Introduzione programmazione

Esistono diversi linguaggi di programmazione di diverso tipo, i linguaggi ci servono per dare delle istruzioni alla nostra macchina.

Immagine che contiene testo

Descrizione generata automaticamente

Cosa sono le first class function? Sappiamo che le function sono blocchi di codice che possiamo riutilizzare quando ne abbiamo bisogno. Con first class function possiamo riassegnare il nome delle variabili delle funzioni ad altre funzioni e fargli fare altre cose, vengono trattate come qualsiasi altra variabile. Tipo:  
a = 10

b = 30

c = b abbiamo assegnato a c un altro valore che è b, lo stesso possiamo farlo con le funzioni.

Esempi di linguaggi:

Procedurale: python, java

Funzionale: python

Orientata agli oggetti: classi sono insiemi di funzioni e proprietà (Studente: proprietà int voto, String nome e avrà una funzione per calcolare il voto). Un oggetto è l’istanza della classe es: io sono un’istanza della classe umano (umano è lo stampino: abbiamo due occhi, un naso, una bocca ecc… ecc..). java, python, c++.

Cosa vuol dire compilare? È il processo di traduzione che fa il programma per tradurre in linguaggio comprensibile alla macchina quello che abbiamo scritto.

Un linguaggio essere di due tipi:

* Statico: il linguaggio COMPILA, legge che tipo di variabile è mentre fa il processo di traduzione (compile time). Il tipo quindi in questi linguaggi deve essere specificato prima. Es: int a = 10. Non possiamo scrivere int a = “nome” perché da errore e lo fa durante la compilazione, legge e dice “questo non è un integer, errore, e si ferma”.
* Dinamico: il linguaggio prima compila e converte in linguaggio macchina poi il programma parte e durante questo periodo (runtine) va a guardare la variabile e capisce di che tipo è, es: a = 10 capisce che è integer. In questo caso non dobbiamo preoccuparci di specificare prima il tipo di variabile. L’errore qui compare solo quando il programma è in funzione. Un esempio di errore potrebbe essere “ciao” + 10. Perché sono tipi diversi di variabili.

Come è fatta la memoria?

Si divide in heap e stack.

Quando scriviamo a = 10, a è il nostro reference variable e 10 è il nostro oggetto. A è conservato nello stack e il valore della variabile invece è conservato nello heap

ATTENZIONE!

Più reference variable possono puntare allo STESSO oggetto!!! Es:

io sono l’oggetto. per i miei “figlia”, per mia sore “sorella” diversi reference ad un solo oggetto che sono io.

**SE UNA DI QUESTE VARIABILI CAMBIA L’OGGETTO A CUI PUNTANO, L’OGGETTO CAMBIA PER TUTTE LE VARIABILI CHE PUNTANO A QUELL’OGGETTO.**

Se mia mamma mi taglia i capelli sono visibili anche a mai sorella.

Se cambiamo l’oggetto alla variabile e lasciamo l’oggetto precedente senza una variabile di referenza questo oggetto viene rimosso dalla memoria tramite il garbage collection es:

a = 10, il reference variable punta la 10

e poi diventa

a = 30, ora il ref variable punta al 30

il 10 viene eliminato dal garbage collectors

**Flowcharts**

Gli usiamo per visualizzare il nostro programma e qualsiasi algoritmo andiamo a creare.

**Pseudocode**

Lo usiamo per far capire a qualcuno come funziona il nostro algoritmo