

- 1) Scrivere un programma con i seguenti requisiti.
- Utilizza una classe *Persona* che dichiara le variabili *nome*, *cognome*, *eta* (età). Si dichiara inoltre un metodo *toString()* che restituisca in una stringa le informazioni sulla persona in questione. Ricordarsi di utilizzare le convenzioni e le regole per i nomi di classi, metodi e variabili descritte a lezione.
 - Utilizza una classe *Principale* che, nel metodo *main()*, istanzia due oggetti chiamati *persona1* e *persona2* della classe *Persona*, inizializzando per ognuno di essi i relativi campi *nome*, *cognome*, *eta* adoperando l'operatore *dot*.
 - Dichiarare un terzo reference (*persona3*) che punti *persona1*. Controllare che effettivamente *persona3* punti all'oggetto voluto, stampando i campi di *persona3* sempre mediante l'operatore *dot* e che modificando *persona3* si modifica *persona1*.

- 2) I numeri razionali sono quei numeri che si ottengono dividendo due numeri interi e hanno particolari proprietà. Un numero razionale viene solitamente rappresentato sotto forma di una frazione. Per elaborare numeri razionali definite una classe *Razionale*, contenente un *int* per il numeratore e uno per il denominatore. Definite costruttori, funzioni membro per il calcolo delle 4 operazioni aritmetiche fra numeri razionali, il metodo *toString* e tutti i metodi che ritenete utili. Verificate il corretto funzionamento almeno delle seguenti funzioni membro:

```
String toString();  
Razionale somma(Razionale);  
Razionale sottrai(Razionale);  
Razionale moltiplica(Razionale);  
Razionale dividi(Razionale);
```

Le funzioni per le operazioni aritmetiche operano sull'oggetto per cui si chiama la funzione e sul parametro e restituiscono il risultato (ad esempio `c = a.somma(b)`; somma *a* e *b* e restituisce il risultato in un NUOVO oggetto senza modificare né *a* né *b*).

Scrivete un programma che tramite un oggetto della classe *Scanner* e il relativo metodo *nextInt* legge da tastiera un paio di oggetti *Razionale* e ne calcola alcune operazioni usando i metodi di cui sopra.

- 3) Un punto nel piano si può rappresentare tramite una classe con due variabili d'istanza: *x* e *y* di tipo *double*. Scrivete una classe (incapsulata) con tutti i metodi e i costruttori che ritenete utili (incluso il metodo *String toString()* e boolean *equals(Punto)*). Scrivere quindi un programma che realizzi le seguenti operazioni con punti nel piano:

- Crea due punti con dati inseriti da tastiera dall'utente (usate il metodo *nextDouble* di *Scanner*)
- Dati i due punti, ne calcola la loro distanza
- Dati i due punti, determina il punto medio del segmento che li unisce
- Dati i due punti verifica se sono uguali (usando il metodo *equals* che avete implementato)

Note:

Nell'implementare il metodo *equals* non potete verificare l'uguaglianza delle coordinate con `==` perché sono valori *double* e quindi approssimati. Potete dire che sono uguali quando la loro distanza è minore di un valore molto piccolo prefissato, ad esempio di `1E-10`

Quando inserite un *double* da tastiera potreste dover utilizzare la virgola invece del punto per il separatore dei decimali in quanto dipende dai settaggi del computer su cui gira la JVM.

- 4) Scrivere un programma con i seguenti requisiti.
- Utilizza una classe *Persona* che incapsula le variabili *nome*, *cognome*, *eta* (età) e implementa tutti i costruttori e i metodi che si ritengono necessari. Si dichiara inoltre un metodo *toString* che restituisca in una stringa le informazioni sulla persona in questione e il metodo *equals* che verifica se due oggetti rappresentano la stessa persona.
 - Utilizza una classe *Principale* che, nel metodo *main()*, crea una lista (vuota) di oggetti *Persona* con l'istruzione `ArrayList<Persona> miaLista = new ArrayList<>()`;
 - Crea alcuni oggetti *Persona* e li aggiunge alla lista col comando *add*
 - Verifica la dimensione della lista col metodo *size*
 - Prende l'i-esimo elemento dalla lista col metodo *get* e lo stampa
 - Elimina l'i-esimo elemento dalla lista
 - Stampa il contenuto dell'intera lista usando un ciclo *for* e il metodo *size* per conoscere la dimensione della lista oppure usando un ciclo *for* migliorato (che succede se passate l'*ArrayList* alla *println*?)
 - Cancella l'intera lista col metodo *clear* e verifica che si sia effettivamente svuotata col metodo *size*

Nota:

Per usare un'ArrayList dovete inserire il comando `import java.util.ArrayList;` nella classe che lo utilizza. Verificare nella pagina della documentazione ufficiale di Java come si usano i metodi indicati sopra ({ [HYPERLINK "https://docs.oracle.com/en/java/javase/11/docs/api/index.html"](https://docs.oracle.com/en/java/javase/11/docs/api/index.html) }).

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 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)