

Esercizi per il Corso di Metodologie di Programmazione

Capitolo 15

Università degli Studi di Roma "La Sapienza"

Esercizio 1

Modificare la classe generica *Pair* in modo che i due valori siano dello stesso tipo.

Consiglio: all'interno della cartella 1 troverete la classe *Pair* da modificare.

Esercizio 2

Aggiungere alla classe *Pair* dell'esercizio precedente il metodo *swap* che scambi il primo elemento della coppia con il secondo.

Consiglio: all'interno della cartella 2 troverete la classe *PairTester* con cui testare il corretto funzionamento dell'esercizio.

Esercizio 3

Realizzare un metodo statico generico *PairUtil.swap()*, il cui parametro sia un oggetto di tipo *Pair*. Il metodo dovrà restituire una nuova coppia avente il primo ed il secondo elemento scambiati rispetto alla coppia originaria.

Consiglio: all'interno della cartella 3 troverete la classe *Pair* da utilizzare per l'esercizio.

Esercizio 4

Realizzare una versione generica dell'algoritmo di ricerca binaria. Il tipo generico dovrà ereditare dalla classe *Comparable*.

Consiglio: all'interno della cartella 4 troverete la classe *BinarySearchTester* da utilizzare per testare l'esercizio. Maggiori informazioni sull'algoritmo di ricerca binaria sono disponibili in **questa** pagina.

Esercizio 5

Realizzare un metodo statico generico *ArrayListUtil.append()*, i cui parametri sono due oggetti di tipo *ArrayList* il cui tipo generico è condiviso. Il metodo dovrà restituire un nuovo oggetto di tipo *ArrayList* ottenuto tramite l'unione dei due parametri.

Consiglio: all'interno della cartella 5 troverete la classe *AppendTester* da utilizzare per testare l'esercizio.

Esercizio 6

Realizzare un metodo statico generico *ArrayListUtil.reverse()* il quale, dato un *ArrayList* di un tipo generico, restituisce una sua versione i cui elementi sono invertiti.

Consiglio: all'interno della cartella 6 troverete la classe *ReverseTester* da utilizzare per testare l'esercizio.

Esercizio 7

Realizzare un metodo statico booleano *ArrayListUtil.isPalindrome()* il quale, dato un *ArrayList* di un tipo generico, restituisce *true* se la sua sequenza è palindroma, *false* altrimenti.

Consiglio: all'interno della cartella 7 troverete la classe *PalindromeTester* da utilizzare per testare l'esercizio.

Esercizio 8

Dotare di un appropriato metodo *hashCode* la classe *Pair* vista nell'Esercizio 2 e realizzare una classe *HashMap* che usi un esemplare di *HashSet<Pair<K, V> >*.

Consiglio: all'interno della cartella 8 troverete la classe *Pair* da utilizzare per lo svolgimento dell'esercizio.