

MODULO OOP - LINGUAGGIO C# - VERIFICA DI FINE MODULO

Cognome e Nome: Mari Lorenzo Giovanni Roger

Rinominare questo file col vostro cognome (esempio: Rossi.docx), inserire cognome e nome, indicare la risposta corretta fra le quattro elencate per ognuna delle domande da 1 a 5, svolgere con Visual Studio l'esercizio di programmazione.

Consegnare entro l'orario stabilito questo documento contenente le risposte e l'applicazione C# realizzata con Visual Studio in formato compresso (ZIP o RAR).

1. L'istruzione dove si dichiara una variabile **x** di tipo **short** e le si assegna il valore **8** è:

- a. `x = 8;`
- b. `short x = 8;`
- c. `short x;`
- d. `short[] x = {8};`

RISPOSTA: b

2. Data la seguente parte di codice:

```
static void Main()  
{  
    int x = 0;  
    while (x < 10)  
    {  
        x++;  
        continue;  
        System.Console.WriteLine(x);  
    }  
}
```

Che cosa sarà stampato a schermo?

- a. I valori di x da 0 a 9 incolonnati
- b. I valori di x da 0 a 9 nella stessa riga
- c. niente
- d. I valori di x da 0 a 10 incolonnati

RISPOSTA: c

3. Data la seguente definizione della **classe Persona**

```
public class Persona
{
    private string nome;
    private string cognome;

    public Persona()
    {
        nome = "Paolino";
        cognome = "Paperino";
    }
    // ... resto della classe
}
```

l'istruzione che dichiara e istanzia un oggetto di tale classe con il **costruttore di default** è:

- a. `p = new Persona();`
- b. `Persona p = new Persona();`
- c. `Persona p;`
- d. `Persona p = Persona();`

RISPOSTA: b

4. Sempre per la **classe Persona** la dichiarazione della classe pubblica **Impiegato** che **eredita** da **Persona** è:

- a. `public class Impiegato {}`
- b. `public class Impiegato { : Persona }`
- c. `public class Impiegato : Persona {}`
- d. `public class Impiegato {this}`

RISPOSTA: c

5. A proposito di classi astratte e interfacce indicare quale fra le seguenti frasi è interamente vera:

- a. Una classe astratta può essere istanziata da oggetti dichiarati come virtual mentre un'interfaccia no
- b. Una classe astratta può contenere attributi, può contenere l'implementazione di alcuni metodi e proprietà e non può essere istanziata, l'interfaccia contiene solo dichiarazioni di proprietà e metodi senza la loro implementazione
- c. Una classe astratta può contenere attributi, proprietà e metodi, un'interfaccia contiene solo attributi
- d. L'unica differenza tra classi astratte e interfacce è che il nome di un'interfaccia deve iniziare obbligatoriamente con I maiuscola

RISPOSTA: b

PROVA DI PROGRAMMAZIONE

Esercizio 1 (*prioritario*)

Realizzare un'applicazione Windows Form avente come nome il vostro cognome.

Aggiungere al progetto una classe pubblica Figura avente i seguenti attributi:

- tipofigura: string
- numerolati: int
- area: double
- perimetro: double

Devono essere implementati seguenti costruttori:

- Costruttore di default che assegna una stringa vuota a tipo figura e il valore 0 agli altri attributi
- Costruttore con un argomento numerolati che verifichi che sia almeno 3 e, per i valori 3, 4, 5, 6 assegni all'attributo tipo figura "Triangolo", "Quadrilatero", "Pentagono", "Esagono" e, per valori superiori, "Poligono"

Deve essere implementata la Proprietà Get e Set per leggere/scrivere l'attributo numerolati, dove nella sezione set sia utilizzata la stessa regola richiesta per il costruttore con un parametro.

Esercizio 2 (*opzionale*)

Aggiungere poi con il meccanismo dell'ereditarietà le classi Rettangolo, Quadrato e Triangolo.

Per ognuna di queste tre classi derivate lo studente dovrà inserire gli attributi di tipo double necessari per descrivere correttamente la nuova figura, ovvero le misure dei lati e altri dati geometrici.

Dovrà poi aggiungere il costruttore di default e almeno un altro costruttore per classe con parametri che tra le altre cose assegnino anche il nome corretto su tipofigura.

Dovrà quindi aggiungere le proprietà con get e set per accedere ai nuovi attributi.

Infine, in ognuna delle classi derivate, dovranno essere presenti i metodi che calcolino l'area e il perimetro salvandone i risultati nei due attributi definiti nella classe base.

Esercizio 3 (*prioritaria la creazione di un oggetto figura*)

Sul form inserire un button e, nella funzione che gestisce il suo evento click, realizzare la creazione di un oggetto Figura, successivamente un oggetto di tipo Triangolo, uno di tipo Rettangolo e uno di tipo Quadrato e, per ognuno di questi tre, calcolarne l'area e il perimetro.

Esercizio 4 (*opzionale*)

Se possibile, sul form implementare la lettura dei dati per creare Triangolo, Rettangolo e Quadrato da delle textbox e mostrare per ognuna di esse area e perimetro attraverso un MessageBox.