

GUIDA DI BASE ALLA PROGRAMMAZIONE

SISTEMI E LINGUAGGI DI PROGRAMMAZIONE AVANZATI

La programmazione

Abbiamo visto come le informazioni vengono codificate per potere essere elaborate dal computer. Ora passeremo a vedere gli strumenti che abbiamo a disposizione per programmare il computer, per fare in modo, cioè, che il computer elabori le informazioni che gli forniamo in modo utile per noi.

Un po' di storia

All'inizio, negli anni '40, l'unico metodo per programmare era il linguaggio macchina. Il lavoro del programmatore consisteva nel settare ogni singolo bit a 1 o 0 su enormi computer che occupavano stanze intere e pesavano decine di tonnellate. I monitor non esistevano; i dati e i programmi si fornivano al computer su schede perforate e il computer mandava i risultati su telescriventi. Questo tipo di procedimento, non solo era faticoso, ma era riservato ad una cerchia ristretta di persone. Eppure in questo modo gli scienziati riuscirono in pochi mesi a completare i calcoli per costruire la prima bomba atomica, calcoli che se fatti a mano, con calcolatrici meccaniche avrebbero richiesto anni.

Negli anni 50, con il progresso tecnologico e la riduzione delle dimensioni e dei costi dei calcolatori, nacquero due importanti linguaggi di programmazione, il FORtrAN (FORmula trANslator), il cui utilizzo era ed è prettamente quello di svolgere in maniera automatica calcoli matematici e scientifici, e l'ALGOL (ALGORithmic Language), altro linguaggio per applicazioni scientifiche sviluppato da Backus (l'inventore del FORtrAN) e da Naur, i quali oltretutto misero a punto un metodo per rappresentare le regole dei vari linguaggi di programmazione che stavano nascendo.

Nel 1960 venne presentato il COBOL (COmmon Business Oriented Language), ideato per applicazioni nei campi dell'amministrazione e del commercio, per l'organizzazione dei dati e la manipolazione dei file.

Nel 1964 fa la sua comparsa il BASIC, il linguaggio di programmazione per i principianti, che ha come caratteristica fondamentale quella di essere molto semplice e, infatti, diventa in pochi anni uno dei linguaggi più utilizzati al mondo.

Intorno al 1970, però, Niklus Wirth pensò bene di creare il PASCAL per andare incontro alle esigenze di apprendimento dei neo-programmatori, introducendo però la possibilità di creare programmi più leggeri e comprensibili di quelli sviluppati in basic. Si può affermare con certezza che Wirth ha centrato il suo obiettivo, considerando che ancora oggi il Pascal viene usato come linguaggio di apprendimento nelle scuole.

Pochi anni più tardi fa la sua comparsa il C (chiamato così perché il suo predecessore si chiamava B), che si distingueva dai suoi predecessori per il fatto di essere molto versatile nella rappresentazione dei dati. Il C, infatti, ha delle solide basi per quanto riguarda la strutturazione dei dati, però può apparire come un linguaggio assai povero vista la limitatezza degli strumenti a disposizione. Invece la sua forza risiede proprio in questi pochi strumenti che permettono di fare qualsiasi cosa, non a caso viene considerato "il linguaggio di più basso livello tra i linguaggi ad alto livello", per la sua potenza del tutto paragonabile al linguaggio macchina, mantenendo però sempre una buona facilità d'uso.

Ma la vera rivoluzione si è avuta nel 1983 quando Bjarne Stroustrup inventò il C++ (o come era stato chiamato inizialmente "C con classi") che introduceva, sfruttando come base il C, la programmazione Orientata agli Oggetti (OO - Object Oriented) usando una nuova struttura, la classe.

La programmazione orientata agli oggetti consente di dividere l'interfaccia dal contenuto, ottenendo in questo modo tanti "moduli" interagibili tra loro attraverso le interfacce, permettendo così al programmatore di cambiare il contenuto di una classe (se sono stati trovati errori o solo per introdurre delle ottimizzazioni) senza per questo doversi preoccupare di controllare eventuale altro codice che richiami la classe. Questo nuovo stile di programmazione ha completamente stravolto il modo di programmare precedente ad esso che si riduceva ad una programmazione procedurale che lasciava poco spazio al riutilizzo del codice.

Insomma il C++ è riuscito a creare un nuovo modo di programmare, o meglio di progettare un programma, rendendo il codice scritto più chiaro e soprattutto "riutilizzabile", ed è grazie a lui che oggi possiamo usare le finestre colorate di Windows che ci piacciono tanto.

Il modello Object Oriented ha avuto grande successo e dopo il C anche il Pascal, il Basic e altri linguaggi hanno avuto la loro versione OOP. Negli ultimi anni, infine, è nato JAVA che ha due caratteristiche principali:

È il primo linguaggio progettato appositamente per la programmazione orientata agli oggetti (non è cioè l'adattamento di un linguaggio preesistente).

Il programma ottenuto non è un programma destinato a girare in un ambiente specifico (Windows o Macintosh o Linux) ma gira su un computer virtuale, la Java Virtual Machine. Basterà installare la versione specifica della Java Virtual Machine per il sistema operativo specifico e lo stesso programma potrà girare in ambienti completamente diversi.

Altri linguaggi di programmazione da menzionare sono il LISP (1959), l'ADA (1970), lo SMALLTALK (1970) e il LOGO e il PROLOG, ecc .

Il panorama attuale

Il panorama attuale

L'enorme diffusione dei personal computer ha influito anche sul mercato dei linguaggi di programmazione, una volta sicuramente un mercato marginale. Pensiamo solamente alla diffusione di HTML e dei linguaggi di scripting ad esso collegati e ci renderemo conto di quanta più gente rispetto a solo dieci anni fa oggi si avvicina in qualche modo al mondo della programmazione.

Da un lato i produttori di software hanno tentato di assecondare questo processo cercando di offrire prodotti sempre più semplici da usare. Sono così nate le versioni visuali di vari linguaggi di programmazione che rendono enormemente più semplice sviluppare programmi nei moderni ambienti grafici a finestre: Visual Basic, Visual C ++, Delphi (versione visual di Object Pascal), Kilyx (versione di Delphi per il mondo Unix), Visual Java, ecc.

Dall'altro gli strumenti per lo sviluppo di contenuti multimediali on-line e off- line (pagine web, animazioni interattive, ecc.) si sono dotati di linguaggi di programmazione sempre più potenti.

Il nostro interesse è soprattutto rivolto allo sviluppo di contenuti multimediali. Può essere, comunque interessante dare un rapido sguardo agli strumenti che abbiamo a disposizione anche per realizzare programmi orientati ad altri fini. Bisogna infatti tenere conto che i linguaggi di programmazione, per loro natura, possono fare benissimo alcune cose e male altre.

Per Applicazioni Matematiche o di Ricerca i linguaggi più adatti sono ancora i vecchi Fortran e Algol.

Per i grandi progetti software (sviluppo di sistemi operativi, applicativi complessi, applicazioni lato server) normalmente viene impiegato il C o il C++.

Per lo sviluppo di programmi gestionali in ambiente Windows lo strumento più utilizzato è Visual Basic prodotto da Microsoft. Un'ottima alternativa al Visual Basic è, da qualche anno Delphi un prodotto Borland basato sul Pascal che consente anche il porting del programma tra l'ambiente Windows e l'ambiente Unix/Linux grazie a Kilyx versione Unix di Delphi. Negli ambienti diversi da Windows si utilizzano C/C++ e ambienti di sviluppo (i cosiddetti CASE) basati su C/C++

Applicazioni multimediali

Dal punto vista del nostro corso meritano una particolare attenzione gli strumenti per lo sviluppo di applicazioni multimediali on line e off line.

Sviluppo di pagine web

L'attività di programmazione entra a vari livelli nello sviluppo di contenuti da distribuire via Internet. Un primo livello di base è la costruzione delle pagine web. Il linguaggio usato è l'HTML (Hyper Text Markup Language) e descrive come una pagina web debba essere mostrata da un browser. L'HTML è un linguaggio a marcatori: tutte le istruzioni proprie del linguaggio sono inserite tra due segni specifici (nel caso di HTML tra “<” e “>”) e costituiscono i cosiddetti tag. I browser cercano di interpretare i tag come istruzioni mentre il resto viene mostrato come testo. I tag che il browser non riesce ad interpretare vengono semplicemente ignorati.

Questo approccio estremamente flessibile ha consentita un grande sviluppo delle possibilità di HTML nelle varie versioni successive senza la necessità di preoccuparsi troppo della compatibilità con le versioni precedenti.

Specifici tag consentono poi di inserire nella pagine web vari tipi di componenti che a loro volta sono programmabili. Vediamo le più importanti:

Il tag SCRIPT consente di inserire dei nuclei di codice che il browser è in grado di eseguire e che, ad esempio, possono servire ad aumentare il grado di interattività della pagina. I linguaggi a disposizione dello sviluppatore sono due: VBSCRIPT (che deriva dal Visual Basic e viene riconosciuto però solo da Internet Explorer) e JAVASCRIPT che deriva invece da JAVA ed è riconosciuto più o meno da tutti i browser.

Il tag APPLET che consente di inserire in una pagina web un programma scritto in JAVA. La condizione per cui il programma possa essere eseguito è che la Java Virtual Machine sia installata sul computer.

Il tag OBJECT (Internet Explorer su Windows) che insieme al **tag EMBED** (browser vari e Internet Explorer su Mac) consentono di inserire nelle pagine web oggetti di varia natura gestiti da estensioni dei browser o (in Windows) dai oggetti gestiti direttamente dal sistema operativo (i cosiddetti ActiveX). Qui segnaliamo due oggetti/plug-in che negli ultimi anni hanno avuto una grande diffusione: **SHOCKWAVE FLASH** e **SHOCKWAVE DIRECTOR** orientati alla sviluppo di applicazioni multimediali interattive e dotati di un potente linguaggio di programmazione.

Un altro importante livello in cui l'attività di programmazione entra nello sviluppo di pagine web è la programmazione lato server.

I tipi di pagine web di cui abbiamo fin qui parlato sono pagine statiche. Pagine, cioè, che vengono inviate al browser dal server web esattamente come sono state composte. Molti siti web oggi usano, invece, pagine dinamiche, pagine, cioè, il cui contenuto viene composto dal server al momento della richiesta sulla base dei parametri che gli vengono passati dal browser e del contenuto di database che il server può consultare.

Questo tipo di programmazione, un tempo riservato agli specialisti, è oggi alla portata di qualsiasi programmatore. Gli strumenti più diffusi sono PERL e PHP per i server web che girano in ambiente Unix e Linux e ASP e la sua più recente evoluzione ASP.NET per Internet Information Server (il server web Microsoft).

Applicazioni multimediali off line

Per quanto riguarda la Creazione di Giochi gli sviluppatori tendono a usare (per ottenere la massima efficienza) strumenti il più vicino possibile al linguaggio dei processori quindi **C++ e Assembler**. Per giochi non troppo complessi, però, anche *FLASH, DIRECTOR e JAVA* possono essere buone soluzioni.

FLASH è probabilmente il prodotto più abbordabile per i neofiti. offre una potente gestione della grafica e delle animazioni ed è in grado di produrre applicazioni molto leggere che possono essere facilmente distribuiti anche su Internet. Le ultimi versioni (dalla 7.0 in poi) di Flash sono dotate di linguaggio molto potente ed evoluto (ACTION SCRIPT 2.0 derivato da Javascript) che consente di utilizzare tecniche di Object Oriented Programming.

DIRECTOR è l'ideale per produrre applicazioni multimediali off line multiplatforma (Windows e Macintosh). Offre un linguaggio di programmazione molto potente ed è in grado di gestire molto bene risorse multimediali diverse (suono, video, immagini vettoriali e bitmap, filmati flash, ecc.). Nelle ultime versioni, inoltre, lo sviluppatore può scegliere se utilizzare la vecchia sintassi del LINGO (linguaggio di programmazione proprietario sviluppato appositamente per Director) o la sintassi di JAVASCRIPT (sintassi derivata da Java, la stessa utilizzate nella programmazione dei browser e da Action Script).

Concludiamo questa breve panoramica spendendo qualche parola su JAVA. Il punto di forza di JAVA è la portabilità. Se sviluppate correttamente le applicazioni Java possono girare su un'infinità di dispositivi per cui (in forme diverse) è stata sviluppata la Java Virtual Machine (Windows, Macintosh, Unix, Linux, telefonini, palmari, ecc.).

Che cos'è un programma

Un computer è una macchina in grado di eseguire un determinato set di istruzioni. Per poter svolgere un compito il calcolatore ha bisogno di qualcosa che gli dica passo passo cosa deve fare, o meglio, quale delle istruzioni che conosce deve eseguire in un determinato momento. Un programma è costituito da istruzioni che il calcolatore è in grado di eseguire. L'elaboratore è in grado di eseguire istruzioni basiche ovvero a basso livello. Un programma scritto nel linguaggio che il computer è in grado di capire sarebbe per noi incomprensibile. Ci aiutano a scrivere programmi per noi comprensibili i linguaggi di programmazione. Dato un problema o un compito è necessario quindi "tradurlo" in istruzioni nel linguaggio che abbiamo scelto. Prima di iniziare a programmare è bene analizzare il problema e individuarne la soluzione indipendentemente dal quale linguaggio useremo. Descrivere la soluzione di un problema o lo svolgimento di un compito sotto forma di passi discreti, eseguibili e non ambigui significa costruire un ALGORITMO.

L'algoritmo

Un ALGORITMO quindi è un procedimento che consente di ottenere un risultato eseguendo, in un determinato ordine, un insieme di passi semplici corrispondenti ad azioni scelte solitamente da un insieme finito.

Le proprietà fondamentali di un algoritmo sono: non ambiguità (il procedimento deve essere interpretabile in modo univoco da chi lo deve eseguire), eseguibilità (ogni istruzione dell'algoritmo deve poter essere eseguita senza ambiguità da parte di un esecutore reale o ideale), finitezza (sia il numero di istruzioni che compongono l'algoritmo, che il tempo di esecuzione devono essere finiti).

Prendiamo il noto problema del contadino che deve far attraversare il fiume ad una capra, un cavolo e un lupo, avendo a disposizione una barca in cui può trasportare solo uno dei tre alla volta. Se incustoditi, la capra mangerebbe il cavolo e il lupo mangerebbe la capra.

Inizio

Porta la capra sull'altra sponda

Porta il cavolo sull'altra sponda

Riporta la capra sulla sponda di partenza

Porta il lupo sull'altra sponda

Porta la capra sull'altra sponda

Fine

Come secondo esempio vediamo l'algoritmo che sta alla base di qualsiasi computer e che stabilisce come funziona la CPU.

Questo algoritmo è intrinseco nella struttura della CPU e fa in modo che una volta inserito un programma nella memoria, questo possa essere letto istruzione per istruzione ed eseguito. Tale caratteristica sta alla base della programmabilità.

**finché non trovi un'istruzione di halt fai:
preleva un'istruzione dalla memoria
decodifica l'istruzione
esegui l'istruzione**

L'algoritmo

Un ALGORITMO quindi è un procedimento che consente di ottenere un risultato eseguendo, in un determinato ordine, un insieme di passi semplici corrispondenti ad azioni scelte solitamente da un insieme finito.

Le proprietà fondamentali di un algoritmo sono: non ambiguità (il procedimento deve essere interpretabile in modo univoco da chi lo deve eseguire), eseguibilità (ogni istruzione dell'algoritmo deve poter essere eseguita senza ambiguità da parte di un esecutore reale o ideale), finitezza (sia il numero di istruzioni che compongono l'algoritmo, che il tempo di esecuzione devono essere finiti).

Prendiamo il noto problema del contadino che deve far attraversare il fiume ad una capra, un cavolo e un lupo, avendo a disposizione una barca in cui può trasportare solo uno dei tre alla volta. Se incustoditi, la capra mangerebbe il cavolo e il lupo mangerebbe la capra.

Inizio

Porta la capra sull'altra sponda

Porta il cavolo sull'altra sponda

Riporta la capra sulla sponda di partenza

Porta il lupo sull'altra sponda

Porta la capra sull'altra sponda

Fine

Rappresentazione di un algoritmo

Al fine di rappresentare gli algoritmi in modo comprensibile per il programmatore, vengono principalmente utilizzate due tecniche:

Rappresentazione mediante **pseudolinguaggio** o pseudocodice che possiamo definire come un linguaggio informale (ma non ambiguo) per la descrizione delle istruzioni. Ogni riga rappresenta un'istruzione. Se non diversamente specificato, le righe si leggono dall'alto verso il basso.

Rappresentazione mediante **diagramma di flusso** (**flowchart**) che è una rappresentazione grafica in cui ogni istruzione è descritta all'interno di un riquadro e l'ordine di esecuzione delle istruzioni è indicato da frecce di flusso tra i riquadri.

Esempio delle due rappresentazioni dell'algoritmo che calcola la somma di due numeri interi usando solo istruzioni di incremento e decremento

Somma di due numeri interi utilizzando solo incrementi e decremento

Leggi A
Leggi B
Se A=0 vai a 7
Decrementa A
Incrementa B
Vai a 4
Stampa B

