



STRUTTURA CLASSE

```
public class NomeClasse{ //DICHIARAZIONE CLASSE

    private TIPO nomeAttributo; //DICHIARAZIONE ATTRIBUTI

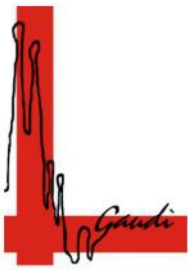
    public NomeClasse (TIPO nomeAttributoGenerico){ // COSTRUTTORE PARAM.
        nomeAttributo = nomeAttributoGenerico;
    }

    public NomeClasse (){ //COSTRUTTORE DI DEFAULT
        nomeAttributo = ""; //NEL CASO STRINGA
        nomeAttributo = -1; //NEL CASO INT/DOUBLE
        nomeAttributo = true; //NEL CASO BOOLEAN
    }

    public TIPO getNomeAttributo() { //STRUTTURA GET
        return nomeAttributo;
    }

    public void setNomeAttributo(TIPO nomeAttributo) { //STRUTTURA SET
        this.nomeAttributo = nomeAttributo;
    }

    public TipoRestituito/void NomeMetodo (TIPO EventualeValoreInIngresso){ //STRUTTURA METODO
        .
        .
        .
        .
    }
}
```



CONTRUCCI PRINCIPALI

IF

```
if ((condizione 1) && (condizione 2) ){  
.....  
}
```

DA RICORDARE:

Diverse condizioni si uniscono con && o ||

Uguaglianza tra numeri si fa con ==

Uguaglianza tra stringhe si fa con .equal()

WHILE

```
while (ESPRESSIONE){  
.....  
}
```

DA RICORDARE:

Il while non viene eseguito sempre!

Deve rispettare la condizione nell'espressione

DO WHILE

```
do{  
.....  
}while(ESPRESSIONE)
```

DA RICORDARE:

Il do while viene eseguito almeno una volta!

FOR

```
for(inizializzazione;terminazione;aggiornamento){  
.....  
}
```

COSTRUTTORE COPIA

```
public NomeClasse (TipoClasse nomeOggetto){  
attributoClasse=nomeOggetto.getAttributoClasse();  
}
```

DA RICORDARE:

Costruttore di copia e getStatus servono per evitare di creare riferimenti alla stessa cella di memoria da parte di due oggetti diversi. Nel costruttore di copia NomeClasse e TipoClasse sono la stessa cosa!

GETSTATUS

```
public TipoClasse getStatus (){  
TipoClasse nomeOggetto = new TipoClasse (nomeAttributo1, nomeAttributo2);  
return nomeOggetto;  
}
```