

Ad-Soyad : CEVAP ANAHTARI
Numara :
İmza :

19 MAYIS ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
BİL204 - NESNE YÖNELİMLİ TASARIM VE PROGRAMLAMA DERSİ ARA SINAV SORULARI

13/04/2011

Soru 1 (4p): “Nesne Yönelimli Tasarım ve Programlama” kavramı nedir ve sizce neden ortaya çıkmıştır? Kısaca açıklayınız.

Algoritma tasarımına ve programlamaya getirilen modern bir yaklaşımdır. Programlama eylemini; günlük hayattaki kavramlar gibi hiyerarşik, doğal, mantığa yakın ve akılda kalıcı bir formda gerçekleştirmeyi amaçlar. (Benzer anlatımlara da tam puan verilecektir.)

Soru 2 (3p): Aşağıda verilen program hatasız bir biçimde derlenmekte ve çalışmaktadır. Buna göre program çalıştırıldığında ekran çıktısı ne olacaktır?

```
using System;

public class Sinif1
{
    static void Main()
    {
        int[] d1 = new int[2];
        d1[0] = 1;
        Console.WriteLine("{0}", d1[d1[0]] - d1[0]);
        Console.ReadLine();
    }
}
```

-1

Soru 3 (3p): Aşağıda verilen program derlenir mi? Derlenmezse sebebi nedir? Derlenirse çalıştırıldığında ekran çıktısı ne olur?

```
using System;

public class Sinif1
{
    static void Main()
    {
        string s1 = "labaratuvar";
        Console.WriteLine("Yanlis sozcuk : {0}", s1);
        s1[3] = 'o';
        Console.WriteLine("Dogru sozcuk : {0}", s1);
        Console.ReadLine();
    }
}
```

Derlenmez. Karakter dizileri sadece okunabilir.

s1[3] = 'o';

ifadesi hatalıdır.

Soru 4 (3p): Aşağıda verilen **Main** fonksiyonu çalıştırıldığında ekran çıktısı ne olur?

```
static void Main()
{
    int i = 7;
    object a = i;
    object b = i;
    object c = a;
    Console.WriteLine(a == b);
    Console.WriteLine(a == c);
    Console.WriteLine(b == c);
    Console.ReadLine();
}
```

**False
True
False**

Ad-Soyad : CEVAP ANAHTARI
Numara :
İmza :

Soru 5 (4p): Aşağıda verilen **Main** fonksiyonunda “#1” ve “#2” ile işaretlenmiş olan satırlarda iki farklı **dönüşüm** yapılmıştır. Bu dönüşümlere ne ad verilir? Aralarındaki fark nedir? Kısaca açıklayınız.

```
static void Main()
{
    int i1, i2;
    long j1;
    i1 = 77;
    j1 = i1;    // #1
    i2 = (int)j1; // #2
}
```

#1 : kapalı dönüşüm. veri kaybı olmaz.

#2 : açık dönüşüm. veri kaybı olabilir.

Soru 6 (3p): Aşağıdaki programın derleme anı hatası vermesinin nedenini açıklayınız.

```
using System;

public class Sinif1
{
    public static int i1 = 5;
}

public class Sinif2
{
    static void Main()
    {
        Sinif1 nesne1 = new Sinif1();
        Console.WriteLine("{0}", nesne1.i1);
        Console.ReadLine();
    }
}
```

Console.WriteLine("{0}", nesne1.i1);

satırında statik bir sınıf üyesi olan i1' e nesne üzerinden erişilmeye çalışılmıştır, hatalıdır.

Soru 7 (4p): Aşağıdaki program çalıştırıldığında ekran çıktısı ne olur?

```
using System;

public class Sinif1
{
    static void fonk1(ref int arg1, int arg2)
    {
        arg1 = arg2;
    }
    static void Main()
    {
        int i1 = 4;
        int i2 = 8;
        fonk1(ref i1, i2);
        Console.WriteLine("{0}\n{1}", i1, i2);
        Console.ReadLine();
    }
}
```

8
8

Soru 8 (3p): Aşağıdaki programın derleme anı hatası vermesinin nedenini açıklayınız.

```
using System;

public class Sinif1
{
    static int topla(int sayi1, int sayi2)
    {
        return (sayi1 + sayi2);
    }
}

public class Sinif2
{
    static void Main()
    {
        Console.WriteLine("{0}", Sinif1.topla(6, 60));
        Console.ReadLine();
    }
}
```

Console.WriteLine("{0}", Sinif1.topla(6, 60));

satırında, private erişimli olan topla fonksiyonu çağırılmak istenmiştir, hatalıdır.

Ad-Soyad : CEVAP ANAHTARI
Numara :
İmza :

Soru 9 (3p): Aşağıdaki programın derleme anı hatası vermesinin nedenini açıklayınız.

```
using System;

protected class Sinif1
{
    public static void fonk1()
    {
        Console.WriteLine("Sinif1.fonk1");
    }
}

public class Sinif2 : Sinif1
{
    static void Main()
    {
        Sinif1.fonk1();
        Console.ReadLine();
    }
}
```

protected erişim belirleyicisi, sadece içsel sınıflarda kullanılabilir. Sinif1 için kullanımı hatalıdır.

Soru 10 (6p): Aşağıda verilen **Sinif1** sınıfında yer alan **topla1** ve **topla2** fonksiyonlarını inceleyiniz. Nesne yönelimli programlama penceresinden baktığınızda, bu fonksiyonların **statik** erişimli olup-olmamları konusundaki düşünceniz nedir (Hangisini nasıl tanımlamak sizce daha uygun olur?)? Her iki fonksiyon için de düşüncelerinizi ayrı ayrı yazınız.

```
public class Sinif1
{
    public int a;
    public int b;

    public Sinif1(int arg1, int arg2)
    {
        a = arg1;
        b = arg2;
    }

    public ..... int topla1()
    {
        return (a + b);
    }

    public ..... int topla2(int sayi1, int sayi2)
    {
        return (sayi1 + sayi2);
    }
}
```

topla1 metodu, Sinif1' in üyeleri üzerinde işlem yapmaktadır, statik olmamalıdır.

topla2 metodu ise kendisine dışarıdan (argüman olarak) verilen sayılar üzerinde işlem yapmaktadır. Sınıf üyesi olan değişken ve fonksiyonlarla ilişkisi yoktur. Statik olmalıdır.

Soru 11 (4p): Aşağıdaki program derlenerek çalıştırıldığında üretilecek olan ekran çıktısını yanına yazınız.

```
using System;

public class Sinif1
{
    static void Main()
    {
        int i = 0, x = 0, y = 0;

        while (i < 5)
        {
            Console.WriteLine("{0}\t{1}", x++, ++y);
            i++;
        }
        Console.ReadLine();
    }
}
```

0	1
1	2
2	3
3	4
4	5

Ad-Soyad : CEVAP ANAHTARI

Numara :

İmza :

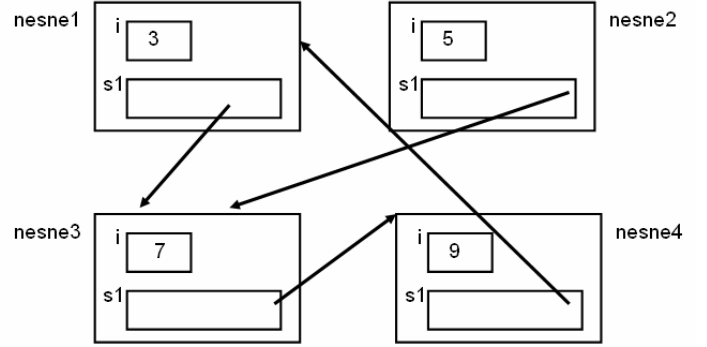
Soru 12 (5p): Aşağıdaki programda bulunan **Main** fonksiyonu çalıştırıldıktan sonra, yaratılan **Sinif1** nesneleri arasındaki ilişkiyi şekil çizerek gösteriniz (4p). Programın üreteceği ekran çıktısını gösteriniz (1p).

```
using System;

public class Sinif1
{
    public int i;
    public Sinif1 s1;

    public Sinif1(int arg1)
    {
        i = arg1;
    }
}

public class Sinif2
{
    static void Main()
    {
        Sinif1 nesne1 = new Sinif1(3);
        Sinif1 nesne2 = new Sinif1(5);
        Sinif1 nesne3 = new Sinif1(7);
        Sinif1 nesne4 = new Sinif1(9);
        nesne3.s1 = nesne4;
        nesne1.s1 = nesne3;
        nesne4.s1 = nesne1;
        nesne2.s1 = nesne4.s1.s1;
        Console.WriteLine("{0}", nesne2.s1.i);
        Console.ReadLine();
    }
}
```



Çıktı:

7

Soru 13 (4p): Aşağıdaki programın üreteceği ekran çıktısını yan tarafına yazınız.

```
using System;

public struct Yapil
{
    public int a;
    public int b;

    public Yapil(int arg1, int arg2)
    {
        a = arg1;
        b = arg2;
    }
}

public class Sinif2
{
    static void Main()
    {
        Yapil ornek1 = new Yapil(3, 5);
        Console.WriteLine("1. durum : {0}, {1}", ornek1.a, ornek1.b);
        Yapil ornek2 = ornek1;
        ornek2.a = 77;
        ornek1.b = 88;
        Console.WriteLine("2. durum : {0}, {1}", ornek1.a, ornek1.b);
        Console.ReadLine();
    }
}
```

1. durum : 3, 5
2. durum : 3, 88

Ad-Soyad : CEVAP ANAHTARI
Numara :
İmza :

Soru 14 (3p): Aşağıdaki program derlenerek çalıştırılmak istendiğinde sonuç ne olur?

```
using System;

public class Sinif1
{
    static void Main()
    {
        int[] dizil = new int[5];
        for (int i = 0; i <= 5; i++)
            dizil[i] = i;
        Console.WriteLine("{0}", dizil[3]);
    }
}
```

for döngüsünde i 5 değerini aldığı zaman:

dizil[5] = 5;

ataması hatalı olur çünkü dizil' in maksimum indeks değeri 4' tür. Program derlenir ancak çalışma anı hatası verir.

Soru 15 (4p): Aşağıdaki sabit listesinde yer alan **Beyaz**, **Sari**, **Kirmizi** ve **Siyah** sabitlerinin sayısal değerlerini yazınız.

```
public enum Renk : int
{
    Beyaz,
    Sari,
    Mavi = 14,
    Kirmizi,
    Yesil = 51,
    Kahverengi = 99,
    Siyah
}
```

Beyaz : 0
Sari : 1
Kirmizi : 15
Siyah : 100

Soru 16 (4p): Aşağıdaki programın üreteceği ekran çıktısını yazınız.

```
using System;

public class Sinif1
{
    private int a;
    private int b;

    public int Ozellik1
    {
        get{ return a;}
        set{ b = value;}
    }
    public Sinif1(int arg1, int arg2)
    {
        a = arg1;
        b = arg2;
    }
}

public class Sinif2
{
    static void Main()
    {
        Sinif1 nesne1 = new Sinif1(42, 25);
        Console.WriteLine("1. durum : {0}", nesne1.Ozellik1);
        nesne1.Ozellik1 = 70;
        Console.WriteLine("2. durum : {0}", nesne1.Ozellik1);
        Console.ReadLine();
    }
}
```

1. durum : 42
2. durum : 42

Ad-Soyad : CEVAP ANAHTARI
Numara :
İmza :

Soru 17 (26p): Aşağıdaki ifadelerin yanlarına “doğru” anlamında “D” ya da “yanlış” anlamında “Y” işareti koyunuz:

- a (2p). Başlarında “**public, internal, protected, private**” erişim belirleyicileri **bulunmayan** tüm sınıf üyeleri (statik olsunlar ya da olmasınlar), **internal** erişimli olarak kabul edilirler. ____ Y
- b (2p). Tıpkı sınıflar gibi isim uzaylarının da nesneleri yaratılabilir ancak isim uzaylarının nesnelerinin yaratılmasında yapıcı metotlar kullanılamaz. ____ Y
- c (2p). Bir sınıfın birden fazla yapıcı metodu bulunabilir. ____ D
- d (2p). Referans türünden olan değişkenler, direkt olarak kendi verilerini bulundururlar. ____ Y
- e (2p). Statik olmayan sınıfların statik üyeleri bulunabilir. ____ D
- f (2p). İçsel sınıflar **private** erişimli olamazlar. ____ Y
- g (2p). Kapalı sınıflar miras bırakabilirler. ____ Y
- h (2p). .NET Framework, işletim sistemi üzerinde doğrudan çalıştırılmayan birtakım uygulamalar için gerekli çalışma ortamını sağlayan bir yazılımdır. ____ D
- i (3p). Makinenin (işlemcinin) anlayabileceği seviyede olan ve sadece "1" ve "0" lardan oluşan dile "ara dil" denir. _ Y
- j (2p). İsim uzayları iç içe bulunabilirler. ____ D
- k (2p). **public** erişimli bir sınıfın tüm üyelerine her koşulda dışarıdan (başka bir sınıftan) erişmek mümkündür. ____ Y
- l (3p). Yapıların (**struct**) örnekleri yaratılabilir ve referanslar vasıtasıyla paylaşılabilirler. ____ Y

Soru 18 (14p): Aşağıda verilen ifadelerde "/" işaretleri ile ayrılmış olan **koyu italik** yazılı ifadelerden doğru olanı bularak elips içerisine alınız. **NOT:** Her soruda tek bir doğru ifade vardır.

- a (2p). **statik / soyut / kapalı** sınıfların nesneleri yaratılabilir.
- b (3p). C# programlama dilinde, herhangi bir isim uzayında yer alan tüm üyelere (sınıf, yapı, arayüz gibi), bulunduğumuz isim uzayı içerisindeymişçesine erişebilmek için ilgili isim uzayının adından önce **using / import / include** anahtar sözcüğünü kullanmak gerekir.
- c (3p). **internal / protected / private / static** erişim belirleyicisi, diğerlerinden farklı olarak erişim iznini değil, erişimin biçimini belirlemede kullanılmaktadır.
- d (3p). Sınıfların nesneleri yaratılırken **object / new / ref / val** anahtar sözcüğü kullanılır.
- e (3p). Bir arayüzden miras alan bir sınıf, bu arayüzde bulunan üyelere **en az birini / tümünü / sadece fonksiyon prototiplerinin tamamını / sadece özelliklerin tamamını** gerçeklemek zorundadır.

-
- Her sorunun cevabını, ilgili sorunun altında bulunan boş alana yazınız. Cevaplarınız mümkün ölçüde; sade, kısa ve anlaşılır olmalıdır.
 - Sınav süresi 135 dakikadır, başarılar...