

PROGRAMMA

CLASSE 4Bi - A.S. 2020/2021

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
L'ereditarietà
Composizione vs Ereditarietà
Overloading ed overriding
Il polimorfismo

La programmazione orientata agli oggetti in Java[‡]

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
Il modificatore d'accesso *private*
L'aliasing
L'operatore *instanceof*
Boxing ed unboxing (con particolare riferimento all'utilizzo di *ArrayList*)
L'overriding di *toString()* ed *equals()*
La classe *ArrayList* (con genericità) ed esempi di utilizzo dei suoi metodi notevoli
Classi astratte ed interfacce
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*)

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()*

[‡] L'ambiente di sviluppo utilizzato è stato *Java™ 2 JDK v.15.X.X* con Eclipse 2021-03 (4.19.XX).

PROGRAMMA

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 con soluzione del problema dei risvegli automatici (simulato con classe *Disturbo*)

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*

L' Insegnante

Gli Studenti