

Object Oriented Programming

Prof. Ing. Loris Penserini, PhD

elpense@gmail.com

<https://dblp.org/pid/p/LorisPenserini.html>

Modellare il Mondo Reale

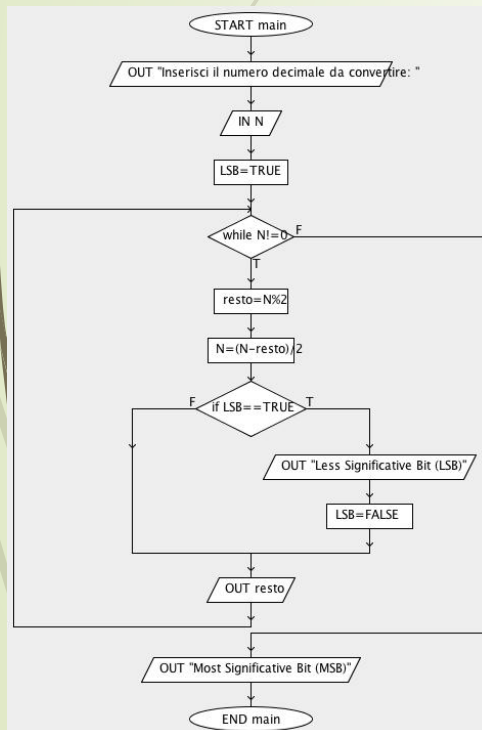
Ciascun paradigma di programmazione fornisce allo sviluppatore un differente approccio concettuale per implementare il pensiero computazionale, cioè la definizione formale della strategia algoritmica utile per affrontare e risolvere un problema del mondo reale.

La OOP è attualmente il paradigma di sviluppo del SW più utilizzato, per questo molti linguaggi del Web che in passato nascevano non ad oggetti ora lo sono diventati, come il C → C++, il PHP dopo la ver. 5, mentre altri sono nati direttamente OO come JAVA (JSP e Servlet).

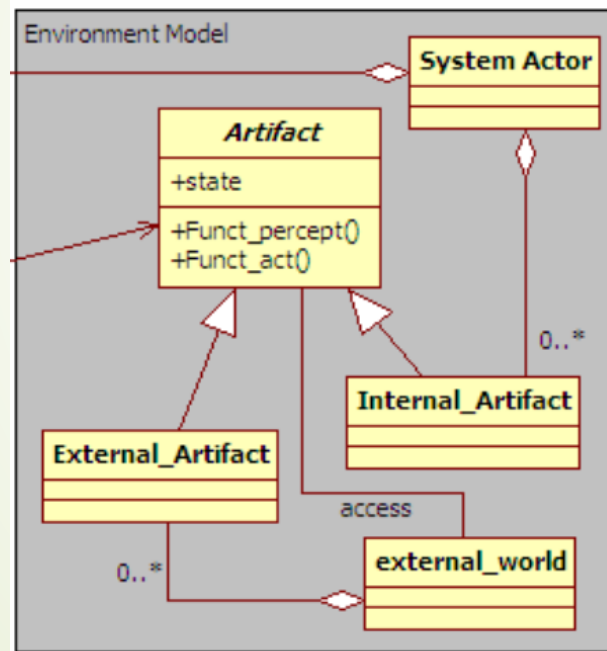
Paradigmi di programmazione...

Alcuni principali paradigmi di programmazione e livelli di astrazione del pensiero computazionale.

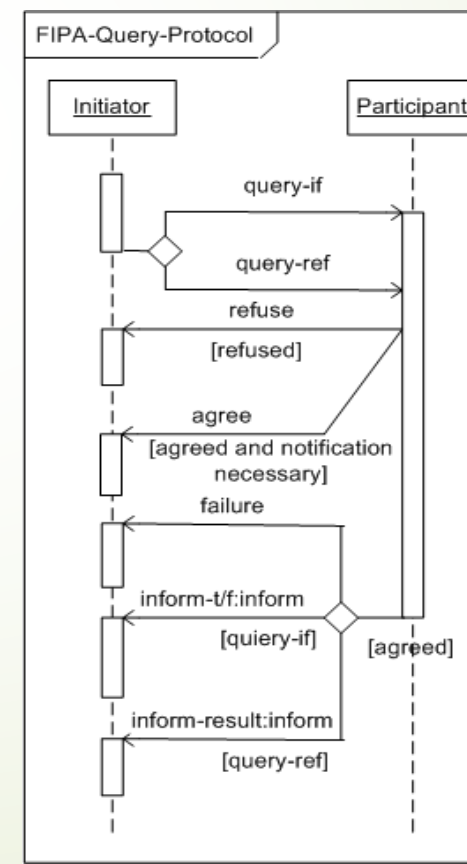
Flow-Chart (Structured Prog.)



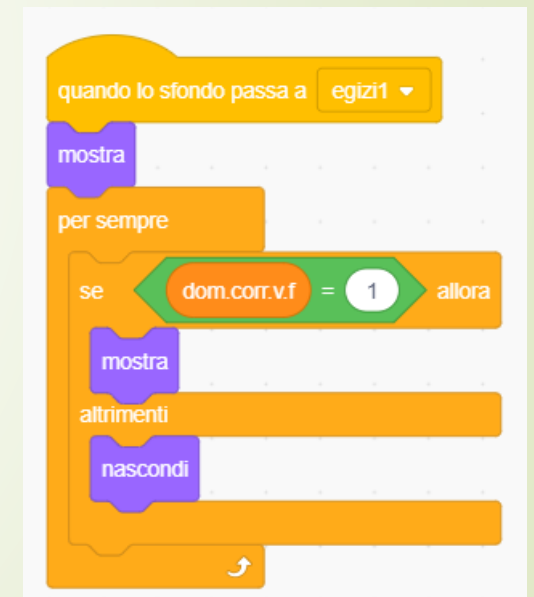
UML (Object Oriented Prog.)



Agent-UML (Agent Oriented Prog.)



Scratch/Blockly (Block based Prog.)



Ambienti visuali per OOP...

BlueJ è uno dei primi ambienti visuali per OOP...

The image shows the BlueJ IDE interface. The main window displays a class diagram with two classes: **TipSaluto** and **Persona**. **TipSaluto** is the superclass, and **Persona** is the subclass, indicated by a hollow triangle arrow pointing from **Persona** to **TipSaluto**. The left sidebar contains buttons for 'Nuova classe', 'Compila tutto', 'Teamwork' (with a 'Share...' button), and 'Testing' (with buttons for 'Esegui i test', 'registra', 'Fine', and 'Annulla'). At the bottom left, a red button labeled 'persona1: Persona' is visible.

Overlaid on the right is a 'Crea oggetto' (Create object) dialog box for the **Persona** class. It shows the constructor **Persona(String tempo, String nome, String cognome)**. The 'Nome dell'istanza' (Instance name) is set to 'persona2'. The 'new Persona(' line is followed by three dropdown menus containing 'mattino', 'Maria', and 'Verdi' respectively, followed by a closing parenthesis. 'OK' and 'Annulla' buttons are at the bottom.

Below the dialog box is a terminal window titled 'BlueJ: BlueJ: Terminale - TestHello'. It shows the output 'Opzioni' followed by 'Buon giorno, mi chiamo Maria Verdi'. A message at the bottom of the terminal reads 'Can only enter input while your programming is ri'.

Perché OOP per lo sviluppo di SW

Ecco alcuni vantaggi di pensare ad un algoritmo in termini di Oggetti:

- Astrarre dalla complessità del codice per concentrarsi maggiormente sulle similitudini tra gli oggetti software e gli oggetti del mondo reale che si vogliono modellare.
- Pensare al comportamento (behavior) di un oggetto software come al suo analogo reale: causalità e relativi effetti.
- Notevole aumento della possibilità di riuso del codice, con conseguente aumento della produttività di qualità:
 - Maggiore stabilità delle applicazioni;
 - Facilità nell'aggiornare mantenere il codice

Tecnologie di Sviluppo

Strumenti necessari per lo sviluppo di applicazioni Web con PHP:

- La parte server, per esempio Apache web server
- La parte client, un qualsiasi browser per pagine HTML
- L'interprete PHP
- Possibilmente un buon IDE

Soluzione tecnologica proposta per queste lezioni:

- **XAMPP → Apache e PHP**
- **NetBeans → IDE per lo sviluppo**
- **Chrome e Edge → Browser**

La «classe»

Nella OOP, la **classe** è il modulo autocontenuto che definisce il progetto (parziale o intero) dell'algoritmo che si vuole realizzare. E' caratterizzata da:

- Proprietà o variabili (in PHP la tipizzazione del dato non è obbligatoria)
- Funzioni o metodi, che contengono la logica dell'algoritmo
- Il nome della classe deve coincidere con quella del file
- I blocchi delimitati dalle parentesi graffe determinano lo scope o campo d'azione delle proprietà
- Lo stato di una classe dipende dai valori di inizializzazione delle sue proprietà nel momento in cui viene caricata in memoria

La classe assomiglia al progetto della casa, ma non è la casa!

Esempio concettuale di «classe» in OOP

//classe

Persona

- nome
- cognome
- età
- Interessi
- pagina di saluto

Buongiorno sono
«nome» +
«cognome»

specializzare

generalizzare

//sottoclassi

Studente

- nome
- cognome
- età
- interessi
- indirizzo_studi
- pagina di saluto_stu

Buongiorno sono
«nome» +
«cognome», e
frequento
«indirizzo_studi»

Insegnante

- nome
- cognome
- età
- interessi
- materia
- pagina di saluto_ins

Buongiorno sono
«nome» +
«cognome» e
insegno «materia»

Cosa è un «oggetto»?

Nella OOP il concetto di «oggetto» è :

- Un processo in esecuzione in memoria
- Una applicazione che fa uso di funzioni/metodi appartenenti a classi diverse (librerie diverse)
- Una applicazione alla quale si può dare uno stato iniziale che ne determina il comportamento (behavior) iniziale.
- Una applicazione che interagisce con il mondo esterno determinando il comportamento che più si adatta per quello scenario

Differenza tra «classe» e «oggetto»

DESIGN time

//classe: è un file

Persona

- nome
- cognome
- età
- Interessi
- pagina di benvenuto



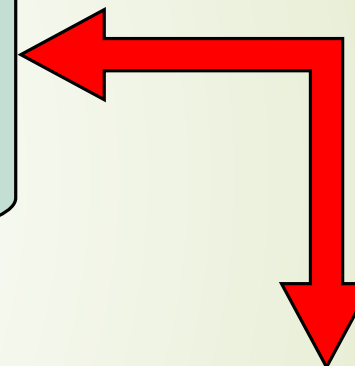
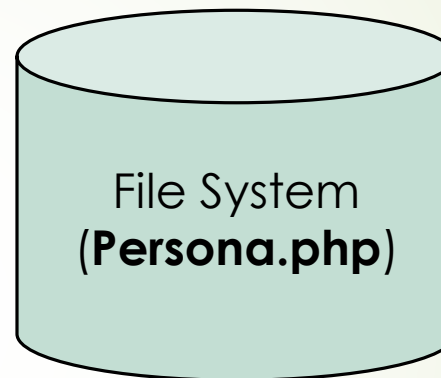
//oggetto: è un processo

Persona1

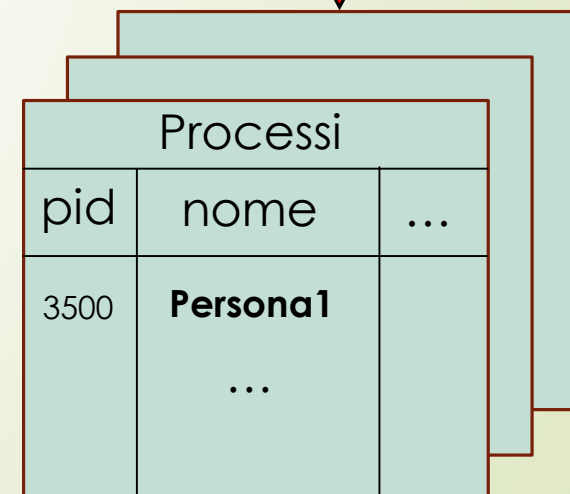
- nome → Mario
- Cognome → Rossi
- Età → 50
- Interessi → robotica
- pagina di benvenuto →
Ciao Mario Rossi

RUN time

Memoria di massa



RAM



Proprietà fondamentali in OOP

Tutti i linguaggi OOP forniscono sempre queste tre proprietà:

- **Ereditarietà**
- **Incapsulamento**
- **Polimorfismo**



EREDITARIETA'

Ereditarietà

In PHP (come in Java) **non** è prevista l'ereditarietà multipla come invece è possibile nel C++, per cui in PHP una classe può al più ereditare da una sola altra classe.

In PHP, dalla versione 5.4, sono state aggiunte delle funzioni per ovviare a questa limitazione (i trait) che vedremo più avanti.

Tuttavia, per bravi programmatori OOP, l'ereditarietà singola non è considerata una limitazione ma un vantaggio per progettare SW efficiente ed modulare.

La classe Persona

```
class Persona {
15     //Proprietà
16     public $nome = "";
17     public $cognome = "";
18     public $eta = "";
19     public $interessi = "";
20     public $saluto = "";
21
22     //costruttore
23     public function __construct($nome,$cognome,$eta,$interessi) {
24         //inizializzazione
25         $this->nome = $nome;
26         $this->cognome = $cognome;
27         $this->eta = $eta;
28         $this->interessi = $interessi;
29         $this->saluto = "Buongiorno sono ".$nome." ".$cognome;
30     }
31
32     //metodo che restituisce la pagina di saluto
33     public function getPagBenvenuto() {
34         $this->saluto = "Buongiorno sono ".$this->nome." ".$this->cognome;
35         return $this->saluto;
36     }
37 }
```


La classe Studente

L'operatore «extends»

```
13 class Studente extends Persona {  
14     public $ind_studio = "";  
15  
16     //metodo per inserire info specifiche per lo studente  
17     public function setIndStudio($ind_studio) {  
18         $this->ind_studio = $ind_studio;  
19     }  
20  
21     public function getPagBenvenutoStud() {  
22         $saluto_persona = $this->getPagBenvenuto();  
23         return $saluto_persona.", e frequento ".$this->ind_studio;  
24     }  
25 }
```

Le Istanze di classi sono Oggetti

```
6 <html>
7   <head>
8     <meta charset="UTF-8">
9     <title></title>
10  </head>
11  <body>
12    <?php
13      include 'Persona.php';
14      //require_once 'Persona.php';
15      include 'Studiante.php';
16
17      $nome = "Loris";
18      $cognome = "Penserini";
19      $eta = "50";
20      $interessi = "DRONI";
21
22      //CREO L'OGGETTO "Personal"
23      $Personal = new Persona($nome, $cognome, $eta, $interessi);
24      $Studentel = new Studiante($nome, $cognome, $eta, $interessi);
25      $Studentel->setIndStudio("Sistemi Informativi Aziendali");
26
27      echo "SALUTO DI PERSONA_1: <br>".$Personal->getPagBenvenuto();
28      echo "<br><br>SALUTO DI STUDENTE_1: <br>".$Studentel->getPagBenvenutoStud();
29
30    ?>
31  </body>
32 </html>
```

Costruttori di classe

Ogni classe dovrebbe avere il metodo costruttore, poiché definisce lo stato iniziale dell'oggetto associato alla classe. Cioè il metodo costruttore crea un punto di partenza nell'esecuzione del codice dell'oggetto.

Allora nella classe Studente da quale blocco di codice si inizia?

Project work 1

Riutilizzate il codice del progetto appena presentato. E con modifiche minimali, aggiungere la classe «Dirigente» (utilizzando come guida la classe Studente), poi dalla pagina index.php lanciare un'istanza e accodare a video il relativo saluto:

output

SALUTO DI PERSONA_1:

Buongiorno sono Loris Penserini

SALUTO DI STUDENTE_1:

Buongiorno sono Mario Rossi, e frequento Sistemi Informativi Aziendali

SALUTO DI DIRIGENTE_1:

Buongiorno sono Giovanna Rossini, e dirigo la scuola IIS POLO3 FANO

Project work (Dirigente.php)

La classe Dirigente (come Studente) eredita dalla classe Persona, ovviamente al contrario di Studente i metodi di Dirigente si specializzano per questo ruolo.

```
14 class Dirigente extends Persona {
15     public $scuola = "";
16
17     //metodo per inserire info specifiche per lo studente
18     public function setScuola($scuola) {
19         $this->scuola = $scuola;
20     }
21
22     public function getPagBenvenutoDir() {
23         $saluto_persona = $this->getPagBenvenuto();
24         return $saluto_persona.", e dirigo la scuola ".$this->scuola;
25     }
26 }
```

Project work (index.php)

```
13 include 'Persona.php';
14 //require_once 'Persona.php';
15 include 'Studente.php';
16 include 'Dirigente.php';
17
18 //Simula una lettura dati che può avvenire tramite FORM HTML,
19 //tramite DB, tramite lettura file, ecc.
20 $nome1 = "Loris";
21 $cognome1 = "Penserini";
22 $eta1 = "50";
23 $interessil = "DRONI";
24
25 $nome2 = "Mario";
26 $cognome2 = "Rossi";
27 $eta2 = "40";
28 $interessi2 = "Pesca";
29
30 $nome3 = "Giovanna";
31 $cognome3 = "Rossini";
32 $eta3 = "40";
33 $interessi3 = "Opera";
34
35 //CREO GLI OGGETTI "Personal", "Studentel" e "Dirigentel"
36 $Personal = new Persona($nome1, $cognome1, $eta1, $interessil);
37 $Studentel = new Studente($nome2, $cognome2, $eta2, $interessi2);
38 $Studentel->setIndStudio("Sistemi Informativi Aziendali");
39 $Dirigentel = new Dirigente($nome3, $cognome3, $eta3, $interessi3);
40 $Dirigentel->setScuola("IIS POLO3 FANO");
41
42 echo "SALUTO DI PERSONA_1: <br>".$Personal->getPagBenvenuto();
43 echo "<br><br>SALUTO DI STUDENTE_1: <br>".$Studentel->getPagBenvenutoStud();
44 echo "<br><br>SALUTO DI DIRIGENTE_1: <br>".$Dirigentel->getPagBenvenutoDir();
```

Sono state cerchiare le
modifiche necessarie al file
«index.php»



GRAZIE!