

- 1) Un punto nel piano si può rappresentare tramite una classe con due variabili d'istanza: *x* e *y* di tipo *double*. Scrivete una classe con tutti i metodi e i costruttori che ritenete utili (incluse *set* e *get*, 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)

Nota: avete già fatto lo stesso esercizio nell'esercitazione 4 ma senza usare l'incapsulamento.
- 2) Scrivete una classe *Persona* come quella dell'esercitazione 3 ma che utilizzi l'incapsulamento e definisca tutti i costruttori e metodi che ritenete utili. Verificate il corretto funzionamento della classe con un programma che crea un paio di oggetti e verifica il corretto funzionamento dei metodi implementati.
- 3) 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 nè *a* nè *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.
- 4) Scrivete una 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 (quindi *Date(int g, int a)*, *Date(int g, int m, int a)* e *Date(int g, String m, int a)*).

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()*).

Prevedete infine il metodo *String toString()* che ritorna un oggetto come stringa nel formato **c** e il metodo *boolean equals(Date d)* che permette di confrontare due oggetti.
- 5) Definite una classe **Studente** avente *String* per nome e cognome, *Date* (vedi esercizio precedente) per data di nascita e *long* 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:

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