

PROGRAMMA

CLASSE 4Bi - A.S. 2015/2016

MATERIA: Informatica

Docente Teorico : *Paolo Minardi*

Docente Tecnico-Pratico : *Franco Moscaritoli*

CONTENUTI

Elementi concettuali relativi all'analisi del problema

La metodologia di progettazione bottom-up

L'integrazione fra bottom-up e top-down

La programmazione orientata agli oggetti

Le classi e gli oggetti

La progettazione delle interfacce di classe: il concetto di stato, i campi, i metodi di alterazione (setter) e i metodi di interrogazione (getter)

I costruttori e l'istanziamento degli oggetti

L'incapsulamento e l'information hiding

I metodi virtuali

La programmazione orientata agli oggetti in Java[‡]

La dichiarazione di classi e la loro istanziazione in oggetti

L'allocazione della memoria ed il garbage collector

La dichiarazione e l'utilizzo di attributi

La dichiarazione e l'utilizzo di metodi

La dichiarazione e l'utilizzo di costruttori

La parola chiave *private*

L'aliasing

L'operatore *instanceof*

Boxing ed unboxing (con particolare riferimento all'utilizzo di *ArrayList*)

L'overriding di *toString()*, *equals()* ed *hashCode()* Implementazione OOP del grafo

La classe *ArrayList* (con genericità) ed esempi di utilizzo dei suoi metodi notevoli

Le eccezioni:

- i tipi di eccezione: checked, runtime ed *Error*
- *try - catch - finally*
- attivazione selettiva dell'exception handler: il concetto di compatibilità (operatore *instanceof*)
- dichiarazione di eccezioni (*throws*)

Serializzazione

Il concetto di stream e di rete di oggetti

Le classi *ObjectOutputStream* e *ObjectInputStream*

Serializzazione su file: le classi *FileOutputStream* e *FileInputStream*

I metodi *writeObject()* e *readObject()*

La terminazione di uno stream: la classe *SignalEOF*

Campi *transient*

Il Multithreading in Java

Il concetto di thread

La definizione di un thread: la classe *Thread* e l'interfaccia *Runnable*

Il metodo *run()*

[‡] L'ambiente di sviluppo utilizzato è stato *Java™ 2 JDK v.1.8.XX* con IDE *Eclipse (Mars 4.5.XX)*.

PROGRAMMA

L'istanziamento di un thread
L'avvio di un thread: il metodo *start()* e lo stack delle chiamate
Gli stati di un thread e le transizioni fra gli stati
Le priorità dei thread
I metodi *sleep()* ed *yield()*
Il metodo *join()*
La sincronizzazione:

- la concorrenza su risorse condivise ed il concetto di semaforo
- i lock
- i metodi ed i blocchi sincronizzati
- lo stallo
- i metodi *wait()*, *notify()* e *notifyAll()*

La grafica

La libreria Swing:

- Gli spazi di coordinate: User Space e Device Space
- Il metodo *paint()* e *Graphics2D*
- Le immagini:
 - la classe *BufferedImage*
 - il caricamento ed il tracciamento
 - le trasformazioni affini di rotazione, traslazione, riscalatura e deformazione, e loro concatenazione

Tipi di Dato Astratti (ADT)

Albero radicato e sua rappresentazione grafica, insiemistica, tabulare e ad espressione
Albero binario
Grafo[†]:

- definizione formale
- definizione di incidenza, di adiacenza di nodi e archi e di grado di un vertice
- lessico specifico: vertice isolato, vertice pendente, archi in serie e grafo nullo
- definizione di isomorfismo fra grafi
- definizione di sottografo

e sua implementazione OOP

La ricorsione

Il principio di induzione matematica
Esempi di dimostrazioni per induzione
Applicazione dell'induzione alla programmazione ricorsiva
Cenni di implementazione ricorsiva relativi al grafo e all'albero

L' Insegnante

Gli Studenti

[†] Lezione CLIL