Citazioni dalla Babele di Martini.

*“a good programming language is a conceptual*

*universe for thinking about programming".*

Ogni linguaggio (naturale o artificiale)) è (innanzitutto) uno strumento per pensare.

I linguaggi di programmazione sono strumenti concettuali per pensare la programmazione come attività di problem solving che prevede di usare (interagire e comunicare con) il computer.

**I linguaggi di programmazione sono strumenti concettuali per realizzare il problem solving**.

*“a language that doesn’t affect the way you think*

*about programming, is not worth knowing”*

Nell’ambito dell’informatica, questo concetto ha generato e guidato nel tempo tutta la produzione dei linguaggi di programmazione da quelli adatti per la costruzione dei computer a quelli di più alto livello adatti per il problem solving; nel campo più ampio della attività scientifica, questo stesso concetto ha generato (come evoluzione del linguaggio naturale) i gerghi delle discipline scientifiche per

* pensare (nelle attività di ricerca) e
* comunicare (nei processi educativi).

Una disciplina senza un proprio gergo non può essere una disciplina scientifica, vedi la filosofia!

**Se un linguaggio di programmazione non caratterizza un metodo di problem solving può essere ignorato.**

*“there will always be things we wish to say in our programs that in all known languages can only be said poorly”.*

Questa tesi lascia aperta la possibilità di trovare linguaggi sempre più specifici e adatti per il problem solving del futuro. Parafrasando Hilbert, questo è uno dei problemi che garantisce vitalità all’informatica!

Ci saranno sempre problemi per la cui soluzione non esiste ancora un efficace linguaggio di programmazione.

[Vannevar Bush](http://en.wikipedia.org/wiki/Vannevar_Bush) (1890-1974), famoso per il proprio trattato **As We May Think** in cui ipotizza una macchina automatica chiamata Memex

Quando si parla di intelligenza artificiale e lavoro, ci si pone spesso il seguente interrogativo: come si salvano i lavori dall'innovazione della Ia? La società di consulenza Accenture ha provato a ribaltare la prospettiva, con una ricerca dove si registrano le figure professionali del tutto inedite che potrebbero emergere grazie all'espansione di modelli commerciali di artificial intelligence. **Le macrocategorie che emergono sono tre: trainers, explainers, sustainers**.

**I trainers** si occupano di “formare” gli algoritmi abilitati dall'intelligenza artificiale per renderli meno asettici e più utili, ad esempio istruendo i chatbot ad esprimersi con un linguaggio più empatico.

**Gli explainers** devono, appunto, spiegare portata e valenza delle innovazioni spinte dalle tecnologie di settore: è il caso di coach chiamati dall'azienda per descrivere in maniera semplice applicazioni e conseguenze dell'uso dell'Ia in un'impresa tradizionale.

**I sustainers**, gli esperti della sostenibilità. Il loro compito è tra i più delicati: valutare gli impatti positivi e negativi delle tecnologie impiegate, dal rapporto costi-benefici ai dilemmi etici che si possono creare.

L’informatica ha una preistoria caratterizzata dalla esigenza di risolvere problemi ed è costellata dalla invenzione di strumenti operativi e concettuali che facilitano le elaborazioni di simboli coinvolte nei processi risolutivi.

L’impatto decisivo su ogni aspetto sociale, economico e culturale ha inizio con la proposta formulata da Alan Turing di un linguaggio (il prototipo dei linguaggi di programmazione) che consente di descrivere qualsiasi algoritmo e con quella avanzata da John von Neumann di una macchina (il computer) capace di eseguire tutte le manipolazioni di simboli descritte (tutti gli ordini impartiti o tutte le istruzioni trasmesse) con un linguaggio di programmazione.

Ma alcuni linguaggi sono più adatti di altri a determinati scopi. Alcuni sono più evocativi, altri più sintetici, altri ancora più astratti, ma tutti con una forte propensione a influire sul modo di pensare ai problemi.