

Modularità

concetto dell'informatica



Questa pagina sull'argomento informatica sembra trattare argomenti unificabili alla pagina Programmazione modulare.

[Ulteriori informazioni](#)

La **modularità** è un concetto che trova applicazione nel contesto dell'[informatica](#), più precisamente nella [programmazione](#), così come le [scienze cognitive](#) nello scoprire la struttura della [mente](#). Un **modulo** può essere definito in molteplici modi, ma di solito si tratta di un componente di un più vasto sistema, che opera in quel sistema indipendentemente dalle operazioni di altri componenti.

I moduli forniscono una separazione tra le [interfacce](#) e la [implementazione](#). Una interfaccia di un modulo esprime gli elementi che sono forniti e necessari al modulo. Gli elementi definiti in una interfaccia sono visibili agli altri moduli. L'implementazione contiene il codice operativo che corrisponde agli elementi dichiarati nell'interfaccia. I linguaggi che supportano esplicitamente il concetto di modulo includono [Ada](#), [D](#), [F](#), [Fortran](#), [Pascal](#) (alcuni derivati), [ML](#), [Modula-2](#), [Python](#) e [Ruby](#). Il sistema IBM i (nominato come AS/400 e iSeries) utilizza i moduli nel suo RPG, [COBOL](#) e CL quando si programma nel suo ambiente ILE.

^ Moduli e classi



La differenza tra le [classi](#) e i moduli sono:

- Le Classi possono essere istanziate per creare [oggetti](#),
- Le Classi possono [Ereditare](#) il loro comportamento e dati da altre classi,
- Il [Polimorfismo](#) permette la relazione tra istanze di classi di poter cambiare a run-time, mentre le relazioni tra i moduli sono statiche.

Le somiglianze con le classi sono:

- Entrambi possono essere usati per nascondere i dettagli dell'implementazione.
- Entrambi possono formare gerarchie di modules/classes.

^ Linguaggi di interconnessione dei moduli (MILs)



I **linguaggi di interconnessione dei moduli (MILs)** forniscono costrutti di grammatica formale per decidere le varie specifiche di interconnessione dei moduli necessarie ad assemblare un completo sistema software. I MILs permettono la separazione tra la **programmazione in piccolo** e la **programmazione in grande**. Scrivere un modulo rappresenta la programmazione in piccolo, mentre assemblare un sistema tramite l'aiuto del MIL rappresenta la programmazione in grande. Un esempio del MIL è il MIL-75.

^ La modularità nell'informatica



La **modularità** è una proprietà dei [programmi](#) che misura la estensione di quanto sono composti in parti separate chiamate [moduli](#). Un approccio modulare sta guadagnando popolarità nel campo della [integrazione di sistemi](#)

[intelligenza artificiale](#), dove un sistema A.I. generale in larga scala system è composto di moduli dei quali ognuno serve a uno specifico scopo e comunica con gli altri per produrre un comportamento generale desiderato.

I programmi che hanno molte correlazioni dirette tra qualsiasi parti del codice di programmazione sono meno **modulari** (più strettamente [accoppiati](#)) dei programmi le cui relazioni avvengono solo tramite ben specificate interfacce di moduli.

Le tecniche di **programmazione modulare** sono quelle che incrementano la modularità. (Vedi anche: [programmazione modulare](#), [programmazione strutturata](#), [programmazione procedurale](#), [Programmazione orientata agli oggetti](#) e [Programmazione orientata agli aspetti](#), [Calcolo distribuito](#).)

^ Lavori citati



- Guimerá, R. & L.A. Nunes Amaral (2005). *Functional Cartography of Complex Metabolic Networks*. *Natura*, **433**: 895-900.
- Yang, A.S. (2001). *Modularity, Evolvability and Adaptive Radiations*. *Evolution and Development*, **3**:2, 59-72.

^ Voci correlate



- [Programmazione modulare](#)
- [Scienze cognitive](#)
- [Coesione \(Informatica\)](#)
- [Accoppiamento \(Informatica\)](#)
- [Incapsulazione](#)
- [Libreria software](#)
- [David Parnas](#)
- [Ilities](#)
- [Modularità della mente](#)
- [Filosofia della mente](#)
- . Un metodo per creare moduli e anche sistemi A.I.

^ Collegamenti esterni



- *Environment Modules* (<http://modules.sourceforge.net/>), su *modules.sourceforge.net*.