Tilldelningsoperator =

Ökningsoperatorer ++, += *=

Minskningsoperarator --, -=, /=

Aritmetiska operatorer +, -, *, /, %

Jämförelseoperatorer ==, !=, <, <=, >, >=

Booleska operatorer (logiska) &&, | |, !

Villkorsoperator (ternary operator, tredelad operator)

Operatorer – en överblick

Tilldelningsoperator =

```
int myAge = 30;
```

Tilldelningsoperatorn används när vi tilldelar en variabel ett värde.

Operatorer

++ Inkrementeringsoperator, -- dekrementeringsoperator

```
    int myAge = 30;
    myAge++; // Ökar värdet med 1
//   myAge = myAge + 1;
    System.out.println(myAge);

    int count = 0;
    while (count < 10){
        System.out.println("Denna text skrivs ut 10 ggr");
        count++; // Ökar värdet med 1
//       count--; // Minskar värdet med 1
    }
```

Inkrementeringsoperatorn används när vi vill öka värdet på en variabel med 1

Dekrementeringsoperatorn används när vi vill minska värdet på en variabel med 1

Operatorer

++ Inkrementeringsoperator, -- dekrementeringsoperator

++, -- före eller efter variabel, vad är skillnaden?

```
int myAge = 30;
myAge++; // Läser först variabeln, utför därefter operation
++myAge; // Utför först operation, därefter läses variabeln
System.out.println(myAge);
System.out.println(myAge++);
System.out.println(myAge);
System.out.println(++myAge);
```

```
"C:\Program Files\Amazon Corretto\jdk11.0.3_7\bin\java.exe"
32
32
33
34

Process finished with exit code 0
```

Operatorer

Operatorer

Tilldelningsoperatorn kan användas tillsammans med aritmetiska operatorer för att utföra en operation, så kallad ökningsoperator eller minskningsoperator

```
        int myAge = 30;
        myAge += 5; // Ökar värdet med 5
//       myAge = myAge + 5;
        System.out.println(myAge);
```

```
"C:\Program Files\Amazon Corretto\jdk11.0.3_7\bin\java.exe"
35

Process finished with exit code 0
```

Varianter

+= -= *= /=

Operatorer

Aritmetiska operatorer +, -, *, /, %

```
myAge = myAge + 5;
myAge = myAge - 5;
myAge = myAge * 5;
myAge = myAge / 5;
System.out.println(myAge - 2);
int a = 2;
int b = 3;
int sum = a + b;
```

Använd inte datatypen int när du räknar med division,
använd förslagsvis double som kan hantera decimaltal

Modulus % - Beräkna resten av en heltalsdivision, cirkulär struktur för cirkulära arrayer.

```
10 % 1 = 0
10 % 2 = 0
10 % 3 = 1
10 % 4 = 2
10 % 5 = 0
10 % 6 = 4
19 % 10 = 9
```

```java
boolean isOdd;
for (int i = 0; i < 11; i++) {
isOdd = i % 2 == 0;
System.out.println(i + " är ett " + (isOdd ? "jämnt":"udda") + " tal");
}
```

```
"C:\Program Files\Amazon Corretto\jdk11.0.3_7\bin\java.exe"
0 är ett jämnt tal
1 är ett udda tal
2 är ett jämnt tal
3 är ett udda tal
4 är ett jämnt tal
5 är ett udda tal
6 är ett jämnt tal
7 är ett udda tal
8 är ett jämnt tal
9 är ett udda tal
10 är ett jämnt tal

Process finished with exit code 0
```

- Kontrollera om ett tal är udda eller jämnt
- Utföra något varannan, var tredje, var fjärde, var... gång
- Bra till animation (sprites-index)

```java
for (int i = 1; i < 10; i++) {
    System.out.print(i);
    if (i % 3 == 0)
        System.out.println();
}
```

```
123
456
789
```

Operatorer

Jämförelseoperatorer används för att jämföra två värden från exempelvis två variabler, ett booleskt värde returneras (sant eller falskt)

== Likhetsoperator

!= Skiljt från-operator

> Större än

>= Större än eller lika med

< Mindre än

<= Mindre än eller lika med

Operatorer

Jämförelseoperatorer används för att jämföra två värden från exempelvis två variabler, ett booleskt värde returneras (sant eller falskt)

Operator	Beskrivning	Exempel	Utfall
==	Likhetsoperator	2 == 5	false
!=	Skiljt från-operator	2 != 5	true
>	Större än	2 > 2	false
>=	Större än eller lika med	2 >= 2	true
<	Mindre än	5 < 2	false
<=	Mindre än eller lika med	2 <= 5	true

Operatorer

Booleska operatorer (logiska operatorer) används för att jämföra två+ uttryck eller för att invertera en boolean
AND, OR, NOT

Operator	Beskrivning	Exempel	Utfall
&&	AND	2 == 2 && 3 <= 6	true
||	OR	2 != 5 || 3 == 2	true
!	NOT	!(2 > 2)	true

Operatorer

Villkorsoperatorn kan användas istället för en enkel if-sats

condition ? värde1 : värde2

```
int age = 55;
int retirementAge = 67;
String whereabouts;
if(age < retirementAge)
    whereabouts = "arbetet";
else
    whereabouts = "hemmet";
```

```
int age = 55;
int retirementAge = 67;
String whereabouts = age < retirementAge ? "arbetet" : "hemma";
```

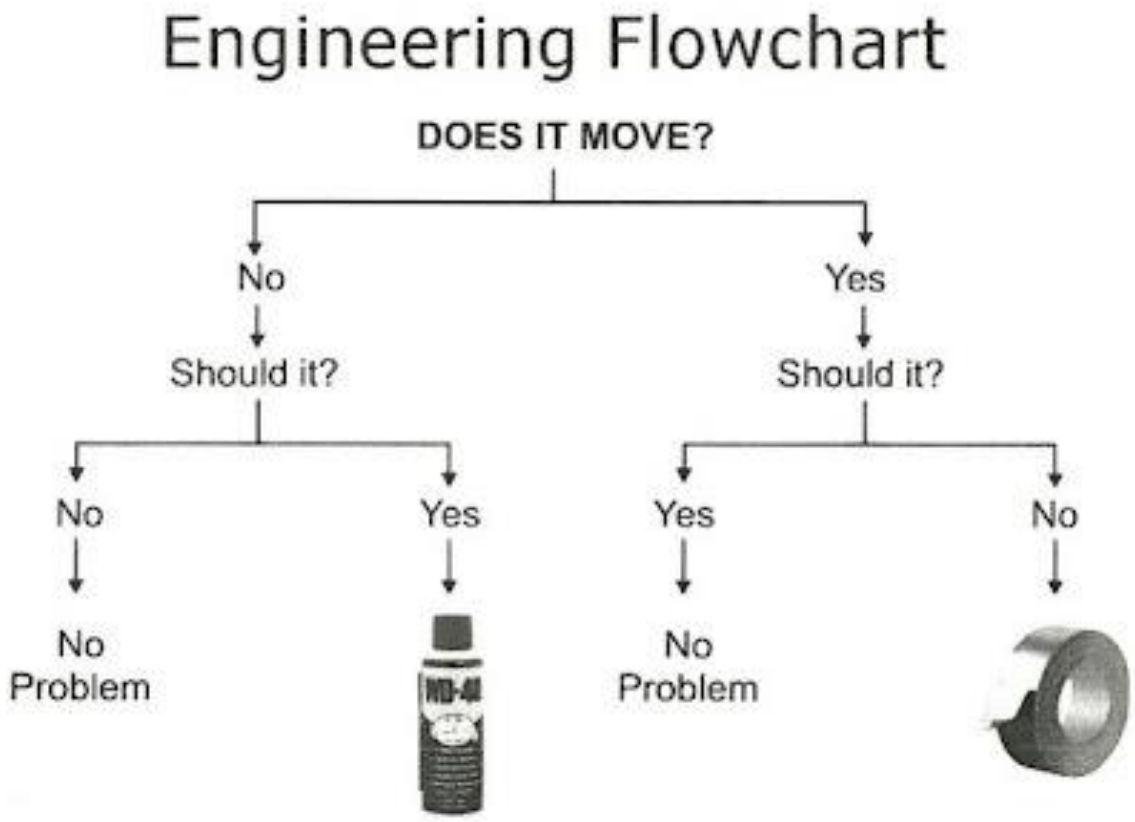
Operatorer

Flödeskontroll

Med hjälp av loopar och villkorssatser kan vi få ett dynamiskt flöde i vår kod:

- Upprepa en sekvens av instruktioner ett visst antal gånger eller tills ett visst villkor uppfyllts
- Hoppa över vissa instruktioner pga ett villkor
- Utföra instruktioner om ett visst villkor uppfylls

Detta gör vi med hjälp av iterationer och selektioner



```
7           if(direction.equals("up")){
8               System.out.println("Going up");
9           } else if(direction.equals("down")){
10              System.out.println("Going down");
11          } else if(direction.equals("left")){
12              System.out.println("Going left");
13          } else if (direction.equals("right")) {
14              System.out.println("Going right");
15          } else {
16              System.out.println("Standing still");
17          }
```

```
19          switch (direction){
20              case "up":
21                  System.out.println("Going up");
22                  break;
23              case "down":
24                  System.out.println("Going down");
25                  break;
26              case "left":
27                  System.out.println("Going left");
28                  break;
29              case "right":
30                  System.out.println("Going right");
31                  break;
32              default:
33                  System.out.println("Standing still");
34                  break;
35          }
```

Selektioner

If-sats

```
5           int variable1 = 2, variable2 = 5;
6           if(variable1 < variable2){
7               System.out.println("Värdet i variable1 är mindre än i variable2");
8           }

13          if(variable1 < variable2){
14              System.out.println("Värdet i variable1 är mindre än i variable2");
15          } else {
16              System.out.println("Värdet i variable1 är inte mindre än i variable2");
17          }

23          if(variable1 < variable2){
24              System.out.println("Värdet i variable1 är mindre än i variable2");
25          } else if(variable1 > variable2){
26              System.out.println("Värdet i variable1 är större än i variable2");
27          } else {
28              System.out.println("Variable1 har samma värde som variable2");
29          }
```

Selektioner

Indentering bristfällig

```
15          boolean myCondition = true;
16          String password = "commodore64";
17
18          if(myCondition){
19              System.out.println("If condition is true, do this:");
20              if(password.equals("commodore64")){
21                  System.out.println("You have access!");
22              }
23          else {
24                  // kodrad
25                  // kodrad
26                  // kodrad
27                  // kodrad
28                  // kodrad
29                  // kodrad
30                  // kodrad
31          }
32          }else {
33              // kodrad
34          }
```

Indentering OK

```
15          boolean myCondition = true;
16          String password = "commodore64";
17
18          if(myCondition){
19              System.out.println("If condition is true, do this:");
20              if(password.equals("commodore64")){
21                  System.out.println("You have access!");
22                  // kodrad
23              } else {
24                  // kodrad
25                  // kodrad
26                  // kodrad
27                  // kodrad
28                  // kodrad
29                  // kodrad
30                  // kodrad
31                  // kodrad
32              }
33          } else {
34              // kodrad
35          }
```

Selektioner

Switch-sats

```
19          switch (direction){
20              case "up":
21                  System.out.println("Going up");
22                  break;
23              case "down":
24                  System.out.println("Going down");
25                  break;
26              case "left":
27                  System.out.println("Going left");
28                  break;
29              case "right":
30                  System.out.println("Going right");
31                  break;
32              default:
33                  System.out.println("Standing still");
34                  break;
35          }
```

Enhanced switch-sats

```
54          switch (direction) {
55              case "up" -> System.out.println("Going up");
56              case "down" -> System.out.println("Going down");
57              case "left" -> System.out.println("Going left");
58              case "right" -> System.out.println("Going right");
59              default -> System.out.println("Standing still");
60          }
```

Selektioner