La programmazione : guida per non addetti

ANDREA BIANCHINI

DEDICA

Dedico questo libro a tutti i miei insegnanti, a partire dalla mia maestra delle elementari agli insegnanti delle medie, delle superiori e dell'università. In particolare, a coloro tra questi che mi hanno inculcato l'amore per la divulgazione scientifica.

CONTENUTI

RINGRAZIAMENTI

Un ringraziamento particolare a tutti i miei insegnanti di elettronica, informatica ed italiano da cui ho avuto il piacere di apprendere dalle scuole medie all'università.

1 UN PO DI STORIA

Nel 1936 un matematico britannico di nome Alan Touring ideò un modello per eseguire calcoli ed elaborazioni in modo meccanico noto come Macchina di Touring. Tale modello diede vita ad una tecnologia tutt' oggi in via di sviluppo ed alla ricerca di una teoria unificante; la programmazione.

Negli anni quaranta i primi programmi per computer venivano scritti in linguaggio macchina su elaboratori che occupavano stanze intere. Il linguaggio macchina non è altro che il codice binario, per intenderci, il numero, che rappresenta all'interno dell' elaboratore, una istruzione.

La macchina di Touring non è altro che un nastro di lunghezza teoricamente illimitata con una finestra attraverso la quale noi possiamo leggere e scrivere sul nastro; questo è un programma.

Dagli anni quaranta ad oggi si è via via passati dal linguaggio macchina ai linguaggi di programmazione, via via, si può dire ?, più evoluti. Il parametro che ha contraddistinto questa evoluzione è stato via via un aumento del grado di astrazione. Se nel linguaggio macchina ogni istruzione rappresentava una singola operazione dell'elaboratore, con l'astrazione e l'associato aumento della potenza di calcolo e della integrazione dei circuiti elettronici, una singola istruzione di un moderno linguaggio evoluto determina l'esecuzione di molteplici istruzioni macchina.

È avvenuto, per i linguaggi di programmazione, quello che un po' avviene nel nostro cervello nell' arco del periodo evolutivo dell' apprendimento; si parte dai mattoni, dalle basi, per arrivare, con una parola, a racchiudere una storia intera.

Non starò ad elencare i vari linguaggi che si sono susseguiti sequenzialmente e spesso parallelamente nel corso degli anni, sarebbe un tecnicismo. Ma la maggior astrazione via via raggiunta dai vari linguaggi di programmazione ha consentito di risolvere problemi di ingegnerizzazione, produttività, modellazione, sicurezza; ma allo stato dell'arte il percorso dei linguaggi di programmazione si trova di fronte lo stesso ostacolo di cui è vittima la burocrazia dei sistemi moderni; l'eccessiva astrazione non riesce ad avere controllo sul singolo dettaglio, e così, un semplice dettaglio, può mandare in crisi un intero sistema.

2 LA MIA STORIA DI PROGRAMMAZIONE

Ho iniziato ad interessarmi di programmazione non appena entrato in possesso del mio primo computer, un Olivetti M20. Il personal computer Olivetti M20 è stato uno dei primissimi computer disponibili al grande pubblico nei primi anni ottanta, insieme all'Amiga, all'Apple ed al Lemmon, per intenderci. Mi ricordo, allora avevo sedici anni, che il suo costo era l'equivalente di un'automobile; infatti, mio padre, fece un leasing per acquistarlo. Subito imparai a scrivere programmi e a svolgere qualche lavoretto commissionatomi dal titolare del concessionario Olivetti di zona Silvano Bellazzecca. Nel frattempo stavo frequentando l'ITIS "E.Mattei" di Urbino indirizzo elettronica industriale ed ebbi modo di approfondire alcuni aspetti della programmazione grazie ad alcuni miei docenti, tutti ingegneri elettronici, tra cui cito in particolare il caro Ing. Giuseppe della Chiara che ho ancora saltuariamente modo di incontrare per motivi musicali. Peppe mi chiamava "little white"….. Grazie Peppe.

Una volta diplomato come perito elettronico mi sono iscritto alla facoltà di ingegneria elettronica all'università degli studi di Bologna e qui ho avuto modo di affinare la mia teoria nei confronti dell'informatica. Mi ricordo che già studente di ingegneria ero titolare di partita Iva e svolgevo qualche lavoretto di programmazione. Addirittura mi proposero di abbandonare gli studi di ingegneria per lavorare come programmatore dipendente presso una azienda a cui avevo effettuato un lavoro. Non so se ho fatto bene a rifiutare…. Ancora oggi me lo chiedo… Infine, una volta laureato in ingegneria elettronica ho iniziato quasi subito a lavorare, il

mio primo lavoro, a meno di un mese di distanza dalla laurea è stato quello di insegnante di elettronica presso l'IPSIA "G.Benelli" di Pesaro e, contemporaneamente ho iniziato, anzi proseguito, il mio lavoro di programmatore che mi ha accompagnato per gran parte della mia carriera lavorativa. Attualmente continuo a programmare ma non a tempo pieno, on demand e contemporaneamente insegno, scrivo e faccio video su Youtube.

3 CHE COSA E' UN PROGRAMMA ?

Si. Va bene, la macchina di Touring, Ma che cosa è un programma? La domanda potrebbe sembrare banale, ma non lo è affatto, almeno per la maggioranza delle persone che stanno leggendo questo libro. Avete presente un documento? Si, un documento, tipo una lettera di quelle che il postino vi imbuca nella cassetta delle lettere di tanto in tanto. Una lettera riporta un testo, ciò che chi vi scrive ha intenzione di comunicarvi nell'inviarvela. E voi cosa fate? Aprite la busta, spiegate la lettera e cominciate a leggere. Successivamente, dopo aver letto la lettera, in genere eseguirete dei compiti in funzione di ciò che avete letto nella lettera, che saranno; pagare una bolletta, pagare una multa, o semplicemente rispondere con un'altra lettera a chi vi ha inviato la sua. Ecco, un programma è un documento, una lettera che noi inviamo al computer, più precisamente al microprocessore del nostro computer, affinché questi esegua delle operazioni. E più precisamente il programma è un insieme sequenziale di istruzioni che noi inviamo al nostro microprocessore affinché questi svolga delle operazioni; cioè, scriviamo sul nastro della nostra macchina di Touring. E per scrivere questo elenco di istruzioni da inviare al nostro microprocessore avremo bisogno di un linguaggio che sia comprensibile al nostro computer, che lo tradurrà in linguaggio macchina e a noi che lo dobbiamo utilizzare per scrivere il nostro programma. Esistono diversi tipi di linguaggio di programmazione in funzione del compito che il programma dovrà assolvere. Non solo, col trascorrere del tempo vengono ideati nuovi linguaggi, alcuni vengono rimpiazzati, altri, per le loro peculiarità, resistono ai nuovi arrivati mantenendo la loro nicchia di campo di applicazione. Non per niente il termine

linguaggio deriva da lingua e cioè insieme di termini e regole atte a comunicare, con la lingua comunichiamo tra esseri umani, con un linguaggio di programmazione comunichiamo da essere umano a macchina, il computer, indicandogli quali sono le operazioni che deve eseguire.

4 I LINGUAGGI DI PROGRAMMAZIONE

Credo che l'argomento linguaggi di programmazione meriti un paragrafo tutto suo. Abbiamo già avuto modo di appurare come il primo linguaggio di programmazione per elaboratori sia stato il linguaggio macchina. Ma vediamo un esempio di cosa significhi macchina è cioè nel nostro caso microprocessore, la nostra macchina di Touring. Immaginiamo di avere una scatola magica che effettua operazioni matematiche; ora immaginiamo che questa scatola magica, al pari di una cassetta delle lettere, abbia una feritoia dalla quale noi possiamo inserire dei fogli con sopra scritto cosa deve fare la scatola magica. Ad esempio se volessimo effettuare la somma di due numeri, come 2+3, potremmo inserire nella feritoia della nostra scatola magica un primo foglio con su scritto 2. Successivamente inseriremmo nella feritoia un secondo foglio con su scritto 3. Infine dovremmo dire alla scatola magica; esegui la somma dei due numeri che ho inserito con i fogli; quindi, a questo scopo potremmo inserire il terzo ed ultimo foglio con su scritto 'più'. A questo punto la scatola magica ci potrebbe rispondere facendo uscire un foglio dalla feritoia con su scritto 5.

Un microprocessore questo fa, esegue delle somme, delle sottrazioni, moltiplicazioni e divisioni. 'più' è l'istruzione macchina che noi, grazie al nostro linguaggio macchina, impartiamo al nostro processore. Ed il nostro linguaggio macchina comprenderà anche altre istruzioni, come 'meno',

‘per’, ‘diviso’. E se noi inventassimo un ufficio che raccoglie lettere di chi non conosce il linguaggio macchina ma vuole scrivere al processore, questo ufficio si occuperebbe di tradurre dal linguaggio naturale dei suoi interlocutori in linguaggio macchina da imbucare nella feritoia della nostra scatola magica. In questo caso il linguaggio naturale dei clienti dell' ufficio andrebbe a costituire il linguaggio di programmazione del nostro processore.

Da quest' ultima rappresentazione possiamo dedurre che un linguaggio di programmazione è uno strumento che ha come mattoni costituenti le istruzioni macchina, è una piramide di astrazioni che si ergono sul linguaggio macchina. Una istruzione di un linguaggio di programmazione evoluto determina l’esecuzione di innumerevoli istruzioni in linguaggio macchina. A partire dagli anni quaranta sino ad oggi si sono susseguiti molteplici linguaggi di programmazione, ne cito alcuni; Fortran, Pascal, Ada, Smalltalk, Basic, Cobol, C, C++, C#, Java, Lisp, Prolog, per passare ai linguaggi di programmazione in ambiente Web come HTML, Asp, PHP, JSP, Python, e via dicendo…

Interessante la classificazione che gli studiosi, ma che è anche una conseguenza naturale, hanno fatto di tutti i linguaggi di programmazione sino ad oggi ideati, e cioè; imperativi e dichiarativi. I primi, quelli imperativi,sono quei linguaggi in cui le istruzioni altro non sono che ordini al processore, direttive; fai questo, prendi quell' altro, esegui. Mentre nei linguaggi dichiarativi non vengono impartiti dei comandi ma vengono dichiarate delle definizioni che l’interprete del

linguaggio elaborerà determinando delle reazioni. La maggior parte dei linguaggi sino ad oggi ideati sono di tipo imperativo, pochi altri come Lisp e Prolog sono di tipo dichiarativo. Ora qui esporro' una mia opinione, quindi prendetela come tale; i linguaggi dichiarativi sono una mera illusione, infatti sotto l'interprete del linguaggio dichiarativo i fatti si risolvono in comandi imperativi, perché la macchina di Touring questo è. Quindi secondo me il linguaggio dichiarativo è una chimera. A questo proposito è bene citare che qualche anno fa vennero annunciati microprocessori di quinta generazione basati su una programmazione dichiarativa come Prolog e Lisp, in particolare furono le nazioni orientali a preannunciare tale generazione di processori, ma purtroppo non se ne è saputo più niente, come la fusione fredda…. Che questa tecnologia sia finita nelle mani di qualche organizzazione segreta o dei militari o dell'intelligence o della Korea del Nord?

La sfida più grande dell' informatica è la programmazione attraverso il linguaggio naturale, con le sue contraddizioni, paradossi, poemi e sentimenti, di questo se ne occupa una branca dell'informatica che si chiama intelligenza artificiale. Infine per terminare questo paragrafo, parlando di macchine di Touring, linguaggio macchina e relativi codici binari, è il caso di citare il relativamente recente arrivo in casa; il qbit. Se i processori ed i relativi elaboratori degli anni quaranta erano basati sul bit, unità di informazione che può assumere due valori; zero o uno, il qbit è una unità di informazione quantistica rappresentata da un vettore nello spazio di Hilbert ed è una combinazione tra i due stati zero e uno. In soldoni,

un singolo qbit, a differenza di un bit che puo assumere due valori, può rappresentare tutto lo scibile umano.

5 DATI E FUNZIONI

Per entrare più nel dettaglio, diciamo nell' architettura di un programma, possiamo dire che molto genericamente un programma è costituito da un un insieme di dati ed un insieme di funzioni che operano su di essi. Per esempio potremmo avere un programma che chiameremo BilancioMensile, che avrà come dati l'elenco degli importi dei prodotti che abbiamo acquistato in un mese; e come funzioni tutte le operazioni di calcolo per determinare le varie voci di bilancio della nostra spesa. Così ad esempio potremmo avere la funzione SpesaMediaGiornaliera, la funzione SpesaTotaleMensile e via dicendo. Le funzioni anche conosciute come procedure o subroutines altro non sono che un insieme di istruzioni, racchiuse in una porzione di programma, che assolvono ad un determinato compito. Ogni volta che avremo bisogno di assolvere a quel determinato compito non faremo altro che evocare quella determinata funzione. È chiara l'utilità di una funzione, infatti questa potrà essere evocata in più punti diversi del programma senza avere la necessità ogni volta di riscrivere tutto il codice, riducendo la probabilità di errori e aumentando la manutenibilita' del codice.

In un programma i dati possono essere di tipo diverso, così avremo ad esempio, per citarne alcuni; i numeri interi, i numeri reali, le stringhe o testi, le date, i numeri logici o booleani, e via dicendo. Insieme ai dati troveremo le strutture

di dati che sono insiemi che contengono diversi dati di tipo diverso tra loro. Il tipo di dato identifica univocamente i valori che esso può assumere e le operazioni che possono essere effettuate su di esso.

Vediamo un piccolo esempio:

```
programma esempio ()
{
    Intero i;
    i=10;
    incrementa_i ();
// commento : a questo punto del programma i vale 11
        fine();
        funzione incrementa_i(
        {
                i=i+1;
          }
}
```

6 LA PROGRAMMAZIONE AD OGGETTI

Sicuramente la più interessante ed utile innovazione introdotta nel corso degli anni di evoluzione dei linguaggi di programmazione è stata il paradigma della programmazione ad oggetti (OOP - Object Oriented Programming)

Un programma OOP è organizzato utilizzando una entità che prende il nome di classe. La classe altro non è che una struttura contenente dati e funzioni. Non entrerò nel dettaglio ma vediamo quali sono le tre caratteristiche fondamentali che caratterizzano l'uso delle classi:

i. Incapsulamento.
ii. Ereditarietà.
iii. Polimorfismo.

Per incapsulamento si intende la proprietà per cui una classe ha la capacità di racchiudere e nascondere al resto del mondo i suoi dati e le sue procedure interni. Per comunicare ed asservire ai suoi compiti una classe esporrà solo i metodi e le proprietà strettamente necessari. La classe opera un po' come una scatola nera; l'utilizzatore sa cosa fa ma non sa come lo fa.

Attraverso l'ereditarietà noi possiamo progettare una nuova classe ereditando tutte le proprietà definite in una seconda o

molteplici classi. La nuova classe avrà tutte le capacità della o delle classi ereditate ed in più potrà personalizzarne metodi e proprietà o addirittura ovviamente crearne di nuovi. È chiaro che una tale caratteristica produce una nuova abilità a processi di ingegnerizzazione, manutenibilta', e produttività.

Il polimorfismo è una proprietà di non immediata comprensione, molti programmatori lo utilizzano ma non saprebbero spiegarlo. A questo scopo immaginiamo una chiave inglese che riesca ad adattarsi alla dimensione dei bulloni; non avremmo bisogno di cambiare chiave al variare della dimensione dei bulloni; cioè potremmo usare la stessa chiave per ogni dimensione dei bulloni. Questo è il polimorfismo; in termini informatici è la proprietà secondo cui con la stessa classe o metodo, possiamo operare su tipi di dati diversi.

Sinceramente il paradigma OOP è stata un po' una certificazione, formalizzazione, di caratteristiche che i programmatori più avveduti adottavano naturalmente. La scienza a volte ha questo gusto di scoprire l'acqua calda. Ma ovviamente scherzo… In una ottica di ingegneria del software, questo paradigma ha reso possibili progressi in ambito informatico altrimenti difficilmente raggiungibili.

7 GLI ALGORITMI

Parallelamente alla ingegneria del software, grazie all'avvento degli elaboratori e dei linguaggi di programmazione sempre più evoluti, si è sviluppata una nuova branca della matematica, la matematica applicata, che trova uno dei suoi campi di applicazione nella ricerca operativa e che consiste nella risoluzione di problemi matematici attraverso l'ausilio degli elaboratori elettronici. La matematica applicata, la ricerca operativa, non richiedono di essere degli esperti in tecnologie informatiche, se non come semplici utilizzatori. L'obiettivo è il problema matematico che viene risolto attraverso un programma che mette in opera un algoritmo.

La definizione di algoritmo è la seguente: "Numero finito di passi da intraprendere per risolvere un problema ".

Ma vediamo subito un esempio di un semplice algoritmo; il nostro algoritmo sarà così definito dai seguenti passi:

Algoritmo: "Portare a spasso il cane"

1. Prendere due sacchetti per le feci.
2. Prendere la paletta per raccogliere le feci.
3. Prendere il guinzaglio.
4. Applicare il guinzaglio al cane.
5. Uscire di casa con il cane e chiudere la porta.

6. Iniziare a passeggiare.
7. Il cane è andato di corpo? Se si vai al punto 8. Altrimenti vai al punto 7.
8. Raccogli le feci con la paletta ed introducile in un sacchetto.
9. Al primo bidone deposita il sacchetto con le feci.
10. Vuoi tornare a casa? Se si vai al punto 11 altrimenti continua la passeggiata e vai al punto 10.
11. Incamminati verso casa.
12. Quando arrivi a casa apri la porta, togli il guinzaglio al cane e riponilo nel suo alloggio.
13. Fine algoritmo.

Ovviamente questo è un banale esempio è può sicuramente essere migliorato ma rende l'idea di cosa sia un algoritmo. L'uomo ha sempre adottato algoritmi da quando è apparso sulla terra, solo che non li chiamava così.

Per concludere vediamo un algoritmo serio atto alla risoluzione di un semplice problema : determinare il più grande tra due numeri.

Algoritmo : determinare il maggiore tra i numeri A e B

1. A è maggiore di B ? se si vai al punto 3
2. B è maggiore di A ? si si vai al punto 4
3. Stampa : "A è maggiore di B". Vai al punto 5
4. Stampa : "B è maggiore di A".
5. Fine algoritmo.

8 DIFFERENZA TRA UNA APP E UNA WEBAPP

Che differenza c'è tra una App che viene eseguita sul nostro computer e una WebApp cioè quelle App che eseguiamo per mezzo di un browser Internet? In tutti due i casi è sempre un processore, eventualmente più di uno, che esegue l'App o la WebApp, solo che nel caso della App il processore e quindi il computer che esegue l'App risiedono nello stesso computer, mentre quando noi eseguiamo una WebApp questa viene eseguita in computer diverso da quello dove la stiamo visualizzando attraverso il nostro browser. Tutto ciò comporta che una WebApp richieda un linguaggio di programmazione idoneo a tale scopo che sarà diverso dal linguaggio di programmazione che utilizzeremo per sviluppare una normale App.

Uno dei primi, più noti e semplici linguaggi di programmazione per contenuti web è l'HTML. HTML sta per Hyper Text Markup Language, e cioè linguaggio a marcatori per gli ipertesti. HTML è molto semplice, consente di collegare pagine web tramite links, collegamenti, semplicemente cliccando su una parola come potrebbe essere 'informatica', viene indirizzata la visualizzazione su di un'altra pagina che, sensatamente, fornirà dei dettagli sul concetto di informatica. Altri linguaggi di programmazione Web, come asp, forniscono potenzialità più avanzate, hanno come cornice sempre HTML, ma riescono a svolgere operazioni 'lato server' ben più complesse ed avanzate di quanto non

possa fare un semplice programma HTML. Infatti con asp noi possiamo collegarci con un database, eseguire operazioni locali al server, anche sul sistema e tutto ciò che ne consegue. Ovviamente asp è stato uno dei primi linguaggi web lato server, ne sono seguiti molti altri, Asp.net, PHP, JSP e via dicendo.

9 BIG DATA

Dall'avvento dei computer una delle problematiche che ha richiesto parecchie risorse per la sua gestione è stata la gestione dei dati. Sino all'avvento di internet ma anche in anni successivi le grandi quantità di dati venivano gestite da applicazioni software specializzate in questo; i database. Per citare i più famosi, Sql Server, MySql, Oracle. All'inizio i database non esistevano e bisognava gestire tutto in file separati o alla meglio in un unico file indicizzato. Ma l'evolversi di internet ci ha messo via via di fronte alla problematica di gestire moli di dati enormi, si parla dell'ordine dello Zettabyte, ovvero, miliardi di terabyte. E' noto che un normale sistema operativo e database annesso per computer sia in grado di gestire moli di dati dell'ordine del terabyte. Non solo, si parla di una eterogeneità dei dati veramente vasta, si tratta di collegare immagini, testi, informazioni, musica, presenti nella rete in misura enorme. Questo richiede nuove tecnologie, il classico database non è più sufficiente, occorrono nuovi strumenti, questa è la sfida dei Big data. Solo una elaborazione distribuita su più server a loro volta multiprocessore potrà far fronte alla complessità ed enormità del problema.

10 LA TEORIA DELLA COMPLESSITA'

La teoria della complessità è una branca della ricerca operativa che si occupa di quantificare la complessità di un algoritmo. La complessità di un algoritmo viene misurata in funzione del numero di passi necessari a risolvere un determinato problema da parte di un algoritmo. In genere la complessità di un algoritmo viene espressa in funzione della dimensione dei suoi dati di ingresso. Così si dirà che un algoritmo ha complessità O(n), cioè proporzionale ad n, se il tempo necessario per la sua esecuzione è proporzionale a n, dove n è il numero di dati del problema. Si è dimostrato che alcuni problemi sono intrinsecamente più complessi di altri, alcuni addirittura che richiedano un tempo di elaborazione infinito per la loro soluzione. Così sono state individuate due classi di problemi, quelli polinomiali, e quelli non polinomiali, cioè quelli che richiedono un tempo polinomiale per essere risolti e quelli che invece richiedono un tempo esponenziale. Un problema tutt'ora aperto è dimostrare se esista un modo per concepire un algoritmo che sia in grado di risolvere in un tempo polinomiale un problema di complessità esponenziale. Cioè, è possibile che i problemi siano tutti facili da risolvere o esistono problemi più difficili che con l'attuale matematica non riusciremo mai a risolvere in un tempo polinomiale ?

11 ESEMPI DI SEMPLICI PROGRAMMI

Di seguito fornisco alcuni esempi dello stesso programma realizzato in linguaggi diversi. Si tratta del programma "Hello World !", il programma che tutti i manuali di un linguaggio di programmazione propongono come primo esercizio. Infine un esempio di algoritmo in pseudocodice, più precisamente l'algoritmo Bubble Sort per l'ordinamento di un vettore di numeri in ordine crescente o decrescente.

BASIC

```
10 PRINT "Hello World!"
```

PASCAL

```
program HelloWorld;

begin
  writeln('Hello World!');
end.
```

C++

```
#include <iostream>

int main()
{
    std::cout << "Hello, world!\n";
}
```

ASP

```
<%
    HelloWorldLabel.Text = "Hello, world!";
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:Label runat="server"
id="HelloWorldLabel"></asp:Label>
    </div>
    </form>
</body>
</html>
```

PHP

```
<html>
 <head>
  <title>Test PHP</title>
 </head>
 <body>
 <?php echo "Hello World!<p>"; ?>
 </body>
</html>
```

JSP

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;


public class HelloWorld extends HttpServlet {


  public void doGet(HttpServletRequest req,
HttpServletResponse res)
                          throws ServletException,
IOException {


        res.setContentType("text/html");
        PrintWriter out = res.getWriter();

        out.println("<HTML>");
        out.println("<HEAD><TITLE>Hello World !
</TITLE></HEAD>");
        out.println("<BODY>");
        out.println("<H1>Hello World</H1>");
        out.println("Today is: " + (new
java.util.Date().toString()) );
        out.println("</BODY></HTML>");
  } // doGet
} // HelloWorld
```

VB.NET

```
Public Class Form1


    Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

        MessageBox.Show("Hello World !", "Message Box
Title")

    End Sub


End Class
```

C#

```
// A Hello World! program in C#.
using System;
namespace HelloWorld
{
    class Hello
    {
        static void Main()
        {
            Console.WriteLine("Hello World!");

            // Keep the console window open in debug mode.
            Console.WriteLine("Press any key to exit.");
            Console.ReadKey();
        }
    }
}
```

ASP.NET

```
<%
    HelloWorldLabel.Text = "Hello, world!";
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:Label runat="server"
id="HelloWorldLabel"></asp:Label>
    </div>
    </form>
</body>
</html>
```

Algoritmo di ordinamento (Bubble Sort).

```
procedure BubbleSort(A:lista di elementi da ordinare)
  scambio ← true
  n ← lenght (A) - 1
   while (scambio) do
    scambio ← false
     for i ← 0 to n + 1 do
       if (A[i] > A[i + 1]) then       //sostituire '>' con '<' per
ottenere un ordinamento decrescente
         swap ( A[i], A[i+1] )
         scambio ← true
    n ← n-1                          //ad ogni passaggio si accorcia
di uno il ciclo di for
```

INFORMAZIONI SULL'AUTORE

La pluriennale esperienza maturata come Docente di Informatica e Programmatore mi permette di seguire in modo indipendente sia l'insegnamento di informatica rivolto agli studenti degli Istituti presenti in Italia che le attività di sviluppo di applicazioni IT realizzate utilizzando molteplici linguaggi di programmazione e piattaforme tecnologiche. Dal 2015 scrivo inoltre libri disponibili nei canali multimediali di Google, Amazon e Ilmiolibro. Il know-how professionale acquisito nel corso del tempo, la buona padronanza della lingua inglese, l'ottima conoscenza dei principali applicativi informatici dedicati alla mansione e il conseguimento del Master in Management Innovativo delle Organizzazioni Sanitarie costituiscono le competenze che vanno a completare il mio background professionale ed educativo.

Potete trovarmi su…

https://es-andreabianchini.it