

## 2 Primitive datatyper

Det er nemt at se forskel på primitive og ikke primitive<sup>1</sup> datatyper i Java. Primitive datatyper staves med lille begyndelses bogstav, ikke primitive datatyper staves med stort begyndelses bogstav. Primitive datatyper kan repræsenteres i en eller flere bytes og kan sammenlignes med en operator '=='. Ikke primitive datatyper er klasser som har tilknyttet funktioner. Hver gang man møder en ikke primitiv datatype, bør man se i dokumentationen, om der er metoder som man kan bruge. Ikke primitive datatyper kan ikke sammenlignes direkte men kun ved hjælp af en funktion fx. equals().

Tabel 1: Liste over datatyper.

Ikke primitive	Primitive
String	integer
Array	float
Klasser	char
Interfaces	boolean
	byte
	short
	long
	double

Denne opgave handler om at forstå de forskellige datatyper.

### 2.1 Opgave

Undersøg for hver primitiv datatype, hvor meget plads (i bytes) hver datatype bruger. Det kan du finde her: <https://data-flair.training/blogs/java-data-types/>. Eksempel: Datatypen integer fylder 4 bytes og at den maksimale værdi er:  $2^{31} - 1 = 2.147.483.648$ . Noter dine resultater for hver af de 8 primitive datatyper.

### 2.2 Opgave

Skriv et program som beviser hvilke minimums- og maksimumsværdier for de primitive datatype kan indeholde. Find evt. inspiration i programmet i figur 5 på side: 6. Noter dine resultater for hver af de 8 primitive datatyper.

<sup>1</sup><https://data-flair.training/blogs/java-data-types/>

## 2.3 Opgave

De to datatyper float og double er ikke lige nøjagtige. Det kan de se ved følgende opgave: Hvad giver kvadratroden af 2 gange med kvadratroden af 2? ( $\sqrt{2} * \sqrt{2}$ ). Lav et først et program med `sqrt()` som returnerer en float og herefter med `Math.sqrt()` som returnerer en double.

- Slå funktionen `sqrt()` op i Processings dokumentation. Deklarer en variabel `f` af type float og initier med værdien af `sqrt(2)`. Brug kommandoen `println(f)`; til at udskrive værdien af `f`.
- Du kan ikke slå funktionen `Math.sqrt()` op i Processings dokumentation. Men den fungerer på helt samme måde som `sqrt()`, den returnerer bare en double. Deklarer en variabel `d` af type double og initier med værdien af `Math.sqrt(2)`. Brug kommandoen `println(d)`; til at udskrive værdien af `d`.

Forklar forskellen på de to resultater.

## 3 Ikke-primitive datatyper

Vi skal i dette kapitel arbejde med ikke-primitive datatyper. Det gælder for de primitive datatyper at de kan repræsenteres i en byte. Ikke primitive datatyper kræver meget mere plads. Primitive datatyper er foruddefineret i java, det er ikke primitive datatyper ikke, på nær String. Ikke primitive datatyper kan bruges til at kalde funktioner og bestemte operationer.

Den ikke primitive datatype String, er en række af karakterer (Chars). Der er til datatypen knyttet en række metoder:

- `toUpperCase()` Gør alle karakterer til store bogstaver
- `toLowerCase()` Gør alle karakterer til små bogstaver
- `substring()` Returnerer en ny streng som er en del af den originale streng.
- `length()` Returnerer en heltalsværdi som repræsenterer antallet af karakterer i strengen.
- `indexOf()` Returnerer indeks af den første forekomst af substreng i strengen.
- `equals()` Sammenligner to strenge
- `charAt()` Returnerer karakteren på den af index angivne plads.

Eksempler på andre ikke primitive datatyper: Strings, Arrays, Classes, Interface, etc.

### **3.1 Opgave**

Lav opgaver på [codingbat.com](https://codingbat.com): String 1

## 14 OOP

OOP betyder objekt orienteret programmering.

## 15 Opgave

Læringsmål: Du skal i denne opgave arbejde med klasser, arrays og input fra mus. På github, kan du finde et program house, med en klasse Room. Programmet tegner et hus. For hvert objekt af klassen Room, du opretter, tegner programmet et ekstra værelse. Et værelse er 100x100 pixels. Der er i alt plads til tre rum i bredden og tre rum i højden.

1. Undersøg programmet house. Hvor deklarerer jeg en variabel af datatypen Room, og hvad hedder min variabel?
2. Hvor mange funktioner er der i programmet house?
3. Når jeg oprettet objektet, altså initierer variabelen med en værdi, hvor mang parameter skal der så bruges?
4. Undersøg klassen Room, attributter.
  - (a) Hvor mange er der og hvad bruges de til?
  - (b) Hvor mange er konstanter og hvor mange er variabler?
  - (c) Hvad er sammenhængen mellem antallet af parameter og klassens tilstand?
5. Find i programmet house, den eller de linjer kode som tegner taget på huset. Dette ansvar bør ligge i klassen. Opret en funktion drawRoof() i klassen Room og flyt de linjer kode over i klassen.
6. For at kunne håndtere flere rum, kan vi oprette et array. Ændre variabelen house, til et array af datatype Room. Jeg kunne godt tænke mig to etager, så derfor skal længden på house være 6. `Room[] house = new Room[6];`
7. Tilføj nu følgende rum: Kitchen, Livingroom, Toilet, Bedroom og Bathroom. Husk at korrigere X og Y positionen.
8. For at få programmet til at udskrive alle rum, skal du loop'e igennem dit array og kalde `"house[i].drawRoom();"` brug `"for(int i = 0;i<house.length;i++)"` som løkke struktur.

9. Find ud af, hvad du skal ændre i funktionen `mouseClicked()` for at kunne tænde og slukke lyset i alle rum.
10. Hvor vil du tilføje funktionen `checkHouse()`; (se Figure 11.), som kan finde ud af om du kan gå fra huset med ro i sindet og vide at alt lys er slukket?
11. I `mousedClicked()` er det funktionen selv, som udskriver om lyset er tændt eller slukket. Dette ansvar vil jeg gerne have skrevet over i klassen.

```

void checkHouse(){
    boolean found=false;
    for (int i=0; i< house.length; i++) {
        if (house[i].isLightOn() == true) {
            println ("WOW turn off the light in the "+house[i].getRoomName());
            found = true;
        }
    }
    if (!found) {
        println("All right! You'r ready to go!");
    } else {
        println("you forgot something");
    }
}

```

Figur 11: funktion som undersøger om alt lys er slukket