**ESAME**

APPUNTI E COSE UTILI

IMPORT E EXPORT

* Import
  + Import->Existing projects from workspace->copia e incolla nome della cartella(più veloce)
  + NB importa cartella, non file zip
* Export
  + Export->Archive file->Scegli la cartella e il nome->Esporta

LIBRERIA

* Build path->Projects->Add->UniBSLib
* NB non lo zip, il profe fa così quindi fallo anche tu
* NB **import** it.unibs.fp.mylib.\*; //Lo inserisce da solo il programma in realtà

JAVADOC

* /\*\* \* \* @param @returns \*\*/ (Vedi esempio tamaSbagliato)

ESERCIZI DEL PROFE:

-Punto-retta

- Cuori solitari

-Tamabase

-Polveri Sottili

-Archivio CD

-Tamazoo

-Carta più alta

-Lavatrici

-Titoli Azionari

-Conto Corrente

Simulazioni:  
-Morra cinese

-Massaia

-Azienda sanitaria

APPUNTI SULL’ESAME

-commenta classi e metodi

-crea metodi static di utilità fuori dal main(?)

-Main snello ma non troppo

-NO stringhe che girano, fai costanti che abbiano senso

-Override toString, segna sopra @override

-Costruttore con tutti gli attributi, poi fai un metodo in cui inserire i dati da passare al costruttore

Appunti da punto-retta

-quanti decimali-> %.n nel printf

-Override del metodo String toString() nella classe che deve usare il metodo

-String format funziona come un printf ma restituisce una stringa

-Può essere apprezzato il for abbreviato

Appunti cuori solitari

-costruttore che passa tutti gli attributi

-metodo static che prende in input tutti gli attributi in variabili e li posiziona negli argomenti del costruttore

-variabile da inizializzare successivamente, metti null all’inizio (Fai un controllo Exception NullPointer?)

-Vedi la classe enum

-interazione tra due oggetti puoi farla nella classe stessa dell’oggetto (Come avevo fatto con la battaglia golem, forse meglio fare una nuova classe, però valuta…)

TAMABASE

-Costruttore con tutti gli attributi, poi fai un metodo in cui inserire i dati da passare al costruttore

-Esempio javadoc

METODI UTILI NELLA LIBRERIA UNIBS

CLASSE INPUTDATI

- *leggiUpperChar(Messaggio, elenco caratteri consentiti) //Verifica maiuscole e minuscole, non fa ignoreCase*

*-leggiIntero(Messaggio, min, max)*

*-leggiStringaNonVuota(Messaggio) //Abbastanza intuitivo che controllo faccia, verifica se funziona con spazio e tab*

ESTRAZIONI CASUALI

*-Abbastanza intuitivo… (solo con int)*

MYMENU

*-NB è una classe istanziabile, utilizza il costruttore mettendo il titolo e un vettore con tutte le alternative.*

*-Per evitare errore out of bounds magari inizializza il vettore direttamente es:*

String [] voci = {"scelta1", "scelta2"};

MyMenu menu=**new** MyMenu("PROVA", voci);

menu.stampaMenu();

*-è ordinato e fa risparmiare spazio nel codice, vedi su momento se utilizzarlo o no…*