

Vediamo come usare le enum. NOTA: questo è un argomento avanzato, utile ma non essenziale.

Definiamo nel nostro package due enum, che sono dei tipi di dati simili alle classi ma più semplici, senza ereditarietà e con una serie di simboli associati.

Enum Seme

```
package it.unitn.lingprogmod1.esercitazione2;

public enum Seme {C, Q, F, P}
```

Enum Valore

Dato che i simboli seguono le stesse regole dei nomi delle variabili, non possono iniziare con un numero.

```
package it.unitn.lingprogmod1.esercitazione2;

public enum Valore {A, _2, _3, _4, _5, _6, _7, _8, _9, _10, J, Q, K}
```

Variazioni nella Class Carta

In Carta scriveremo ci basterà cambiare la definizione delle due variabili di istanza ed il costruttore. Era:

```
package it.unitn.lingprogmod1.esercitazione2;
public class Carta {
    String valore;
    String seme;
    public Carta(String valore, String seme) {
        this.valore = valore;
        this.seme = seme;
    }
    ...
}
```

e diviene:

```
public class Carta {
    Valore valore;
    Seme seme;
    public Carta(Valore valore, Seme seme) {
        this.valore = valore;
        this.seme = seme;
    }
}
```

Variazioni nella classe Mazza

Nella classe Mazza non ci serve più di definire i simboli, che abbiamo portato nei due files esterni Valore.java e Seme.java. Quindi non abbiamo più:

```
String[] valore = {"A", "2", "3", "4", "5", "6", "7", "8", "9", "10",
"J", "Q", "K"};
String[] seme = {"C", "Q", "F", "P"};
```

Cambierà il ciclo che creava le carte. Era:

```
for (String v : valore) {  
    for (String s: seme) {  
        Carta c = new Carta(v, s);  
        this.add(c);  
    }  
}
```

e diviene:

```
for (Valore v : Valore.values()) {  
    for (Seme s: Seme.values()) {  
        Carta c = new Carta(v, s);  
        this.add(c);  
    }  
}
```