

# Kodekonventioner

## Navngivning

Generelt skal alle navne være så beskrivende, at man kan forstå indholdet eller funktionen nogenlunde ud fra navnet.

## Klasser

Klassenavne skrives i Pascalcase:

```
1 | public class SomeClass {}
```

Builder-klassers navn skal slutte med *Builder*.

```
1 | public class SomeClassBuilder {}
```

## Metoder

Metodenavne skrives i camelcase:

```
1 | public void someMethod() {}
```

## Felter

Felter skrives i camelcase:

```
1 | private int someInt;
```

Hvis en feltvariabel er ufærdig eller ufuldstændig prependeres navnet med "\_":

```
| private String _notComplete;
```

## Lokale variable

Som udgangspunkt frit valg, men skal helst være nogenlunde forståeligt og sammenhængende.

## Konstanter (static final)

Konstanter skrives altid med all-caps med "\_" mellem ord:

```
1 | public static final SOME_CONSTANT;
```

## Loops

**Normalt:** *i, j, k, ..., etc.*

```
1 | for (int i=0; i<n; i++) {
2 |     for (int j=0; j<n; j++) {
3 |         for (int k=0; k<n; k++) {
4 |             }
5 |     }
6 | }
```

**Ved reference til koordinater (eg. i matrix):** *x, y, ..., etc.*

**Ved reference til rækker og kolonner:** *r, c.*

## Parametre

I konstruktorer skal parametre have samme navne som de felter de repræsenterer:

```
1 | public class Model {
2 |     private int someData;
3 |
4 |     public Model(int someData) { this.someData = someData }
5 | }
```

Generelt skal parametre skrives i camelcase.

## Struktur

### Placering af tuborgklammer

Den første tuborgklamme efter definition eller control-flow statement skrives på samme linje som dette:

```
1 | if (something) {  
2 |     // code  
3 | }
```

Ved meget korte metodedefinitioner (f.eks. gettere / settere) eller korte control-statements:

```
1 | public String getValue() { return value; }
```

### Field-keywords

Keywords før feltdefinitioner skrives i følgende rækkefølge:

```
1 | public/private/protected transient static final
```

### Indention

Al kode skal indenteres med ét tab-tegn pr. niveau:

```
1 | if (something) {  
2 |     while (something-else) {  
3 |         // code  
4 |     }  
5 | }
```

### Gettere/settere

Alle felter i klasser skal være private (undtagen i særlige tilfælde)

og skal have en getter og kan have en setter:

```
1 | private int count;  
2 |  
3 | public int getCount() { return count; }  
4 | public void setCount(int count) { this.count = count; }
```

Setteres parametre skal navngives efter feltnavnet den setter.

## Builder-klasser

Alle klasser, hvis setter-metoder returnerer *this*, og som bruges til at opbygge en anden type objekt, er Builder-klasser.

Eksempel på setter i builder-klasse:

```
1 | public SomeClassBuilder setCount(int count) { this.count = count; }
```

## Dokumentation / kommentarer

### Kodekommentarer

Normale kommentarer er altid single-line og der skal være mellemrum mellem `//` og kommentaren:

```
1 | // somecomment  
2 | // othercomment
```

### JavaDoc

Al ikke-triviel kode skal dokumenteres med JavaDoc-kommentarer.

Ved beskrivelse af metode:

```
1 | /**  
2 |  * Beskrivelse af funktion
```

```
3 | * gerne flere linjer
4 | *
5 | * @param parameternavn beskrivelse
6 | * @param parameternavn2 beskrivelse
7 | *
8 | * @return beskrivelse
9 | *
10 | * @throws type beskrivelse
11 | */
```

Ved beskrivelse af klasser:

```
1 | /**
2 | * Beskrivelse af, hvad klassen gør
3 | */
```