Esercizi ARRAY - OOP

Esercizio 5.1: Scrivere un programma **StampaZigZag** che prevede un array di 10 numeri interi contenente valori a piacere (senza bisogno di chiederli all'utente) e ne stampa gli elementi secondo il seguente ordine: il primo, l'ultimo, il secondo, il penultimo, il terzo, il terz'ultimo, ecc... Il nome dell'array può essere scelto a piacere. (Il programma deve essere scritto facendo finta di non sapere quali siano i valori inseriti nell'array)

Esercizio 5.2: Scrivere un programma **SommaPariDispar**i che prevede un array di 10 numeri interi contenente valori a piacere (senza bisogno di chiederli all'utente) e stampa *"Pari e dispari uguali"* se la somma dei numeri in posizioni pari dell'array è uguale alla somma dei numeri in posizioni dispari, altrimenti il programma stampa *"Pari e dispari diversi"*. (Il programma deve essere scritto facendo finta di non sapere quali siano i valori inseriti nell'array)

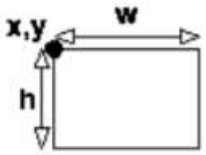
Soluzioni proposte

Esercizio OOP: Progettare ed implementare la classe **Rettangolo** che risponda all'esigenza seguente:

costruire un rettangolo partendo da una base, un'altezza e dalle coordinate del piano.

Si desiderano due costruttori:

- uno ***di default***: crea un rettangolo con le dimensioni definite (base = 1, altezza = 1, **x**=0, **y**=0)
- uno ***parametrico***: permette all'utente di assegnare i valori



Si progettino i seguenti metodi:

- metodi di accesso
- metodi che eseguono operazioni specifiche:
  - traslazione delle coordinate
  - calcolo dell'area
  - calcolo del perimetro

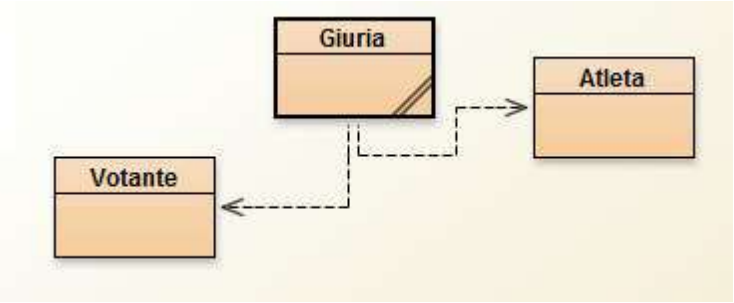
Si proponga un'applicazione Java **RettangoloTester** che testa i metodi progettati.

Chiarimento: per test dei metodi progettati si intende anche l'uso dei due costruttori istanziando sia con costruttore di default che parametrico

Suggerimento: nel definire le dimensioni di default, si suggerisce **unica ascissa** ed **unica ordinata** (le coordinate del punto in alto a sinistra nella grafica con PC) ed è richiesta la *traslazione solo dell'ascissa e dell'ordinata* (di quel punto)

Soluzione

Esercizio (*array di oggetti*):
In una gara il punteggio di ciascun atleta è dato dal pubblico. I voti possono andare da 1 a 10. Progettare ed implementare un'applicazione Java che per ogni atleta rilevi il numero di occorrenze dei vari voti.



Per soluzione si veda Esercizio 7C

Testo e soluzioni http://pages.di.unipi.it/milazzo/teaching/AA1415-ProgJava/dispensa-compitino/Dispensa-esercizi-compitino.pdf

Soluzione esercizio 5.1

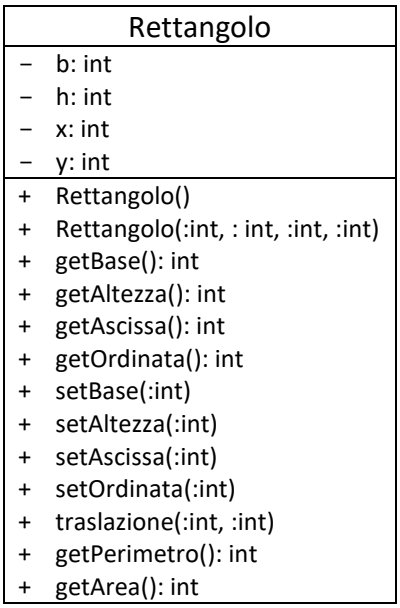
```
public class StampaZigZag {
        public static void main ( String [] args ) {
                int [] valori = { 4, 5, 2, 5, 7, 6, 3, 1, 3, 6 };
                for (int i =0; i <5; i ++) {  // sostituire  5  con  valori.length /2
                   int j=9-i;                 // sostituire 9 con dimensione array -1
                                              // cioè valori.length-1
                   System . out . println ( valori [i]);
                   System . out . println ( valori [j]);
                }
        }
}
```

Soluzione esercizio 5.2 (uso di operatori di ***assegnamento composto***: forma compatta o abbreviata a scapito della leggibilità)

```
public class SommaPariDispari {
        public static void main ( String [] args ) {
                int [] valori = { 4, 5, 2, 5, 7, 6, 3, 1, 3, 2 };
                int sommaPari = 0;
                int sommaDispari = 0;
                for (int i =0; i < valori.length; i +=2) {   // pari  con i inizialmente 0 :
                                                             // i ←i+2
                    sommaPari += valori [i];
                    sommaDispari += valori [i +1];       // dispari … se i pari :   i+1
                }
                if ( sommaPari == sommaDispari )
                    System . out . println (" Pari e dispari uguali ");
                else
                    System . out . println (" Pari e dispari diversi ");
        }
}
```

Forma compatta	Scrittura equivalente
`x += y`	`x = x + y`
`x -= y`	`x = x - y`
`x *= y`	`x = x * y`
`x /= y`	`x = x / y`
`x %= y`	`x = x % y`

Testo e possibile soluzione da http://www.webalice.it/paolo.latella/eserciziario_java.pdf




```
class Rettangolo{
	private int b;
	private int h;
	private int x;
	private int y;
	// si costruisce un rettangolo con i parametri
	predefiniti
	public Rettangolo(){
		b = 1;
		h = 1;
		x = 0;
		y = 0;
	}
	/** si costruisce un rettangolo con i parametri
	acquisiti dall'esterno
	@param base la base del rettangolo
	@param altezza l'altezza del rettangolo
	@param ascissa l'ascissa del rettangolo
	@param ordinata l'ordinata del rettangolo
	 */
	public Rettangolo(int base, int altezza, int
	ascissa, int ordinata){
		b = base;
		h = altezza;
		x = ascissa;
		y = ordinata;
	}
	/** si acquisisce la base
	@return la base del rettangolo
	*/
	public int getBase(){
		return b;
	}
	/** si acquisisce l'altezza
	@return l'altezza del rettangolo
	*/
	public int getAltezza(){
		return h;
	}
	/** si acquisisce l'ascissa
	@return l'ascissa del rettangolo
	*/
	public int getAscissa(){
		 return x;
	}
	/** si acquisisce l'ordinata
	@return l'ordinata del rettangolo
	*/
	public int getOrdinata(){
		return y;
	}
	/** si modifica la base
	@param nuovaBase la nuova misura della base
	*/
	public void setBase(int nuovaBase){
		b = nuovaBase;
	}
	/** si modifica l'altezza
	@param nuovaAltezza la nuova misura dell'altezza
	*/
	public void setAltezza(int nuovaAltezza){
		h = nuovaAltezza;
	}
```

```
	/** si modifica l'ascissa
	@param nuovaAscissa la nuova ascissa
	*/
	public void setAscissa(int nuovaAscissa){
		x = nuovaAscissa;
	}
	/** si modifica l'ordinata
	@param nuovaOrdinata la nuova ordinata
	*/
	public void setOrdinata(int nuovaOrdinata){
		y = nuovaOrdinata;
	}
	/** si traslano le coordinate nel piano
	@param trX lo spostamento in ascissa
	@param trY lo spostamento in ordinata
	*/
	public void traslazione(int trX, int trY){
		x = x + trX;
		y = y + trY;
	}
	/** si calcola il perimetro
	@return il perimetro
	*/
	public int getPerimetro(){
		return (b + h)*2;
	}
	/** si calcola l'area
	@return l'area
	 */
	public int getArea(){
		return b * h;
	}
}
```

RettangoloTester.java

```
public class RettangoloTester{

	public static void main(String[] args){

		Rettangolo r = new Rettangolo();

		System.out.println("Perimetro: " + r.getPerimetro());
		System.out.println("Area: " + r.getArea());
		System.out.println("Expected perimetro: 4");
		System.out.println("Expected area: 1");

		Rettangolo rr = new Rettangolo(5, 3, 9, 2);

		System.out.println("Perimetro: " + rr.getPerimetro());
		System.out.println("Area: " + rr.getArea());
		System.out.println("Expected perimetro: 16");
		System.out.println("Expected area: 15");
	}
}
```

Nb: vedremo in seguito esempi di GUI usando in Java *primitive grafiche* (metodi per disegno diretto)