



POLITECNICO
MILANO 1863

Ingegneria del Software

Ingegneria Informatica – Sez. P-Z

Esercitazione:

Java

– esercizio introduttivo –

Poligoni

Si scriva una classe che rappresenti un poligono nel piano.

Ogni poligono è costituito da una successione di punti, ognuno descritto tramite le sue coordinate cartesiane.

Per ogni poligono deve essere possibile calcolare il perimetro.

– esercizio introduttivo –

Poligoni

Si scriva una classe che rappresenti un poligono nel piano.

Ogni poligono è costituito da una successione di punti, ognuno descritto tramite le sue coordinate cartesiane.

Per ogni poligono deve essere possibile calcolare il perimetro.

Una volta creato, un poligono deve essere immutabile.

– esercizio introduttivo –

Poligoni nD

Si scriva una classe che rappresenti un poligono **nell'iperspazio**.

Ogni poligono è costituito da una successione di punti, ognuno descritto tramite le sue coordinate cartesiane.

Per ogni poligono deve essere possibile calcolare il perimetro.

Una volta creato, un poligono deve essere immutabile.

– esercizio introduttivo –

Poligoni nD mutabili

Si scriva una classe che rappresenti un poligono nell'(iper)spazio.

Ogni poligono è costituito da una successione di punti, ognuno descritto tramite le sue coordinate cartesiane.

Per ogni poligono deve essere possibile calcolare il perimetro.

Una volta creato, un poligono deve essere **immutabile**.

– esercizio introduttivo –

Poligoni nD mutabili «ottimizzati»

Si scriva una classe che rappresenti un poligono nell'(iper)spazio.

Ogni poligono è costituito da una successione di punti, ognuno descritto tramite le sue coordinate cartesiane.

Per ogni poligono deve essere possibile calcolare il perimetro.

Una volta creato, un poligono deve essere immutabile e il perimetro deve essere calcolato solo una volta per ogni modifica.

Esercizi

Esercizio 1

- Indicare il comportamento del seguente programma nello heap:

```
public static void main(String[] args) {  
    String s1 = "abc";  
    String s2 = "abc";  
    String s3 = new String("abc");  
    String s4 = s3;  
}
```


Esercizio 1

(soluzione)

- Indicare il comportamento del seguente programma nello heap:

```
public static void main(String[] args) {  
    String s1 = "abc";  
    String s2 = "abc";  
    String s3 = new String("abc");  
    String s4 = s3;  
}
```

Puntano allo stesso
oggetto immutabile nello heap

Esercizio 1

(soluzione)

- Indicare il comportamento del seguente programma nello heap:

```
public static void main(String[] args) {  
    String s1 = "abc";  
    String s2 = "abc";  
    String s3 = new String("abc");  
    String s4 = s3;  
}
```

s3 punta ad un oggetto diverso da s1 e s2
ma è lo stesso oggetto puntato da s4

Esercizio 2 - PointStackedSet

Implementare una classe PointStackedSet:

- Che raccolga un numero arbitrario di Point (punti nello spazio bidimensionale)
- (*push*) Che permetta di aggiungere alla raccolta un Point, mantenendo l'ordine di inserimento
- (*pop*) Che permetta di rimuovere dalla raccolta un Point, nell'ordine inverso rispetto a quello di inserimento.

Esercizio 2 - PointStackedSet

Si estendano le funzionalità precedenti in modo da permettere:

- Di creare un nuovo PointStackedSet come copia di un PointStackedSet esistente

Esercizio 3 - Gara Canora

- Scrivere un insieme di classi per gestire una gara canora.
- Per ogni cantante bisogna indicare le generalità e la canzone in gara
- La competizione deve poter gestire i cantanti e la classifica generale