

Uppräkningstyper, enumeration types

```
public static final int SEASON_WINTER = 0;  
public static final int SEASON_SPRING = 1;  
public static final int SEASON_SUMMER = 2;  
public static final int SEASON_FALL   = 3;
```

Inte bra!

```
JLabel lab = new JLabel("Konstigt", SEASON_WINTER);
```

Använd uppräkningstyper istället:

```
enum Season { WINTER, SPRING, SUMMER, FALL }
```

- En *klass* med fyra konstanta fördefinierade object.
- Privat konstruktör.

```
Season se = Season.SUMMER;
```

```
If (se == Season.SPRING)
```

```
se = new Season();    // FEL!!
```

```
public class Hiss {
    public enum Rörelse {NERÅT, STILLA, UPPÅT}
    // instansvariabler
    private Rörelse riktning = Rörelse.STILLA;
    private int våning;
    // Metoder
    public Rörelse avläsRiktning() {
        return riktning;
    }
    ...
}
```

```
Hiss h = new Hiss();
if (h.avläsRiktning() == Hiss.Rörelse.STILLA)
    System.out.println("Hissen är stilla");
```

```

enum Size {
    SMALL("S"), MEDIUM("M"), LARGE("L"), EXTRA_LARGE("XL");

    private String abbreviation;           // egenskap för varje värde

    private Size(String abbreviation) {    // konstruktor
        this.abbreviation = abbreviation;
    }

    public String getAbbreviation() {      // avläs egenskap
        return abbreviation;
    }
}

public class EnumTest {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a size: (SMALL, MEDIUM, LARGE, EXTRA_LARGE) ");
        String input = in.next().toUpperCase();
        Size size = Enum.valueOf(Size.class, input);
        System.out.println("size=" + size);
        System.out.println("abbreviation=" + size.getAbbreviation());
    }
}

```