

C# ile Programlama Ara Sınav - 20 Soru - 60 Dakika

Ad Soyad/ Sınıf Numarası :

00 - Aşağıdaki kod parçasında ile işaretli yere şıklardaki hangi anahtar kelime(keyword) gelmelidir?

```
..... averagePoint = 50.89;
```

```
Console.WriteLine($"Ortalama puan {averagePoint}");
```

- A - int
- B - string
- C - f32
- D - double

01 - Aşağıdaki ifadelerden hangisi doğrudur?

- A - C# yorumlayıcı (**interpreter**) kullanan bir programlama dilidir.
- B - C# ile yazılan kodlar doğrudan makine diline çevrilir.
- C - Nesne yönelimli dillerin önemli karakteristikleri soyutlama (**Abstraction**), kapsülleme (**Encapsulation**), kalıtım (**Inheritance**) ve çok biçimlilik (**Polymorphism**).
- D - C# kod yazım stili küçük/büyük harf duyarlı değildir.

02 - Main fonksiyonu içeren bir konsol/terminal uygulaması oluşturmak için aşağıdaki komutlardan hangisi kullanılır.

- A - dotnet new console --use-program-main -o hello_world
- B - dotnet new console -o hello_world
- C - dotnet create project console
- D - dotnet new web -o hello_world

03 - Aşağıdaki kod neden çalışmaz?

```
namespace HelloWorld;
internal class Program
{
    static void main(string[] args)
    {
        Console.WriteLine("Hello, World!");
    }
}
```

- A - Program sınıfı internal erişim belirleyicisi ile tanımlanmadığı için.
- B - args dizisi main metodunda kullanılmadığı için.
- C - main metodunun adı hatalı yazıldığı için. Main şeklinde yazılmalıdır.
- D - main metodu geriye sonuç dönmediği için.

04 - Aşağıdaki kodda ile boş bırakılan yerlere sırasıyla hangi anahtar kelimeler gelmelidir? (..... ifadelerini ilk kod satırından itibaren değerlendirip cevaplayınız)

```
namespace ShopApp;
internal ..... Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Lütfen ürünün adını giriniz");
        var productName = Console.ReadLine();
        var currentCategory = Category.Electronic;
        Console.WriteLine("Ürünün fiyatını giriniz");
        var unitPrice = Convert.ToDecimal(Console.ReadLine());
        Console.WriteLine(....."Ürünün adı {productName}({unitPrice} TL) , Kategorisi {currentCategory}");
    }
}
..... Category
{
    Electronic,
    Book,
    Game
}
```

- A - struct, \$, enum**
- B - class, #, enum**
- C - class, #, struct**
- D - class, \$, enum**

05 - Aşağıdaki programın çalışma zamanı çıktısı şıklardan hangisidir?

```
namespace ShopApp;
public static class Program {
    static void Main() {
        var products = new Product[3];
        products[0] = new Product{
            Id = 1,
            Title = "Advanced C# Programming",
            Category = Category.Book,
            Price = 40.50M
        };
        products[1] = new Product{
            Id = 2,
            Title = "Rust Programming",
            Category = Category.Book,
            Price = 35.90M
        };
        products[2] = new Product{
            Id = 3,
            Title = "C# ile Programlamaya Giriş",
            Category = Category.Book,
            Price = 25.55M
        };
        for (int i = 0; i < products.Length; i++){
            Console.WriteLine($"Fiyat : {products[i].Price}");
            products[i].DecreasePrice(0.5M);
            Console.WriteLine($"İndirimli fiyat : {products[i].Price}");
        }
    }
}
```

```

}
class Product {
    public int Id { get; set; }
    public string Title { get; set; } = string.Empty;
    public Category Category { get; set; }
    public decimal Price { get; set; }
    public void DecreasePrice(decimal rate)
    {
        Price -= Price * rate;
    }
}
enum Category {
    Electronic,
    Book,
    Game
}

```

A -

Fiyat : 40,50
 İndirimli fiyat : 20,250
 Fiyat : 35,90
 İndirimli fiyat : 17,950
 Fiyat : 25,55
 İndirimli fiyat : 12,775

B -

Fiyat : 40.50
 İndirimli fiyat : 20.250
 Fiyat : 35.90
 İndirimli fiyat : 17.950
 Fiyat : 25.55
 İndirimli fiyat : 12775

C - Kod tip dönüştürme hatası (Type Cast Exception) vererek kırılır.

D - Satırlar arasında fazladan boşluk oluştuğuna dikkat edin!

Fiyat : 40.50

 İndirimli fiyat : 20.250

 Fiyat : 35.90

 İndirimli fiyat : 17.950

 Fiyat : 25.55

 İndirimli fiyat : 12775

06 - Aşağıdaki program çalıştığında oluşabilecek olası hata verilen şıklardan hangisi olabilir?

```

namespace ShopApp;
public static class Program
{
    static void Main(){
        var product = new Product {
            Id = 1,
            Title = "Filips 4K Monitor",
            Category = Category.Electronic

```

```

    };
    Console.WriteLine($"{product.Id}-{product.Title} {product.Price} $");
    Console.WriteLine("Ürün fiyatı?");
    product.Price = Convert.ToDecimal(Console.ReadLine());
    Console.WriteLine($"{product.Id}-{product.Title} {product.Price} $");
}

}

class Product{
    public int Id { get; set; }
    public string Title { get; set; } = string.Empty;
    public Category Category { get; set; }
    public decimal Price { get; set; }
    public void DecreasePrice(decimal rate) {
        Price -= Price * rate;
    }
}

enum Category {
    Electronic,
    Book,
    Game
}

```

A - Product sınıfına ait nesne örneği (**object instance**) oluşturulurken Price bilgisi verilmediğinden program derlenmeyecek ve bu sebeple çalışmayacaktır.

B - Ürün fiyatı bilgisi decimal olarak dönüştürülemeyecek bir ifade olarak girilebilir. Bu

System.FormatException istisnasının fırlatılmasına neden olur.

C - **Convert.ToDecimal** metoduna parametre olarak **Console.ReadLine()** metodu atanamaz. Bu nedenle program derleme zamanı hatası vererek sonlanır.

D - Ürün fiyatı bilgisi decimal olarak dönüştürülemeyecek bir ifade olarak girilebilir. Bu

System.ArgumentException istisnasının fırlatılmasına neden olur.

07 - Geriye değer döndürmeyen bir metot aşağıdaki hangi anahtar kelime ile tanımlanır.

A - void

B - C# dilinde tüm fonksiyonlar bir dönüş tipine sahip olmalıdır.

C - null

D - nothing

08 - Aşağıdaki programın çalışma zamanı çıktısı şıklardan hangisidir ?

```

class Program {
    static void Main() {
        for (int i = 0; i < 10; i++) {
            Console.Write(Fibonacci.Get(i) + ", ");
        }
    }
}

public static class Fibonacci {
    public static int Get(int n) {
        if (n <= 1)
            return n;
        else
            return Get(n - 1) + Get(n - 2);
    }
}

```

```
}  
}
```

A - 1, 1, 2, 3, 5, 8, 13, 21,

B - 1, 1, 2, 3, 5, 8, 13, 21, 34,

C - 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

D - Program **IndexOutOfRangeException** vererek kırılır ve sonlanır.

09 - Aşağıdaki program derleme zamanında hata almaktadır. Bunun sebebi şıklardan hangisidir?

```
namespace ShopApp;  
public static class Program  
{  
    static void Main() {  
        var products = new List<Product>() {  
            new Product {  
                Id = 1,  
                Title = "Advanced C# Programming",  
                Category = Category.Book,  
                Price = 40.50M  
            },  
            new Product {  
                Id = 2,  
                Title = "Rust Programming",  
                Category = Category.Book,  
                Price = 35.90M  
            },  
            new Product {  
                Id = 3,  
                Title = "C# ile Programlamaya Giriş",  
                Category = Category.Book,  
                Price = 25.55M  
            }  
        };  
        foreach (var product in products) {  
            Console.WriteLine($"Fiyat : {product.Price}");  
            product.DecreasePrice(0.5);  
            Console.WriteLine($"İndirimli fiyat : {product.Price}");  
        }  
    }  
}  
class Product {  
    public int Id { get; set; }  
    public string Title { get; set; } = string.Empty;  
    public Category Category { get; set; }  
    public decimal Price { get; set; }  
    public void DecreasePrice(decimal rate) {  
        Price -= Price * rate;  
    }  
}  
enum Category{  
    Electronic,  
    Book,  
    Game  
}
```

A - **List** koleksiyonu oluşturulurken kapasite değeri verilmediği için derleme zamanı hatası alınır.

B - **DecreasePrice** metod çağrısında gönderilen 0.5 değeri decimal türe çevrilemediği için derleme zamanı hatası alınır.

C - Kod başarılı bir şekilde derlenir, hata almadan çalışır.

D - **Product** ve **enum** sınıfları bir **namespace** içerisinde olmadığından derleme zamanı hatası alınır.

10 - Stok Dışı Ürünler

Geliştirilmekte olan program CSV uzantılı bir dosyadan ürün bilgilerini okumakta ve bunları Product sınıfına ait nesne örneklerinden oluşan bir dizide tutmaktadır. GetOutOfStock isimli bir fonksiyon ile bu ürünlerden stok seviyesi belli bir değerin altında olanların bulunması ve yeni bir Product nesne dizisi olarak geriye döndürülmesi istenmektedir. Aşağıdaki şıklardan hangisi bu fonksiyon olabilir? (*Generic koleksiyon, LINQ sorgularını destekleyen genişletme metodları/extension methods olmadığıni düşünerek cevaplamaya çalışınız*)

A -

```
public Product[] GetOutOfStock(Product[] products, int stockLevel) {
    var outOfStockCount = 0;
    foreach (var product in products) {
        if (product.StockLevel <= stockLevel)
        {
            outOfStockCount++;
        }
    }
    if (outOfStockCount > 0) {
        var result = new Product[outOfStockCount];
        var counter = 0;
        for (int i = 0; i < products.Length; i++) {
            if (products[i].StockLevel <= stockLevel) {
                result[counter] = products[i];
                counter++;
            }
        }
        return result;
    }
}
```

B -

```
public Product[] GetOutOfStock(Product[] products, int stockLevel) {
    var outOfStockCount = 0;
    foreach (var product in products) {
        if (product.StockLevel <= stockLevel)
        {
            outOfStockCount++;
        }
    }
    if (outOfStockCount > 0) {
        var result = new Product[outOfStockCount];
        for (int i = 0; i < products.Length; i++) {
            if (products[i].StockLevel <= stockLevel)
            {
                result[i] = products[i];
            }
        }
        return result;
    } else {
        return [];
    }
}
```

C -

```

public Product[] GetOutofStock(Product[] products, int stockLevel) {
    var result = new Product[products.Length];
    var counter = 0;
    for (int i = 0; i < products.Length; i++) {
        if (products[i].StockLevel <= stockLevel) {
            result[counter] = products[i];
            counter++;
        }
    }
    return result;
}

```

D -

```

public Product[] GetOutofStock(Product[] products, int stockLevel) {
    var outOfStockCount = 0;
    foreach (var product in products) {
        if (product.StockLevel <= stockLevel) {
            outOfStockCount++;
        }
    }
    var result = new Product[outOfStockCount];
    var counter = 0;
    for (int i = 0; i < products.Length; i++) {
        if (products[i].StockLevel <= stockLevel) {
            result[counter] = products[i];
            counter++;
        }
    }
    return result;
}

```

11 - Fizz Buzz FizzBuzz Oyunu

Programlamada popüler kod katalarından birisi de FizzBuzz serisidir. Buna göre 1 ile belli bir aralıktaki sayıların 3 ve 5 ile tam bölünebilme durumlarına istinaden ekrana çıktı basılır. 3 ile tam bölünen sayılar için Fizz, 5 ile tam bölünen sayılar için Buzz, hem 3 hem de 5 ile tam bölünebilen sayılar içinse FizzBuzz yazdırılır. Buna göre şıklardan hangisi aşağıdaki ekran görüntüsünde yer alan kodu üretir.

```

1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
Fizz
22
23
Fizz
Buzz

```

A -

```
public void FizzBuzz(uint maxValue) {
    for (int i = 1; i <= maxValue; i++) {
        if (i % 3 == 0 && i % 5 == 0) {
            Console.WriteLine("FizzBuzz");
        } else if (i % 3 == 0) {
            Console.WriteLine("Fizz");
        } else if (i % 5 == 0) {
            Console.WriteLine("Buzz");
        } else {
            Console.WriteLine(i);
        }
    }
}
```

B -

```
public void FizzBuzz(uint maxValue) {
    for (int i = 0; i < maxValue; i++) {
        if (i % 3 == 0 && i % 5 == 0) {
            Console.WriteLine("FizzBuzz");
        } else if (i % 3 == 0) {
            Console.WriteLine("Fizz");
        } else if (i % 5 == 0) {
            Console.WriteLine("Buzz");
        } else {
            Console.WriteLine(i);
        }
    }
}
```

C -

```
public void FizzBuzz(uint maxValue) {
    for (int i = 1; i <= maxValue; i++) {
        if (i % 3 == 0 && i % 5 == 0) {
            Console.WriteLine("FizzBuzz");
        } else if (i % 3 == 0) {
            Console.WriteLine("Fizz");
        } else if (i % 5 == 0) {
            Console.WriteLine("Buzz");
        }
    }
}
```

D -

```
public void FizzBuzz(uint maxValue) {
    for (int i = 1; i <= maxValue; i++) {
        if (i % 3 == 0 || i % 5 == 0) {
            Console.WriteLine("FizzBuzz");
        } else if (i % 3 == 0) {
            Console.WriteLine("Fizz");
        } else if (i % 5 == 0) {
            Console.WriteLine("Buzz");
        } else {
            Console.WriteLine(i);
        }
    }
}
```


12 - Pell Sayı Serisi

Hesaplanan değerleri itibariyle altın oranla yakın ilişkisi olan, hızlı oranlarda büyüyen, kriptografi gibi alanlarda da kullanılan Pell serisine ait sayıların hesaplanmasında ilk iki terim 0 ve 1 olmak suretiyle $P(n) = 2 * P(n-1) + P(n-2)$ formülü kullanılır. Buna göre aşağıdaki kod parçası şıklardan hangi seriyi üretir.

```
int[] series = new int[10];
series[0] = 0;
series[1] = 1;
for (int i = 2; i < series.Length; i++) {
    series[i] = 2 * series[i - 1] + series[i - 2];
}
Console.WriteLine(string.Join(", ", series));
```

A - 0, 1, 3, 8, 21, 55, 144, 377, 987, 2584

B - 0, 1, 2, 5, 12, 29, 70, 169, 408, 985

C - 0, 1, 4, 10, 22, 46, 94, 190, 382, 766

D - Program **IndexOutOfRangeException** fırlatarak sonlandığından sayı serisi üretilmeyecektir.

13 - Birim Testler(*Unit Test*) için aşağıda belirtilen ifadelerden hangisi yanlıştır.

A - Belli işlevsellikleri belli kabul kriterlerine göre test etmek için birim testlerden yararlanılır.

B - xunit, .Net tarafından kullanılan Unit Test şablonlarından birisidir.

C - Birim test metodları Fact veya Test gibi niteliklerle(**Attribute**) işaretlenir ve bu sayede çalışma zamanı tarafından test metodu oldukları anlaşılır.

D - Birim test, Code Coverage değerlerinin yükselmesine ve güvenilir kod çıktılarının oluşturulmasına katkı sağlar.

E - Dotnet projeleri birim testleri yazılmadan çalışmazlar.

14 - Donkey Kong ve 80ler

Arkadaşlarımızın sevdiği oyunların bilgilerini saklayacağınız bir program geliştiriyorsunuz. Her oyunun belli başlı karakteristik özellikler var. Bu amaçla Game isimli bir sınıf geliştiriyoruz ve ilk gün tasarımı aşağıdaki kod parçasında görüldüğü gibi yapıyoruz.

```
public class Game {
    public string Title { get; set; }
    public int Year { get; set; }
    public double Rating { get; set; }
    public Game(string title, int year, double rating) {
        Title = title;
        Year = year;
        Rating = rating;
    }
    public override string ToString() {
        return $"{Title} ({Year}) - Rating: {Rating}";
    }
}
```

Game nesnesinin kullanımını görmek içinde onu örnekleyip ekrana yazdırmak istiyoruz. Bu amaçla 1984 yılında piyasaya çıkan ve hali hazırda kullanıcılardan ortalama 4.8 puan almış olan Donkey Kong isimli oyunu gözümüze kestiriyoruz. Bu bilgiler ışığında aşağıdaki kod parçalarından hangisi doğrudur?

A -

```
Game myGame = new Game();
myGame.Title = "Donkey Kong!";
myGame.Year = 1984;
myGame.Rating = 4.8;
Console.WriteLine(myGame.ToString());
```

B -

```
Game myGame;
myGame.Title = "Donkey Kong";
myGame.Year = 1984;
myGame.Rating = 4.8;
Console.WriteLine(myGame.ToString());
```

C -

```
Game myGame = new Game("Donkey Kong", 1984, 4.8);
Console.WriteLine(myGame);
```

D -

```
Game myGame = new Game();
myGame.Title = "Donkey Kong";
myGame.Year = 1984;
myGame.Rating = 4.8;
Console.WriteLine(myGame);
```

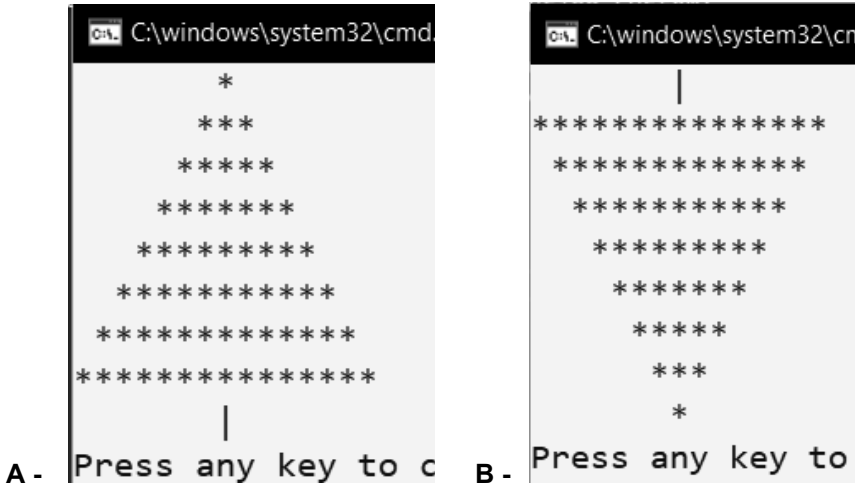
15 - Aşağıdakilerden hangisi bir sınıfın sahip olabileceği üyelerdendir.

A - Yapıcı metotlar (Constructors)**B -** Alanlar (Fields)**C -** Özellikler (Properties)**D -** Metodlar (Methods)**E -** Hepsi

16 - Yıldızlı sürprizler.

```
using System;
namespace Games {
    public static class Program {
        static void Main() {
            Draw(8);
        }
        static void Draw(int level) {
            for (int i = 0; i < level; i++) {
                for (int j = 0; j < level - i - 1; j++) {
                    Console.Write(" ");
                }
                for (int k = 0; k < 2 * i + 1; k++) {
                    Console.Write("*");
                }
                Console.WriteLine();
            }
            for (int i = 0; i < level - 1; i++) {
                Console.Write(" ");
            }
            Console.WriteLine("|");
        }
    }
}
```

Sizce yukarıdaki kodun çıktısı aşağıdakilerden hangisidir?



17 - Geliştirmekte olduğumuz projenin solution ağacı içerisinde yer alan birim testleri çalıştırmak için aşağıdaki terminal komutlarından hangisi kullanılır?

- A - dotnet start
- B - dotnet check
- C - dotnet build
- D - dotnet test

18 - Aşağıdaki ifadelerden hangisi yanlıştır?

- A - DotNet dünyasında her şey bir nesnedir. Bir başka deyişle tipler ata sınıf olan Object'ten türer. Bu nedenle her primitive tipin ardından ToString metoduna erişilebilir, nitekim bu metod Object sınıfında tanımlanmış olup türetme yoluyla alt türlerin erişimine açıktır.
- B - Aynı konuya dair türler, alt türler ve bu türlerin elemanları genellikle bir namespace başlığı altında toplanır. Örneğin matematiksel fonksiyonlara ait tüm türler System.Math isim alanı altında toplanmıştır.
- C - Main, özel bir fonksiyondur ve programın giriş noktasını (Entry Point) ifade etmektedir. Çalıştırılabilir tüm .Net uygulamalarında Main fonksiyonu bulunur. Main fonksiyonu dışarıdan parametre de alabilir. Buna göre örneğin bir terminal uygulaması komut satırı parametreleri ile başlatılabilir.
- D - .Net dünyasında tipler iki ana kategoriye ayrılır. Değer türlü (value types) olanlar ve referans türlü olanlar. Referans türlü (Reference Types) tipler belleğin Stack bölgesinde tutulurlar ve Last In First Out (LIFO) ilkesine göre yerleştirilip kaldırılırlar.

19 - Generic List Kullanımı

Aşağıdaki kod parçalarından hangisinde Category nesne örneklerinden oluşan categories isimli koleksiyon doğru şekilde oluşturulabilmiştir.

A -

```
using Warehouse;
class Program {
    static void Main() {
        var categories = new List<Category>{
            new Category{Id=1,Title="Book"},
            new Category{Id=2,Title="Electronics"},
        };
    }
}
```

```

    }
}
namespace Warehouse {
    public class Category {
        public int Id { get; set; }
        public string Title { get; set; } = string.Empty;
    }
}

```

B -

```

using Warehouse;
class Program {
    static void Main() {
        var categories = new List{
            new Category{Id=1,Title="Book"},
            new Category{Id=2,Title="Electronics"},
        };
    }
}
namespace Warehouse {
    public class Category {
        public int Id { get; set; }
        public string Title { get; set; } = string.Empty;
    }
}

```

C -

```

using Warehouse;
class Program {
    static void Main() {
        categories = new List<Category>{
            new Category{Id=1,Title="Book"},
            new Category{Id=2,Title="Electronics"},
        };
    }
}
namespace Warehouse {
    public class Category {
        public int Id { get; set; }
        public string Title { get; set; } = string.Empty;
    }
}

```

D -

```

using Warehouse;
class Program {
    static void Main() {
        var categories = new List<Category>{
            new {Id=1,Title="Book"},
            new {Id=2,Title="Electronics"},
        };
    }
}
namespace Warehouse {
    public class Category {
        public int Id { get; set; }
        public string Title { get; set; } = string.Empty;
    }
}

```