

- 1) Rivedere l'esercizio 3 dell'esercitazione 7 (Abbonato, AbbonatoPremium) e lo si implementi utilizzando un array eterogeneo di Abbonato.
- 2) Analogamente a quanto fatto nell'esercitazione 5 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 $a + i b$.
Dovreste implementare i seguenti costruttori e metodi:
Complex(double re, double im)
Complex(double re)
Complex()
Complex somma(Complex altro)
Complex sottrai(Complex altro)
String toString()
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.
- 3) Si implementi una gerarchia di classi per rappresentare Razionale e Complex come classi derivate di una classe astratta Numero. Si definisca un'interfaccia Aritmetica con le operazioni aritmetiche somma e sottrai tra coppie di elementi Numero che saranno opportunamente implementate nelle classi Razionale e Complex. Nel caso in cui si provi a fare operazioni fra elementi di tipo diverso visualizzate un messaggio di errore.
Verificate il corretto funzionamento con un programma che crea alcuni oggetti delle due classi e fa alcune operazioni aritmetiche.
- 4) Si utilizzi la documentazione di Java 8 per studiare le caratteristiche dell'interfaccia `java.lang.CharSequence`. Quali sono i metodi che una classe che implementa `CharSequence` deve necessariamente implementare?
Si scriva quindi una classe **ReversedString** che implementa `CharSequence` e la si utilizzi per operare su una stringa scorrendo i caratteri da destra verso sinistra (piuttosto che da sinistra a destra come si fa di solito).

Per esempio, si vuole che l'output del seguente codice in una classe Main:

```
String s = "Questo è solo un esempio";  
ReversedString rs = new ReversedString(s);  
System.out.println(rs.subSequence(0, 7));
```

sia: "oipmese".

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

```
javac -d . es09src/nomeClasse.java    compila e genera il bytecode  
java nomePackage.nomeClasse          esegue il bytecode sulla JVM
```