

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

قابلیت همکاری

مدرس : مهندس افشین رفوآ

قابلیت همکاری

مقدمه

یک از اهداف اصلی که در **NET** دنبال می شود، فراهم کردن امکان همکاری بین زبان های مختلف است (مانند به اشتراک گذاری کد). یکی از روش هایی که این امر را امکان پذیر می کند، توانایی استفاده از قابلیت های یک زبان در برنامه ای است که توسط زبان برنامه نویسی دیگر نوشته شده. به طور مثال، می توان از (کتابخانه ی غنی) توابع **Visual Basic** در یک برنامه ی کاربردی **C#** بهره جست. به این خاطر که هیچ کتابخانه ای صد در صد کامل نیست، ممکن است به قابلیتی نیاز پیدا کنید که در خود زبان انتخابی (زبانی که هم اکنون برای برنامه نویسی به کار می برید) موجود نباشد. همچنین ممکن است با تیم برنامه نویسی همکاری داشته باشید که از زبان های برنامه نویسی مختلفی استفاده کرده و توابع یا عملیات پیچیده ای نوشته باشند که شما مجبور به استفاده در برنامه ی خود هستید.

کتابخانه ی Win32

اصل سیستم عامل میکروسافت ویندوز در محیط زبان **C** نوشته شده است (زبان والد **C++**، **C#** و **Java**). برای ایجاد امکان برنامه نویسی (ایجاد برنامه های کاربردی)، میکروسافت کتابخانه ای به نام **Win32** را منتشر کرد. کتابخانه ی نام برده متشکل از یک سری تابع و کلاس هست که از قبل استفاده می کردید. امروزه، شما دیگر برای ایجاد برنامه ی کاربردی ویندوز ملزوم به استفاده از **Win32** نیستید. با این وجود، هنوز شمار زیادی از عملیاتی که ملزم با انجام آن ها در برنامه کاربردی ویندوز هستید تنها در **Win32** موجود هستند. خوشبختانه، به کاربردن تعداد زیادی از این توابع در زبان **C#** دشوار نیست و تنها قوانینی دارد که باید رعایت کرد.

مثال

```
using System;
using System.Runtime.InteropServices;
```

```

class Program
{
    [DllImport("Kernel32.dll")]
    public static extern bool SetConsoleTitle(string
strMessage);

    static void Main()
    {
        SetConsoleTitle("C# Programming");
    }
}

```

کتابخانه ی CLI/Visual C++

در زمان های قبل ایجاد یک کتابخانه (به خصوص در C++) کار بسیار دشواری بود. خوشبختانه، برنامه ی **Microsoft Visual C++** اکنون ایجاد کتابخانه را به فرایندی فوق العاده ساده تبدیل کرده. برای این منظور، ابتدا باید پنجره ی محاوره ی **New Project** را باز کنید. پس از انتخاب **Visual C++**، در لیست میانی، روی گزینه ی **Class Library** کلیک کرده و اسم آن را مشخص کنید. حال می توانید در بدنه ی فایل، کلاس ها / توابع دلخواه را ایجاد کنید.

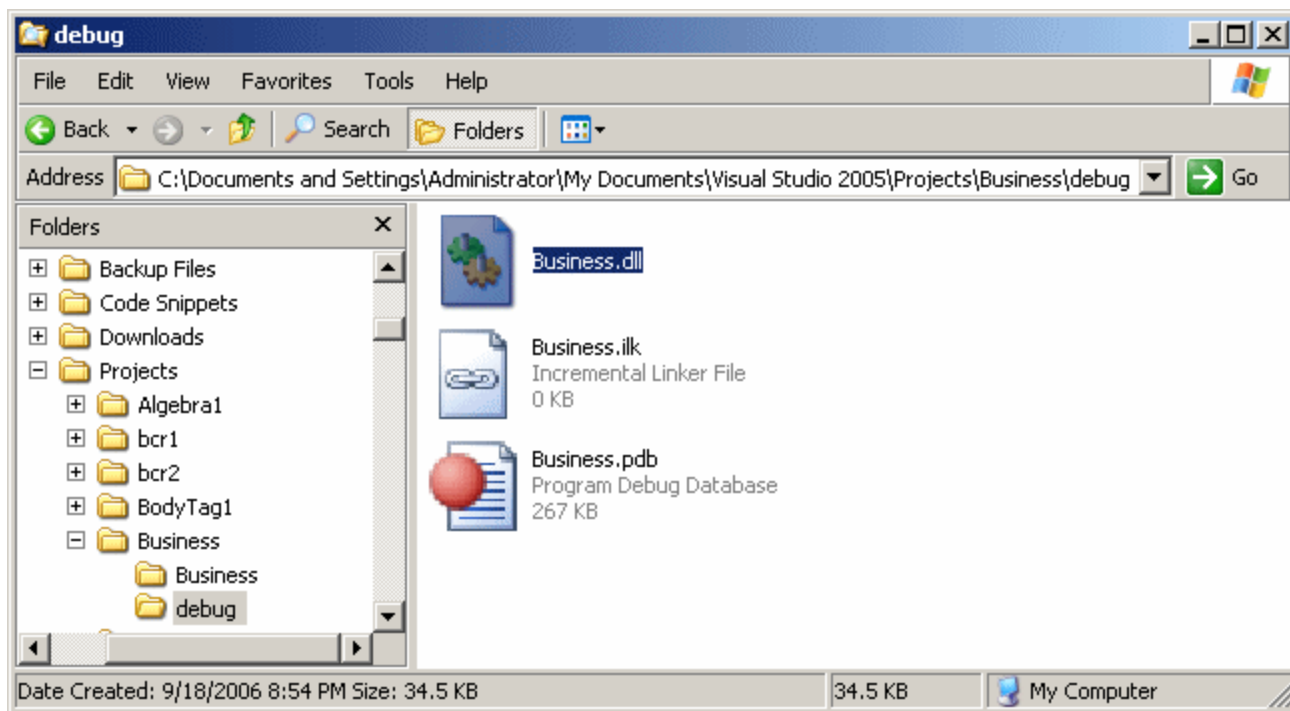
مثال

```

// Business.h
#pragma once
using namespace System;
public ref class Finance
{
public:
    double CalculateDiscount(double MarkedPrice,
double DiscountRate)
    {
return MarkedPrice * DiscountRate / 100;
    }
};

```

پس از آماده شدن پروژه، باید به ساختن آن پرداخت (فهرست گزینه ی اصلی، **Build -> Build Business**). در نتیجه، **compiler** فایل ی با پسوند **dll** ایجاد می کند.



به کاربردن کتابخانه

ایجاد کتابخانه در محیط ++C بسیار آسان است. برای استفاده از آن، قواعدی هست که باید رعایت شود. در وهله ی اول، باید مطمئن شوید که پروژه ی شما می تواند کتابخانه ی موردنظر را پیدا کند. آسان ترین روش برای این منظور، کپی کردن فایل **dll** و جای گذاری آن در فولدر دربردارنده ی فایل اجرایی (**executable**) پروژه است. این کار را می توان به طور مستقیم (از طریق **Microsoft Visual Studio**) با وارد کردن فایل کتابخانه نیز انجام داد.

لازم است فضای نام **System.Runtime.InteropServices** را در پروژه ی خود دخیل کنید. پیش از کتابخانه (پیش از ("**Business.dll**")), صفت مشخصه ی **DllImport** را وارد کنید. این ویژگی (**Attribute**) اسم کتابخانه را که در قالب رشته (**string**) ارسال شده، به جای / به عنوان آرگومان می گیرد.

مثال

```
using System;
    using System.Runtime.InteropServices;
    using Business;
```

```

class Exercise
{
    [DllImport("Business.dll")]
    public static extern double CalculateDiscount(double
price,
    double discount)

    static int Main()
    {
        Finance fin = new Finance();

        double markedPrice = 275.50;
        double discountRate = 25.00; // %
        double discountAmount =
fin.CalculateDiscount(markedPrice,
discountDate);
        double netPrice = markedPrice - discountAmount);
        Console.WriteLine("Marked Price: ");
        Console.WriteLine("markedPrice);
        Console.WriteLine("Discount Rate: ");
        Console.WriteLine("discountRate / 100);
        Console.WriteLine("Discount Amount: ");
        Console.WriteLine("discountAmount);
        Console.WriteLine("Net Price: ");
        Console.WriteLine("netPrice);

        return 0;
    }
}

```