LABORATORIO DI INFORMATICA 1

2014-2015

- parte 5 -

Classi

- **Costrutto base della programmazione ad oggetti**
- **Le useremo come strutture dati**

- **Le Stringhe sono una classe, non un tipo nativo** (int, boolean, double, …)
- **Contenitore di informazioni**

Oggetti

Una classe instanzia (crea) oggetti

Es: nel mondo reale una classe può essere il concetto di persona

le cui istanze sono Marco, Francesca, …

Ciasuno di loro ha i propri dati/proprietà/caratteristiche (età, luogo di nascita, professione, …)

```
class Persona {
    private String nome;
    private int anni;
    private String luogoNascita;

    public Persona(String nome, int anni, String luogoNascita) {
        this.nome = nome;
        this.anni = anni;
        this.luogoNascita = luogoNascita;
    }

    public Persona() {
        this.nome = "Anonimo";
        this.anni = 0;
        this.luogoNascita = "Sconosciuto";
    }
}
```

VARIABILI DELL'OGGETTO

COSTRUTTORE CON PARAMETRI

COSTRUTTORE VUOTO

Costruttore

Serve per **inizializzare gli oggetti che vengono creati**

- **Costruttore vuoto (senza parametri):**

  assegna valori pre-impostati alle variabili

  Come usarlo:

  Persona sconosciuta = new Persona();

- **Costruttore con parametri:**

  assegna i valori che gli vengono passati alle variabili

  Come usarlo:

  Persona mario = new Persona(“Mario”, 12, “Trento”);

Visibilità delle variabili

- **private -> difficilmente accessibili**

  Per leggerle e scriverle è necessario creare dei metodi che restituiscono e modificano il loro valore (questi metodi vengono chiamati **get**ter e **set**ter)

```
private String nome;

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}
```

- **public -> facilmente accessibili**

  Per leggerle e scriverle non sono necessari metodi, ma è buona norma usare nelle classi variabili private

Classi

- Ogni classe risiede in un file separato che ha lo stesso nome della classe

- **Solo il file principale (quello che contiene la logica del programma) contiene il metodo main**

- Le classi hanno l’iniziale maiuscola e utilizzano la notazione CamelCase

- Due classi all’interno dello stesso progetto non possono avere lo stesso nome

Progetto

Classe principale

Classe “secondaria”
Es: Animale

Classe “secondaria”
Es: Persona

- Nella classe principale abbiamo I metodi necessari alla logica del programma
- Nella classe principale è presente il metodo main che dà inizio all’esecuzione del programma

- La classe principale istanzia oggetti invocando il costruttore delle classi “secondarie”

- Le classi secondarie possono creare oggetti delle altre classi secondarie
  (ad esempio una persona può possedere degli animali e salvare i loro dati all’interno di un array)

Esercizio

Create una *class Persona* che abbia:

- Una variabile per il nome
- Una variabile per l'età
- Una variabile per il luogo di nascita
- Un costruttore senza parametri formali che imposti di default l'indirizzo

  Sconosciuto, 15 anni, nato a *Trento*
- Un costruttore con i tre parametri
- I get e set delle variabili

Create inoltre un programma che crei un array contenente 3 persone con dati a vostra scelta e successivamente le stampi

Esercizio

1. Una volta creata la classe persona dell'esercizio precendete e istanziate le persone in un array, create un programma che calcoli l'eta media delle persone all'interno dell'array e la stampi a video

2. Successivamente, aggiungete alle variabili una Stringa che sarà "M" per i maschi e "F" per le femmine, e calcolate l'età media per ciascun sesso.

(per controllare che una stringa sia uguale ad un'altra non potete usare il doppio uguale ==, ma dovete usare il metodo equals, come visto nelle precedenti lezioni

Es: String sesso = "M";

sesso.equals("M") ritornerà true

sesso.equals("F") ritornerà false

…fate attenzione alle maiuscole e alle minuscole!)