CATEGORIA 1

Lettura e commento codice

Il candidato legga il codice sorgente in C# predisposto dalla commissione e descriva il comportamento del software.

Esercizio1

ESTRATTA

```
4   static void Main(string[] args)
5           {
6               int num1, num2,sayac=0;
7               Console.Write("Numero inferiore: ");
8               num1 = Convert.ToInt32(Console.ReadLine());
9               Console.Write(" superiore: ");
10              num2 = Convert.ToInt32(Console.ReadLine());
11              Console.WriteLine("tra {0} e {1} ci sono: ",num1,num2);
12              Console.WriteLine("==============================");
13              for(int i=num1;i<num2;i++)
14              {
15                  sayac = 0;
16                  if(i>1)
17                  {
18                      for(int j=2;j<i;j++)
19                      {
20                          if(i%j==0)
21                          {
22                              sayac = 1;
23                              break;
24                          }
25                      }
26                      if(sayac==0)
27                      {
28                          Console.WriteLine(i);
29                      }
30                  }
31              }
32              Console.ReadKey();
33          }
34
```

Lettura e commento codice

Il candidato legga il codice sorgente in C# predisposto dalla commissione e descriva il comportamento del software.

Esercizio 2

ESTRATTA

```
38  class Program
39      {
40          static void Main(string[] args)
41          {
42              int number, sum=0;
43
44              Console.Write("Numero: ");
45              number = Convert.ToInt32(Console.ReadLine());
46
47              if(number<0)
48              {
49                  Console.Write("Errore");
50              }
51              else
52              {
53                  while(number>0)
54                  {
55                      sum += number;
56                      number -=1;
57                  }
58              }
59              Console.WriteLine("Risultato:  "+sum);
60
61              Console.ReadKey();
62          }
63      }
```

Lettura e commento codice

Il candidato legga il codice sorgente in C# predisposto dalla commissione e descriva il comportamento del software.

Esercizio 3

```
71    static void Main(string[] args)
72          {
73              int i, number, fact;
74              Console.WriteLine("Numero");
75              number = int.Parse(Console.ReadLine());
76              fact = number;
77              for (i = number - 1; i >= 1; i--)
78              {
79                  fact = fact * i;
80              }
81              Console.WriteLine("Risultato: "+fact);
82              Console.ReadLine();
83
84          }
```

Lettura e commento codice

Il candidato legga il codice sorgente in C# predisposto dalla commissione e descriva il comportamento del software.

Esercizio 4

ESTRATTA

```
92      class Program
93      {
94          static void Main(string[] args)
95          {
96              int baseNumber, expNumber, result = 1;
97              Console.Write("Base: ");
98              baseNumber = Convert.ToInt32(Console.ReadLine());
99              Console.Write("Esponente: ");
100             expNumber = Convert.ToInt32(Console.ReadLine());
101             while (expNumber != 0)
102             {
103                 result *= baseNumber;
104                 --expNumber;
105             }
106             Console.WriteLine("Risultato = {0}", result);
107             Console.ReadLine();
108         }
109     }
```

Lettura e commento codice

Il candidato legga il codice sorgente in C# predisposto dalla commissione e descriva il comportamento del software.

Esercizio 5

ESTRATTA

```
116   class Program
117       {
118           static void Main(string[] args)
119           {
120               int[] numbers = new int[10];
121               Random rnd = new Random();
122               int min, max;
123                for (int i = 0; i < numbers.Length; i++)
124               {
125                   numbers[i] = rnd.Next(1, 100);
126                   Console.WriteLine(numbers[i]);
127               }
128               min = numbers[0];
129               max = numbers[0];
130               for (int i = 1; i < numbers.Length; i++)
131               {
132                   if (min > numbers[i])
133                       min = numbers[i];
134                   if (max < numbers[i])
135                       max = numbers[i];
136               }
137               Console.WriteLine("====================================");
138               Console.WriteLine("Risultato1: " + max);
139               Console.WriteLine("Risultato2: " + min);
140               Console.ReadKey();
141           }
142       }
143
```

Lettura e commento codice

Il candidato legga il codice sorgente in C# predisposto dalla commissione e descriva il comportamento del software.

Esercizio 6

ESTRATTA

```
150   static void Main(string[] args)
151       {
152           Console.Write("Inserisci la stringa:");
153           string text = Console.ReadLine();
154           int len = text.Length;
155           bool flag = true;
156           for (int i = 0; i < len/2; i++)
157           {
158               if (text[i] != text[len - (i + 1)])
159               {
160                   flag = false;
161                   break;
162               }
163           }
164           if (flag)
165           {
166               Console.WriteLine("{0} lo è", text);
167           }
168           else
169           {
170               Console.WriteLine("{0} non lo è", text);
171           }
172           Console.ReadLine();
173       }
```

Lettura e commento codice

Il candidato legga il codice sorgente in C# predisposto dalla commissione e descriva il comportamento del software.

Esercizio 7

ESTRATTA

```
class Program
{
    static void Main(string[] args)
    {
    string[] arr = { "Area", "Di","Ricerca","Science park" };
    string longWord = "";
    int Wordcount = 0;
    foreach (string item in arr)
    {
        if (item.Length > Wordcount)
        {
            Wordcount = item.Length;
            longWord = item;
        }
    }
    Console.WriteLine("Risultato: {0} \nConteggio : {1}", longWord, Wordcount);
    Console.ReadKey();
    }
}
```

Lettura e commento codice

Il candidato legga il codice sorgente in C# predisposto dalla commissione e descriva il comportamento del software.

Esercizio 8

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main()
        {
            int i       = 0;
            int[] arr   = new int[5];

            Console.WriteLine("Anni: ");
            for (i = 0; i < arr.Length; i++)
            {
                Console.Write("Anni[" + (i + 1) + "]: ");
                arr[i] = int.Parse(Console.ReadLine());
            }

            Console.WriteLine("Risultati: ");
            for (i = 0; i < arr.Length; i++)
            {
                if ((arr[i] % 4 == 0) && (arr[i] % 100 != 0) )
                    Console.Write(arr[i] + " ");
                else if ((arr[i] % 4 == 0) && (arr[i] % 100 == 0) && (arr[i] % 400 == 0) )
                    Console.Write(arr[i] + " ");
            }
            Console.WriteLine();
        }
    }
}
```

Lettura e commento codice

Il candidato legga il codice sorgente in C# predisposto dalla commissione e descriva il comportamento del software.

Esercizio 9

ESTRATTA

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main()
        {
            int i = 0;
            int j = 0;
            int[] arr1  = new int[5];
            int[] arr2 = new int[5];
            Console.WriteLine("Inserisci dei numeri: ");
            for (i = 0; i < 5; i++)
            {
                Console.Write("Elementi[" + (i + 1) + "]: ");
                arr1[i] = int.Parse(Console.ReadLine());
            }
            for (i = 0,j=arr1.Length-1; i < arr1.Length; i++)
            {
                arr2[i] = arr1[j--];
            }
            Console.WriteLine("Risultato: ");
            for (i = 0; i < 5; i++)
            {
                Console.WriteLine("Elementi[" + (i + 1) + "]: "+ arr2[i]);

            }
            Console.WriteLine();
        }
    }
}
```

Lettura e commento codice

Il candidato legga il codice sorgente in C# predisposto dalla commissione e descriva il comportamento del software.

Esercizio 10

ESTRATTA

```
using System;

class Demo
{
    static void Main()
    {
        int firstNumber=0;
        int secondNumber=0;

        int temp1=0;
        int temp2=0;

        int lcm=0;

        Console.Write("Numero 1: ");
        firstNumber = Convert.ToInt32(Console.ReadLine());

        Console.Write("Numero 2: ");
        secondNumber = Convert.ToInt32(Console.ReadLine());

        temp1 = firstNumber;
        temp2 = secondNumber;

        while (firstNumber != secondNumber)
        {
            if (firstNumber > secondNumber)
            {
                firstNumber = firstNumber - secondNumber;
            }
            else
            {
                secondNumber = secondNumber - firstNumber;
            }
        }
        lcm = (temp1 * temp2) / firstNumber;

        Console.WriteLine("Risultato : " + lcm);
    }
}
```

CATEGORIA 2

Bug hunting

All'interno del codice proposto nell'esercizio precedente è stato inserito intenzionalmente un errore; il candidato identifichi il bug e proponga una soluzione correttiva.

ESTRATTA

```
================================== 1 ==================================
bool launchReady = false;
int fuelLevel = 17000;

if (fuelLevel >= 20000
{
   Console.WriteLine("Fuel level cleared.");
   launchReady = true;
}
else
{
   Console.WriteLine("WARNING: Insufficient fuel!");
   launchReady = false;
}
```

Bug hunting

All'interno del codice proposto nell'esercizio precedente è stato inserito intenzionalmente un errore; il candidato identifichi il bug e proponga una soluzione correttiva.

```
================================== 2 ==================================

bool launchReady = false;
bool crewStatus = true;
string computerStatus = "green";

if (crewStatus &&& computerStatus == "green")
{
   Console.WriteLine("Crew & computer cleared.");
   launchReady = true;
}
else
{
   Console.WriteLine("WARNING: Crew or computer not ready!");
   launchReady = false;
}

if (launchReady)
{
   Console.WriteLine(("10, 9, 8, 7, 6, 5, 4, 3, 2, 1...");
   Console.WriteLine("Fed parrot...");
   Console.WriteLine("Ignition...");
   Console.WriteLine("Liftoff!");
}
else
{
   Console.WriteLine("Launch scrubbed.");
}
```

Bug hunting

All'interno del codice proposto nell'esercizio precedente è stato inserito intenzionalmente un errore; il candidato identifichi il bug e proponga una soluzione correttiva.

ESTRATTA

========== 3 ==========

```
bool launchReady = false;
int fuelLevel = 17000;

if (fuelLevel >= 20000)
{
   Console.WriteLine("Fuel level cleared.");
   launchReady = true();
}
else
{
   Console.WriteLine("WARNING: Insufficient fuel.");
   launchReady = false();
}
```

Bug hunting

All'interno del codice proposto nell'esercizio precedente è stato inserito intenzionalmente un errore; il candidato identifichi il bug e proponga una soluzione correttiva.

ESTRATTA

========== 4 ==========

```
bool launchReady = false;
int fuelLevel = 17000;
bool crewStatus = true;
string computerStatus = "green";

if (fuelLevel >= 20000)
{
   Console.WriteLine("Fuel level cleared.");
   launchReady = true;
}
else
{
   Console.WriteLine("WARNING: Insufficient fuel!");
   launchReady = false;
}

if (launchReady.status()==true)
{
   Console.WriteLine("10, 9, 8, 7, 6, 5, 4, 3, 2, 1...");
   Console.WriteLine("Liftoff!");
}
else
{
   Console.WriteLine("Launch scrubbed.");
}
```

Bug hunting

All'interno del codice proposto nell'esercizio precedente è stato inserito intenzionalmente un errore; il candidato identifichi il bug e proponga una soluzione correttiva.

ESTRATTA

========== 5 ==========

```
bool launchReady = false;
int fuelLevel = 17000;
bool crewStatus = true;
string computerStatus = "green";

if (fuelLevel !>= 20000)
{
   Console.WriteLine("Fuel level cleared.");
   launchReady = true;
}
else
{
   Console.WriteLine("WARNING: Insufficient fuel!");
   launchReady = false;
}

Console.WriteLine(launchReady);

if (crewStatus== true && computerStatus == "green")
{
   Console.WriteLine("Crew & computer cleared.");
   launchReady = true;
}
else
{
   Console.WriteLine("WARNING: Crew or computer not ready!");
   launchReady = false;
}

Console.WriteLine(launchReady);
```

Bug hunting

All'interno del codice proposto nell'esercizio precedente è stato inserito intenzionalmente un errore; il candidato identifichi il bug e proponga una soluzione correttiva.

ESTRATTA

========== 6 ==========

```
bool launchReady = false;
int fuelLevel = 17000;

if (fuelLevel >= 20000);
{
   Console.WriteLine("Fuel level cleared.");
   launchReady = true;
}
else
{
   Console.WriteLine("WARNING: Insufficient fuel!");
   launchReady = false;
}
```

Bug hunting

All'interno del codice proposto nell'esercizio precedente è stato inserito intenzionalmente un errore; il candidato identifichi il bug e proponga una soluzione correttiva.

ESTRATTA

========== 7 ==========

```
bool launchReady = false();
bool crewStatus;
string computerStatus = "green";

if (crewStatus && computerStatus == "green")
{
   Console.WriteLine("Crew & computer cleared.");
   launchReady = true;
}
else
{
   Console.WriteLine("WARNING: Crew or computer not ready!");
   launchReady = false;
}

if (launchReady)
{
   Console.WriteLine(("10, 9, 8, 7, 6, 5, 4, 3, 2, 1...");
   Console.WriteLine("Fed parrot...");
   Console.WriteLine("Ignition...");
   Console.WriteLine("Liftoff!");
}
else
{
   Console.WriteLine("Launch scrubbed.");
}
```

Bug hunting

All'interno del codice proposto nell'esercizio precedente è stato inserito intenzionalmente un errore; il candidato identifichi il bug e proponga una soluzione correttiva.

========== 8 ==========

```
bool launchReady = false;
int fuelLevel = 17000;

if (fuelLevel >= 20000)
{
   Console.WriteLine("Fuel level cleared.");
   launchReady = true();
}
else
{
   Console.WriteLine("WARNING: Insufficient fuel.");
   launchReady = false();
}

end if
```

Bug hunting

All'interno del codice proposto nell'esercizio precedente è stato inserito intenzionalmente un errore; il candidato identifichi il bug e proponga una soluzione correttiva.

ESTRATTA

```
================================== 9 ================================
bool launchReady = false;
time fuelLevel = 17000;
bool crewStatus = true;
string computerStatus = "green";

if (fuelLevel >= 20000)
{
   Console.WriteLine("Fuel level cleared.");
   launchReady = true;
}
else
{
   Console.WriteLine("WARNING: Insufficient fuel!");
   launchReady = false;
}

if (launchReady==true)
{
   Console.WriteLine("10, 9, 8, 7, 6, 5, 4, 3, 2, 1...");
   Console.WriteLine("Liftoff!");
}
else
{
   Console.WriteLine("Launch scrubbed.");
}
```

Bug hunting

All'interno del codice proposto nell'esercizio precedente è stato inserito intenzionalmente un errore; il candidato identifichi il bug e proponga una soluzione correttiva.

ESTRATTA

```
================================== 10 ================================

bool launchReady = false;
int fuelLevel = 17000;
bool crewStatus = true;
string computerStatus = "green";

if (fuelLevel >= 20000)
{
   Console.WriteLine("Fuel level cleared.");
   launchReady = true().toint;
}
else
{
   Console.WriteLine("WARNING: Insufficient fuel!");
   launchReady = false;
}

Console.WriteLine(launchReady);

if (crewStatus == true && computerStatus == "green")
{
   Console.WriteLine("Crew & computer cleared.");
   launchReady = true;
}
else
{
   Console.WriteLine("WARNING: Crew or computer not ready!");
   launchReady = false;
}

Console.WriteLine(launchReady);
```

CATEGORIA 3

Sviluppo base dati a partire da case study

Il candidato realizzi la struttura di database idonea a memorizzare i dati per un'applicazione descritta come segue.

Esercizio 1. ESTRATTA

In casa ci sono vari dispositivi audiovisivi di memorizzazione: cd, dvd e videocassette, ognuno di essi appartiene a una persona che vive in casa, ma può essere in uso a una sola persona della casa per volta (che potrebbe non essere chi lo possiede).

Le persone sono caratterizzate da un nome, un cognome e una data di nascita; i dispositivi audiovisivi sono tutti caratterizzati da un titolo, i cd anche da un interprete (gruppo o singolo) e da un numero di canzoni, i dvd possono essere o dvd musicali di un concerto con le stesse caratteristiche dei cd o dvd cinematografici caratterizzati dal nome del regista, le videocassette hanno le stesse caratteristiche dei dvd.

Sviluppo base dati a partire da case study

Il candidato realizzi la struttura di database idonea a memorizzare i dati per un'applicazione descritta come segue.

Esercizio 2. ESTRATTA

In una nazione ci sono vari zoo, in ogni città ce ne possono essere al massimo due, ogni zoo ha un proprio nome, un indirizzo, un numero di telefono.

In ogni zoo vivono vari animali ognuno caratterizzati da un nome, una specie, un peso e un'età ogni animale vive all'interno di un recinto che rappresenta un tipo di habitat, in un recinto possono vivere più animali anche di specie diverse.

Sviluppo base dati a partire da case study

Il candidato realizzi la struttura di database idonea a memorizzare i dati per un'applicazione descritta come segue.

Esercizio 3.

In una scuola ci sono varie aule caratterizzate dal piano e dal numero dell'aula; in ogni aula fa lezione una classe caratterizzata da anno, sezione e indirizzo; gli studenti sono caratterizzati da nome, cognome, eventuale numero di cellulare e data di nascita e ogni studente frequenta una classe. Ogni docente, caratterizzato da nome cognome e indirizzo mail, insegna una o più materie in alcune classi e ad alcuni studenti.

Non tutti gli studenti frequentano tutte le materie insegnate alla propria quella classe (si pensi ad esempio religione cattolica che è una materia opzionale o alcune materie di potenziamento).

Sviluppo base dati a partire da case study

Il candidato realizzi la struttura di database idonea a memorizzare i dati per un'applicazione descritta come segue.

Esercizio 4. ESTRATTA

Ad un consorzio partecipano varie aziende agricole tutte caratterizzate da un nome, una partita iva, un indirizzo mail. Le aziende agricole possono avere produzione di verdure, produzione di frutta, allevamento di animali o una combinazione di queste tipologie.

Le aziende agricole che producono frutta e/o verdura hanno dei campi, ogni campo è caratterizzato da: dimensione, indirizzo, tipo di coltura, tutti i campi sono coltivati.

Le aziende agricole che hanno allevamenti hanno dei recinti, ogni recinto è caratterizzato da: dimensione, indirizzo, tipologia (pascolo erboso, terra, laghetto,...), tipologia di animali che vi sono allevati, alcuni recinti potrebbero non essere utilizzati.

Sviluppo base dati a partire da case study

Il candidato realizzi la struttura di database idonea a memorizzare i dati per un'applicazione descritta come segue.

Esercizio 5. ESTRATTA

I giochi in scatola sono caratterizzati da un nome, l'anno della prima edizione, uno o più autori e se sono solitari o per più giocatori, questi ultimi sono caratterizzati da un numero minimo e uno massimo di giocatori e una tipologia di interazione tra i giocatori (cooperativo, competitivo, semi-cooperativo). Per ogni autore di giochi è memorizzato il nome e cognome, il numero di giochi realizzati e il numero di premi vinti.

Ogni collezionista di giochi, caratterizzato da un nome e un nick name, possiede vari giochi e per ogni gioco deve essere memorizzata la data di acquisto, inoltre ogni giocatore può avere il desiderio di acquistare ulteriori giochi.

Sviluppo base dati a partire da case study

Il candidato realizzi la struttura di database idonea a memorizzare i dati per un'applicazione descritta come segue.

Esercizio 6. ESTRATTA

In un'azienda ci sono vari dipendenti, tutti caratterizzati da un nome e un numero di matricola, gli operai, i segretari, addetti alla ricerca e sviluppo e i dirigenti. L'azienda è organizzata in reparti caratterizzati da un nome, a ogni reparto afferisce almeno edificio caratterizzato da tipologia (ufficio, impianto industriale, magazzino), metratura, indirizzo e numero civico, alcuni edifici possono essere in condivisione tra più reparti.

Gli operai sono caratterizzati da una mansione e lavorano in un solo reparto, i segretari possono occuparsi della segreteria commerciale o di quella amministrativa e lavorano in almeno un reparto, ma possono lavorare contemporaneamente in più reparti.

Sviluppo base dati a partire da case study

Il candidato realizzi la struttura di database idonea a memorizzare i dati per un'applicazione descritta come segue.

Esercizio 7. ESTRATTA

Sono trattate diverse specie di piante. Per ciascuna specie sono noti sia il nome latino che il nome comune, ed un codice univoco attraverso cui la specie viene identificata. Per ciascuna specie `e inoltre noto se sia tipicamente da giardino o da appartamento o entrambi e se sia una specie esotica o no. Le piante possono essere verdi oppure fiorite. Nel caso di specie di piante fiorite, sono note tutte le colorazioni in cui ciascuna specie `e disponibile.

Sviluppo base dati a partire da case study

Il candidato realizzi la struttura di database idonea a memorizzare i dati per un'applicazione descritta come segue.

Esercizio 8. ESTRATTA

I clienti sono identificati attraverso un codice cliente e sono costituiti da privati e da rivendite. Per ciascun privato sono noti il codice fiscale, il nome e l'indirizzo della persona, mentre per ogni rivendita sono noti la partita iva, il nome e l'indirizzo della rivendita. Ogni cliente può acquistare diversi prodotti di cui si conoscono marca, tipologia di prodotto, modello e prezzo. Nell'acquisto vengono tracciati data, cliente, prodotti acquistati.

Sviluppo base dati a partire da case study

Il candidato realizzi la struttura di database idonea a memorizzare i dati per un'applicazione descritta come segue.

Esercizio 9.

Nella gestione dei fornitori, questi sono identificati attraverso un codice fornitore e sono inoltre noti il nome, il codice fiscale e l'indirizzo. Il fornitore può fornire diversi prodotti. I prodotti afferiscono a diverse categorie e sono caratterizzati da codice prodotto, descrizione, prezzo, deperibilità ed eventuale data di scadenza. Ogni fornitore vuole gestire un catalogo di offerte con prodotti e periodo di validità delle offerte e sconto associato al periodo.

Sviluppo base dati a partire da case study

Il candidato realizzi la struttura di database idonea a memorizzare i dati per un'applicazione descritta come segue.

Esercizio 10. ESTRATTA

Una catena di distribuzione possiede diversi negozi e in ogni città ce ne possono essere al massimo ree, ogni negozio ha un proprio nome, un indirizzo, un numero di telefono ed un codice identificativo costituito da caratteri alfanumerici.

In ogni negozio vengono venduti diversi prodotti caratterizzati da codice prodotto, descrizione, importo unitario, limite inferiore di riordino e posizione nel reparto; in un reparto possono essere posizionati più prodotti. Gli addetti alle vendite hanno in gestione diversi reparti.

CATEGORIA 4

Roleplay analisi requisiti

Sulla base del progetto proposto, il candidato effettui un'intervista ai membri della commissione cercando di identificare tutti i vincoli funzionali e le logiche come se si trattasse di una reale fase di sviluppo software.

ESTRATTA

1. Progetto di realizzazione di un software per la gestione di una biblioteca

Roleplay analisi requisiti

Sulla base del progetto proposto, il candidato effettui un'intervista ai membri della commissione cercando di identificare tutti i vincoli funzionali e le logiche come se si trattasse di una reale fase di sviluppo software.

2. Progetto di realizzazione di un software per la gestione di una piscina

Roleplay analisi requisiti

Sulla base del progetto proposto, il candidato effettui un'intervista ai membri della commissione cercando di identificare tutti i vincoli funzionali e le logiche come se si trattasse di una reale fase di sviluppo software.

ESTRATTA

3. Progetto di realizzazione di un software per la gestione degli appuntamenti per una ditta che fa assistenza

Roleplay analisi requisiti

Sulla base del progetto proposto, il candidato effettui un'intervista ai membri della commissione cercando di identificare tutti i vincoli funzionali e le logiche come se si trattasse di una reale fase di sviluppo software.

ESTRATTA

4. Progetto di realizzazione di un software per la gestione della mensa

Roleplay analisi requisiti

Sulla base del progetto proposto, il candidato effettui un'intervista ai membri della commissione cercando di identificare tutti i vincoli funzionali e le logiche come se si trattasse di una reale fase di sviluppo software.

5. Progetto di realizzazione di un software per la gestione di una società sportiva

Roleplay analisi requisiti

Sulla base del progetto proposto, il candidato effettui un'intervista ai membri della commissione cercando di identificare tutti i vincoli funzionali e le logiche come se si trattasse di una reale fase di sviluppo software.

ESTRATTA

6. Progetto di realizzazione di un software per la gestione di un negozio che vende ricambi per biciclette

Roleplay analisi requisiti

Sulla base del progetto proposto, il candidato effettui un'intervista ai membri della commissione cercando di identificare tutti i vincoli funzionali e le logiche come se si trattasse di una reale fase di sviluppo software.

ESTRATTA

7. Progetto di realizzazione di un software per la gestione dell'inventario del magazzino hardware informatico

Roleplay analisi requisiti

Sulla base del progetto proposto, il candidato effettui un'intervista ai membri della commissione cercando di identificare tutti i vincoli funzionali e le logiche come se si trattasse di una reale fase di sviluppo software.

8. Progetto di realizzazione di un software per la gestione di una clinica ortopedica ESTRATTA

Roleplay analisi requisiti

Sulla base del progetto proposto, il candidato effettui un'intervista ai membri della commissione cercando di identificare tutti i vincoli funzionali e le logiche come se si trattasse di una reale fase di sviluppo software.

9. Progetto di realizzazione di un software per la gestione di un cinema ESTRATTA

Roleplay analisi requisiti

Sulla base del progetto proposto, il candidato effettui un'intervista ai membri della commissione cercando di identificare tutti i vincoli funzionali e le logiche come se si trattasse di una reale fase di sviluppo software.

10. Progetto di realizzazione di un software per la gestione di una paninoteca ESTRATTA

INGLESE

Methods can be either *instance* or *static*. Invoking an instance method requires that you instantiate an object and call the method on that object; an instance method operates on that instance and its data.

You invoke a static method by referencing the name of the type to which the method belongs; static methods don't operate on instance data. Attempting to call a static method through an object instance generates a compiler error.

ESTRATTA

Calling a method is like accessing a field. After the object name (if you're calling an instance method) or the type name (if you're calling a `static` method), add a period, the name of the method, and parentheses. Arguments are listed within the parentheses and are separated by commas.

ESTRATTA

You can also use *named arguments* instead of positional arguments when invoking a method. When using named arguments, you specify the parameter name followed by a colon (":") and the argument.

Properties behave like fields when they're accessed. However, unlike fields, properties are implemented with accessors that define the statements executed when a property is accessed or assigned.

ESTRATTA

The syntax shown above is the *auto property* syntax. The compiler generates the storage location for the field that backs up the property. The compiler also implements the body of the `get` and `set` accessors.

ESTRATTA

The property definition shown above is a read-write property. Notice the keyword `value` in the set accessor. The `set` accessor always has a single parameter named `value`.

ESTRATTA

Arguments to the method can appear in any order, as long as all required arguments are present. The following example uses named arguments to invoke the `TestMotorcycle.Drive` method.

ESTIMATTA

In addition to the members that are explicitly defined in a type, a type inherits members defined in its base classes. Since all types in the managed type system inherit directly or indirectly from the Object class, all types inherit its members, such as Equals(Object), GetType(), and ToString().

ESTIMATTA

You can extend this same syntax to anything needed in your scenario. You can check the relationships between different properties, or validate against any external conditions. Any valid C# statements are valid in a property accessor.

ESTIMATTA