

## Esercitazione sul Polimorfismo

### (Fondamenti di Informatica – Emilio Di Giacomo)

**Esercizio 1.** Sia data la seguente interface **Funzione** che rappresenta funzioni reali di variabile reale. Il metodo **valore(double x)** restituisce il valore della funzione calcolata nel punto **x**. Il metodo **derivata()** restituisce un oggetto **Funzione** che rappresenta la derivata della funzione rappresentata dall'oggetto su cui il metodo è stato invocato.

```
public interface Funzione{
    /* restituisce il valore della Funzione nel punto x */
    public double valore(double x);

    /* restituisce la derivata della Funzione */
    public Funzione derivata();
}
```

Si scrivano due classi **SommaDiFunzioni** e **ProdottoDiFunzioni** che implementano l'interface **Funzione** e rappresentano rispettivamente funzioni della forma  $f(x)=g(x)+h(x)$  e  $f(x)=g(x)\cdot h(x)$ , dove  $g(x)$  e  $h(x)$  sono funzioni qualsiasi. Entrambe le classi avranno un costruttore che riceve in ingresso due oggetti **Funzione** che rappresentano le due funzioni  $g(x)$  e  $h(x)$  ed il metodo **toString()** che restituisce una descrizione della funzione.

*Si ricorda che la derivata della funzione  $f(x)=g(x)+h(x)$  è la funzione  $f'(x)=g'(x)+h'(x)$ , mentre la derivata della funzione  $f(x)=g(x)\cdot h(x)$  è la funzione  $f'(x)=g'(x)\cdot h(x) + g(x)\cdot h'(x)$  in entrambi i casi  $g'(x)$  è la derivata di  $g(x)$  e  $h'(x)$  è la derivata di  $h(x)$ .*

**Esercizio 2.** Con riferimento alla interface **Funzione** dell'Esercizio precedente, si scriva una classe **FunzionePolinomiale** che implementa l'interface **Funzione** e rappresenta funzioni della forma  $p(x)=a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\dots+a_1x+a_0$ . La classe avrà un costruttore che riceve in ingresso un array di **double a**, il cui elemento **a[i]** rappresenta il coefficiente  $a_i$  della funzione polinomiale  $p(x)$ . Si scrivano poi le classi **FunzioneSinusoidale** e **FunzioneCosinusoidale** che implementano l'interface **Funzione** e rappresentano funzioni della forma  $s(x)=a \sin(b\cdot x+c)$  e  $c(x)=a \cos(b\cdot x+c)$ , rispettivamente. Entrambe le classi avranno un costruttore che riceve in ingresso tre valori **double a**, **b** e **c** che rappresentano i tre coefficienti  $a$ ,  $b$  e  $c$  della funzione  $s(x)$  e  $c(x)$ , rispettivamente. Tutte e tre le classi inoltre saranno dotate del metodo **toString()** che restituisce una descrizione della funzione.

*Si ricorda che la derivata della funzione  $p(x)=a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\dots+a_1x+a_0$  è la funzione  $p'(x)=b_{n-2}x^{n-2}+b_{n-3}x^{n-3}+\dots+b_1x+b_0$ , con  $b_i=a_{i+1}(i+1)$  (per  $0 \leq i \leq n-2$ ); la derivata della funzione  $s(x)=a \sin(b\cdot x+c)$  è la funzione  $s'(x)=ab \cos(b\cdot x+c)$ ; e la derivata della funzione  $c(x)=a \cos(b\cdot x+c)$  è la funzione  $c'(x)=-ab \sin(b\cdot x+c)$ .*

**Esercizio 3.** Utilizzando le classi definite negli Esercizi precedenti scrivere una classe **ProvaFunzione** dotata del solo metodo **main** che svolge le seguenti azioni:

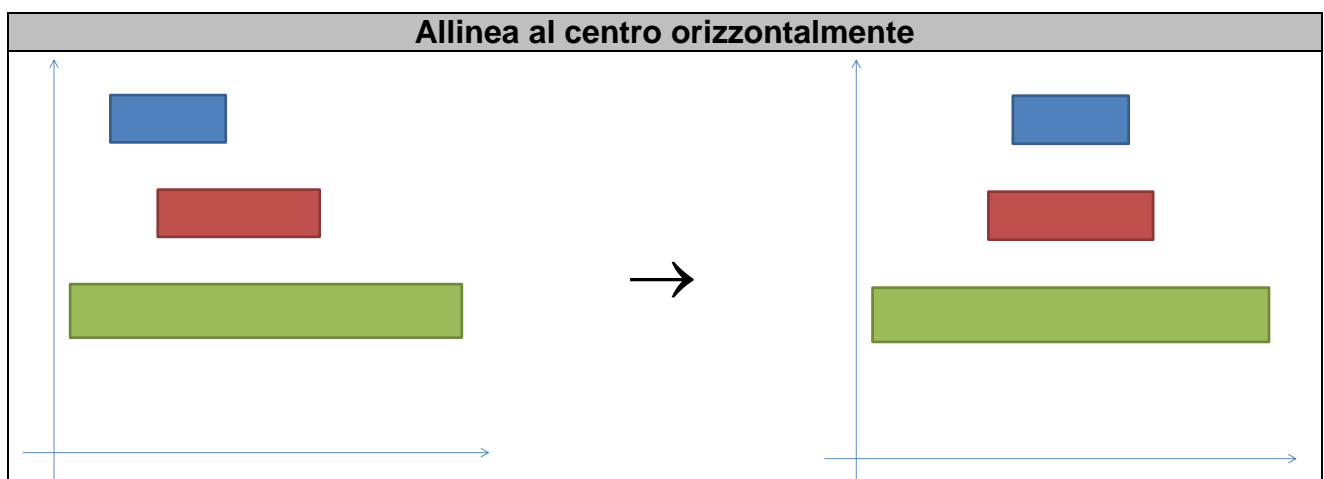
- crea un oggetto **FunzionePolinomiale g** ed un oggetto **FunzioneSinusoidale h** chiedendo i parametri all'utente;

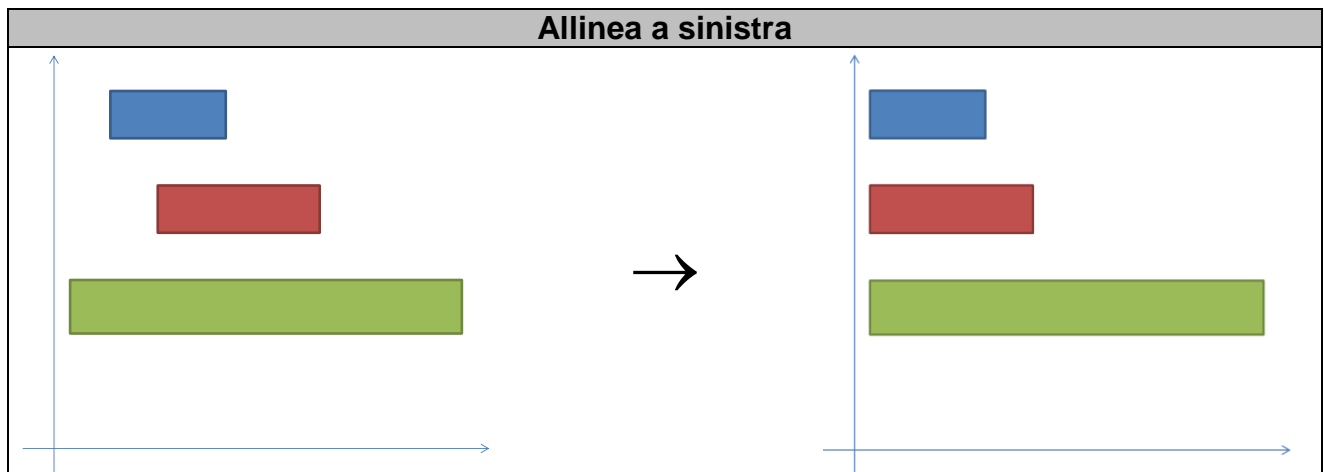
- utilizza i due oggetti **g** ed **h** per creare un oggetto **SommaDiFunzioni f1** ed un oggetto **ProdottoDiFunzioni f2**;
- stampa le derivate delle funzioni rappresentate dagli oggetti **g**, **h**, **f1** ed **f2**;
- stampa il valore delle funzioni rappresentate dagli oggetti **g**, **h**, **f1** ed **f2** e delle loro derivate in un punto **x** scelto dall'utente.

**Esercizio 4.** Si vuole realizzare un modulo da integrare in un editor di grafica vettoriale. In particolare il modulo dovrà occuparsi dell'allineamento degli oggetti grafici presenti nell'editor (comando "Allinea").

Il modulo dovrà avere le seguenti funzionalità:

- "allinea al centro orizzontalmente": l'esecuzione di questo comando allinea tutti gli oggetti grafici a cui il comando è applicato in maniera che i loro centri abbiano la stessa coordinata x;
- "allinea a destra": l'esecuzione di questo comando allinea tutti gli oggetti grafici a cui il comando è applicato in maniera che i loro estremi destri abbiano la stessa coordinata x;
- "allinea a sinistra": l'esecuzione di questo comando allinea tutti gli oggetti grafici a cui il comando è applicato in maniera che i loro estremi sinistri abbiano la stessa coordinata x;
- "allinea al centro verticalmente": l'esecuzione di questo comando allinea tutti gli oggetti grafici a cui il comando è applicato in maniera che i loro centri abbiano la stessa coordinata y;
- "allinea in alto": l'esecuzione di questo comando allinea tutti gli oggetti grafici a cui il comando è applicato in maniera che i loro estremi superiori abbiano la stessa coordinata y;
- "allinea in basso": l'esecuzione di questo comando allinea tutti gli oggetti grafici a cui il comando è applicato in maniera che i loro estremi inferiori abbiano la stessa coordinata y.





Gli oggetti grafici presenti nell'editor possono essere di diverso tipo: rettangoli, cerchi, ecc. ed è possibile che il numero di tipi di oggetti grafici aumenti in futuro. Si vuole realizzare il modulo che realizza l'allineamento in maniera che sia possibile aggiungere nuovi tipi di oggetti senza doverlo modificare. A tale fine si utilizzi la seguente interface **Allineabile** per realizzare una classe **Allineatore** che implementi le operazioni di allineamento descritte sopra.

```
public interface Allineabile{
```

```
// restituisce la coordinata x del centro della figura
```

```
public double getCX();
```

```
// restituisce la coordinata y del centro della figura
```

```
public double getCY();
```

```
// modifica la coordinata x del centro della figura
```

```
public void setCX(double x);
```

```
// modifica la coordinata y del centro della figura
```

```
public void setCY(double y);
```

```
// restituisce la larghezza della figura
```

```
public double getLarghezza();
```

```
// restituisce l'altezza della figura
```

```
public double getAltezza();
```

```
}
```

Si scrivano due classi **Rettangolo** e **Cerchio** che implementano l'interface **Allineabile** e si scriva una classe di test che:

- fa inserire all'utente un certo insieme di figure allineabili (il numero di figure e il tipo di ciascuna di esse è scelto dall'utente);
- mostra all'utente le figure inserite;
- chiede ripetutamente all'utente quale delle seguenti operazioni intende effettuare: allineamento orizzontale, allineamento verticale, uscire dall'applicazione;
- esegue ogni operazione scelta dall'utente.