```
for (years=0 ; years<5 ; years++) {
    interest = principal * rate;
    principal += interest;
    System.out.println(principal);
}
```

```
years = 0; // init
while (years<5) { // condition
    interest = principal * rate;
    principal += interest;
    System.out.println(principal);

    years++; // update
}
```

▼ **Nestede loops**

Et loop inde i et andet loop

```
for (int y = 1 ; y <= 10 ; y++) {
    for (int x=1 ; x<=y; x++)
    { //
        System.out.printf("%4d", x*y);
    }
    System.out.println("");
}
```

▼ **Opgaver**

4.1 Definition

Hvad er en boolean?

En boolean er en statement som er TRUE eller FALSE

4.2 Oprettelse

Hvilke måder kan man få en boolsk værdi i spil?

```
boolean goalFailed = false;
boolean timeLeft = false;
boolean succes = !goalFailed && timeLeft;
```

4.3 Købsbeslutning

```
double price = 599.95;
double budget = 1000.0;
boolean requiredReading = true;
boolean shouldBuy= price < budget && requiredReading;
```

Forklar sidste linje og fokuser på:

• I hvilken rækkefølge bliver hvad udregnet?

Først regnes det ud, og price < budget er TRUE, derefter om requiredReading er TRUE, og til sidst om de begge er TRUE

- **Hvilke værdier (navngivne eller ej) udføres de enkelte operatorer på?**

Price og budget, er double, og requiredReading er en boolsk type der angiver TRUE

- **Hvilke typer har disse værdier?**

Hhv. **double** og **boolean**

- **Hvad repræsenterer variablen shouldBuy?**

Den fortæller både om prisen er mindre end budgettet, og om 'bogen' er requiredReading. Hvis begge er sande, siger variablen TRUE

4.4 Terninger

Skriv et program, hvori
1. Værdien af et terningeslag er gemt i en variabel.
2. Opret en boolsk variabel og tildel den en værdi der repræsenterer hvorvidt denne værdi er lige og større end 3.
3. Udskriv den boolske variabel

```
byte terning = 4;
        boolean terningslag = terning >= 3;
        System.out.println(terningslag);
```

Med tilfældige tal, hvor den roller 6 gange:

```
public class terninger {
    public static void main(String[] args) {
        for (int i = 1; i <= 6; i++)
        {
            int terning = (int) (Math.random()*(6-1))+1;

            boolean terningslag = terning >= 3;
            System.out.println(terning + " " + terningslag);
        }
    }
}
```

5.3 Juleudsalg

Skriv et program, hvori
1. En variabel indeholder et antal sekunder siden nytår (alle måneder antages at være 30 dage lange).
2. En anden variabel indeholder en pris på 599,95 dkr.

3. Der gives et 30% tilskud hvis det er Jul.
4. Udregn den gældende pris (eventuelt tilskud medregnet) og gem denne i en variabel.
5. Udskriv denne variabel.

```
  public static void main(String[] args) {
         //sekunder fra nytår til jul = 30585600
         double sekunder_siden_nytaar = 30585600;
         double dage_siden_nytaar = sekunder_siden_nytaar / 60 / 60 / 24;
         double pris = 599.95;
         double rabat = 0.7;

         if((dage_siden_nytaar >= 354) && (dage_siden_nytaar <= 355))
         {
             pris *= rabat;
         }

         System.out.println(pris);
     }
 }
```

6.1 Celcius til Fahrenheit

Skriv et program, hvori
• Der udskrives en tabel af matchende Celcius of Fahrenheit værdier.
• Der skal være ét sæt matchende værdier per linje.
• Listen skal starte med -5◦C og slutte ved 40◦C.
• Listen skal have én linje for hver 0,5◦C.

```
public class Main {

  public static void main(String[] args) {
    double fahrenheit;
    for (double celsius = -5; celsius <= 40; celsius+=0.5) {
      fahrenheit = 32 + (celsius * 9 / 5);
      System.out.println(celsius + " ºC = " + fahrenheit + " ºF");

    }
  }
}
```

6.2 Celcius til Fahrenheit i Omvendt Rækkefølge

Omskriv programmet fra opgave 6.1 til at vende rækkefølgen om.

```
public class fahrenheitcelsius {
    public static void main(String[] args) {
        double celsius;
        for (double fahrenheit = -5; fahrenheit <= 40; fahrenheit += 0.5) {
            celsius = (fahrenheit - 32) / 1.8;
```

```
            System.out.println(fahrenheit + " F =" + celsius + " C");
        }
    }
}
```

6.4 Areal af cirkler

Skriv et program der udregner og udskriver arealet (π · r2) af tre cirkler med radius på hhv. 1, 3 og 5.

```
public class cirkler {
    public static void main(String[] args) {
        double areal;
        double pi = 3.14159;
        for (double radius = 1; radius <= 5; radius += 2) {
            areal = radius * radius * pi;
            System.out.println("area is " + areal + " radius is " + radius);
        }
    } //eks. area is 78.53 radius is 5.0
}
```

▼ 6.5 Længden af en Måned

Skriv et program, hvori
1. En måneds nummer gemmes i en variabel.
2. Udregn og udskriv antallet af dage i denne måned (vi antager at det ikke er skudår).

```
public static void main(String[] args) {
        byte january = 31;
        byte february = 28;
        byte march = 31;
        byte april = 30;
        byte may = 31;
        byte june = 30;
        byte july = 31;
        byte august = 31;
        byte september = 30;
        byte october = 31;
        byte november = 30;
        byte december = 31;

        int month = 5;
        String monthString;
        switch (month) {
            case 1:  monthString = "Days in January = " + january;
                break;
            case 2:  monthString = "Days in February = " + february;
                break;
            case 3:  monthString = "Days in March = " + march;
                break;
            case 4:  monthString = "Days in April = " + april;
                break;
            case 5:  monthString = "Days in May = " + may;
                break;
```

```
            case 6:  monthString = "Days in June = " + june;
                break;
            case 7:  monthString = "Days in July = " + july;
                break;
            case 8:  monthString = "Days in August = " + august;
                break;
            case 9:  monthString = "Days in September = " + september;
                break;
            case 10: monthString = "Days in October = " + october;
                break;
            case 11: monthString = "Days in November = " + november;
                break;
            case 12: monthString = "Days in December = " + december;
                break;
            default: monthString = "Invalid month = ";
                break;
        }
        System.out.println(monthString);
    }
}
```

▼ **I et array:**

```
public static void main(String[] args) {
        int index = 5;
        String[] months = { "January", "February", "March", "April", "May",
"June", "July", "August", "September", "October", "November", "December"};
        int[] day_count = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
        System.out.println(months[index-1] + " has " + day_count[index-1] + " days!");
    }
}
```

6.6 Primtal

Skriv et program, der udregner alle primtal under 1.000.000, og udskriver det største.

```
public class numberprime {
    public static void main(String[] args) {
        for (int number = 1000000; number > 1; number--) {
            boolean isPrime = true;
            //kontrol om tallet ikke er prim:
            for (int divisor = 2; divisor < (number / 2); divisor++) {
                if ((number % divisor) == 0) {
                    isPrime = false;
                    break;
                }
            }
            //Hvis tallet er prim, print tallet
            if (isPrime) {
                System.out.println(number);
                break;
            }
        }
    }
}
```