RIASSUNTO INFORMATICA

L'ARCHITETTURA DEI CALCOLATORI: tutta la teoria dei calcolatori si basa sulla macchina di Von Neumann, che può essere sintetizzata cosi:

- Cpu, unità centrale di elaborazione (processore)
- Memoria di tipo eletronico o magnetico
- Unità periferiche
- Componenti BUS, mettono in collegamento i vari elementi.

IL **SISTEMA OPERATIVO**: il sistema operativo opera con l'iniziale funzione di rendere utilizzabili le risorse fisiche presenti nel sistema informatico. Tra i suoi compiti troviamo il controllo dell'esecuzione di un'applicazione, l'accesso ai dispositivi di ingresso e di uscita, l'archiviazione dei programmi (gestite in apposite cartelle o file), il controllo di accesso (con relativi meccanismi di protezione come username o password), la contabilizzazione (monitoraggio dell'uso delle risorse) e infine la gestione dei malfunzionamenti. Grazie al sistema operativo, che funge da intermediario tra utente e software, è possibile sviluppare programmi in modo semplice a prescindere dal calcolatore. Al giorno d'oggi si sono sviluppati tre tipi di sistemi operativi: quelli distribuiti (la computazione divisa tra più processori), quelli embedded (una sola applicazione supportata, elettrodomestici e automobili) e quelli multimediali e portatili (con supporto per l'iterazione multimediale e interfaccia multitouch).
2. Un sistema operativo è un software di sistema che gestisce le risorse hardware e software della macchina, fornendo servizi di base ai software applicativi. Esso rende utilizzabile le risorse fisiche presenti nel sistema informatico, fornisce all'utente una serie di comandi e servizi per usufruire al meglio della potenza di calcolo degli elaboratori elettronici.
Esso gestisce l'operatività di base di un calcolatore coordinando e gestendo le risorse hardware di processamento e memorizzazione periferiche, software e interfaccia con l'utente, e applicazioni o programmi specifici. Il sistema operativo è un componente essenziale che si interfaccia con altri software che appoggiandosi a lui, dovranno essere progettati per essere riconosciuti e supportati dal SO.
Esistono tre principali elementi di un sistema operativo: sistema di gestione del processore, sistema di gestione della memoria e sistema di gestione delle periferiche.
I VANTAGGI SI UN SO: sono legati alla disponibilità di definire modalità standard di interfaccia con i dispositivi fisici, cosicché sia possibile:

- Sviluppare programmi in modo semplice, modulare ed indipendente dallo specifico calcolartore su cui viene fatto funzionare il sistema operatico
- Aggiornare il software di base e l'hardware in modo trasparente ai programmi applicativi e all'utente, ossia senza che vengano influenzati dall'operazione

LA **MACCHINA VIRTUALE**: è un software che rappresenta schematicamente e idealmente il funzionamento dell'hardware di un qualsiasi calcolatore. Nella sua rappresentazione schematica quindi troviamo il nucleo (si occupa dell'esecuzione di programmi), la gestione della memoria (maschera la collocazione fisica dei dati e protegge i programmi caricati nella memoria di lavoro, inoltre espande lo spazio di memoria in maniera irreale per gestire la sovrapposizione dei dati), il gestore delle periferiche, il File System (gestisce la memoria di massa e i file) e l'interprete dei comandi (attraverso i comandi di lettura interpreta i comandi che gli giungono dai dispositivi periferici). A questo segue il controllo degli accessi (un account per utente con user e password) e infine il middleware, insieme di librerie di utilità che viene standardizzato (ad esempio il Java Runtime Enviroment).

COSA SI INTENDE PER **FILE**: per file si intende un contenitore logico di informazioni che permette di conservare informazioni anche dopo la terminazione del processo che lo ha generato.
Per file si intende un contenitore logico di informazioni che permette di conservare informazioni anche dopo la terminazione del processo che lo ha generato. Per ogni file è presente un identificatore, una periferica, una data di creazione, una dimensione ed altre informazioni. I nomi dei file sono generalmente composti da

2 parti, il nome che viene assegnato dall'utente e l'estensione, che consente di identificare la tipologia di dati contenuti nel file. (Si può allargare il discorso parlando del file system e della gestione della memoria).

FILE SYSTEM: indica un meccanismo con il quale i file sono posizionati e organizzati su dispositivi di archiviazione. Esso quindi si può considerare un gestore di file e gestisce anche la memoria di massa.
È un componente del sistema operativo che gestisce l'organizzazione logica delle informazioni memorizzate sui dispositivi di memoria secondaria, come disco rigido, cd o floppy. La gestione della memoria di massa ha come obiettivo quello di presentare all'utente l'organizzazione logica dei dati e le operazioni che è possibile compiere su di essi. Le operazioni di base di un file system sono: - recupero di dati precedentemente memorizzati, - eliminazione di dati obsoleti, - modifica/aggiornamento di dati pre-esistenti, -copia di dati. Il file system è un contenitore logico di informazioni utilizzatom per conservare anche dopo la terminazione del processo che lo ha generato. Ogni file ha: un identificatore, una periferica, una data di creazione, una dimensione, una posizione, una posizione effettiva dei dati nella memoria di massa e altre informazioni specifiche.

LA **MEMORIA**: deve fornire alla CPU i dati nel modo più veloce possibile e contemporaneamente deve anche garantire l'archiviazione di dati e programmi garantendone la conservazione. I criteri per cui si misura la memoria sono: la velocità, la capacità, il costo per bit e la volatilità (basata sulla necessità o meno di alimentazione per mantenere i dati archiviati). La memoria centrale è detta RAM, mantiene al proprio interno i dati e le istruzioni, è sviluppata con tecnologia elettronica (veloce ma volatile e costosa) che contiene: ROM (elettronica e di sola lettura) e Flash (elettronica ma non volatile e riscrivibile).
Le due tipologie di memorie: le memorie si dividono in grandi e lente, che possiedono grande capacità e bassa velocità e sono economiche e accessibili tramite il bus, e le memorie piccole e veloci, che vengono integrate nello stesso chip della cpu ma sono molto costose. Mettendo assieme queste due tipologie di memorie è possibile combinare una memoria grande (cosicché possa fornire grande spazio di archiviazione), e veloce (cosiché possa rifornire adeguatamente le richieste della cpu), nella memoria piccola e veloce saranno contenute una parte delle informazioni, attraverso alcuni principi statistici (principi di località spaziale e temporale) quelli più richiesti dal processore e nella memoria grande e lenta gli altri dati.
2. è l'elemento di un computer che ha il compito di memorizzare dei dati e programmi, esistono vari supporti di memorizzazioni. La memoria deve avere il compito di supportare la CPU, deve consentire l'archiviazione dei dati. Essa esegue questi compiti con precise esigenze e diverse tecnologie: elettronica (veloce, ma volatile e costosa), magnetica e ottica (non volatile, lenta ed economica). I parametri di caratterizzazione di una memoria sono: volatilità o meno, velocità, costo per bit e capacità.
La RAM (memoria centrale) mantiene al proprio interno i dati e le istruzioni dei programmi in esecuzione. Essa utilizza la tecnologia elettronica, quindi volatile e costosa ma con 2 eccezioni: ROM e Flash, la prima permanente e di sola lettura, la seconda volatile e riscrivibile.
Le informazioni vengono contenute in celle e indirizzi di memoria, i bit sono raggruppati in celle e tutte le celle sono formate dallo stesso numero di bit, ogni cella composta da k bit può contenere una qualunque tra 2^k combinazioni diverse di bit. Ogni cella ha un indirizzo che serve come accesso all'informazione. Esse sono numerate da 0 e sono numeri positivi interi. La cella è l'unità indirizzabile più piccola ed è formata da 8 bit (1 byte) ed a loro volta i byte vengono raggruppati in parole (32/64 bit) dove la CPU lavora ed esegue le operazioni. Una memoria può essere organizzata in diversi modi, con una certa quantità di bit si possono avere combinazioni differenti di celle e bit.

GERARCHIA DI MEMORIA: le memorie si differenziano per MGL e MPV, memoria grande e lenta (economica e accessibile tramite i bus) e memoria piccola e veloce (costosa e integrata nella cpu). L'obiettivo è quello di realizzare una memoria grande e veloce. La soluzione sta in una gerarchia, essa è una memoria formata da una MPV e da una MGL. La MPV contiene una copia di alcune celle della MGL. Quando la cpu chiede una particolare cella di memoria, la richiesta va ad entrambe le memorie, se il dato si trova nella MPV viene passato direttamente alla cpu, invece se il dato si trova nella MGL viene caricato anche nella MPV. Un sistema di memoria gerarchico può essere reso efficiente se la modalità di accesso ai dati ha caratteristiche prevedibili. Il principio di località si basa sul fatto che in un dato istante i programmi fanno accesso ad una porzione relativamente piccola del loro spazio di indirizzamento.

Esistono 2 tipi diversi di località:

- Località temporale: è probabile che un oggetto a cui si è fatto riferimento venga nuovamente richiesto in tempi brevi
- Località spaziale: è probabile che gli oggetti che si trovano vicini ad un oggetto a cui si è fatto riferimento vengano richiesti in breve tempo.

Con la località temporale i dati prelevati dalla MGL vengono conservati nella MPV il più a lungo possibile. Con la località spaziale quando si copia un dato dalla MGL alla MPV, si copiano anche i dati vicini. La frequenza di successo cresce fino al 90%.

MEMORIA DI MASSA: sono dispositivi che vengono collegati esternamente all'architettura di von neumann. Esso raccoglie tipicamente grandi quantità di dati rispetto alla memoria primaria e in maniera non volatile, cioè permanente almeno fino a quando non vuole l'utente. La caratteristica principale della memoria di massa è la non volatilità ovvero la possibilità di memorizzare permanentemente i dati. La memoria di massa ha un costo inferiore rispetto alla memoria principale poiché i tempi di accesso sono notevolmente più lunghi di quelli della memoria principale. La memoria di massa è un dispositivo che consente la registrazione, la conservazione e la rilettura dei dati. Può essere di tipo fisso o removibile, installato all'interno o all'esterno. Esempi di memorie di massa sono i dischi magnetici, floppy disk, memorie flash e dischi ottici.

DISCO MAGNETICO: è una tipologia di supporto di memorizzazione a memoria magnetica, di forma discoidale. È utilizzato come supporto di memoria di massa, esso è un dispositivo che viene collegato esternamente all'architettura di Von Neumann. Sono dei piatti di alluminio (o di altro materiale) ricoperti di materiale ferromagnetico. Questo materiale viene opportunamente magnetizzato permettendo di scrivere le informazioni sul disco magnetico. La forma, quindi il diametro del disco è importante perché più si risulta essere piccolo più risulterà avere una velocità di rotazione elevata.
Viene chiamato testina di un disco lo strumento di lettura/scrittura. È sospeso sopra la superficie magnetica e al passaggio di corrente sopra la testina magnetizza la superficie (scrittura). Invece il passaggio sopra un'area magnetizzata induce una corrente nella testina. La traccia è la sequenza circolare di bit scritta mentre il disco compie una rotazione completa. Il settore è una parte di traccia corrispondente a un settore circolare del disco. Esistono principalmente 2 tipi di dischi: disco rigido e disco flessibile. Disco rigido (o anche hard disk) è un disco caratterizzato da una buona rigidità fisica, ed è un supporto di memoria non removibile. Si può sia scrivere (sull'unità a disco rigido) sia leggere. Il disco flessibile o floppy disk è un disco magnetico caratterizzato da una buona flessibilità fisica, è un supporto di scrittura e lettura removibile.

PROCESSORE: il processore è un dispositivo elettronico organizzato con un'architettura BUS ORIENTED composta da:

- Registri: unità di memorizzazione temporanee, tra cui il PC-program counter che contiene l'indicazione su quale sia l'informazione prossima da svolgere. L'IR-instruction register che presenta l'informazione corrente e il PSW-program status word che memorizza gli esiti dei processi.
- UNITÀ ARITMETICO-LOGICHE: prendono in imput le informazioni dai registri e lì danno in output ad altri registri.
- UNITÀ DI CONTROLLO: che coordinano l'attività di elaborazione, prelevano nella fase di lettura l'informazione da eseguire.
- DATA PATH: è un insieme di unità di calcolo tipo ALU che prende le informazioni e le elabora.

2. È un tipo di dispositivo hardware che è dedicato all'esecuzione di istruzioni. Esso provvede all'esecuzione materiale dell'elaborazione dati di un programma tipicamente sotto la supervisione del sistema operativo attraverso il ciclo fetch-execute. Il processore è un elemento base dell'architettura degli elaboratori. L'unità di elaborazione centrale, cpu, è il processore che sovraintende tutte le funzionalità del computer digitale basato sull'architettura di von neumann. Questa nasce con l'unione di due processori: l'ALU (unità aritmetico logica) e l'unità di controllo. Un computer multiprocessore può avere anche più unità di elaborazione centrale che collaborano tra loro.

CPU: l'unità di elaborazione centrale è un tipo di microprocessore digitale che si contraddistingue per eseguire gran parte delle funzionalità del computer digitale basato sull'architettura di Von Neumann. È detta centrale perché coordina tutte le altre attività di elaborazione presenti nell'architettura hardware dei computer per la gestione delle varie periferiche. Il compito della cpu è quello di eseguire le istruzioni di un programma presente in memoria centrale (RAM) dopo averlo prelevato dalla memoria di massa (ROM). Durante l'esecuzione del programma presente la cpu legge o scrive dati in memoria centrale. Una cpu comprende 2 componenti principali: data path e unità di controllo. Il data path è la parte che si occupa dell'effettiva elaborazione dei dati, inoltre essa comprende le unità aritmetico-logiche dette ALU, che si occupano di eseguire le operazioni logiche e aritmentiche.
L'unità di controllo invece coordina le operazioni del processore, regola il flusso dei dati e indica quali registri debbano essere collegati agli ingressi e quali all'uscita dell'alu, memorizza il risultato scrivendolo in memoria o in un registro della cpu. Inoltre, comprende alcuni registri specifici. La cpu è in grado di eseguire solo istruzioni codificate in linguaggio macchina. Il suo ciclo si basa su fetch-decode-execute.

CICLO FETCH-DECODE-EXECUTE:

1. Prendi l'istruzione corrente dalla memoria (quella individuata dal contenuto del pc) e mettila nell'IR (instruction register), contemporaneamente incrementa il program counter (PC) in modo che contenga l'indirizzo dell'istruzione successiva (FETCH)
2. Determina il tipo di istruzione da eseguire (DECODE)
3. Se l'istruzione usa dei dati presenti in memoria, determinare la posizione
4. Carica i dati, se necessario, in uno o due registri della CPU
5. Esegui l'istruzione (EXECUTE)
6. Torna al punto 1 e inizia a eseguire l'istruzione successiva

JAVA

PROGRAMMAZIONE AD OGGETTI: la programmazione ad oggetti è una metodologia di programma che considera il programma come costruito da oggetti (o istanze) che si possono interagire da soli o fra loro. Un oggetto ha diverse caratteristiche dette attributi. Ad esempio, un oggetto automobile potrebbe avere attributi come il nome, la velocità corrente e il livello di carburante. Le azioni che un oggetto può effettuare sono dette comportamenti, e ciascun comportamento è definito in una porzione di codice detta metodo. Una classe invece viene definita come una sorta di stampo che consente di creare (istanziare) oggetti.

INTERPRETI: particolari tipi di linguaggi di alto livello non vengono tradotti in linguaggi di basso livello da compilatori ma da questi interpreti, che seguono ogni singola porzione di codice subito dopo averla tradotta invece di tradurre l'intero programma in una sola passata. Java combina questi due aspetti.

COMPILATORI: traducono il linguaggio di alto livello (java, visual basic, c++) in linguaggio di basso livello (quello comprensibile dalla macchina) tutto insieme.
Il compilatore in java (nel nostro caso Eclipse) non traduce il linguaggio alto del programma nel linguaggio macchina specifico del computer, ma lo traduce in un linguaggio specifico detto Bytecode. Il bytecode non è un linguaggio macchina di alcun computer, ma è il linguaggio macchina di una macchina virtuale simile a tutte quelle più diffuse. Il vantaggio è nel fatto che il bytecode non è leggibile (con il passaggio successivo), solo nel linguaggio macchina del computer su cui è in esecuzione ma in quello di tutti i computer, rendendo a Java il vantaggio della portabilità. Successivamente il Bytecode viene tradotto in linguaggio macchina dalla JVM (java virtual machine). Questa è contenuta nel pacchetto JDK, da noi installato contenente l'ambiente di sviluppo, le librerie, il louncher, il compilatore e il debugger.

POLIMORFISMO: il polimorfismo è una proprietà del linguaggio java che sfruttando l'ereditarietà delle classi permette di definire metodi che a seconda del contesto in cui sono evocati assumeranno specifiche diverse. Si andrà a progettare un metodo nella classe padre contenente un metodo proprio ad ogni classe figlia. Questo avviene grazie al binding dinamico ovvero durante la fase di compilazione del codice java non assegna direttamente ad un codice un diretto riferimento, solo nel momento dell'invocazione del codice java assegna

al metodo una definizione. A differenza del binding statico che assegna immediatamente ad ogni invocazione di metodo la sua definizione.
Il polimorfismo permette di modificare la definizione dei metodi nella classe derivata e far sì che questi cambiamenti siano effettivi anche per il codice della classe base. Per realizzarlo java adopera un meccanismo chiamato binding dinamico che significa che l'associazione tra invocazione e definizione del metodo viene prodotta quando il metodo è invocato. Se l'associazione venisse prodotta quando il codice è compilato si tratterebbe di binding statico. La definizione di polimorfismo sfrutta naturalmente l'ereditarietà delle classi.
2. L'ereditarietà permette di definire una classe base il cui codice può essere utilizzato non solo dagli oggetti creati da tale classe, ma anche dagli oggetti creati da ogni classe da essa derivata. Il polimorfismo permette di modificare la definizione dei metodi nella classe derivata e far si che questi cambiamenti siano effettivi anche per il codice della classe base. Il termine polimorfismo fa riferimento alla capacità di assegnare più significati a uno stesso nome di metodo, sfruttando il meccanismo di binding dinamico. Quindi essi fanno riferimento lo stesso concetto. Il binding dinamico fa sì che l'invocazione di un metodo venga associata alla sua definizione solamente a run-time, quindi nel momento in cui l'invocazione viene effettivamente eseguita. In Java tutti metodi non private, per default, solo ridefinibili ma è possibile specificare la keyword "final" per istruire il compilatore a non ammettere la ridefinizione.

CICLI: Vengono usati quando il programma ha necessità di ripetere una istruzione più volte. La parte di programma che ripete un'istruzione o un gruppo di istruzioni è chiamata ciclo. L'istruzione o il gruppo di istruzioni che vengono ripetuti nel ciclo sono chiamati corpo del ciclo. Ogni ripetizione del ciclo viene chiamato iterazione. Poi parla di while/do/do while.
2. la parte di programma che ripete un'istruzione o un gruppo di istruzioni è chiamata ciclo (loop). L'istruzione o il gruppo di istruzioni che vengono ripetuti nel ciclo sono chiamati corpo del ciclo. Ogni ripetizione del corpo del ciclo è chiamata iterazione del ciclo. È necessario definire un meccanismo che permetta di determinare quando la ripetizione del ciclo deve terminare. In java i cicli sono sostanzialmente 3: while, do-while e for.
Il ciclo while esegue una istruzione o un blocco di codice finchè rimane verificata una certa condizione. Quando l'espressione diventa falsa, la ripetizione termina. Il corpo del ciclo viene ripetuto fin tanto che l'espressione booleana di controllo rimane vera, spesso il corpo del ciclo è costituito da un'istruzione composta, racchiusa tra parentesi graffe.
Il ciclo do-while è molto simile all'istruzione while. La differenza principale consiste nel fatto che il corpo di un ciclo do-while viene sempre eseguito almeno una volta, mentre nel caso del ciclo while il corpo del ciclo potrebbe non essere mai eseguito.
Il ciclo for permette di scrivere facilmente un ciclo controllato da un contatore. La sintassi è la seguente:

- Un'espressione di inizializzazione eseguita solo una volta, prima di iniziare il ciclo
- Un'espressione di aggiornamento (incremento) da eseguire al termine di ogni esecuzione del blocco
- Una condizione di terminazione dell'esecuzione

Cicli difettosi o infiniti o sbagliati di una unità.

STRINGHE: per elaborare e creare stringhe di caratteri java usa una classe chiamata string. Un valore di tipo string è una stringa racchiusa tra doppi apici, si tratta quindi di una sequenza di caratteri considerati come se fossero un singolo elemento. Una stringa può contenere un numero qualsiasi di caratteri (vedi stringa vuota). Le stringhe possono essere concatenate. Poiché una variabile string non è primitiva (int, double) né un oggetto appartenente ad una classe string, possiede metodi. (charAt, IndexOf, Lenght, substring…).
2. una stringa è una catena di caratteri, questa catena di caratteri è in realtà un oggetto. Ogni qualvolta si dichiara una variabile di tipo string, java alloca dello spazio in memoria per quella variabile. Inoltre java fornisce una classe chiamata string che può essere utilizzata per creare ed elaborare stringhe di caratteri. Questa classe definisce decine di metodi. È possibile creare stringhe contenenti zero caratteri anche chiamate stringhe vuote. Una stringa deve essere delimitata da doppi apici. Le stringhe sono costanti perciò sono immutabili, una volta creati non può essere modificato il loro contenuto, una volta creato l'oggetto di tipo string, avremo modo solo di leggerlo o di leggere le sue parti.
Per la classe string non è necessario invocare il metodo new, infatti si possono dichiarare variabili della classe string e assegnare a loro le stringhe arbitrarie. Si possono concatenare 2 stringhe per crearne una di maggiori dimensioni, questa operazione di concatenazione avviene con l'utilizzo del +.

L'invocazione di un metodo si ottiene scrivendo il nome dell'oggetto seguito dal punto e dal nome del metodo ed infine da una coppia di parentesi tonde. Con il termine substring (sottostringa) si indica una porzione di stringa. Una variabile string non è una variabile semplice come int, ma si tratta di un oggetto appartenente alla classe string. Gli oggetti possiedono metodi e dati. Gli oggetti della classe string memorizzano dati costituiti da stringhe di caratteri, e i metodi forniti dalla classe string consentono di elaborare questi dati. La maggior parte dei metodi di string restituisce un valore.

COSTRUTTORI: un costruttore è un particolare metodo che viene invocato quando si usa l'operatore new per creare un nuovo oggetto. I costruttori hanno essenzialmente lo stesso compito dei metodi set. Ma a differenza dei metodi set, i costruttori creano un oggetto oltre ad inizializzarlo. Come i metodi set, i costruttori possono avere parametri. Una proprietà di costruttori è che ogni costruttore ha lo stesso nome della sua classe. Le intestazioni dei costruttori non contengono la parola void e nemmeno altro tipo di ritorno. Un costruttore senza parametri è chiamato costruttore di default.
2. quando si crea un oggetto di una classe utilizzando l'operatore new, si invoca un particolare tipo di metodo chiamato costruttore. In quel momento spesso è opportuno compiere operazioni di inizializzazione, come assegnare specifici valori alle variabili di istanza. Un costruttore eseguirà queste inizializzazioni. Un costruttore è un particolare metodo che viene invocato quando si utilizza l'operatore new per creare un oggetto. Il costruttore è quel metodo di una classe il cui compito è quello di creare istanze, oltre a essere il punto del programma in cui un nuovo elemento viene creato ed è reso disponibile per l'interazione con il resto del sistema. In java possono esserci molteplici costruttori per la stessa classe e ne esiste sempre almeno uno. Se infatti per una data classe non viene specificato alcun costruttore, il compilatore ne crea automaticamente uno e viene detto costruttore di default (esso è senza parametri). I costruttori hano essenzialmente lo stesso compito dei metodi set, ma a differenza di questi ultimi, i costruttori creano un oggetto oltre ad inizializzarlo.

EREDITARIETÀ: l'ereditarietà permette di definire una classe in una forma molto generale e, in un secondo momento, di utilizzarla come base di partenza per definire nuove classi, che sono specializzazioni della classe generale. Queste nuove classi ereditano i metodi e le variabili di istanza della classe generale e definiscono nuove variabili di istanza e nuovi metodi. L'ereditarietà rafforza il riutilizzo del software. La classe principale viene chiamata classe base o superclasse, quella secondaria classe figlia.
2. essa permette di definire una classe in una forma molto generale e in un secondo momento di utilizzarlo come base di partenza per definire nuove classi, che sono specializzazioni della classe generale. Queste nuove classi ereditano i metodi e le variabili di istanza della classe generale e definiscono nuove variabili di istanza e nuovi metodi. Per questo motivo l'ereditarietà rafforza il riutilizzo del software. L'ereditarietà permette di definire una classe più in generale e di definire in seguito classi specializzate che aggiungono nuovi dettagli alla classe generale. Esso permette di risparmiare tempo perché la classe specializzata eredita tutte le proprietà della classe generale e quindi il programmatore deve solo realizzare le nuove caratteristiche.
La classe derivata o sottoclasse è una classe definita aggiungendo variabili di istanza e metodi a una classe esistente. La classe esistente dalla quale è stata definita la classe derivata è chiamata classe base o superclasse. Una classe derivata può inoltre aggiungere nuove variabili di istanza e nuovi metodi a quelli ereditati dalla sua classe base. Quando si parla di ereditarietà, viene comunemente adottata una terminologia che deriva dalle relazioni familiari. Una classe base è spesso chiamata classe genitore o padre. Una classe derivata è chiamata classe figlia. La gerarchia si può estendere all'infinito, una classe "nonno" è chiamata classe antenato.
In java la relazione di derivazione viene resa con la parola keyword "extends" che deve essere usata nella dichiarazione della classe. La classe derivata dovrà chiamare il costruttore esplicitamente con la sintassi "super".

COLLEZIONI, MAPPE E ITERATORI: le collezioni sono contenute da java nel JCF (JAVA COLLECTIONS FRAMEWORK) e vengono definite come insieme di classi che implementano strutture dati che si espandono e contraggono dinamicamente in base agli elementi in essa contenuti, di soli oggetti e non di valori di tipo primitivo (per i quali bisognerà ricorrere alle corrispondenti classi wrapper). Con le liste è possibile l'accesso

posizionale, oppure si possono utilizzare gli iteratori, oggetti di supporto utilizzati per accedere agli elementi di una collezione, uno alla volta e in sequenza. (hasNext(), next(), remove sono solo alcuni dei suoi metodi).

COLLEZIONI JAVA: le liste sono degli array che quando hanno esaurito la capienza compiono un processo del genere: creano un array più grande, copiano gli elementi dell'array originale nel nuovo array e poi rinominano il nuovo array. Quindi la lista usa questo accorgimento.
Viene definita collezione un insieme di oggetti o istanze senza una dimensione fissa, che non possono contenere variabili di tipo positivo (problema ovviato con l'utilizzo delle classi wrapper). Una collezione Java è una qualunque classe che implementi l'interfaccia Collections.
Il più classico esempio di collezione è la lista. La lista è un insieme di oggetti o istanze senza una dimensione fissa, e in grado di contenere variabili di tipo non. Primitivo (int, double, long,…). Spesso la lista si scorre attraverso l'uso di iteratori. La lista inizialmente è di 10 elementi quando c'è bisogno di spazio raddoppia automaticamente. Un iteratore è qualcosa che consente di esaminare in ordine sequenziale una lista ed ha senso se associato ad una lista.
2. esse stanno alla base del JFC (JAVA COLLECTIONS FRAMEWORK) insieme di dati e classi che si espandono e contraggono dinamicamente.

ITERATORI: un iteratore è un oggetto di supporto usato per accedere agli elementi di una collezione, uno alla volta e in sequenza.

LISTE: una lista è una sequenza di elementi (oggetti) dove è possibile l'accesso posizionale ed è implementabile con arrayList. Una struttura dati è un costrutto, per esempio una classe o un array. Una struttura dati le cui dimensioni aumentano o diminuiscono durante l'esecuzione del programma viene chiamata dinamica.

ARRAY: viene utilizzato per memorizzare collezioni di dati e tutti i dati memorizzati devono essere dello stesso tipo. È un insieme ordinato, di dimensione fissa di oggetti dello stesso tipo. Il primo indice è 0 e l'ultimo è n-1.

OVERLOADING VS OVERRIDING: non si deve confondere l'overriding con l'overloading (di un metodo). Quando si effettua l'overriding della definizione di un metodo, la nuova definizione del metodo nella classe derivata ha lo stesso nome, lo stesso tipo di ritorno e gli stessi parametri in termini di tipo, ordine e numero. Se invece il nome e il ritorno del metodo fossero uguali, ma un numero differente di parametri o anche un solo parametro di tipo differente del metodo nella classe base, saremmo di fronte ad un caso di overloading, in questa situazione la classe derivata avrebbe entrambi i metodi.