

Inserire nel package **it.unipa.community.nomecognome.prg.n07.es0X** le seguenti applicazioni.

- 1) Modificare la classe **Date** (esercitazione n04) in modo da rendere confrontabili date diverse (la data 11/11/2016 deve essere maggiore della data 11/11/2015) implementando l'interfaccia **Comparable** (documentazione: <https://docs.oracle.com/javase/8/docs/api/java/lang/Comparable.html>). Tale interfaccia consente il confronto tra oggetti della stessa gerarchia attraverso il metodo **compareTo()** che dovete implementare voi.
- 2) Analogamente a quanto fatto nell'esercizio precedente Modificare la classe **Razionale** (esercitazione n03) in modo da rendere confrontabili numeri razionali diversi.
- 3) Si consideri la gerarchia di classi dell'esercizio 5 dell'esercitazione n05 che rappresenta **Shape, Circle, Rectangle, Square** e aggiungete l'implementazione, in ciascuna classe, dell'interfaccia **Comparable**. Il confronto tra forme sia fatto attraverso il valore dell'area. Verificate il corretto funzionamento con un programma che crea alcuni oggetti e li confronta.
- 4) Si considerino le classi **Razionale** (esercizio precedente) e **Complesso** (esercitazione n04). Si implementi un'opportuna interfaccia **Aritmetica** che implementi le operazioni aritmetiche tra coppie di numeri razionali, e coppie di numeri complessi. Verificate il corretto funzionamento con un programma che crea alcuni oggetti delle due classi e fa alcune operazioni aritmetiche.
- 5) Si implementi adesso una gerarchia di classi (senza usare l'interfaccia Aritmetica) per rappresentare **Razionale e Complesso** come classi derivate di una classe astratta **Numero**. Verificate il corretto funzionamento con lo stesso **main()** dell'esercizio 4 che crea alcuni oggetti delle due classi e fa alcune operazioni aritmetiche e verificate che i risultati siano gli stessi.
- 6) Riprendete la gerarchia di classi dell'esercizio 1 e implementate in ogni classe la propria versione della funzione membro **draw()** per il tracciamento della figura (per semplicità supponete che la funzione **draw()** invece di tracciare la figura ne debba soltanto stampare il tipo e i relativi parametri con **toString()**). Utilizzate le vostre classi per creare un array eterogeneo di **Shape** in cui memorizzare alcuni oggetti delle varie classi. Dopo aver inserito gli oggetti nell'array attraversate l'intero array chiamando il metodo **draw()** di ciascun oggetto, per stampare i dettagli di tutti gli oggetti. Verificate che venga chiamato per ciascun oggetto la versione corretta del metodo **draw()**.
- 7) 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.

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

Creare una cartella col proprio cognome sulla scrivania con all'interno le directory **classes** e **src**.  
Aprire una finestra di **terminale** e digitare:

```
cd Desktop/cognome/classes oppure cd Scrivania/cognome/classes
```

Creare i file sorgente con **gedit** e salvarli nelle sottodirectory di **src** corrispondenti al package.

Digitare:

```
javac -d . ../src/it/unipa/.../prg/n07/es0X/*.java (compila e genera il bytecode)
```

```
java it.unipa.community....prg.n07.es0X.Main (esegue il bytecode sulla JVM)
```