Kodestil i Java

Mathias Ciarlo Thorstensen
Kristian Gregorius Hustad
Sist oppdatert av Silje Merethe Dahl

Innholdsfortegnelse

Indentering (innrykk)

Kortversjon: Det anbefales å bruke 4 mellomrom som innrykk.

Lang versjon: I mange teksteditorer fungerer innrykk slik at når du trykker TAB så settes det egentlig inn et spesielt "tomrom"-symbol. Problemet er at det finnes flere av disse symbolene, og hvilket som brukes kan variere avhengig av hvilket program som brukes for å redigere tekstfilen.

Hvis mulig bør du derfor justere innstillingene i din teksteditor slik at et trykk på TAB-knappen setter inn 4 vanlige mellomrom heller enn et stort.

Hvor skal jeg indentere?

Innrykk på linjen etter en krøllparentes-blokk starter, altså inni følgende:

- Klasser
- Metoder
- Løkker
- If-tester

Eksempel på korrekt indentering:

```
class Test {
    public static void main(String[] args) {
        System.out.println("This program is correctly indented.");

        // Here's a while loop
        while (true) {
            if ("abcde".equals("abcde")) {
                System.out.println("Yay!");
            }
            else {
                System.out.println("Oh no!");
            }
        }

        System.out.println(someFancyMethod());
    }

    // Here's another method
    static String someFancyMethod() {
        return "Fancy.";
    }
}
```

Store og små bokstaver i sammensatte ord

Sammensatte ord i Java-kode skal skrives i camelCase.

- Klasser skal ha stor forbokstav
- Metoder og variabler skal ha liten forbokstav
- Hvert nye ord etter det skal ha stor forbokstav.

Eksempel:

```
class KaffedrikkendeStudentMedDeltidsjobb {
    private int kopperDrukket = 0;

    public void drikkKoppMedKaffe() {
        kopperDrukket++;
    }
}
```

Variabelnavn

- Bruk variabelnavn som sier noe om innholdet. a, foo, tmp, tmp2 er dårlige variabelnavn – sum, stringResult, sunnyDayCount er gode.

Unngå for lange linjer

Dette er kjempeviktig for å holde programmene ryddige og enkle å lese.

- Prøv å holde deg under 100 tegn, historisk var 80 linjer standarden på grunn av bredden i terminalen.
- Sett kommentarer på egen linje
- Lange utskrifter kan splittes på flere linjer ved å sette et + på slutten av linjen og så bare fortsette på linjen under

Luft i koden

Luft i koden generelt og "avsnitt" gjør koden mer oversiktelig og mye enklere å lese. Bruk mellomrom foran og bak operatorer (+, -, =, *, /, ==, osv.).
Hvor skal jeg ha avsnitt?
Sett avsnitt naturlige steder i koden. Jeg foretrekker følgende steder:

- Før du definerer en ny klasse
- Over metoder (Avsnitt, kommentar, metode)
- Over kommentarlinjer
- Over if-tester
- Over løkker

Hvis du vil kommentere noen av elementene over, sett avsnittet over kommentaren slik at det blir avsnitt, kommentar, element.

Eksempel på luft og gode avsnitt:

```java
import java.util.Scanner;
class Test {
    public static void main(String[] args) {
        System.out.println("This program has air and is easy to read.");

        // Get user input and convert it to a String array
        Scanner scan = new Scanner(System.in);
        String scannedString = scan.nextLine();

        // Print the average word length
        System.out.println(averageWordLength(scannedString));
    }

    // This method returns the average length of the words in the input string
    static String averageWordLength(String input) {
        int averageLength = 0;

        // Convert string to array by splitting it on space
        String[] arr = input.split();

        // Loop through the array and sum the word lengths
        for (int i = 0; i < arr.length; i++) {
            averageLength += arr[i].length();
        }

        // Divide by the number of words
        return averageLength / arr.length;
    }
}
```