

Esercitazioni di Laboratorio

10. Laboratorio 10

Nota: sono pochi esercizi ma, a parte l'esercizio 1, richiedono tutti impegno e tempo per la progettazione e l'implementazione. Ricordo anche, per chi non li avesse già svolti, che anche nel Lab 9 sono disponibili un paio di esercizi sull'ereditarietà.

Esercizio 1

Argomento: ripasso

Rispondere al [Quiz di Autovalutazione 5](#) disponibile nel sito moodle nella corrispondente sezione.

Esercizio 2

Argomento: interfaccia Comparable, algoritmi di ordinamento

Modificare la classe `BankAccount` in modo da farle implementare l'interfaccia `Comparable` (dichiarazione nell'intestazione + implementazione di `compareTo`). In una classe `ArrayAlgorithms` implementare le versioni "Comparable" degli algoritmi di ordinamento visti a lezione (`selection sort`, `insertion sort`, `mergesort`) oltre che i metodi di ricerca visti a lezione (`linearSearch`, `binarySearch`). Creare una classe eseguibile `TestCompare.java` in cui si crea un array di 10 elementi di tipo `BankAccount` e lo si inizializza con un saldo a caso tra 1 e 1000 euro. Visualizzare il saldo dei conti in banca nell'ordine in cui sono stati creati. Ordinare poi l'array con ciascuno dei metodi di ordinamento (ricordarsi di non passare l'array ordinato al secondo metodo di ordinamento invocato, ma di farne prima una copia e passare quella).

Esercizio 3

Argomento: ereditarietà, sovrascrivere metodi, metodi polimorfici

Realizzare una classe **`Square`** che estenda la classe **`Rectangle`** della libreria standard. La classe deve realizzare i seguenti comportamenti:

- Il costruttore crea un quadrato ricevendo come parametri espliciti le coordinate (**`x,y`**) del centro del quadrato, e la dimensione del quadrato (ovvero la lunghezza del lato). Per realizzare il costruttore sarà necessario invocare costruttori e/o metodi della classe **`Rectangle`** (si consiglia in particolare di studiare la documentazione dei metodi **`setLocation`** e **`setSize`** della classe **`Rectangle`**). Osservare anche che, dal momento che che i campi **`x`**, **`y`**, **`width`**, **`height`** di **`Rectangle`** sono tutti di tipo **`int`**, il posizionamento del centro del quadrato avrà una approssimazione di ± 1 .
- La classe possiede un nuovo metodo **`getArea()`**, che calcola e restituisce l'area del quadrato.
- La classe sovrascrive il metodo **`setSize(int width, int height)`** della classe **`Rectangle`** (studiarne la documentazione): se **`width=height`**, il metodo esegue correttamente ridimensionando il quadrato alla nuova dimensione **`width`**, altrimenti lancia un'eccezione di tipo **`IllegalArgumentException`**.
- La classe possiede un nuovo metodo **`setSize(int dim)`**, che esegue il ridimensionamento del quadrato sulla base dell'unico parametro esplicito **`dim`**.

Collaudare la classe scrivendo un programma di test che

- Riceve da standard input due triple di numeri interi (una tripla per riga), rappresentanti le coordinate (**`x,y`**) del centro e la dimensione di ciascuno dei due quadrati.
- Crea due oggetti di tipo **`Square`** usando tali valori, e stampa gli oggetti a standard output in ordine di area (il primo oggetto stampato è quello di area più piccola).
- Riceve da standard input due coppie di numeri interi (una coppia per riga), rappresentanti i nuovi valori (**`width,height`**) di larghezza e altezza per ciascuno dei due quadrati.
-
- ridimensiona i due quadrati usando il metodo

`setSize(int width, int height)` sovrascritto nella classe **`Square`**

- Se **`setSize`** termina correttamente la propria esecuzione (ovvero se **`width=height`**), stampa nuovamente i due oggetti a standard output in ordine di area; altrimenti segnala l'errore e termina l'esecuzione.

Suggerimenti: non dimenticate che la classe **`Rectangle`** ha già sovrascritto i metodi **`toString`** e **`equals`** di **`Object`**. Inoltre, poniamoci questa domanda: la nostra classe **`Square`** ha oppure no necessità di avere dei nuovi campi di esemplare rispetto a **`Rectangle`**? Quindi è necessario oppure no sovrascrivere **`equals`**?

Esercizio 4

Per concludere, prima delle vacanze vi auguriamo: **buone feste!**

