**Pensiero computazionale**

In un famoso articolo del 2006 Jeannette M. Wing, docente di *Computer science* nel Computer Science Department della Carnegie Mellon University, scrive che il *computational thinking* «[…] represents a universally applicable attitude and skill set everyone, not just computer scientists» (“rappresenta un atteggiamento e delle abilità universalmente applicabili a tutti, non solo agli scienziati del computer”) (2006).

Il c.t. si basa sull’informatica (*computer science*), tuttavia dovrebbe essere un’abilità per tutti e non solo per gli scienziati e dovrebbe essere fortificata su qualunque bambino, così come avviene per le abilità dello scrivere, del leggere, per quelle matematiche.

Esso implica:

* da un lato: problem solving, progettazione di sistemi, comprensione del pensiero umano,
* ma, anche: prendere in considerazione set di istruzioni, dispositivi hardware, ambienti operativi.

Favorisce:

* il pensiero ricorsivo, l’astrazione di procedimenti e di strutture dati in modelli, l’utilizzare ragionamenti euristici per la soluzione di problemi, l’affrontare situazioni di incertezza, il cercare una strategia per vincere in un gioco,
* ma anche: il saper raccogliere e gestire i dati ai fini di calcoli, , ma anche il rendere automatici queste creazioni.

L’autrice poi elenca poi una serie di caratteristiche del c.t.:

* Conceptualizing, not programming.

Il c. t. non è la programmazione di computer. Pensare come uno scienziato di computer pensa è più che programmare un computer. L’aspetto significativo è pensare per poi programmare e ciò richiede pensare a diversi livelli di astrazione.

* Fundamental skill.

il c. t. non deve e non può essere un’abilità di routine, ma un’abilità fondamentale che va coltivata.

* A way that humans, not computers, think.

Il c.t. è una modalità umana di risolvere i problemi, non un tentativo di ragionare come i computer. Essi sono noiosi e monotoni, mentre noi siamo intelligenti e creativi. Con l’aiuto del computer risolviamo problemi che prima non saremmo riusciti a risolvere, tuttavia la soluzione è nostra.

Ideas, not artifacts.

Entriamo in contatto con artefatti software e hardware che produciamo, ma ancor di più con i concetti che usiamo per affrontare e risolvere i problemi, per gestire la nostra vita quotidiana, e per comunicare e per interagire con altre persone.

For everyone, everywhere.

Il c.t. sarà perfettamente integrato in qualunque posto e per chiunque, quando scomparirà come filosofia esplicita. Ciascuno deve incarnare questo atteggiamento che diviene un habitus che permette di costruire e realizzare attraverso spontanei meccanismi.

Questo articolo è estremamente attuale; riflette le necessità e risponde ai bisogni che la scuola non è riuscita, negli ultimi anni, a soddisfare. L’esigenza di formare studenti che sappiano costruire con le tecnologie è una necessità oramai improcrastinabile; l’autrice ci ricorda come l’informatica, in particolare il *computational thinking*, sia perfettamente incardinata in diversi settori della vita sociale e del mondo scientifico, tuttavia la scuola non è riuscita, nel tempo, a percepire questa evidenza e a soddisfarla; in modo più o meno approfondito, ha recepito solo gli aspetti che possiamo inquadrare in alcuni settori delle scienze sociali. Ci riferiamo, in particolare, a quelli del web nelle varie articolazioni. Altri settori sono stati completamente abbandonati e, in questo caso, l’errore è ancor più grave in quanto non si trattava e non si tratta di una scelta didattica, non può trattarsi di una scelta, poiché è un obbligo offrire una visione approfondita di spaccati della realtà che ci circonda e che sono imbevuti di *computational thinking*.