

Programmazione a oggetti

1 Paradigma

Un **paradigma** è un insieme di regole e convenzioni adottate per svolgere determinati compiti.

In particolare, un **paradigma di programmazione** determina le astrazioni utilizzate per

- concettualizzare il processo di computazione
- organizzare e strutturare i compiti che un calcolatore deve svolgere

Alcuni esempi di paradigmi di programmazione sono:

- *imperativo* (o *procedurale*)
 - *orientato agli oggetti*
- *funzionale*
- *logico*

2 Programmazione orientata agli oggetti

La programmazione orientata agli oggetti (OOP, *Object Oriented Programming*) è un paradigma basato su quello imperativo, al quale aggiunge alcune astrazioni: in particolare, l'esecutore di un programma si può pensare come più esecutori virtuali (**oggetti**) che compiono azioni e si scambiano **messaggi** per svolgere determinati compiti.

2.1 Esempio

Vogliamo ordinare una torta per una festa di compleanno:

1. andiamo dal pasticciere (`daPinoPasticcino`)
2. gli comunichiamo il tipo di torta e la data in cui passeremo a ritirarla

In questo esempio ci sono due oggetti, (`cliente` e `daPinoPasticcino`) che si scambiano un messaggio, `ordinaTorta(torta, data)`.

`daPinoPasticcino`:

- deve rispondere correttamente al messaggio (*contratto*)
- per farlo usa un *metodo*

3 Metodo

Un **metodo** è una procedura eseguita da un oggetto per rispondere a un determinato messaggio che ha ricevuto.

L'oggetto che invia il messaggio non ha bisogno di sapere come è fatto il metodo corrispondente: gli interessa solo il risultato.

4 Stato e comportamento di un oggetto

Un oggetto è caratterizzato da

stato: l'insieme delle proprietà dell'oggetto in un determinato istante

- è descritto da una lista di variabili
- in linea di principio non è necessario che sia conosciuto al di fuori dell'oggetto

comportamento: l'insieme delle azioni che può eseguire

- ogni azione viene iniziata inviando un messaggio all'**agente** (oggetto) responsabile di svolgerla
- il messaggio codifica la richiesta dell'azione, corredata con le informazioni necessarie a soddisfarla (*argomenti*)
- il ricevente si assume la responsabilità di compiere l'azione (*contratto*), la quale viene svolta eseguendo un metodo

5 Protocollo

Un **protocollo** (o **interfaccia**) definisce

- l'*insieme dei messaggi* che un oggetto può riconoscere
- il *formato* di ciascun messaggio

cioè le regole di comunicazione con l'oggetto.

6 Contratto

A ogni messaggio è associato un **contratto** che descrive come l'oggetto garantisce di rispondere a tale messaggio.

7 Principi impliciti della OOP

La programmazione OO ha alcuni principi impliciti:

- se c'è un compito da svolgere, *trova qualcuno che lo svolga per te*
- chi si assume una responsabilità *deve onorarla*

Da tali principi si ricavano i concetti di **scomposizione e riutilizzo del codice**: uno degli obiettivi principali della OOP è lo sviluppo di **componenti riutilizzabili**.

8 Classi

L'interpretazione di un messaggio (cioè le azioni eseguite) *dipende dal ricevente*.

Una **classe** è un modello dello stato e del comportamento delle sue **istanze** (gli oggetti basati su tale modello).

- Ogni oggetto è istanza di una classe.
- Il metodo eseguito per rispondere a un messaggio dipende dalla classe da cui l'oggetto è stato istanziato: tutte le istanze di una particolare classe rispondono agli stessi messaggi eseguendo le stesse azioni.

8.1 Esempio

L'oggetto `daPinoPasticcino` dell'esempio precedente è un'istanza della classe `Pasticceria`. Tutte le altre istanze di `Pasticceria` eseguiranno le stesse azioni in risposta al messaggio `ordinaTorta(torta, data)`.

Le istanze di un'altra classe possono invece rispondere diversamente allo stesso messaggio: ad esempio, un'istanza della classe `Segretaria` non preparerebbe direttamente la torta, ma comunicherebbe l'ordine alla pasticceria per conto del cliente.

9 Programma

Un **programma** ad oggetti è un insieme di oggetti che cooperano per realizzare un obiettivo.

Per poter realizzare un programma bisogna

- conoscere le classi (fondamentali) messe a disposizione da linguaggio e librerie
- costruire e memorizzare gli oggetti necessari per svolgere il compito
- inviare messaggi a tali oggetti