

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

متغیر های Boolean

مدرس : مهندس افشین رفوآ

متغیر های Boolean

مقدمه

هنگام تعامل با کامپیوتر، کاربر ممکن است مقادیری را به برنامه ی در حال اجرا عرضه کند که برخی از این مقادیر معتبر هستند ولی برخی دیگر نیاز به اصلاحاتی دارند یا توسط رایانه رد می شوند. برای جلوگیری از این رخداد مقادیر باید مرتب مورد بررسی قرار بگیرند. اعتبار یک مقدار در رابطه با نوعش بررسی می شود. برای مثال، یک شرط می تواند درست باشد یا یک عدد مساوی عدد دیگری باشد.

برای اعتبار سنجی مقادیر، زبان **C#** یک سری نشانه عرضه می کند که از آن ها به عنوان عملگرهای **Boolean** یاد می شود.

تعریف متغیر های Boolean

1. **Microsoft Visual Studio** را اجرا کنید.
2. برای ایجاد برنامه ی جدید، به فهرست اصلی مراجعه کرده، روی **File -> New Project...** کلیک کنید.
3. در فهرست میانی روی گزینه ی **Empty Project** کلیک کنید.
4. اسم مورد نظر را به **NationalBank2** تغییر دهید.
5. برای ایجاد فایل جدید، در پنجره ی **Solution Explorer**، راست کلیک کرده سپس **NationalBank2 -> Add -> New Item...**
6. گزینه ی **Code File** را از لیست میانی انتخاب کنید.
7. اسم فایل را به **Employee** تغییر دهید.
8. **Add** را کلیک کنید.
9. فایل **Employee.cs** را به شکل زیر اصلاح کنید.

```
public class Employee
{
    public string EmployeeNumber;
```

```

    public string FirstName;
    public string LastName;
    public double HourlySalary;
    public Employee(string emplNbr = "00-000", string
fName = "John",
                                string lName = "Doe", double salary
= 0.00)
    {
        EmployeeNumber = emplNbr;
        FirstName = fName;
        LastName = lName;
        HourlySalary = salary;
    }
}

```

10. برای ایجاد فایل جدید، در پنجره ی **Solution Explorer**، راست کلیک کرده سپس : **NationalBank2**

.-> Add -> New Item...

11. توجه داشته باشید که در لیست میانی گزینه ی **Code File** انتخاب شده باشد. اسم آن را به

Management تغییر دهید و کلید **Enter** را بزنید.

12. فایل را به صورت زیر اصلاح کنید.

```

using System;
public class Management
{
    private static Employee HireEmployee()
    {
        Employee empl = new Employee();
        Console.Title = "National Bank";
        Console.WriteLine("=====");
        Console.WriteLine("== National Bank ==");
        Console.WriteLine("To hire a new employee, enter the
following information");
        Console.WriteLine("-----");
        Console.Write("Employee #: ");
        empl.EmployeeNumber = Console.ReadLine();
        Console.Write("First Name: ");
        empl.FirstName = Console.ReadLine();
        Console.Write("Last Name: ");
        empl.LastName = Console.ReadLine();
        Console.Write("Hourly Salary: ");
        empl.HourlySalary =
double.Parse(Console.ReadLine());
    }
}

```

```

        Console.WriteLine("=====");
        return empl;
    }
    private static void ShowEmployeeRecord(Employee
empl)
    {
        Console.Title = "National Bank";
        Console.WriteLine("=====");
        Console.WriteLine("== National Bank ==");
        Console.WriteLine("    Employee Record");
        Console.WriteLine("-----");
        Console.WriteLine("Employee #:    {0}",
empl.EmployeeNumber);
        Console.WriteLine("First Name:    {0}",
empl.FirstName);
        Console.WriteLine("Last Name:     {0}",
empl.LastName);
        Console.WriteLine("Hourly Salary: {0}",
empl.HourlySalary);
        Console.WriteLine("=====");
    }

    public static int Main()
    {
        Employee clerk = null ;
        clerk = HireEmployee();
        Console.Clear();
        ShowEmployeeRecord(clerk);
        Console.ReadKey();
        return 0;
    }
}

```

13. برای اجرای برنامه، به فهرست اصلی مراجعه کرده، سپس روی **Debug -> Start Debugging** کلیک کنید.

14. اطلاعات لازم را به صورت زیر اصلاح کنید.

Employee #	88-602
First Name	Joan
Last Name	Gergman
Hourly Salary	16.85

15. کلید **Enter** را فشار دهید.

```

=====
==== National Bank ====
      Employee Record
-----
Employee #:      88-602
First Name:     Joan
Last Name:      Bergman
Hourly Salary:  16.85
=====

```

16. حال **Enter** را زده تا پنجره ی **DOS** بسته شود.

تعریف متغیر Boolean

به متغیری **Boolean** گفته می شود که مقدار آن از دو حالت **true** یا **false** خارج نباشد. برای تعریف متغیر **Boolean**، می توان هر یک از دو کلیدواژه ی **bool** یا **var** را به کار برد. به مثال زیر توجه کنید.

```

using System;
public class Exercise
{
    static int Main()
    {
        bool drinkingUnderAge;
        return 0;
    }
}

```

می توان از نوع داده ی **Boolean** برای تعریف متغیر **Boolean** استفاده کرد. نوع داده ی **Boolean** بخشی از فضای نام **System** می باشد.

مثال

```

using System;
public class Exercise
{
    static int Main()
    {
        bool drinkingUnderAge;
        return 0;
    }
}
using System;
public class Exercise

```

```

{
    static int Main()
    {
        bool drinkingUnderAge;
        return 0;
    }
}

```

پس از تعریف متغیر، باید آن را با مقدار **true** یا **false** مقداردهی اولیه کرد. در حقیقت، چنانچه متغیر نام برده را به عنوان **var** تعریف کنید، چاره ای جز مقداردهی اولیه ی آن ندارید. توجه خود را به مثال زیر جلب کنید.

```

using System;
public class Exercise
{
    static int Main()
    {
        var drinkingUnderAge = true;
        return 0;
    }
}

```

به منظور نمایش مقدار متغیر **Boolean** در کنسول، می توانید اسم آن را در پرانتزهای متد **Write()** یا **WriteLine()** کلاس **console** تایپ کنید. مثال زیر را در نظر بگیرید.

```

using System;
public class Exercise
{
    static int Main()
    {
        var drinkingUnderAge = true;
        Console.WriteLine("Drinking Under Age: {0}",
drinkingUnderAge);
        return 0;
    }
}

```

نتیجه

```

Drinking Under Age: True
Press any key to continue...

```

می توانید هر زمان که لازم می دانید، مقدار متغیر **Boolean** را با تخصیص مقادیر **true** یا **false** تغییر دهید.

مثال

نتیجه ی زیر حاصل می گردد.

```
Drinking Under Age: True
Drinking Under Age: False
Press any key to continue...
```

بازیابی مقدار متغیر **Boolean**

می توان مقدار یک متغیر **Boolean** را از کاربر در خواست کرد. در این مورد، لازم است کاربر **True** (یا **true**) یا **False** (یا **false**) را تایپ کند. برای بازیابی آن باید از متدهای **Read()** یا **ReadLine()** کلاس **Console** کمک گرفت.

مثال

```
using System;
public class Exercise
{
    static int Main()
    {
        var drivingUnderAge = false;
        Console.WriteLine("Were you driving under
age?");
        Console.Write("If Yes, enter True. Otherwise
enter False: ");
        drivingUnderAge =
bool.Parse(Console.ReadLine());
        Console.WriteLine("\nWas Driving Under Age:
{0}\n", drivingUnderAge);
        return 0;
    }
}
```

مثال

```
Were you driving under age?
If Yes, enter True. Otherwise enter False: true

Was Driving Under Age: True
```

Press any key to **continue**...

ایجاد فیلد Boolean

مثل دیگر نوع متغیر هایی که در مباحث پیشین با آن برخورد داشتیم، می توان متغیر **Boolean** را به فیلد یک کلاس تبدیل کرد. متغیر مذکور را مثل هر نوع متغیر دیگر، البته با کلید واژه **bool** یا نوع داده ی **Boolean** تعریف می کنید.

مثال

```
public class House
{
    char typeOfHome;
    int beds;
    float baths;
    byte stories;
    bool hasCarGarage;
    int yearBuilt;
    double value;
}
```

هنگام مقداردهی اولیه ی شی ای که متغیر **Boolean** یک عضو آن محسوب می شود، کافی است **true** یا **false** به متغیر مربوط اختصاص دهید. به همین نحو، می توانید مقدار متغیر عضو **Boolean** را با دسترسی به آن بازیابی یا چک کنید. نمونه های آن را در زیر مشاهده می کنید.

```
using System;
public class House
{
    public char typeOfHome;
    public int beds;
    public float baths;
    public byte stories;
    public bool hasCarGarage;
    public int yearBuilt;
    public double value;
}
public class Program
{
    static int Main()
    {
        var Condominium = new
```



```

        {
            hasCarGarage = false,
            yearBuilt = 2002,
            baths = 1.5F,
            stories = 18,
            value = 155825,
            beds = 2,
            typeOfHome = 'C'
        };
        Console.WriteLine("=//= Altair Realtors =//=");
        Console.WriteLine("=== Property Listing ===");
        Console.WriteLine("Type of Home:      {0}",
Condominium.typeOfHome);
        Console.WriteLine("Number of Bedrooms:  {0}",
Condominium.beds);
        Console.WriteLine("Number of Bathrooms: {0}",
Condominium.baths);
        Console.WriteLine("Number of Stories:   {0}",
Condominium.stories);
        Console.WriteLine("Year Built:          {0}",
Condominium.yearBuilt);
        Console.WriteLine("Has Car Garage:      {0}",
Condominium.hasCarGarage);
        Console.WriteLine("Monetary Value:     {0}\n",
Condominium.value);
        return 0;
    }
}

```

نتیجه ی زیر به دست می آید.

```

=//= Altair Realtors =//=
=== Property Listing ===
Type of Home:      C
Number of Bedrooms:  2
Number of Bathrooms:  1.5
Number of Stories:   18
Year Built:         2002
Has Car Garage:     False
Monetary Value:     155825

Press any key to continue...

```

آرگومان های Boolean

درست مثل پارامترهای انواع دیگر، می توان آرگومان (از) نوع **bool** یا **Boolean** به متد ارسال کرد. چنین آرگومانی می تواند تنها دو مقدار **true** یا **false** را داشته باشد.

www.tahilidadeh.com