

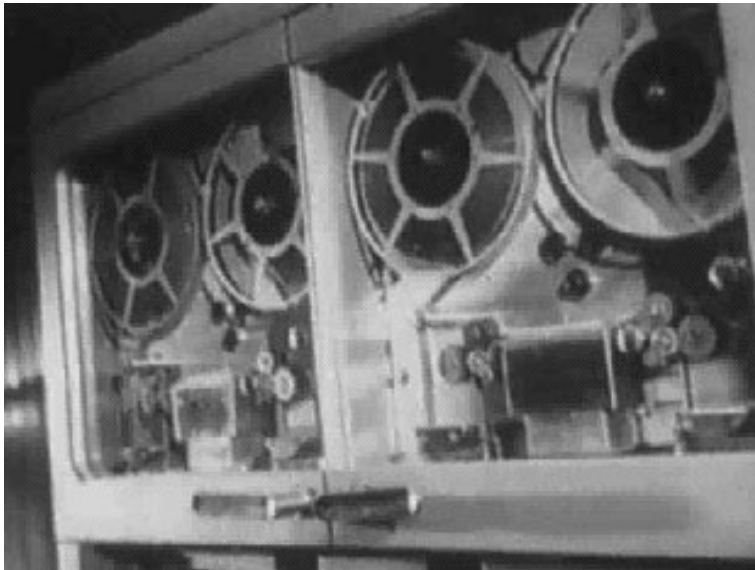
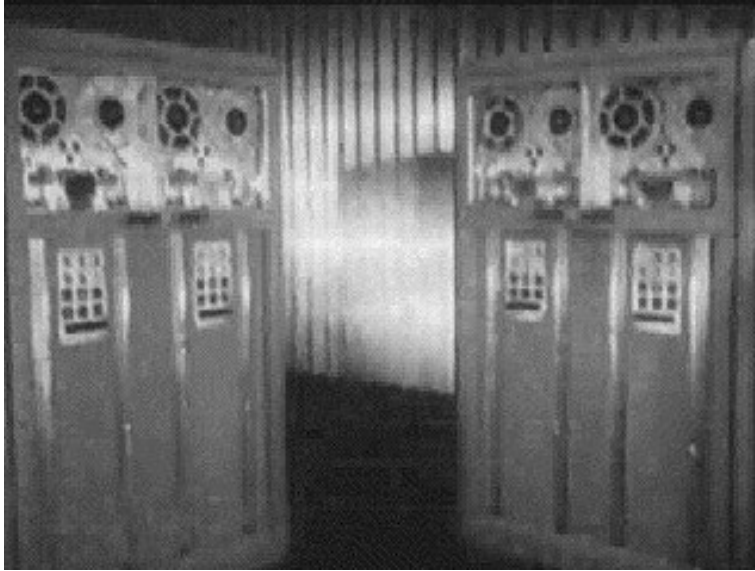
STORIA DELL'INFORMATICA

***Storia dei linguaggi
informatici***

Classe di concorso A042

Corsista D'ELIA WANDA

In principio era il bit...



ENIAC

Quello che può essere considerato il progenitore di tutti i computer è sicuramente l'ENIAC (Electronic Numerical Integrator And Calculator), un mastodontico calcolatore pesante 30 tonnellate, al cui progetto partecipò il brillante matematico J. von Neumann. Noi ora, "viziati" dalle semplici tecniche di programmazione di Basic sicuramente sorridiamo pensando all'ENIAC. Per quel tempo era una vera e propria innovazione. Infatti fu il primo elaboratore programmabile interamente a circuiti elettronici e senza parti meccaniche in movimento, ma questa è un'altra storia...

FORTRAN (FORmula TRANslation)



Ideato negli anni '50, fu uno dei primi linguaggi di programmazione. Il **FORTRAN** è stato sviluppato da un gruppo di programmatori della **IBM** guidati da **John Backus** e pubblicato per la prima volta nel 1957. Il FORmula TRANslator era stato progettato per facilitare la traduzione in codice di formule matematiche.

Spesso definito come linguaggio scientifico, il FORTRAN è stato anche il primo **linguaggio di alto livello**, e quello che ha per primo utilizzato un **compilatore**. Prima di allora i programmi andavano sviluppati esclusivamente in linguaggio macchina.

FORTRAN (FORmula TRANslation)

L'obiettivo era di creare un linguaggio **semplice da imparare**, impiegabile in un ampio raggio di esigenze, che fosse **indipendente dalla macchina**, e che consentisse l'esecuzione di complesse funzioni matematiche, scritte il più possibile in modo analogo a quello usato normalmente.

Alcuni anni dopo la sua prima comparsa si sono sviluppati diversi "dialetti" derivanti dal FORTRAN originale.

Diverse sono le versioni Fortran apparse: il FORTRAN I nel 1957, il FORTRAN II nel 1958, il FORTRAN III nel 1958 (usato da una ventina di clienti dell'IBM, ma mai rilasciato come prodotto commerciale per la sua mancanza di **portabilità**), il FORTRAN IV nel 1961, il FORTRAN 66 nel 1966, il FORTRAN 77 nel 1977, il Fortran 90 nel 1990, il Fortran 95 nel 1995 e il Fortran 2003 nel 2003.

FORTRAN (FORmula TRANslation)

Il FORTRAN II introdusse la possibilità di compilazioni separate, il FORTRAN III la possibilità di usare espressioni **booleane** (o logiche) e quella di inserire "in linea" codice in linguaggio assembler (cioè di mescolare istruzioni Fortran e istruzioni in **linguaggio assembler**). Fu questo a renderlo non portabile, poiché ciascun assembler ovviamente è specifico di un dato calcolatore, non essendo altro, in ultima analisi, che una forma simbolica del suo linguaggio macchina. Le espressioni booleane furono introdotte anche nel FORTRAN IV, insieme alla nuova istruzione condizionale IF in grado di eseguire un test sulla veridicità di un'espressione booleana, che si aggiungeva alla precedente istruzione condizionale IF presente nel FORTRAN II (che a sua volta l'aveva ereditata dal FORTRAN I).

FORTRAN (FORmula TRANslation)

I primi programmi in FORTRAN venivano perforati per lo più su schede a 80 colonne, con regole molto stringenti. Una riga di codice non poteva superare i 72 caratteri, che venivano perforati nelle colonne da 1 a 72. In particolare se la colonna 1 conteneva una C i caratteri successivi costituivano un commento; le colonne da 1 a 5 potevano contenere un'etichetta numerica che identificava l'istruzione e che poteva essere usata per saltare all'istruzione stessa da un altro punto del programma; le colonne da 7 a 72 contenevano l'istruzione; la colonna 6 era normalmente vuota e, se conteneva un carattere, indicava che l'istruzione era il seguito dell'istruzione perforata nella scheda precedente. Le colonne da 73 a 80 venivano spesso usate per numerare le schede e permettere così di riordinarle nel caso fossero state accidentalmente mescolate.

FORTRAN (FORmula TRANslation)

Con l'avvento del Fortran 90, una revisione "**principale**" del linguaggio, queste regole stringenti sono state abbandonate in favore del codice in formato libero. Altre notevoli innovazioni introdotte dal Fortran 90 sono l'allocazione **dinamica della memoria**, le operazioni sugli array, i tipi di dati astratti, l'overloading (o sovraccarico) degli operatori, i **puntatori** e i moduli (questi ultimi consentono di raggruppare sottoprogrammi e dati correlati).

Il Fortran 95, una revisione "minore" del linguaggio, ha introdotto essenzialmente la possibilità della programmazione parallela.

Lo standard formale più recente del linguaggio è il Fortran 2003 (anche se il documento che lo definisce è stato pubblicato nel 2004). Si tratta di un'estensione propria del Fortran 95, che include l'aritmetica in virgola mobile, la gestione delle eccezioni, costrutti per la programmazione orientata agli oggetti e una migliorata interoperabilità col linguaggio C.

FORTRAN (FORmula TRANslation)

Data la sua semplicità di scrittura i programmatori riuscivano ad essere fino a 500 volte più veloci in FORTRAN che in altri linguaggi, dedicando così maggiormente la loro attenzione sui problemi reali piuttosto che sulla trascrizione nel linguaggio.

Si può quindi affermare che FORTRAN è stato anche il primo linguaggio "**problem oriented**" anziché "machine oriented".

Anche per il FORTRAN, come per altri linguaggi, sono state redatte delle specifiche di standardizzazione ANSI, ma che di volta in volta non hanno scongiurato il proliferare di nuove versioni non proprio compatibili tra loro.

L'ALGOL



L'ALGOL è stato sviluppato congiuntamente da un comitato di informatici statunitense ed europeo. Esistono tre differenti sintassi: una sintassi di riferimento, una sintassi per la pubblicazione ed una per l'implementazione.

Le differenti sintassi permisero di usare differenti parole chiave e convenzioni per i punti decimali (punti o virgole) a seconda della lingua utilizzata. La citazione completa è: "Qui c'è un linguaggio così avanzato che non è solo un miglioramento rispetto ai predecessori ma anche rispetto ai propri successori".

L'ALGOL

A volte viene attribuito erroneamente a Edsger Dijkstra, conosciuto per le acute osservazioni, che collaborò all'implementazione del primo compilatore per ALGOL 60.

Per una trentina d'anni è stato lo standard de facto con cui si rappresentavano gli algoritmi. L'ALGOL utilizza blocchi di istruzioni delimitati da coppie di begin ed end (come il Pascal); è stato il primo linguaggio ad usare questa sintassi molto utilizzata anche in didattica: frammenti di sintassi simile all'ALGOL sono tuttora utilizzati a volte come notazione per gli algoritmi chiamata Pidgin Algol.

L'ALGOL

Esistono tre versioni ufficiali di ALGOL: ALGOL 58, che è stato sviluppato da John Backus; ALGOL 60, creato da Backus e Naur e rivisto ed ampliato da Peter Naur, ed ALGOL 68. Di queste, solo ALGOL 60 si è diffusa. I nomi ufficiali delle versioni di ALGOL sono dovuti all'anno di pubblicazione. L'ALGOL 58 era conosciuto originariamente come **IAL** (l'acronimo di **I**nternational **A**lgorithmic **L**anguage.) L'ALGOL 60 ispirò molti dei linguaggi che seguirono; la citazione canonica a questo proposito è dovuta a Hoare: "ALGOL fu un grande progresso sui suoi successori".

COBOL (COmmon Business Oriented Language)



Progettato nel 1959, nasce ufficialmente nel 1961, grazie ad un gruppo di lavoro composto dai membri dell'industria americana e da alcune agenzie governative degli Stati Uniti con lo scopo di creare un linguaggio di programmazione adatto all'elaborazione di dati commerciali. Grace Murray Hopper ebbe un ruolo primario nello sviluppo e nella progettazione di questo linguaggio. Serviva soprattutto per sviluppare programmi gestionali, cioè atti alla risoluzione di problemi aziendali (fatturazione, contabilità, stipendi etc.).

Uno dei motivi della sua enorme diffusione è stato la sua discreta facilità, dovuta anche al fatto che le istruzioni somigliano molto a frasi inglesi. Ad esempio, la frase "aggiungi il valore di importo al valore totale" si tradurrebbe in COBOL così: "add import to totale". Trovando una grandissima applicazione nelle aziende, ebbe molto successo.

COBOL (COmmon Business Oriented Language)

Dagli anni '60 ad oggi, il COBOL ha subito continue evoluzioni: negli anni 1968, 1974 e 1985 l'American National Standards Institute (ANSI) ha definito gli standard Cobol68, Cobol74 e Cobol85, adottati anche dall'International Organization for Standardization (ISO).

Con l'ISO/IEC 1989-2002, iniziato nel 1989 e terminato nel 2002, si è giunti allo standard internazionale definitivo.

Gli applicativi Cobol, noti per la loro stabilità, sono alla base del funzionamento dei Bancomat e dell'operatività di molte banche.

COBOL (COmmon Business Oriented Language)

Il COBOL è un linguaggio ad alto livello di tipo procedurale, orientato allo sviluppo di applicazioni commerciali e per l'elaborazione di grossi volumi di dati. Le istruzioni, molto simili a frasi della lingua inglese, sono organizzate in uno schema dalla struttura rigida, composto di sezioni e paragrafi.

È disponibile sulle più diffuse piattaforme (Unix, Microsoft Windows, GNU/Linux) e permette l'accesso alle più diffuse basi di dati relazionali, come ad esempio DB2 e Oracle.

COBOL (COmmon Business Oriented Language)

Caratteristica dei programmi scritti in COBOL è la struttura sempre formata da quattro DIVISION, (divise a loro volta in SECTION), che debbono sempre essere collocate nel seguente ordine:

- IDENTIFICATION DIVISION: contiene i dati identificativi del programma (nome, autore, data ecc.)
- ENVIRONMENT DIVISION: contiene i dati relativi alle macchine su cui è stato sviluppato e su cui girerà il programma.
- DATA DIVISION: Contiene la struttura dei dati che è organizzata in modo gerarchico.
- PROCEDURE DIVISION: Contiene le istruzioni.

BASIC (Beginner's All purpose Symbolic Instruction Code).



Il **BASIC** (**B**eginner's **A**ll purpose **S**ymbolic **I**nstruction **C**ode cioè "Codice di istruzioni simboliche di uso generale per principianti") è un linguaggio di programmazione ad alto livello sviluppato a partire dal 1963 nell'Università di Dartmouth (Dartmouth College) per il calcolatore GE-225. Ne furono inventori i professori John George Kemeny e Thomas Eugene Kurtz. Il primo programma in BASIC "girò" il 1° maggio 1964, alle ore 4:00.

BASIC (Beginner's All purpose Symbolic Instruction Code).

Il BASIC (almeno nelle versioni di Kemeny e Kurtz, che costituiscono il cosiddetto Dartmouth BASIC) fu progettato per essere un linguaggio semplice da imparare. Per questo si differenziò dalla maggior parte dei linguaggi suoi coetanei enfatizzando la semplicità d'uso piuttosto che la potenza espressiva. Le istruzioni erano poche e non complesse. Il BASIC offriva inoltre pochi costrutti strutturati, quindi orientati alla creazione di programmi chiari e comprensibili. Fu pensato inoltre come un linguaggio algoritmico, adatto cioè a programmare algoritmi.

BASIC (Beginner's All purpose Symbolic Instruction Code).

Il BASIC fu anche progettato per essere un linguaggio facilmente trasportabile su calcolatori diversi dal GE-225 su cui giravano le prime versioni. Concepito e realizzato originariamente come compilatore, molte delle sue versioni più note fanno tuttavia uso di interpreti.

Una delle più famose versioni è il Microsoft BASIC, sviluppato da Bill Gates e Monte Davidoff per l'Altair 8800, poi evolutasi nel GW-BASIC per il primo MS-DOS. Negli anni dei primi microcomputers, tra cui il famoso Commodore 64, il linguaggio BASIC faceva parte del firmware della macchina.

BASIC (Beginner's All purpose Symbolic Instruction Code).

Con gli anni il BASIC ha subito notevoli evoluzioni e cambiamenti, diventando un linguaggio strutturato con potenzialità molto simili a quelle di altri linguaggi evoluti per professionisti. Le numerose versioni attuali del BASIC affiancano ai concetti originari nuovi concetti, quali la gestione degli oggetti, la ricorsività, la strutturazione, ecc. (ad esempio Visual Basic o REALbasic).

Sempre dal BASIC discende tutto il filone dei vari Visual BASIC, linguaggi visuali orientati agli oggetti, con l'interfaccia grafica (bottoni, finestre etc.). La più recente versione di BASIC visuale è il REALBasic della Real Software.

Il Pascal

Wirth era un docente di programmazione e sentiva la mancanza di un vero linguaggio di programmazione adatto alla didattica e che fosse dotato di strutture dati avanzate. Il BASIC, creato nel 1964, era facile da imparare ma non aveva strutture dati avanzate e soprattutto non incoraggiava affatto ad analizzare il problema da risolvere prima di scrivere il codice. Perciò, Wirth creò da zero il Pascal, inserendo nel linguaggio il concetto di programmazione strutturata.

Niklus Wirth, quindi, ideò e realizzò il Pascal affinché facilitasse l'applicazione delle regole e delle tecniche di programmazione. Il suo scopo era ottenere un linguaggio adatto per l'insegnamento della scrittura di programmi e centrò benissimo il suo obiettivo, tanto che il Pascal è ancora oggi molto usato nelle scuole.

Il Pascal

Il Pascal si basa sul linguaggio ALGOL e fu così chiamato in onore del matematico e filosofo francese **Blaise Pascal**, inventore della prima macchina calcolatrice automatica.

La prima implementazione del linguaggio divenne operativa nel 1970, ma raggiunse una discreta diffusione nel campo industriale nel 1973 con la pubblicazione del testo: "Pascal user manual and Report". Questo è il testo di riferimento standard per il linguaggio.

Le caratteristiche principali del Pascal sono una sintassi molto chiara e molto rigida, l'obbligo di dividere il programma in sezioni ben definite (uses, declaration, implementation) e di dichiarare in anticipo tutte le variabili usate nel programma; permette l'uso di tipi di dati complessi e definibili dal programmatore (i record); permette, inoltre, (con alcune limitazioni) l'uso dei puntatori e l'allocazione dinamica della memoria (new e dispose).

Il Pascal

Il Pascal è pertanto classificato come linguaggio semidinamico, in quanto gestisce i puntatori, permettendo di creare liste di variabili di dimensione non nota all'atto della compilazione, ma questi possono essere solo di un tipo alla volta, il tipo usato per definire la variabile puntatore stessa. Al posto delle subroutine di uso comune nei linguaggi di allora, il Pascal introduce le procedure e le funzioni, che estendono il concetto di subroutine incapsulando al loro interno le variabili e i dati locali, ed esponendo solo i valori in ingresso ed in uscita. Inoltre, nella sua formulazione originale, era privo dell'istruzione GOTO, aggiunta poi in seguito con riluttanza.

Il Pascal

Essendo un linguaggio pensato per degli utenti inesperti, spesso i compilatori Pascal hanno un atteggiamento materno nei confronti del programmatore, partendo dal presupposto che qualunque irregolarità del codice scritto sia un errore, che infastidisce un po' gli esperti. In cambio è un linguaggio completo e potente, e soprattutto educa fin dall'inizio ad adottare un buono stile di programmazione e ad analizzare a fondo il problema prima di cominciare a scrivere il codice, cosa che rende i programmi Pascal più chiari e più facili da leggere.

Data la diffusione negli ambienti scolastici, ne sono state create versioni ad oggetti, come il Turbo Pascal e l'Object Pascal (utilizzato nell'ambiente di sviluppo Delphi di Borland).

Il linguaggio C



Fu ideato nei Bell Laboratories della AT&T nel 1972 da Dennis Ritchie come evoluzione del linguaggio B di Ken Thompson usato per la scrittura dei primi sistemi operativi UNIX. Lo stesso Thompson nel 1970 si era a sua volta ispirato al linguaggio BCPL di Martin Richards, anch'esso pensato per scrivere sistemi operativi e software di sistema.

La definizione formale si ha nel 1978 a cura di B. W. Kernighan e D. M. Ritchie.

Nel 1983 iniziò il lavoro di definizione di uno standard da parte dell'American National Standards Institute, che rilasciò nel 1990 lo Standard ANSI C (ISO C89).

Il linguaggio C

È stato talvolta definito come "il linguaggio di più basso livello tra i linguaggi ad alto livello", infatti nasce per lo sviluppo di sistemi operativi, quindi per software di basso livello, ma riesce a mantenere semplicità d'uso. Il trucco usato dai suoi sviluppatori per mantenere compatto il linguaggio, e nel contempo estenderne le funzionalità, sta nell'affidare le funzioni più complesse a un'insieme di librerie esterne: esattamente come il MacOS, al quale, per aggiungere funzionalità aggiungiamo estensioni e librerie.

Il C è tecnicamente un linguaggio di programmazione ad alto livello. Tuttavia, poiché esso mantiene evidenti relazioni semantiche con il linguaggio macchina e l'assembly, risulta molto meno astratto di linguaggi anche affini, come per esempio il Pascal.

Il linguaggio C

Per questo motivo, talvolta viene anche identificato con la locuzione (più ambigua) linguaggio di medio livello, se non addirittura (in modo certamente improprio) come macro-assembly, o assembly portabile.

Il C è rinomato per la sua efficienza, e si è imposto come linguaggio di riferimento per la realizzazione di software di sistema su gran parte delle piattaforme hardware moderne. La standardizzazione del linguaggio (da parte dell'ANSI prima e dell'ISO poi) garantiscono la portabilità dei programmi scritti in C (standard, spesso detto **ANSI C**) su qualsiasi piattaforma.

Il linguaggio C

Oltre che per il software di sistema, il C è stato a lungo il linguaggio dominante in tutta una serie di altri domini applicativi caratterizzati da forte enfasi sull'efficienza. Esempi tipici sono le telecomunicazioni, il controllo di processi industriali e il software real-time. Oggi il predominio del C in questi contesti è in parte diminuito a seguito dell'avvento di competitor significativi, primo fra tutti il C++; tuttavia, il tempo in cui il C si potrà considerare obsoleto appare ancora molto lontano.

Il linguaggio C

Il C ha, e continua ad avere, anche una notevole importanza didattica, sebbene, per la sua complessità semantica e per le forti relazioni di tale semantica con il funzionamento dell'hardware dei computer, non si tratti di un linguaggio particolarmente intuitivo per i principianti, e in special modo a quelli sprovvisti di un adeguato background sull'elettronica dei calcolatori. Se un tempo scuole superiori e corsi universitari adottavano il C come linguaggio di riferimento a causa della sua importanza tecnica, oggi questa scelta trova un'ulteriore motivazione nella crescente importanza di linguaggi che dal C derivano (per esempio C++, Java e C#).

Java



Java, la cui creazione risale a metà degli anni '90, è stata un'invenzione rivoluzionaria e indispensabile nel mondo della programmazione e non solo per i browser.

La SUN, qualche anno fa presentò il Java come un linguaggio di programmazione Object Oriented semplice e familiare a chi conosce il C++, indipendente dall'architettura, e sicuro per l'uso in rete. In effetti, uno stesso programma funziona su Mac, PC , Unix e SGI, senza richiedere modifiche o ricompilazioni. Però, per chi non conosce il C++, Java risulta difficile da imparare come qualsiasi altro linguaggio.

Java

Essendo un linguaggio multiplatforma, il compilatore non crea degli applicativi eseguibili (APPL nel Mac o .exe in Windows) dal computer, ma dei file.class che devono essere interpretati dalla JVM (Java Virtual Machine, un microprocessore virtuale) implementata sui vari sistemi. Ciò ha sempre reso l'esecuzione di un'applicazione Java molto più lenta dei programmi normali, e solo a partire dalla versione 3 di Netscape Navigator ed Explorer, grazie al compilatore Just In Time, l'esecuzione ha iniziato ad avere tempi accettabili. Dal punto di vista economico-aziendale, il Java, si è rivelato vincente, abbattendo i costi e il tempo per la conversione di programmi C/C++ per piattaforme differenti.

Java

Java venne creato per soddisfare quattro scopi:

- essere orientato agli oggetti
- essere indipendente dalla piattaforma
- contenere strumenti e librerie per il networking
- essere progettato per eseguire codice da sorgenti remote in modo sicuro.

Per facilitare il passaggio a questo linguaggio per i programmatori old-fashioned legati in particolare a linguaggi come il C la sintassi di base (cicli, operatori, ...) è stata mantenuta pressoché identica.

Java

Rispetto alla tradizione dei linguaggi a oggetti da cui deriva (e in particolare rispetto al suo diretto progenitore, il C++), Java ha introdotto una serie di notevoli novità rispetto all'estensione della sua semantica. Fra le più significative si possono citare probabilmente la possibilità di costruire GUI (interfacce grafiche) con strumenti standard e non proprietari (per il C++ e altri linguaggi analoghi solitamente le GUI non fanno parte del linguaggio, ma sono delegate a librerie esterne), la possibilità di creare applicazioni multi-thread, ovvero che svolgono in modo concorrente molteplici attività, e il supporto per la riflessione, ovvero la capacità di un programma di agire sulla propria struttura e di utilizzare classi caricate dinamicamente dall'esterno.

HTML (Hyper Text Mark-Up Language)



È stato sviluppato alla fine degli anni '80 da Tim Berners-Lee al CERN di Ginevra. Verso il 1994 ha avuto una forte diffusione, in seguito ai primi utilizzi commerciali del web. È un linguaggio usato per descrivere i documenti ipertestuali disponibili nel Web. Non è un linguaggio di programmazione, ma un linguaggio di markup, ossia descrive il contenuto, testuale e non, di una pagina web. Punto HTML (.html) o punto HTM (.htm) è anche l'estensione comune dei documenti HTML.

Durante gli anni l'HTML ha subito molte revisioni e miglioramenti, che sono stati indicati secondo la classica numerazione usata per descrivere le versioni dei softwares. Attualmente l'ultima versione disponibile è la versione 4.01, resa pubblica il 24 dicembre 1999.

UML (Unified Modeling Language)

È un linguaggio di modellazione e specifica basato sul paradigma object-oriented. Il nucleo del linguaggio fu definito nel 1996 da Grady Booch, Jim Rumbaugh e Ivar Jacobson (detti "i tre amigos") sotto l'egida dello OMG, che tuttora gestisce lo standard di UML. Il linguaggio nacque con l'intento di unificare approcci precedenti (dovuti ai tre padri di UML e altri), raccogliendo le best practices nel settore e definendo così uno standard industriale unificato.

UML svolge un'importantissima funzione di lingua franca nella comunità della progettazione e programmazione a oggetti. Gran parte della letteratura di settore usa UML per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico.

UML (Unified Modeling Language)



L'ultima versione del linguaggio, la 2.0, è stata consolidata nel 2004 e ufficializzata da OMG nel 2005. UML 2.0 riorganizza molti degli elementi della versione precedente (1.5) in un quadro di riferimento ampliato e introduce molti nuovi strumenti, inclusi alcuni nuovi tipi di diagrammi.

Sebbene OMG indichi UML 2.0 come la versione "corrente" del linguaggio, la transizione è di fatto ancora in corso; le stesse specifiche pubblicate da OMG sono ancora non completamente aggiornate e il supporto dei tool a UML 2.0 è, nella maggior parte dei casi, appena abbozzato.

SITOGRAFIA

- www.wikipedia.it
- www.windoweb.it
- www.ilmac.net
- www.prometheo.it
- www.dsi.uniroma1.it

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.