

- 1) Analogamente a quanto fatto nell'esercitazione precedente con la classe **Razionale** scrivete una classe **Complex** che permetta di fare le operazioni di somma, sottrazione, distanza, confronto e conversione in String in forma binomiale $a + i b$.

Potreste ad esempio implementare i seguenti costruttori e metodi:

```
Complex(double re, double im)
Complex(double re)
Complex()
Complex somma(Complex altro)
Complex sottrai(Complex altro)
double distanza(Complex altro)
String toString()
boolean equals(Complex altro)
```

Scrivete un programma di prova in cui create alcuni oggetti con valori letti da tastiera utilizzando uno *Scanner* e verificate il corretto funzionamento dei metodi e costruttori implementati.

- 2) Definire una classe **Stack** che implementi una pila di 100 String tramite un *ArrayList<String>*. Le funzioni membro della classe devono essere:

```
void push(String s)
String pop()
boolean isEmpty()
boolean isFull()
```

Scrivete un programma che crea un oggetto **Stack** e, tramite un menu testuale, verifica il corretto funzionamento della classe.

Implementate infine i metodi *toString* e *equals* e verificatene il corretto funzionamento.

- 3) Create la classe **IntegerSet** (insieme di interi). Ogni oggetto di **IntegerSet** può memorizzare interi fra 0 e 99. Un insieme è rappresentato internamente da un array di booleani. L'elemento dell'array **a[i]** è **true** se l'intero **i** è contenuto nell'insieme. L'elemento dell'array **a[j]** è **false** se l'intero **j** non è contenuto nell'insieme. Il costruttore di default inizializza un insieme all'insieme vuoto, il cui array contiene soltanto **false**.

Scrivete i metodi che effettuano le comuni operazioni sugli insiemi. Per esempio scrivete la funzione **unionOfIntegerSet** che crea un terzo insieme che è l'unione di due insiemi esistenti: un elemento del terzo insieme è **true** se è presente in almeno uno dei due insiemi origine. Scrivete il metodo **intersectionOfIntegerSet** che crea un terzo insieme che è l'intersezione di due insiemi esistenti: un elemento del terzo insieme è **true** se è presente in entrambi gli insiemi origine. Scrivete la funzione membro **insertElement** che inserisce un nuovo intero **k** nell'insieme (impostando **a[k]** a **true**). Scrivete la funzione **deleteElement** che elimina dall'insieme l'intero **m** (impostando **a[m]** a **false**). Scrivete la funzione membro **toString()** che restituisce una stringa che rappresenta un insieme come lista di numeri separati da spazi. Visualizzate – per l'insieme vuoto. Scrivete un programma di prova che faccia uso della classe **IntegerSet**. Istanziare diversi oggetti **IntegerSet**. Verificate la correttezza dei metodi.

- 4) Si scriva un programma interattivo che, utilizzando la classe *Studente* dell'esercitazione 4 e due *ArrayList<Studente>*, dia la possibilità, tramite un semplice menu testuale, di archiviare gli studenti (*ArrayList* che chiameremo *archivio* che contiene tutti gli studenti) e gestire le iscrizioni agli esami (*ArrayList* che chiameremo *esame* che contiene solo gli studenti iscritti all'esame). L'utente dovrà poter scegliere fra le seguenti opzioni:

- inserire un nuovo studente nella lista *archivio* con dati inseriti dall'utente
- cercare uno studente nella lista *archivio* in base alla matricola
- cercare uno studente nella lista *archivio* in base al cognome (se ce ne è più di uno bisogna stamparli tutti)
- iscrivere uno studente all'esame (inserirlo quindi nella lista *esame*) in base alla matricola
- rimuovere uno studente dall'esame in base alla matricola
- stampare l'intera lista degli iscritti agli esami
- cancellare l'intera lista degli iscritti all'esame
- esci

Nota:

Scrivere nel *main* un ciclo infinito che chiede all'utente cosa vuole fare e tramite uno *switch* esegue l'operazione richiesta.

Fate in modo che la matricola venga assegnata univocamente e automaticamente utilizzando una variabile statica nella classe *Studente* che tiene conto di quanti oggetti sono stati creati e usa questo numero progressivo come matricola.

- 5) Riorganizzate i programmi degli esercizi precedenti utilizzando i packages. Ciascun esercizio sarà definito in un package avente nome derivato dal vostro cognome. Ad esempio l'esercizio 1 apparterrà al package **cognome.prg.n05.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 2, dopo aver copiato i sorgenti nella cartella corrispondente, vi posizionerete nella cartella *classes* e digiterete:

```
javac -d . ../src/cognome/prg/n05/es02/*.java
```

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

```
java cognome.prg.n05.es02.Main
```

NOTE PER COMPILAZIONE E TEST A RIGA DI COMANDO:

Digitare per ciascuna classe:

```
javac nomeClasse.java
```

(compila e genera il bytecode)

Digitare per la classe che contiene il main:

```
java nomeClassePrincipale
```

(esegue il bytecode sulla JVM)