

Esercitazioni del corso di:

ALGORITMI E STRUTTURE DATI

Tutor: Francesca Piersigilli

email: francesca.piersigilli@unicam.it

Java (Sun Microsystem)

Esercitazioni in Java:

- Modo efficace per rappresentare gli algoritmi e strutture dati
- Implementazioni efficienti, compatte, portabili
- Senza l'utilizzo dei “package”

Vantaggi

- Applicazioni sicure
- Applicazioni indipendenti dalla piattaforma
- Utilizzabili da qualsiasi punto della rete
- Da affiancare all'HTML, rendendolo sicuro

Particolarità

- Creazione *.java*
- Il sorgente compilato diventa un codice detto bytecode (diverso dal linguaggio macchina): *.class*
- Per eseguire il bytecode serve uno strumento particolare: la *JVM* che interpreta ed esegue
- Grazie alla JVM abbiamo portabilità
- JVM implementata nei browser per eseguire le *applet* (programmi incontrati nella rete)

La programmazione Java

- D'obbligo la programmazione ad oggetti: dati e funzioni racchiusi in *classi*
- *Classi*: strutture astratte che possono essere istanziate e quindi creare oggetti
- La classe definisce tutte le proprietà degli oggetti appartenenti a quella classe, detti attributi, e le funzioni che verranno usate per agire su di essi, detti metodi.

Classe “Persone”

Inizio classe persone

Attributo annodinata

Metodo calcolaetà (annoattuale)

Fine classe persone

```
class persone
{
    public int annodinata;
    public int calcolaeta ( int annoattuale )
    {
        return ( annoattuale - annodinata );
    }
}
```

Costruttore

Metodo particolare che può inizializzare alcuni attributi della classe, quando essa viene istanziata.

Particolarità:

- stesso nome della classe
- non ha un valore di ritorno
- esiste il costruttore di default
- i costruttori possono essere più di uno (stesso nome, ma parametri diversi)

Costruttore di “Persone”

```
class persone
{
    public int annodinascita;
    public String Cognome=new String();
    public persone(int annonascita)
    {
        this.annodinascita=annonascita;
    }

    public persone(String Cognome)
    {
        this.Cognome=Cognome;
    }

    public persone(int annonascita , String Cognome)
    {
        annodinascita=annonascita;
        this.Cognome=Cognome;
    }
}
```

Commenti

//questo è un commento su una linea

/*questo è un commento che può stare su più linee
basta che venga terminato con i seguenti due caratteri */

I commenti vengono ignorati dal compilatore

Instanziare oggetti

persone Pietro=new persone(1974);

oppure

persone Pietro=new persone("Castellucci");

oppure

persone Pietro=new persone(1974,"Castellucci");

persone Lina=new persone(1975);

oppure

persone Lina=new persone("Marucci");

oppure

persone Lina=new persone(1975,"Marucci");

Manipolare oggetti

copiare i riferimenti degli oggetti:

```
persone Pietro2=Pietro;
```

Costruiti gli oggetti ne posso invocare i metodi, questo si fa indicando *nomeOggetto.nomeMetodo*, ad esempio è possibile invocare i metodi:

```
Pietro.calcolaeta(2000);  
Pietro2.calcolaeta(2000);  
Lina.calcolaeta(2000);
```