وزارت علوم تحقیقات و فناوری
دانشگاه جامع علمی کاربردی
مرکز آموزش عالی علمی کاربردی جهاددانشگاهی شعبه ساوه
نیمسال اول ۸۹ - ۸۸

نظام آموزشی: ترمی | مقطع: کاردانی | رشته : نرم افزار | درس: برنامه‌نویسی شیئ‌گرا (برنامه‌نویسی پیشرفته) | مدرس: حمید رضا نیرومند

مدت زمان امتحان: ۱۲۰ دقیقه | تعداد صفحات : ۲ | شماره دانشجویی: | نام و نام خانوادگی:

۱- کدام گزینه در مورد C# غلط است؟ (۲ نمره)

a. کدهای زبان C# در اصطلاح Type-Safe است.
b. C# یک زبان Strongly-Typed است.
c. C# به کوچک و بزرگ بودن حروف حساس نیست.
d. آرایه‌های C# در اصطلاح Zero-Based هستند.

۲- در مورد وراثت، کدام گزینه غلط است؟ (۲ نمره)

a. در C# می‌توان کلاسی تعریف کرد که از هیچ کلاسی به ارث نبرد.

توضیح: در C# هیچ کلاسی وجود ندارد که از کلاس دیگر به ارث نبرد. حتی اگر کاربر برای یک کلاس، پدر یا Parent تعیین نکند، آن کلاس از کلاس System.Object به ارث می‌برد.

b. به کلاس اصلی که کلاس‌های دیگر از آن به ارث می‌برند، Base Class یا Parent گفته می‌شود.
c. با ایجاد یک شیئ از کلاس A که از کلاس B به ارث برده است، ابتدا سازنده کلاس A فراخوانی می‌شود و به محض فراخوانی، سازنده کلاس B اجرا می‌شود و بعد از اجرای کامل سازنده B کنترل مجددا به سازنده کلاس A بر می‌گردد.
d. اگر کلاس A از کلاس B به ارث ببرد، می‌توان روابط بین آن‌ها را A ISA B تعریف کرد.

۳- شما هم اکنون در پایان یک دوره «برنامه‌نویسی شیئ‌گرا» قرار دارید. اگر از شما بپرسند «شیءگرایی» به چه معناست، پاسخ شما چه خواهد بود؟ (۵ نمره)

پاسخ: مفهوم شیئ‌گرایی: شیئ‌گرایی مفهوم جدیدی در زبان‌های برنامه‌نویسی پیشرفته است که هدف اصلی آن ساده‌تر کردن درک برنامه‌نویسی است. برای رسیدن به این هدف، برنامه‌ها با مفاهیم دنیای واقعی تطبیق داده می‌شود. به طور مثال مفهوم «کتاب» در برنامه، به عنوان یک شیئ (Object) در نظر گرفته می‌شود که مانند یک کتاب واقعی، دارای صفاتی مثل «نام»، «نویسنده» و ... است که در زبان‌های شیئ‌گرا به این صفات، Property گفته می‌شود. همچنین ممکن است دارای عکس العمل‌هایی در مقابل تغییرات باشد که آن‌ها Event و ممکن است عملکردهایی در مورد آن قابل تصور باشد که بدان‌ها Method گفته می‌شود.

در برنامه‌نویسی شیئ‌گرا،کلاسی از اشیاء تعریف شده و هر شیئ یک نمونه از این کلاس در نظر گرفته می‌شود. مثل کلاسی به نام «انسان» که شخصی به نام «حسین» یک نمونه از کلاس «انسان» است.

مفاهیم دیگری مثل «به ارث بردن رفتارها و صفات یک کلاس از کلاس دیگر» نیز مانند دنیای واقعی است که برنامه‌نویسی را بسیار سریع‌تر می‌کند.

برنامه‌نویسی شیئ‌گرا، همچون برنامه‌نویسی رویه‌ای نیست که توابع به صورت جداگانه و بدون ارتباط با یکدیگر باشند و مجبور باشیم برای یک مفهوم، هر کدام را به صورت جداگانه فراخوانی کرده و به آن مفهوم نسبت دهید. بلکه در شیئ‌گرا، مفهومی به نام Encapsulation (کپسوله سازی) وجود دارد که اجازه می‌دهد تمام صفات و متودها و هر چه یک شیئ نیاز دارد را در قالب یک کلاس تعریف کنید و با ایجاد یک نمونه از یک کلاس، تمام این مفاهیم نیز به شیئ جدید نسبت داده شود.

در کل، شیئ‌گرایی با تطبیق مفاهیم زبان برنامه‌نویسی با دنیای واقعی و ایجاد امکانات پیشرفته، باعث تسهیل و تسریع در برنامه‌نویسی می‌شود.

۴- .NET Framework چیست؟ اجزای آن را ضمن اشاره به ماشین مجازی جاوا نام برده و توضیح دهید. (۶ نمره)

پاسخ: چارچوب کاری .NET یا دات.نت فریم‌ورک، مجموعه‌ای از مؤلفه‌ها، ابزارها و امکانات، است که به کاربر امکان ایجاد برنامه‌های کاربردی (تحت .NET) را می‌دهد. این چارچوب (یا چهارچوب)، شامل یک ماشین مجازی، مجموعه‌ای از صدها کلاس مورد نیاز، کامپایلر و ... است.

اجزای مهم .NET Framework عبارتند از:

۱- CLR یا Common Language Runtime : سی.ال.آر جزء اصلی چارچوب .NET است که همچون ماشین مجازی جاوا (JVM = Java Virtual Machine) بستری را برای اجرای برنامه‌های تحت دات.نت فراهم می‌کند. این مؤلفه که برای هر ماشین به صورت خاص ساخته می‌شود، با استفاده از اجزای خود، کدهای زبان میانی مایکروسافت یا MSIL را به زبان قابل فهم برای بستر نهایی (مثلاً سیستم عامل لینوکس یا گوشی موبایل) تبدیل می‌کند.

۲- CL یا Class Library : همانطور که از نام این بخش پیداست، یک کتابخانه شامل کلاس‌هایی می‌شود که مایکروسافت زحمت نوشته و تهیه آن‌ها را کشیده است. به طور مثال کلاس‌هایی برای تبدیلات داده‌ای. کلاسی‌هایی برای کار با دیتابیس. اگر این کلاس‌ها نبود، برنامه‌نویس مجبور خود دست به کار شده و تک تک آن‌ها را بنویسد!

۳- CTS یا Common Type System : شامل انواع داده‌ای مشترک در بین زبان‌های .NET است. مایکروسافت برای هماهنگ کردن زبان‌های برنامه‌نویسی خود (مثل VB، C++، C#، J# و...) این بخش را ایجاد کرد تا انواع داده‌ای تعریف شده در دستورات هر زبان را در نهایت به این انواع داده‌ای مشترک تبدیل کند تا برنامه‌های مختلف در هماهنگی داده‌ای بین یکدیگر، ساده‌تر عمل کنند.

۵- تقریباً در تمامی زبان‌های برنامه‌نویسی سطح بالا ساختار switch وجود دارد. کاربرد این دستور چیست؟ ساختار کلی آن را با یک مثال بیان کنید. (۵ نمره)

پاسخ: در صورتی که نیاز باشد نسبت به مقادیر مختلف یک متغیر، تصمیمات مختلفی گرفته شود، یکی از روش‌ها، استفاده از دستور if…else if …. else if …. else است.

که اگر این تصمیمات زیاد باشد، تعداد بسیار زیادی if … else و نوشتن مکرر نام متغیر، خواهیم داشت که موجب سردرگمی برنامه‌نویس و عدم بهینه بودن برنامه خواهد شد.

برای رفع این مشکل از ساختار Switch استفاده می‌شود. به شکل زیر:

```
switch (نام متغیر)
{
    case مقدار اول :
            دستورات در صورت صحت مقدار اول ;
    break;

    case مقدار دوم :
            دستورات در صورت صحت مقدار دوم ;
    break;
.
.
.
    default :
            دستورات، در صورتی که هیچ کدام از مقادیر برقرار نباشد ;
    break;
}
```

مثال:

```
string x = textBox1.Text;
switch (x){
        case "Jahad":
                MessageBox.Show("The name of the Uni");
        break;
        case "Hamid Reza":
                MessageBox.Show("My first name");
        break;
        case "Niroomand":
                MessageBox.Show("My last name");
        break;
        default:
                MessageBox.Show("Unknown  name");
        break;
}
```

۶- توضیح دهید که برنامه زیر در هر خط چه کاری انجام می‌دهد. (در پاسخنامه، ذکر شماره هر خط کافی است) (۱۵ نمره)

راهنمایی: با فراخوانی متود Next() از شیئ rand یک عدد رندوم بین محدوده مشخص شده ایجاد می‌شود.

```
10  static void Main(string[] args)
11  {
12      int UB1 = Int32.Parse(Console.ReadLine());
13      int UB2 = Int32.Parse(Console.ReadLine());
14      int[,] ARR = new int[UB1, UB2];
15      Random rand = new Random();
16      int i, j;
17      for (i = 0; i < UB1; i++)
18          for (j = 0; j < UB2; j++)
19              ARR[i, j] = rand.Next(1,10);
20
21      int search = Int32.Parse(Console.ReadLine());
22      int count = 0;
23
24      foreach (int x in ARR)
25          if (x == search)
26              count++;
27
28      Console.WriteLine(count);
29
30      Console.ReadLine();
31  }
```

در مورد برنامه بالا، به سؤالات زیر پاسخ دهید:

الف. اگر مقادیر UB1=2 و UB2=3 وارد شوند، خروجی برنامه چه خواهد بود؟ (هر کجا به یک عدد رندوم نیاز بود، یک عدد دلخواه وارد نمایید)

ب. شکل انتزاعی داده‌های ذخیره شده در حافظه را رسم کنید.

پاسخ:

۱۰- متود Main که آغاز اجرای تمام برنامه‌هاست.

۱۲- عددی از نوع int را به عنوان تعداد سطرهای آرایه دو بعدی ARR دریافت می‌کند و در متغیری به نام UB1 قرار می‌دهد.

۱۳- عددی از نوع int را به عنوان تعداد ستون‌های آرایه دو بعدی ARR دریافت می‌کند و در متغیری به نام UB2 قرار می‌دهد.

۱۴- یک آرایه دو بعدی با نام ARR به تعداد UB1 سطر و UB2 ستون تعریف می‌شود.

۱۵- یک شیئ به نام rand از کلاس Random ایجاد می‌شود.

۱۶- تعریف دو متغیر از نوع int

۱۷ – ۱۸ و ۱۹ – با دو حلقه تو در تو تک تک خانه‌های آرایه مرور می‌شود و در آن‌ها عددی رندوم بین یک و ده قرار داده می‌شود.

۲۱- مقدار یک متغیر به نام search و از نوع int را از ورودی دریافت می‌کند تا در آرایه دنبال آن بگردد.

۲۲- یک متغیر از نوع int به عنوان شمارنده تعریف می‌شود.

۲۴- با استفاده از یک حلقه foreach تک‌تک عناصر آرایه چک می‌شود و هر خانه به نوبت در متغیری محلی به نام x ریخته می‌شود.

۲۵- اگر مقدار خانه جاری آرایه برابر با مقدار جستجو شده بود،

۲۶- یک واحد به شمارنده افزوده می‌شود.

۲۸- مقدار متغیر count که تعداد وقوع متغیر search در آرایه ARR است در خروجی نمایش داده می‌شود.

۳۰- کنسول منتظر دریافت یک رشته از ورودی می‌ماند که از این ترفند برای باز نگه داشتن پنجره کنسول استفاده می‌شود.

الف: فرض می‌کنیم اعداد رندومی که در آرایه قرار گرفته‌اند، به صورت زیر باشد:

9	3	5
1	4	3

و عددی قرار است جستجو شود، ۳ باشد. در این حالت، خروجی، ۲ خواهد بود. (۲ بار عدد ۳ در آرایه تکرار شده است)

ب: نحوه‌ی ذخیره سازی آرایه دو بعدی در حافظه به صورت زیر است که یک «تقسیم منطقی» در محل مشخص شده داریم:

9
3
5
1
4
3

۶- تصور کنید قرار است یک رشته را رمزگذاری کنیم. به این صورت که به کد اسکی مربوط به تک‌تک کاراکترهای رشته، یک مقدار معین اضافه کنیم و سپس کاراکتر مربوط به کد جدید را نمایش دهیم. اگر کد رمز رشته، برابر با "طول رشته منهای یک" باشد، برنامه‌ای برای اجرای این طرح بنویسید. (۱۰ نمره)

پاسخ:

راهنمایی: برای پیدا کردن کد اسکی یک کاراکتر، بعد از آنکه آن کاراکتر را از رشته جدا کردید، آن را به نوع داده Char و سپس به Int16 یا Int32 تبدیل کنید.

```
string str = textBox1.Text;
            string encodded = "";
            char c;
            for (int i = 0; i <= str.Length; i++)
            {
                c = Convert.ToChar(str.Substring(i, 1));
                encodded += Convert.ToString(Convert.ToChar(Convert.ToInt16(c) + (str.Length - 1)));
            }
MessageBox.Show(encodded);
```

۷- برنامه‌ای بنویسید که از یک textBox یک عدد int دریافت کند و به یک متود ارسال کند. (۵ نمره)

آن متود، مشخص کند که عدد اول است یا خیر، سپس نسبت به نتیجه‌ی متود، پیغام مناسبی در یک label نمایش داده شود.

پاسخ: کلاسی با نام mth شامل متود تشخیص دهنده اول بودن یا نبودن عدد، به صورت زیر تعریف می‌کنیم:

```
public class mth
    {
        public static bool IsPrime(int num){
            for (int i = 2; i < num / 2; i++)
            {
                if (num % i == 0)
                    return false;
            }
            return true;
        }
    }
```

در متود Main داریم:

```
static void Main(string[] args)
        {
            int NUM = Int16.Parse(textBox1.Text);
            if (mth.IsPrime(NUM) == true)
                label1.Text = "است اول عدد";
            else
                label1.Text = "نیست اول عدد";
        }
```

۹- یک کتابخانه را در نظر بگیرید. می‌توان سه شیئ زیر را برای آن تصور کرد:

عضو (مشتری) - کتاب - کتابدار

برنامه‌ای شامل کلاس‌های لازم و روابط بین آن‌ها بنویسید.

در ادامه، یک شیئ از کلاس «عضو» تعریف کنید، اطلاعات مربوط به وی را از چندین textBox دریافت کنید و به لیست‌باکس‌هایی اضافه کنید. (۱۰ نمره)

پاسخ: کلاس person یک کلاس پایه (base) برای کلاس عضو و کتابدار به حساب می‌آید. پس صفات مشترک این دو کلاس را در کلاس person به صورت زیر تعریف می‌کنیم:

```
public class person
{
    string f_name, l_name;
    public string FirstName
    {
        get
        {
            return f_name;
        }
        set
        {
            f_name = value;
        }
    }
    public string LastName
    {
        get
        {
            return l_name;
        }
        set
        {
            l_name = value;
        }
    }
}
```

کلاس عضو را می‌توان به صورت زیر تعریف کرد:

```
public class Member:person
    {
        private int id; // شماره عضویت
        private int registration_year; // سال عضویت

        public int ID
        {
            get
            {
                return id;
            }
            set
            {
                id = value;
            }
        }
        public int Registration_Year
        {
            get
            {
                return registration_year;
            }
            set
            {
                registration_year = value;
            }
        }
    }
```

کلاس مربوط به کتابدار (ketaabdaar) نیز مانند Member است. فقط صفاتی مثل «شماره کارمندی» جایگزین صفت «شماره عضویت» می‌شود.

در متود Main چنین کدی خواهیم داشت:

```
static void Main(string[] args)
        {
            Member mem1 = new Member();
            mem1.FirstName = textBox1.Text;
            mem1.LastName = textBox2.Text;
            mem1.ID = textBox3.Text;
            mem1.Registration_Year = textBox4.Text;

            listBox1.Items.Add(mem1.FirstName);
            listBox2.Items.Add(mem1.LastName);
            listBox3.Items.Add(mem1.ID);
            listBox4.Items.Add(mem1.Registration_Year);
        }
```

روابط بین کلاس‌ها:

Member IS-A person

Ketaabdaar IS-A person

۱۰- **پاسخ به این سؤال اختیاری است. در صورتی که به درستی پاسخ دهید، جایگزین سؤالاتی خواهد شد که نمره کامل نگرفته‌اید: (+ ۸ نمره)**

مفاهیم زیر را تعریف کنید:

Encapsulation : به معنی کپسوله‌سازی است با این مفهوم که: کپسوله سازی همان مفهوم کلی کلاس‌هاست که شما در یک کلاس، همه اجزای لازم را تعریف می‌کنید و در قالب کلاس (مثل یک کپسول که حاوی همه چیز است) ارائه می‌کنید. صفات، متودها، اعضای داده‌ای و ... در قالب یک کلاس قرار می‌گیرند و مثل یک کپسول به کاربر نهایی ارائه می‌شوند.

مثلاً شما یک کلاس به نام Convert دارید که مایکروسافت زحمت تعریف متودها و صفات آن را کشیده است. یعنی مفاهیم مربوط به تبدیلات داده‌ای را در کپسولی به نام Convert به شما ارائه کرده است. آیا شما خبر دارید که متود ToString یا ToChar که متودهای کلاس Convert هستند، چطور عمل می‌کنند؟ خیر. پس یکی از فواید کپسوله سازی، مخفی نگه داشتن عملیاتی است که کاربر نیاز چندانی به دانستن آن‌ها ندارد.

Polymorphism : به معنی «چند ریختی» و یکی از اصلی‌ترین ویژگی‌های زبان‌های شیئ‌گراست. با استفاده از مفهوم «چند ریختی» شما از یک کلاس در اشکال مختلف استفاده می‌کنید. به طور مثال، یک کلاس پایه با نام «Car» (برای ماشین) در نظر می‌گیرید، اما در حین برنامه ممکن است از آن برای تعریف ماشین‌هایی با ریخت‌های مختلف استفاده کنید. ماشین سواری، ماشین تریلی و ... . در حقیقت صفات مشترک در کلاس پایه قرار داده می‌شوند و در اشکال مختلف ظاهر می‌شوند.

Inheritance : مفهوم «وراثت» نیز یکی از زیرمجموعه‌های مفهوم «چند ریختی» است. «چند ریختی» از طریق «وراثت» قابل پیاده سازی است. کلاسی که مربوط به ماشین سواری است، صفات مشترک خود را از کلاس Car که برای تمامی انواع ماشین تعریف شده است، به ارث می‌برد. (مثل صفت مشترکی به نام مدل موتور، شماره پلاک و ... که به هر حال برای تمامی ماشین‌ها وجود دارد). پس کلاس‌ها با استفاده از امکان Inheritance می‌توانند سفات و متودهای عمومی خود را از یکدیگر به ارث ببرند که این امکان از دوباره کاری‌ها و بهینه نبودن برنامه‌ها جلوگیری می‌کند.

Constructor: به معنی «سازنده» است. به متودی گفته می‌شود که هم‌نام کلاس است. به محض ایجاد یک شیئ از یک کلاس حتماً یکی از سازنده‌های کلاس فراخوانی و اجرا می‌شوند. یعنی در حقیقت بدون سازنده، امکان ساخت شیئ از یک کلاس وجود ندارد. سازنده‌ها بیشتر وظیفه مقداردهی اولیه به متغیرهای محلی کلاس را دارند.