

PROVA PRATICA – LINGUAGGIO JAVA

Esercizio (da realizzare in un UNICO FILE .java)

Nella soluzione, prediligere il maggior riutilizzo di codice e la maggiore protezione possibile.

- Si realizzi un componente interfaccia **Collezione** che rappresenta la tipologia di una collezione di oggetti. In tale interfaccia si definiscono i metodi:
  - **public boolean member(Object X)**  
che ricevendo il riferimento a un oggetto **X**, stabilisce se esso appartiene alla collezione;
  - **public Collezione add(Object X)**  
che ricevendo il riferimento a un oggetto **X**, lo aggiunge alla **Collezione**.
- Si realizzi poi una classe **ListaInteri** che implementa l'interfaccia precedente. Ciascun **ListaInteri** (se non **null**) ha un elemento **Integer** in testa e un riferimento al resto della sequenza (**ListaInteri**). Il metodo **member** restituisce vero se l'oggetto **X** è l'elemento in testa alla lista (**ListaInteri**) o appartiene al resto della lista, falso altrimenti (se la lista è vuota). Il metodo **add** ricevendo il riferimento a un oggetto **X**, lo aggiunge in testa alla **ListaInteri** e restituisce un riferimento alla lista stessa.
- Si realizzi poi una funzione **main** in una classe **Prova** che:
  - Letti due valori interi da input, dichiarati due oggetti **i** e **j**, istanze della classe **Integer**, rispettivamente inizializzati con i valori letti.
  - Istanzi un oggetto **l** della classe **ListaInteri**, con **i** come primo elemento.
  - Aggiunga con il metodo **add** l'oggetto **j** alla lista **l** e verifichi con il metodo **member** se **j** appartiene alla lista **l**, stampando un messaggio a video (trovato, non trovato).

```

import java.io.*;

interface Collezione {

    public boolean member(Object X);

    public Collezione add(Object X);
}

class ListaInteri implements Collezione {

    private Integer head;
    private ListaInteri resto;

    //costruttore
    public ListaInteri(Integer n, ListaInteri l) {
        this.head = n;
        this.resto = l;
    }

    public boolean member(Object X) {

        if((this.head!=null)&&(this.head.compareTo((Integer) X) == 0)){
            return true;
        } else if (this.resto != null) {
            return resto.member(X);
        } else {
            return false;
        }

    }

    public Collezione add(Object X) {
        ListaInteri L = new ListaInteri((Integer) X, this);
        return L;
    }

}

public class Prova {

    public static void main(String[] args) {
        try{
            /*(1) PRIMA PARTE*/
            /*Apertura dello std in*/
            BufferedReader stdin=new BufferedReader(new InputStreamReader(System.in));
            System.out.println("+ Inserisci il primo numero intero: ");
            String temp=stdin.readLine();
            /*letto l'intero sottoforma di stringa creo un Integer associato*/
            Integer i=new Integer(temp);

            System.out.println("- Inserisci il secondo numero intero: ");
            temp=stdin.readLine();
            /*letto l'intero sottoforma di stringa creo un Integer associato*/
            Integer j=new Integer(temp);
            /*chiusura del stdin*/
            stdin.close();

            /*(2) SECONDA PARTE*/
            ListaInteri l=new ListaInteri(i,null);

            /*(3) TERZA PARTE*/
            l=(ListaInteri) l.add(j);
        }
    }
}

```

```
/*e verifichi con il metodo member se j appartiene alla lista l,
stampando un messaggio a video (trovato, non trovato).*/
boolean result=l.member(j);

    if (result) System.out.println("trovato");
    else      System.out.println("non trovato");
}catch(Exception e) {System.err.println(e); System.exit(1);}
}
}
```