

- 1) Riorganizzate i programmi dell'esercitazione precedente utilizzando i packages. Se non avete immediatamente a disposizione il codice potete scaricarlo dal sito del corso *prgpa.blogspot.it*. Ciascun esercizio sarà definito in un package avente nome derivato dal vostro account community. Ad esempio l'esercizio 1 apparterrà al package **it.unipa.community.nomecognome.prg.n03.es01**. Ciascun esercizio avrà il main in una classe chiamata **Main**. Create una cartella principale per la vostra esercitazione all'interno della quale creerete le cartelle *classes* e *src*. All'interno di *src* create la struttura delle cartelle (che riflette quella dei package) che conterranno i vostri sorgenti. Per compilare, ad esempio l'esercizio 6, dopo aver copiato i sorgenti nella cartella corrispondente, vi posizionerete nella cartella *classes* e digiterete:


```
javac -d . ../src/it/unipa/community/nomecognome/prg/n03/es06/*.java
```

 Per eseguire il programma, sempre dalla cartella *classes*, digiterete:


```
java it.unipa.community.nomecognome.prg.n03.es06.Main
```
- 2) Analogamente a quanto fatto nell'esercitazione precedente con la classe **Razionale** scrivete una classe **Complex** che permetta di fare le operazioni di somma, sottrazione e stampa di numeri complessi in forma binomiale, $a + i b$. Implementate le operazioni della classe **Complex** e la funzione di visualizzazione anche come metodi *static* della classe che ricevono come parametri i due operandi e ritornano il risultato come valore di ritorno del metodo. Verificate, con un programma di prova, che effettuando le operazioni con metodi o con metodi static si producono gli stessi risultati. (`ris=a.somma(b); ris.stampa();` deve essere equivalente a `ris=Complex.somma(a, b); Complex.stampa(ris);`). Nel main utilizzate anche oggetti della classe **Razionale** dell'esercitazione precedente dopo averla opportunamente importata.
- 3) Definire una classe **Stack** che implementi una pila di 100 caratteri tramite un array. Le funzioni membro della classe devono essere:


```
void push(char), char pop(), boolean isEmpty(), boolean isFull().
```

 Scrivete un programma che crea un oggetto **Stack** e verifica il corretto funzionamento della classe.
- 4) Create la classe **Date** secondo queste specifiche:
 Effettua l'output della data secondo più formati, come
 - a. GGG AAAA
 - b. GG/MM/AA
 - c. 14 Giugno 1992
 Usa costruttori in overloading per creare oggetti **Date** inizializzati con dati secondo i formati del punto precedente. Considerando questa classe come una classe di uso generico, immaginate quali altri metodi potrebbero essere utili e implementateli. Si ignorino i problemi relativi agli anni bisestili. Scrivete un programma per testare la creazione di oggetti **Date** con tutti i costruttori e delle funzioni di stampa (che potreste ad esempio chiamare *stampa1()*, *stampa2()* e *stampa3()*).
- 5) Definite una classe **Studente** avente oggetti *String* per nome e cognome, *Date* (vedi esercizio precedente) per data di nascita e data di immatricolazione e *int* per la matricola. Definite funzioni set e get e tutti i metodi che ritenete utili prevedendo nei metodi set i controlli di validità che ritenete sensati. Scrivete un costruttore parametrizzato che riceve tutti i parametri necessari per creare l'oggetto. Scrivete infine un programma che crea un paio di oggetti **Studente** e ne stampa i dettagli.

NOTE PER COMPILAZIONE E TEST A RIGA DI COMANDO IN AMBIENTE LINUX:

Creare una cartella col proprio cognome sulla scrivania e i relativi file sorgenti al suo interno.

Aprire una finestra di **terminale** e digitare:

cd Desktop/cognome oppure **cd Scrivania/cognome** (si posiziona nella directory)

Creare i file sorgente con **gedit** e salvarli nella propria directory.

Digitare:

javac nomeClasse.java (compila e genera il bytecode)

java nomeClasse (esegue il bytecode sulla JVM)