بسم اللّه الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

تبدیل مقادیر

مدرس : مهندس افشین رفوآ

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 561 - واحد 7
88146323 - 88446780 - 88146330

تبدیل مقادیر

تبدیل ضمنی

با نحوه ی تعریف نوع های رشته، ممیز شناور و صحیح (integral) آشنا شدیم. طریقه ی مقداردهی اولیه ی آن ها نیز تشریح شد. در صورت داشتن برنامه ای که متغیرهای مختلفی در آن به کار گرفته شده، می توان مقدار یک متغیر را به (مقدار) متغیر دیگری تبدیل کرد. در درس 1 با مقدار حافظه ی مورد نیاز برای ذخیره سازی متغیر هر نوع داده آشنا شدیم.

Data Type	Name	Memory Size
byte	Byte	8 bits
sbyte	Signed Byte (بایت با علامت)	8 bits
char	Character	16 bits
short	Small Integer (عدد صحیح کوچک)	16 bits
ushort	Unsigned Small Integer	16 bits
int	Signed Integer(عدد صحیح علامت دار)	32 bits
uint	Unsigned Integer(عدد صحیح بدون علامت)	32 bits
float	Single-Precision Floating-Point Number (عدد با ممیز شناور با تنها یک رقم اعشار)	32 bits
double	Double-Precision Floating-Point Number (عدد با ممیز شناور با دو رقم اعشار)	64 bits
long	Signed Long Integer (عدد صحیح بزرگ علامت دار)	64 bits
ulong	Unsigned Long Integer	64 bits
decimal	Extended Precision Floating-Point Number(عدد ممیز اعشار با رقم اعشار	128 bits

	طولانی)	

همان طور که در جدول بالا مشاهده می کنید، مقدار یک متغیر Byte می تواند در مقدار حافظه ی تخصیص یافته برای متغیر int جای گیرد و سرانجام مقدار متغیر int خود در متغیر long قابلیت ذخیره شدن را دارد. بر این اساس، می توان مقدار یک Byte را به متغیر int اختصاص داد و یا متغیر int را به متغیر long. همچنین به این خاطر که حافظه ی رزرو شده برای متغیر int بیشتر از مقدار حافظه ی تخصیص یافته به متغیر double می باشد، می توان متغیر اولی را به متغیر دومی اختصاص داد. مثال زیر را در نظر بگیرید.

```
using System;
class Program
{
            static int Main()
            {
                int iNumber = 2445;
                double dNumber = iNumber;
                Console.WriteLine("Number = {0}", iNumber);
                Console.WriteLine("Number = {0}\n", dNumber);
                return 0;
            }
        }
```

نتیجه

```
Number = 2445
        Number = 2445
        Press any key to continue...
```

از این ویژگی با عنوان تبدیل ضمنی یاد می شود.

تبدیل صریح

به دلیل مقررات و الزامات حافظه، عکس فرایند تبدیل ضمنی امکان پذیر نمی باشد. به این خاطر که حافظه ی تخصیص یافته برای متغیر short کمتر از مقدار حافظه است که برای متغیر int رزرو شده، نمی توان مقدار متغیر int را به short اختصاص داد. برنامه ی زیر را در نظر بگیرید.

```
using System;
class Program
```

```
{
            static int Main()
            {
                int iNumber   = 168;
                short sNumber = iNumber;
                Console.WriteLine("Number = {0}", iNumber);
                Console.WriteLine("Number = {0}\n", sNumber);
                return 0;
            }
        }
```

با اخطار زیر مواجه می شوید.

```
Cannot implicitly convert type 'int' to 'short'.
```

تبدیل (نوع) مقدار عبارتند از تبدیل یک مقدار به مقداری با نوع متفاوت. برای مثال، مقداری integer داریم و می خواهیم آن مقدار مورد نظر را در عبارتی که مقدار short می طلبد داشته باشیم. از این فرایند با عنوان تبدیل صریح نیز یاد می شود.

به منظور تبدیل یک مقدار یا متغیر، نوع داده ی دلخواه را پیش از مقدار مورد نظر داخل پرانتز تایپ می کنید.

مثال

```
using System;
class Program
{
            static int Main()
            {
                int iNumber   = 168;
                short sNumber = (short)iNumber;
                Console.WriteLine("Number = {0}", iNumber);
                Console.WriteLine("Number = {0}\n", sNumber);
                return 0;
            }
        }
```

نتیجه ی زیر حاصل می گردد.

```
Number = 168
        Number = 168
        Press any key to continue...
```

هنگام اجرای تبدیل صریح، توجه دقیق به نوع تبدیلی (نوعی که می خواهید تبدیل کنید) از اهمیت بسیار بالایی برخوردار است. اگر می خواهید مقدار integer به متغیر short تخصیص یابد، مقدار مورد نظر باید در 16 بیت جای گیرد، به این معنا که باید در محدوده (طیف) عددی -32768 تا 32767 قرار گیرد. هر مقدار دیگری فراتر از این رنج نتیجه ی غیرقابل پیش بینی بدست خواهد داد.

مثال

```
using System;
          class Program
{
              static int Main()
              {
                  int iNumber   = 680044;
                  short sNumber = (short)iNumber;

                  Console.WriteLine("Number = {0}", iNumber);
                  Console.WriteLine("Number = {0}\n", sNumber);
                  return 0;
              }
          }
```

حاصل زیر به دست می آید.

```
Number = 680044
         Number = 24684
         Press any key to continue...
```

همان طور که مشاهده می کنید نتیجه ی نادرست حاصل شده.

کلاس Convert

در درس 13 و 32 توضیح داده خواهد شد که هر نوع داده ی زبان C#، که خود از ساختار .NET Framework اقتباس شده، مجهز به متد ToString() است که قابلیت تبدیل مقدار نوع داده به نوع String را فراهم می کند. البته به امکان تبدیل مقداری از یک نوع اولیه (primitive type) به نوع اولیه ی دیگر پرداخته نشد. برای پشتیبانی از این امکان (تبدیل مقداری از یک نوع به نوع دیگر) .NET Framework کلاسی به نام

Convert را ارائه می دهد. کلاس مذکور مجهز به چندین متد ایستا است که در این مبحث امکان تشریح همه ی آن ها وجود ندارد.

به خاطر داشته باشید که هر نوع داده ی اولیه ی C# از .NET Framework برگرفته شده است.

C# Data Type	Name	.NET Framework Structure
bool	Bollean	Boolean
byte	Byte	Byte
sbyte	Signed Byte	SByte
char	Character	Char
short	Small Integer	Int16
ushort	Unsigned Small Integer	UInt16
int	Integer	Int32
uint	Unsigned Integer	UInt32
long	Long Integer	Int64
ulong	Unsigned Long Integer	UInt64
float	Single-Precision Floating-Point	Single
double	Double-Precision Floating-Point	Double
decimal	Extended Precision Floating-Point Number	Decimal
No Explicit Type	Date/Time Value	DateTime
string	String	String

برای تطبیق دادن کلاس Convert به نوع داده های C#، کلاس مزبور مجهز به متد ایستایی است که نام آن با To آغاز شده و با اسم ساختارش پایان می یابد و به عنوان آرگومان، آن نوعی را می گیرد .NET Framework.

که باید تبدیل شود. بر این اساس، به منظور تبدیل عدد دهدهی از نوع double به عددی از نوع int، می توان متد ToInt32() را فراخواند سپس متغیر double را به عنوان آرگومان ارسال کرد. دستور نحوی آن به صورت زیر است.

```
public static int ToInt32(double value);
```

مثال

```
using System;
class Program
{
            static int Main()
            {
                double dNumber = 34987.68D;
                int iNumber = Convert.ToInt32(dNumber);
                Console.WriteLine("Number: {0}", dNumber);
                Console.WriteLine("Number: {0}", iNumber);
                return 0;
            }
        }
```