
Esercizi: array e ArrayList

Esercizio

- Implementare e testare la nuova classe `Purse` come collezione di monete:
 - una moneta è un'istanza di una classe `Coin`:
 - gli oggetti di `Coin` sono immutabili e hanno un nome (nome della moneta) e un valore (valore della moneta)
 - deve essere possibile recuperare l'informazione contenuta in ogni oggetto `Coin`
 - oltre ai metodi visti a lezione (conteggio, min-max, `getTotal`) implementare anche:
 - `remove(Coin aCoin)`: rimuove una moneta di valore pari a `aCoin` se presente nella borsa
 - `find(Coin aCoin)`: restituisce l'indice di una moneta di valore pari a `aCoin` (private)
 - Modificare l'implementazione di `Purse` in modo che `getTotal()` sia implementato senza ricorsione e iterazione
-

Esercizio

- Implementare e testare una classe Bank che contenga un ArrayList di oggetti di tipo BankAccount e abbia i metodi:
 - ❑ addAccount(initialBalance, customerName)
 - ❑ deposit(accountNumber, amount)
 - ❑ withdraw(accountNumber, amount)
 - ❑ getBalance(accountNumber)
 - ❑ transfer(fromAccNumber, toAccNumber, amount)
-

Esercizio

- Scrivere una classe test per la classe Bank che esegua nell'ordine
 - aggiungere 10 conti correnti utilizzando informazioni inserite direttamente nell'infrastruttura (i saldi e i nomi sono inseriti in due array)
 - collaudare i metodi deposit, withdraw e transfer scegliendo casualmente gli importi
 - verificare la correttezza di transfer testando che la somma dei saldi dei due conti interessati non varia dopo un trasferimento di fondi

Esercizio

- Implementare il concetto di array estendibile visto a lezione
 - aggiungere un metodo per la restituzione dell'elemento alla posizione i
 - aggiungere un metodo per la cancellazione dell'elemento alla posizione i
 - implementare un metodo di servizio (private) `arrayCopy(int daStart, int aStart, int num)`
-

Esercizio

- Realizzare la classe MioArrayList che implementa il concetto di ArrayList con gli array estendibili
 - L'interfaccia pubblica di MioArrayList deve contenere i metodi add(obj), add(obj,i), set(i,obj), remove(i), get(i)
-

Esercizio

- Implementare le classi Bank e Purse con MioArrayList



Esercizio

- Scrivere un metodo `getWinner` per la classe `Tris` che restituisce "X", "0" o "Nessun vincitore" a seconda se vince il giocatore X, il giocatore 0 o non c'è nessun vincitore.
 - Modificare la classe `Tris` data a lezione utilizzando il `for` generalizzato
-