

تعامل MATLAB (متلب) با دات نت - قسمت اول

عنوان:

مسعود مشهدی

نویسنده:

۱۵:۰ ۱۳۹۲/۰۳/۱۴

تاریخ:

www.dotnettips.info

آدرس:

برچسب‌ها: NET, C#.NET, MATLAB.

متلب (MATLAB) یکی از پرکاربردترین نرم افزارهای محاسباتی در حوزه مهندسی بویژه برق، ریاضیات، مکانیک و ... می‌باشد. بدون شک تعامل نرم افزارهای مختلف با هم در جهت کاربردی‌تر کردن یک پروژه کمک بسزایی به کاربران نهایی می‌کند. قطعاً استفاده از علوم روز همچون شبکه‌های عصبی، منطق فازی و الگوریتم‌های تکاملی همچون ژنتیک بدون استفاده از متلب بسیار سخت و پیچیده خواهد بود. دستورات و تابع‌های (functions) آماده و ساده در متلب در جهت استفاده از این علوم تقریباً هر پژوهشگر و کاربری را ترغیب به استفاده از متلب می‌کند. طبقاً استفاده از کتابخانه‌های دانت در متلب کمک بسیاری به توسعه دهندگان این حوزه می‌کند.

در این سری از مطالب سعی بر بررسی این تعامل شده است.

بطور کلی دو نوع تعامل در این زمینه وجود دارد :

1- استفاده از اسمبلی‌های دات نت در متلب تحت عنوان MATLAB .NET Interface

2- استفاده از پکیج تابع‌های متلب در پروژه‌های مبتنی بر دات نت تحت عنوان MATLAB Builder NE

در مورد اول از دات نت فقط در پلت فورم ویندوز استفاده می‌شود. کلیه امکانات دات نت 2 را ساپورت میکند و با ورژن‌های 3 و 3.5 سازگار است اما با ورژن 4 تنها بعضی از امکانات در دسترس است و هنوز مورد تست کلی قرار نگرفته است. کلیه امکانات دات نت در C# در متلب بجز یک سری از موارد که در جدول زیر ذکر شده است در دسترس است.

Features Not Supported in MATLAB

Cannot use `ClassName.propertyname` syntax to set static properties. Use [NET.setStaticProperty](#) instead.

Unloading an assembly

Passing a structure array, sparse array, or complex number to a .NET property or method

Subclassing .NET classes from MATLAB

Accessing nonpublic class members

Displaying generic methods using `methods` or `methodsview` functions. For a workaround, see [Display .NET Generic Methods Using Reflection](#).

Creating an instance of a nested class. For a workaround, see [Working With Nested Classes](#).

Saving (serializing) .NET objects into a MAT-file

Creating .NET arrays with a specific lower bound

Concatenating multiple .NET objects into an array

Implementing interface methods

Hosting .NET controls in figure windows

Casting operations

Calling constructors with `ref` or `out` type arguments

Using `System.Console.WriteLine` to write text to the command window

Pointer type arguments, function pointers, `Dllimport` keyword

.NET remoting

Using the MATLAB: (colon) operator in a `foreach` iteration

Adding event listeners to .NET events defined in static classes

Handling .NET events with signatures that do not conform to the standard signature

Creating empty .NET objects

Creating .NET objects that do not belong to a namespace

به عنوان مثال از کلاس `speech synthesizer` دات نت 3 در متلب بصورت زیر استفاده می‌کنیم :

```
function Speak(text)
    NET.addAssembly('System.Speech');
    speak = System.Speech.Synthesis.SpeechSynthesizer;
    speak.Volume = 100;
```

```
end          Speak(speak,text);
```

سپس برای رندر کردن یک متن به صوت دستور زیر را اجرا می‌کنیم :

```
Speak('You can use .NET Libraries in MATLAB');
```

در ارتباط با استفاده از توابع متلب در یک پروژه مبتنی بر دات نت در قسمت بعد توضیح داده خواهد شد.

منبع : Help متلب

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۳/۱۶ ۰:۵۸

در این کدی که تهیه کردید، Speak داخل متد Speak [چطور فراخوانی شده](#) بدون ارجاع به یک شیء؟ (سطر آخر متد). بعد در متلب نیازی به new نیست؟ فقط فراخوانی NET.addAssembly باعث شناسایی System.Speech.Synthesis.SpeechSynthesizer میشه؟

نویسنده: مسعود مشهدی
تاریخ: ۱۳۹۲/۰۳/۱۶ ۹:۱۸

متلب به حروف کوچک و بزرگ حساس است. دقت کرده باشید speak داخل Speak است. در حقیقت Speak نام تابع است. در متلب تابع NET.addAssembly کار بارگذاری اسمبلی‌های یاد شده را دارد. خیر در متلب احتیاجی به new نیست. در کل برنامه نویسی در متلب کمی متفاوت با زبان‌های معمول (C# و ..) است.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۳/۱۶ ۱۰:۲۰

اینجا در سطر آخر متد نباید نوشته می‌شد speak.Speak ؟ یا اینکه این Speak نوشته شده [با دو پارامتر](#) ، منظور تابعی دیگر هست؟

نویسنده: مسعود مشهدی
تاریخ: ۱۳۹۲/۰۳/۱۶ ۱۲:۱۸

نه دوست عزیز همونطور که گفتم ساختار متلب متفاوت هست.
منظور از Speak همون تابع است.

کافیست تابع فوق را در یک M فایل (کدها در متلب با پسوند ام فایل ذخیره می‌شوند) ذخیره کنید. سپس در داخل متلب مسیر اجرای کد را به مسیر M فایل تغییر دهید و سپس دستور رندر کردن رو که بالا گفته شد در Command Window اجرا کنید. متن فوق به تابع Speak ارجاع داده می‌شود.

نویسنده: علی
تاریخ: ۱۳۹۲/۰۳/۲۳ ۲۲:۳۷

استفاده از Matlab در دات نت نیاز به نصب کلی خود Matlab داره؟

نویسنده: مسعود مشهدی
تاریخ: ۱۳۹۲/۰۳/۲۴ ۲۱:۳۰

در این مورد در قسمت بعد توضیح خواهم داد. بله احتیاج است. اگر تصمیم به نوشتن کدها و کامپایل کردن اونها دارید. خیر، اگر نویسنده کدها یک پکیج از پروژه متلب در اختیار شما قرار بدهد.

در [قسمت قبل](#) در مورد استفاده دات نت در متلب توضیح داده شد. در این قسمت به نحوه استفاده توابع متلب در دات نت بصورت ساده می‌پردازیم.

فرض کنید تیم برنامه‌نویس متلب و تیم برنامه‌نویس دات نت در تعامل با یکدیگر هستند. وظیفه تیم برنامه‌نویس متلب به شرح زیر می‌باشد :

- 1- نوشتن توابع در متلب و تست کردن آنها جهت توسعه و ارائه مناسب به تیم مقابل
- 2- درست کردن کامپوننت دات نت در متلب با استفاده از محیط Deployment Tool GUI (با اجرای دستور deploytool در متلب)
- 3- استفاده از یک پکیج بسته‌بندی شده از فایل‌های قابل ارائه به تیم مقابل (اختیاری)
- 4- کپی پکیج در محل از قبل تعیین شده توسط دو تیم یا ارائه آن به تیم مقابل جهت استفاده

برای مثال M فایل (اصطلاح فایل‌ها در متلب همانند کلاس در دات نت) makesquare.m را که در مسیر

```
matlabroot\toolbox\dotnetbuilder\Examples\VS8\NET\MagicSquareExample\MagicSquareComp
```

است را در نظر بگیرید :

```
function y = makesquare(x)
%MAKESQUARE Magic square of size x.
% Y = MAKESQUARE(X) returns a magic square of size x.
% This file is used as an example for the MATLAB
% Builder NE product.

% Copyright 2001-2012 The MathWorks, Inc.

y = magic(x);
```

تابع magic یک ماتریس در ابعاد x در x درست می‌کند که درایه‌های آن اعداد صحیح از 1 تا x^2 بوده و مجموع سطر و ستون‌های آن با هم برابر است. x باید بزرگتر یا مساوی 3 باشد.

در صورتی که x برابر 5 انتخاب شود خروجی متلب بصورت زیر خواهد بود :

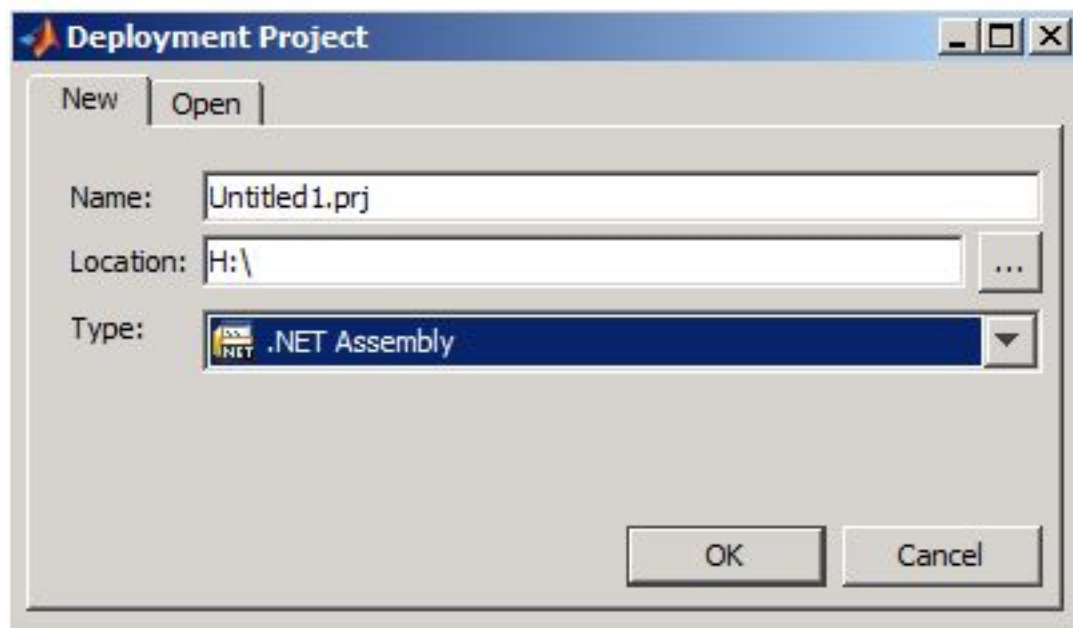
```
17 24 1 8 15
23 5 7 14 16
4 6 13 20 22
10 12 19 21 3
11 18 25 2 9
```

در قسمت تهیه یک کامپوننت دات نت اطلاعات زیر را در نظر بگیرید :

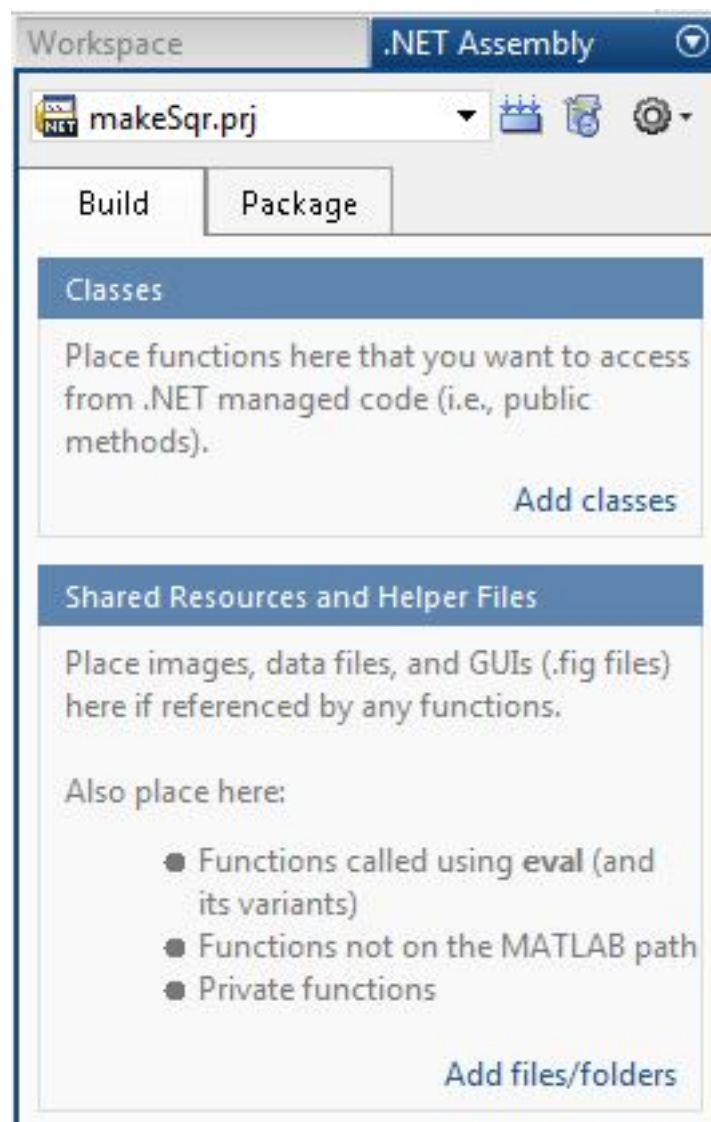
Project Name	makeSqr
Class Name	MLTestClass
File to compile	makesquare.m

سپس برای درست کردن کامپوننت در محیط Deployment Tool GUI برنامه متلب را اجرا کرده و در پنجره command دستور deploytool را اجرا کنید تا پنجره زیر باز شود :

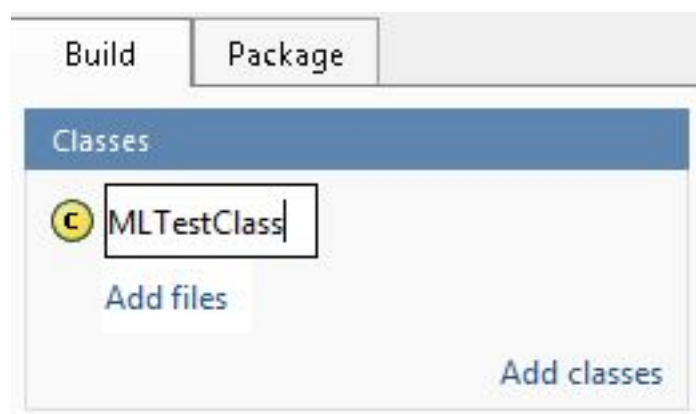
The Deployment Project Dialog Box



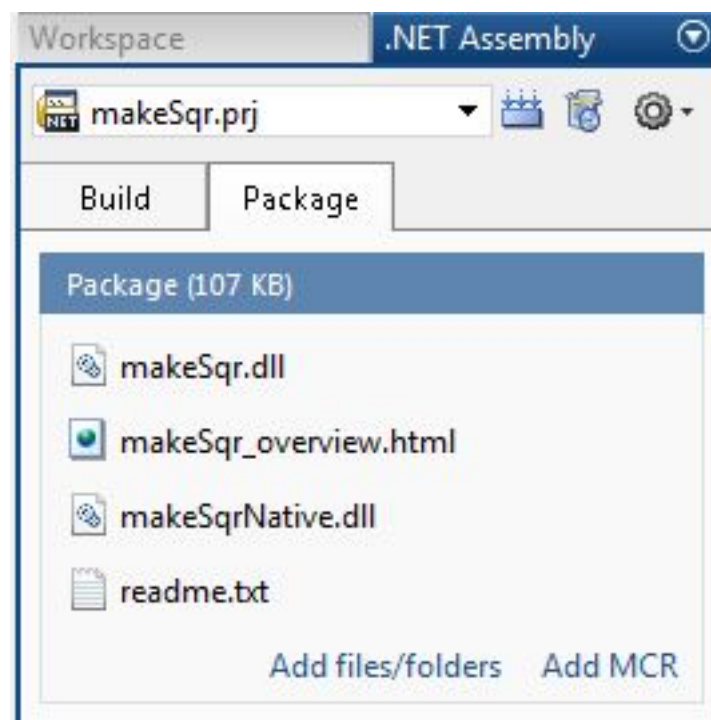
نام و مسیر پروژه را تعیین کنید سپس از منوی کشویی نوع پروژه، که دات نت اسمبلی باشد را انتخاب کنید. پنجره‌ای در به شکل زیر مشاهده خواهد شد :



در تب build اگر قصد استفاده از اپلیکیشن COM را دارید و یا فایل‌هایی جهت تکمیل پروژه قصد پیوست دارید را در قسمت پایین Add files را انتخاب کنید. و اگر قصد استفاده از اپلیکیشن دات نت را دارید قسمت بالایی Add classes را انتخاب کنید و نام کلاس را وارد کنید.

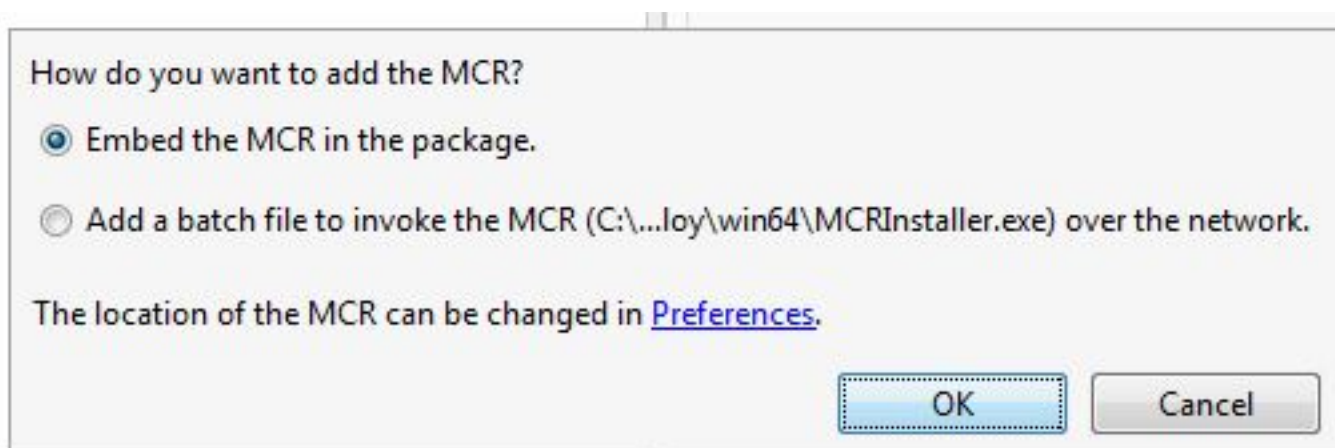


سپس برای کلاس مورد نظر فایل‌های متلبی که قصد کامپایل کردن آنها را دارید از قسمت Add files پیوست کنید. در صورتیکه قصد اضافه کردن کلاس اضافی را داشتید مجدداً مراحل را طی کنید. در انتها دکمه build را زده تا عملیات کامپایل آغاز شود. اما برای استفاده تیم برنامه‌نویسی دات نت احتیاج به کامپایلر متلب می‌باشد که این مهم در پکیجی که به این تیم ارائه خواهد شد مد نظر قرار خواهد گرفت. در قسمت تب Package گزینه Add MCR را انتخاب نمائید :



بعد از انتخاب، دو گزینه برای انتخاب وجود دارد که بطور خلاصه گزینه اول فایل‌های کامپایلر متلب در داخل پروژه جهت ارائه قرار می‌گیرد. همچنین این گزینه جهت استفاده در مواقع درون شبکه‌ای، مواردی که فضای دیسک و عملکرد و چندان اهمیت ندارد مورد استفاده قرار می‌گیرد. اما گزینه دوم عکس قضیه بالا عمل می‌کند و برای تعداد یوزر بالا و شبکه‌ای و ... مورد استفاده می‌باشد.

در اینجا گزینه اول را انتخاب می‌کنیم. در صورتیکه فایل‌های دیگری جهت ضمیمه به پکیج احتیاج است به آن اضافه می‌کنیم.



سپس کلید پکیج را زده تا پکیج مورد نظر آماده شود. دقت داشته باشید که بعد از انتخاب کامپایلر متلب، حجم پکیج نزدیک به 400 مگابایت خواهد شد. پکیج مورد نظر بصورت یک فایل exe فشرده خواهد شد. معمولاً پکیج شامل فایل‌های زیر باید باشد :

componentName.xml	Documentation files
componentName.pdb (if Debug option is selected)	Program Database File, which contains debugging information
componentName.dll	Component assembly file
MCR Installer	MCR Installer (if not already installed on the target machine).

بعد از طی مراحل فوق نوبت به تیم برنامه‌نویسی دات نت می‌رسد. بعد از دریافت پکیج از تیم برنامه‌نویسی متلب در صورتیکه بر روی سیستم هدف کامپایلر متلب و یا خود متلب نصب نیست باید از داخل پکیج این کامپایلر نصب شود. دقت داشته باشید که ورژن کامپایلر بر روی سیستم باید با ورژن پکیج دریافتی یکی باشد. در VS یک پروژه کنسول ایجاد کنید و از فولدر پکیج پروژه دریافتی در زیرفولدر distrib فایل makeSqr.dll را به رفرنس برنامه VS اضافه کنید. در ادامه از مسیر نصب کامپایلر فایل MArray.dll را هم به رفرنس پروژه اضافه کنید. این فایل جهت تبادل داده اپلیکیشن با کامپایلر متلب مورد استفاده قرار می‌گیرد.

installation_folder\toolbox\dotnetbuilder\bin\architecture\framework_version

اسمبلی‌های زیر را به کلاس Program برنامه اضافه کنید :

```
using System;
using MathWorks.MATLAB.NET.Arrays;
using MyComponentName;
```

سپس کدهای زیر را به کلاس فوق اضافه نمایید :

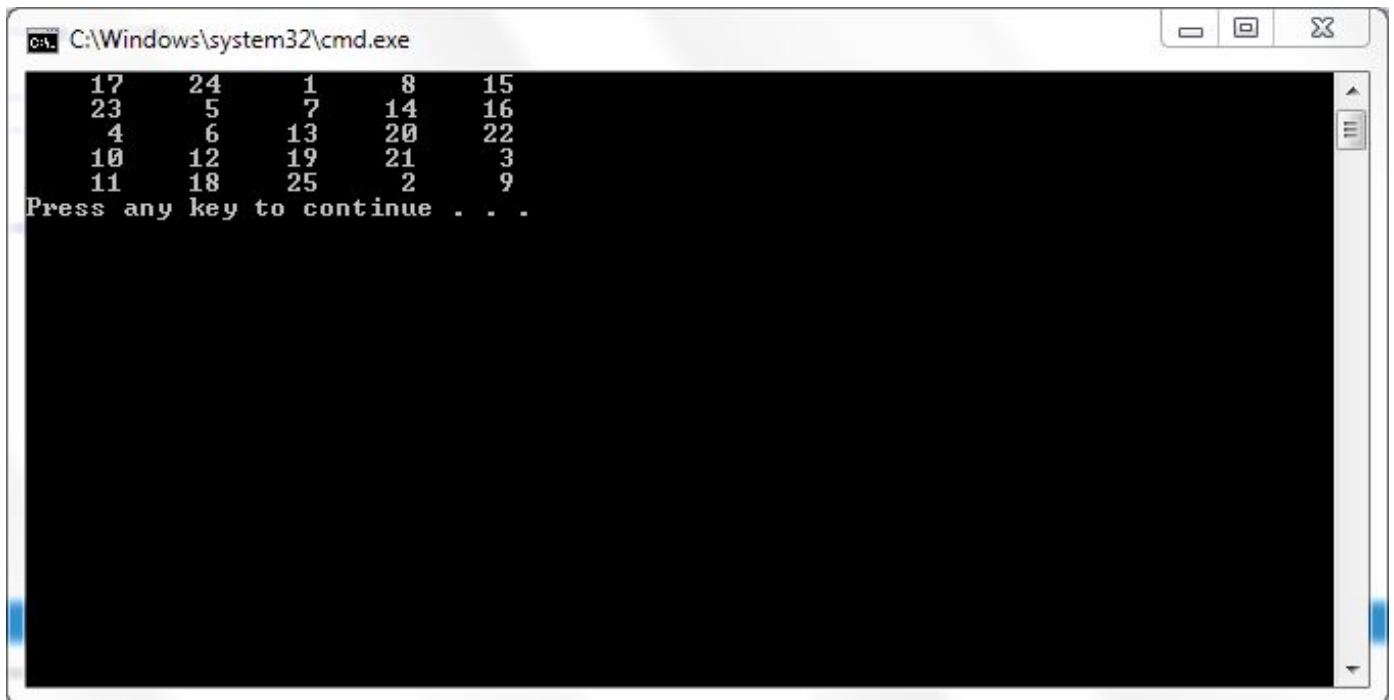
```
static void Main(string[] args)
{
    MLTestClass obj = new MLTestClass();
    MArray[] result = obj.makesquare(1, 5);

    MNumericArray output = (MNumericArray)result[0];
    Console.WriteLine(output);
}
```

توضیحات کدهای فوق :

- 1- MNumericArray یک اینترفیس جهت تعیین و نمایش نوع آرایه‌های عددی در متلب است.
- 2- MArray یک کلاس abstract جهت دسترسی، فرمت‌دهی و مدیریت آرایه‌های متلب می‌باشد.
- 3- عدد 1 مشخص کننده تعداد خروجی تابع متلب و عدد 5 ورودی تابع می‌باشد.

خروجی برنامه همانند خروجی متلب بصورت زیر خواهد بود :



```
C:\Windows\system32\cmd.exe

17    24    1    8    15
23    5    7    14   16
 4    6   13   20   22
10   12   19   21    3
11   18   25    2    9
Press any key to continue . . .
```

نکته:

ورژن فریمورک دات نت در هنگام کامپایل با ورژن Mwarrray.dll باید یکی باشد.

نظرات خوانندگان

نویسنده: سید امیر سجادی
تاریخ: ۱۱:۲۶ ۱۳۹۲/۰۸/۲۸

بسیار عالی بود. ممنون

نویسنده: بهار
تاریخ: ۱۳:۸ ۱۳۹۳/۰۵/۱۲

آموزشتون عالی بود. ممنون

ولی من نمی‌تونم برنامه را اجرا کنم. با پیغام زیر مواجه می‌شم

Could not load file or assembly 'MwArray, Version=2.11.0.0, Culture=neutral, PublicKeyToken=e1d84a0da19db86f' or one of its dependencies

برنامه را با VS 2010 و متلب R2011a در ویندوز 7 ، 64 بیتی نوشتم. از اینکه راهنمایی می‌کنید سپاسگزارم.

سوال دیگه ای که دارم، اینه که، چه جوری می‌تونم به جای برنامه ماتریس جادویی از شبکه عصبی استفاده کنم؟ آیا تغییر این دو برنامه فقط در تب Build است و نیازی به تغییر در تب package ندارد؟

نویسنده: محسن خان
تاریخ: ۱۵:۲۰ ۱۳۹۳/۰۵/۱۲

برنامه‌ی نوشته شده اگر کد خالص دات نت هست، [مشکلی با 64 بیت و 32 بیت نداره](#) . اما اگر داخلش ناخالصی native وجود داره، مثلاً از یک DLL بومی ویندوز استفاده می‌کنه که دات نت نیست و همچنین این DLL از نوع 32 بیتی هست و برنامه روی Any CPU تنظیم شده، حتماً کرش می‌کنه با خطایی که گفتید. راه حلش اینه که در خواص پروژه، any CPU را به X86 تغییر بدید.

نویسنده: مسعود مشهدی
تاریخ: ۱۷:۲۳ ۱۳۹۳/۰۵/۱۲

یک نکته دیگه رو هم دقت کنید که ورژن فریمورک دات نت در هنگام کامپایل با ورژن Mwarray.dll باید یکی باشد. در مورد ماتریس یا شبکه‌های عصبی کلیت تفاوتی نمی‌کند همانطور که این مثال یک تابع ماتریس magic هست شبکه‌های عصبی هم همان تابع هست با این تفاوت که از توابع تو در تو تشکیل شده است.

نویسنده: بهار
تاریخ: ۲۲:۱۷ ۱۳۹۳/۰۵/۱۲

ممنون از اینکه پاسخ دادید. مشکل حل شد. به جای X86 از Any CPU استفاده کردم و برنامه اجرا شد