

## PROGRAMMA

**CLASSE 4Bi - A.S. 2014/2015**

**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: gli oggetti come macchine, 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

L'ereditarietà

Composizione vs Ereditarietà

I metodi virtuali

Il polimorfismo

#### La programmazione orientata agli oggetti in Java<sup>‡</sup>

La dichiarazione di classi e la loro istanziamento 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

L'overriding e l'overloading

La parola chiave *private*

L'aliasing

L'operatore *instanceof*

L'upcasting ed il downcasting

Boxing ed unboxing

Classi astratte ed interfacce (cenni)

L'overriding di *toString()*, *equals()* ed *hashCode()*

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()*

<sup>‡</sup> L'ambiente di sviluppo utilizzato è stato Java™ 2 JDK v.1.7 con IDE Eclipse 4.2 (Juno).

## PROGRAMMA

La terminazione di uno stream: la classe *SignalEOF*

Campi *transient*

### Programmazione di Rete

Creazione lato server di un *ServerSocket* per le richieste di connessione

Creazione lato server e lato client di un *Socket* per la connessione

- Realizzazione della comunicazione tramite creazione dal *Socket* degli stream *ObjectInputStream* ed *ObjectOutputStream* e loro utilizzo

### Il Multithreading in Java

Il concetto di thread

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

Il metodo *run()*

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()*
- il produttore-consumatore: implementazione fortemente accoppiata e debolmente accoppiata con soluzione del problema dei risvegli automatici (simulato con classe *Disturbo*)

### La grafica

La libreria Swing:

- Gli spazi di coordinate: User Space e Device Space
- Il metodo *paint()*
- Gli attributi di stato: le interfacce *Stroke* e *Paint* e le classi che le implementano
- Le immagini:
  - la classe *BufferedImage*
  - il caricamento ed il tracciamento
  - le trasformazioni affini di rotazione, traslazione, riscalatura e deformazione, e loro concatenazione
- La libreria Slick2D: introduzione

### La ricorsione

Il principio di induzione matematica

Dimostrazioni per induzione

Applicazione dell'induzione alla programmazione ricorsiva

Ricorsione e Tipi di Dato Astratti (ADT):

- alberi radicati e loro rappresentazione grafica, insiemistica, tabulare e ad espressione
- alberi binari e alberi binari di ricerca (ABR)
- grafi
- algoritmi ricorsivi: visite degli alberi, percorso pesato minimo di un grafo, colorazione di un grafo

L' Insegnante

Gli Studenti

# PROGRAMMA