

Paradigmi

Tue, 4 Oct

Paradigma Procedurale - *Algoritmo e Struttura Dati*

Le caratteristiche principali dei **linguaggi imperativi** sono legate all'**architettura di Von Neuman**.

I linguaggi imperativi si chiamano così perché "**prescrivono**" delle azioni.

Paradigmi Funzionale e Logico

Sono linguaggi ad **altissimo livello di astrazione**, generati per effettuare **manipolazione simbolica**.

Non esistono strutture dati al di fuori delle **liste**, non si dichiarano variabili. I concetti di "programma" e "struttura dati" non sono nettamente distinti.

Tali linguaggi sono basati su **concetti matematici**. Applicano uno stile "**dichiarativo**".

Paradigma Logico - *Conoscenza e Controllo*

In **Prolog** vengono descritte **formule logiche ben formate**.

L'interprete applica una **regola di inferenza**, il *principio di risoluzione*.

Prolog permette di definire due tipi di FBF (dette **clausole**):

- Asserzioni incondizionate - *fatti*

A.

- Asserzioni Condizionate - *regole*

▼ A :- B, C, D, ... Z.

B and C and D and ... and Z *implica* A

A è detto **conseguente**, B, C, D, ... Z sono detti **antecedenti**.

È poi possibile interrogare la base delle conoscenze con delle **query**:

- :- K, L, M, ... P.

Un **teorema** è una formula che dev'essere dedotta da formule già esistenti.

Paradigma Funzionale - *Funzioni*

Il concetto primitivo in Lisp è quello di **funzione matematica**.

Una **funzione** è una regola di associazione tra due insiemi, dominio e codominio.

L'unica operazione utilizzata nel **paradigma funzionale** puro è l'**applicazione di funzioni**.

L'interprete si limita a **valutare** l'applicazione di una funzione e produce un **valore**.