(به نام خداوند بخشنده ی مهربان)

درس برنامه نویسی پیشرفته

تمرین سری هفتم

دانشکده ی مهندسی کامپیوتر

دانشگاه علم و صنعت ایران

استاد مرضیه ملکی مجد

نیم سال دوم ۱۴۰۰-۱۳۹۹

مهلت ارسال: ۱۴۰۰/۳/۳۰

مبحث تمرینات:

تکمیل مباحث

سوال اول:

مقدمه و شرح سوال

هدف تمرین؛ پیاده سازی کلاس Matrix با استفاده از رابط‌های IEnumerable, IEquatable و سربارگذاری عملگرها. نکته ای که در رابطه با این تمرین وجود دارد این است که میخواهیم با استفاده از مفهوم انواع داده ای عام و واسط‌ها کدی بزنیم که قابل استفاده مجدد باشد و بتوانیم از یک منطق چندبار استفاده کنیم.

کلاس vector:

همان طور که می دانید ه ر ماتریس از تعدادی بردار تشکیل شده است پس برای پیاده سازی کلاس Matrix (ماتریس) باید کلاس Vector (بردار) را پیاده سازی کنیم . لازم است بعد از پیاده سازی کامل کلاس Vector عملیات زیر انجام شود.

کلاس Vector را با استفاده از انواع داده خام پیاده سازی کنید. همان طور که میدانید این کار باعث میشود تا انعطاف پذیری کلاس ساخته شده بالا برود و بتوانید برداری از هر نوع داده بسازید. مثلا برداری از int, double, string یا هر نوع داده فرضی مانند Cell

بخش امتیازی اول:

این کلاس را به طوری پیاده‌سازی کنید که بتوانیم به آن به صورت زیر ورودی دهیم.

```
Vector <int > v1 = new Vector <int >(5) { 1, 2, 3, 4, 5 };
Vector <double > v2 = new Vector <double >(5) {1.1, 2.2., 3.3, 4.4., 5.5};
```

ورودی گرفتن عادی: به این صورت است که هر دفعه یک نوع داده ای را به کلاس add میکنید.

```
V1.addItem(1);
```

واسط‌های زیر را برای کلاس Vector پیاده سازی کنید.

۱. IEnumerable<_Type>

با پیاده‌سازی این واسط می‌توان اعضای یک نمونه از کلاس Vectorرا با توجه به نوع داده ای علمی که نمونه بر روی آن تعریف شده، پیمایش کرد.(پیمایش بر روی نوع داده ای نیست و بر روی اعضای وکتور است.)

۲. <<IEquatable<Vector<_Type>>

با پیاده سازی این واسط می توان هر شی از این کلاس با نوع داده ای مشخص را با شی دیگری از همین کلاس با همان نوع داده ای از لحاظ برابری مقایسه کرد.

Override کردن متد()ToString

پیاده سازی متد ()ToString را به گونه‌ای Override کنید که هر شی از کلاس Vector را به فرمتی که در مثال‌ها آمده نمایش دهد.

```
using System;
namespace A10
{
public class Program
{
public static void Main(string [] args)
{
Vector <int >[] vectors = new[]
{
```

```
new Vector <int >(5) {1, 2, 3, 4, 5},
new Vector <int >(5) {1, 2, 0, 4, 5},
new Vector <int >(6) {1, 2, 3, 4, 5, 6},
};
foreach (Vector <int > vector in vectors)
{
Console.WriteLine(vector.ToString ());
Console.WriteLine("---------------");
}
}
}
}
```

خروجی نمونه :

```
[1,2,3,4,5]
---------------
[1,2,0,4,5]
---------------
[1,2,3,4,5,6]
```

سربارگذاری عملگرها(explicit-implicit)

برای آشنایی بیشتر با این مفاهیم، نکته آخر داک را مطالعه کنید.

عملگر + را به گونه ای پیاده‌سازی کنید که بتوان دو شی از کلاس Vector را با انواع داده یکسان را به روش جمع برداری با هم جمع کرد. در صورت مهیا نبودن شرایط ضرب برداری از Exception مناسب استفاده کنید. دقت کنید که عملگر جمع و ضرب لزوما برای هر نوع داده ای تعریف نشده است. به همین دلیل کامپایلر در برخورد با عملگر + و *، خطای کامپایل خواهد داد. از این جهت کامپایلر اجازه ضرب یا جمع این انواع را نمیدهد. برای رفع این اشکال میتوانید از کلمه کلیدی dynamic استفاده کنید. جزئیات بیشتر استفاده از این کلمه کلیدی را میتوانید جست و جو کنید.

```
using System;
namespace A10
{
public class Program
{
public static void Main(string [] args)
{
Vector <int > v1 = new Vector <int >(5) { 1, 3, 2, 5, 4 };
Vector <int > v2 = new Vector <int >(5) {3, 1, 1, 6, 1};
Vector <int > v3 = v1 + v2;
Console.Write(v3.ToString ());
}
}
}
```

خروجی :

[4,4,3,11,5]

توضیحات :

V1 = [1 3 2 5 4]

V2 = [3 1 1 6 1]

V3 = v1 + v2

=[(1+3) (3+1) (2+1) (5+6) (4+1)]

[۵ ۱۱ ۵ ۴ ۴]=

عملگر == را به گونه ای پیاده سازی کنید که بتوان برابری دو شی از کلاس Vector با نوع داده یکسان را بررسی کرد.

عملگر =! را به گونه ای پیاده سازی کنید که بتوان نابرابری دو شی از کلاس Vector با نوع داده ای یکسان را بررسی کرد.

کلاس Matrix:

کلاس ماتریس را با استفاده از نوع داده عام پیاده سازی کنید . همان طور که می دانید این کار باعث می شود تا انعطاف پذیری کلاس ساخته شده بالا برود و بتوانید ماتریسی از هر نوع داده بسازید . مثلا ماتریسی از int ها ، doubleها ، stringها یا هر نوع داده فرضی ای مانند Cell

واسط‌های زیر را برای کلاس Matrix پیاده سازی کنید.

۱. IEnumerable<_Type>

با پیاده سازی این واسط می توان می توان اعضای هر نمونه از کلاس ماتریس را پیمایش کرد.

2.<<IEquatable<Vector<_Type>>

با پیاده سازی این واسط می توان هر شی از این کلاس با نوع داده ای مشخص را با شی دیگری از همین کلاس با همان نوع داده ای از لحاظ برابری مقایسه کرد.

Override کردن متد()ToString

پیاده سازی متد ToString() را به گونه ای override کنید که هر شی از کلاس Matrix را به فرمتی که در مثال زیر آمده نمایش دهد.

نکته: برای این کلاس هم حالت عادی اضافه کردن عضو به کلاس ماتریس مانند کلاس vector است با این تفاوت در این بخش به عنوان مثال شما باید هر دفعه یک شی از کلاس vector را به این کلاس add کنید. البته توجه کنید که این کلاس هم باید به صورت generic پیاده سازی شود. همانند کلاس vector.

نمونه ورودی(در این مثال تعریف کلاس ماتریس به صورت امتیازی پیاده سازی شده.)

```
using System;
namespace A10
{
public class Program
{
public static void Main(string [] args)
{
Matrix <int >[] matrices = new[]
{
new Matrix <int >(2, 3)
{
new Vector <int >(3) {1, 2, 1},
new Vector <int >(3) {2, -1, 1},
},
```

```
new Matrix <int >(3, 3)
{
new Vector <int >(3) {1, 2, 1},
new Vector <int >(3) {2, 0, 1},
new Vector <int >(3) {2, 0, 1}
},
new Matrix <int >(2, 3)
{
new Vector <int >(3) {1, 2, 1},
new Vector <int >(3) {2, -1, 1},
},
};
foreach (Matrix <int > matrix in matrices)
{
Console.WriteLine(matrix.ToString ());
Console.WriteLine("---------------");
}
}
}
```

{

خروجی نمونه:

[
[1,2,1],
[2,-1,1]
]

[
[1,2,1],
[2,-1,1]
]

[
[1,2,1],
[2,0,1]
[2,0,1]
]

سربارگذاری عملگرها

عمل گرها ی + ، == و =! را برا ی کلاس Matrix سربارگذاری کنید .

عملگر + را به گونه ای پیاده سازی کنید که بتوان دو شی از کلاس Matrix با انواع داده یکسان را به روش جمع برداری با هم جمع کرد. در صورت برقرار نبودن شرایط مناسب Exception مناسب ایجاد کنید.

```
using System;
namespace A10
{
public class Program
{
public static void Main(string [] args)
{
Matrix <int > m1 = new Matrix <int >(2, 3)
{
new Vector <int >(3) {1, 2, 1},
new Vector <int >(3) {2, -1, 1},
};
Matrix <int > m2 = new Matrix <int >(3, 2)
{
new Vector <int >(3) {0,2,1},
new Vector <int >(3) {1,4,1},
};
var m3 = m1 + m2;
Console.WriteLine(m3.ToString ());
}
}
```

{

خروجی نمونه

[

[1,4,2],

[3,5,2]

[

عملگر == را به گونه ایی پیاده سازی کنید که بتوان برابری دو شی از کلاس Matrix با نوع داده یکسان را بررسی کرد .

عملگر != را به گونه ای پیاده سازی کنید که بتوان نابرابری دو شی از کلاس Matrix با نوع داده یکسان را بررسی کرد .

نکته: مفهوم سربارگذاری(implicit, explicit) را میتوانید از صفحات ۴۹۳ الی ۵۱۳ فصل ۲۲ کتاب مرجع درس یا از طریق لینک های زیر مطالعه نمایید.

C# Basics - C# Type Conversions (Implicit and Explicit Conversion) (code-maze.com)

Implicit And Explicit Conversions In C# (c-sharpcorner.com)

توجه شود که باید برای برنامه Main به همراه ورودی و خروجی های مناسب برای هر متد نوشته شود. در غیر این صورت قسمتی از نمره را از دست خواهید داد. مواردی که در مین برنامه باید انجام دهید: ۱. ساختن سه شی از کلاس Vector که دوتا از آن‌ها شبیه هم باشند

. ۲. فراخوانی متد toString برای یکی از آن ها.

۳. جمع کردن دوتا از vector ها و نمایش خروجی با استفاده از متد toString

۴. چک کردن یکی بودن یا نبودن دوتا از vector ها باهم.

۵. ساختن سه شی از کلاس Matrix که دوتا از آنها یکی باشند.

۶. نمایش یکی از آن ها با متد toString

۷. جمع کردن دوتا از ماتریس‌ها باهم و نمایش خروجی با متد toString

۸. چک کردن یکی بودن یا نبودن ماتریس ها باهم

بخش امتیازی: میتوانید به جای نوشتن Main از یونیت تست استفاده کنید. در این صورت به شما نمره امتیازی تعلق خواهد گرفت.

سوال دوم

کریستوفر رابین

کریستوفر رابین به جنگل بازگشته است و قصد دارد دوستانش را ملاقات کند. بنابراین تصمیم میگیرد قرار ملاقات را در کلبه کوچکش بگذارد. به دلیل اینکه کلبه اش کوچک بوده و به اندازه یک نفر ظرفیت دارد، دوستانش باید به ترتیب با او ملاقات کنند. کریستوفر رابین برای ملاقات با دوس تانش امتیازهایی به آنها داده است .اما با توجه به اینکه دوستانش شیطنت هایی دارند لذا هر کدام تلاش میکنند تا زودتر با او ملاقات کنند .کریستوفر رابین اعلام کرده است به دلیل مشغله کاری اش، امروز میتواند ۳ نفر از دوستانش را ببیند.

حال با دوس تان کریستوفر رابین آشنا شوید:

1.پو: یک خرس زرد رنگ مهربان که عاشق عسل است. امتیاز ۵ برای او در نظر گرفته شده است. در واقع بالاترین امتیاز ثبت شده برای اوست. پس او همیشه میتواند اولین نفر کریس توفر را ملاقات کند.

2.پیگلت: خوک صورتی بامزه ای که کمی ترسو است. امتیاز ۴ برای او در نظر گرفته شده است.

3.تیگر: یک ببر شیطون و همیشه خندان است. امتیاز ۳ برای او در نظر گرفته شده است.

4.روو: بچه کانگرو شیطون که رفیق تیگر هست. امتیاز ۲ برای او در نظر گرفته شده است.

5.ایور: خر خسته که همیشه غرغر میکند. امتیاز ۱ برای او در نظر گرفته شده است.

پیاده سازی: در این سوال میخواهیم با کلاس generic بیشتر آشنا شوید. در این سوال شما باید یکinterface و شش کلاس پیاده سازی کنید:

رابط: IPersonality

شامل دو ویژگی(property)، نام از نوع string و امتیاز از نوع int است، همچنین یک متد به نامPersonality بدون پارامتر ورودی و خروجی ای از نوع string است.

Class:

دوستان کریستوفر رابین به صورت کلاس تعریف میشوند. اسامی کلاسها باید بر اساس نوع حیوانی که هستند باشد. مثلا برای پو کلاسی با نام Bear، برای تیگر کلاسی با نام Tiger و غیره داریم. در همه کلاسها لازم است رابط IPersonality را پیاده سازی کنید. متد Personality باید تمامی تعاریفی که از دوستان کریستوفر در فوق به رنگ آبی مشخص شده را به عنوان رشته برگرداند. مثلا برای کلاس پو خواهیم داشت: (پو خرس زرد رنگ، عاشق عسل، امتیاز ۵)

Pooh is yellow bear.

He loves honey.

His name’s rate is 5 => from property

دقت کنید باید برای این قسمت ۵ کلاس بنویسید

کلاس Friend که به صورت generic تعریف شود، تایپ های مورد قبول این کلاس، تایپ هایی هستند که رابط IPersonality را پیاده سازی کرده باشند. تایپ جنریک کلاس را T نام گذاری کنید. در این کلاس یک فیلد از نوع T وجود دارد. سازنده ای با پارامتر ورودی از نوع T نیز تعریف کنید. و در نهایت یک متد تعریف کنید که به عنوان خروجی، همان خروجی متد Personality را برمیگرداند.

پس از پیاده سازی کلاسها، در Main برنامه خواهیم داشت:

Friend<Bear> bear = new Bear(“Pooh”, 5)

Friend<Tiger> tiger = new Tiger(“Tiger”,3)

...(و برای سایر دوستانش به همین ترتیب خواهیم داشت)

خروجی متد موجود در کلاس Friend را برای هر یک از شی های ساخته شده چاپ کنید.

سوال سوم

کار با لینک

موضوع تمرین پردازش برخی از داده های مربوط به فیلم های imdb است. فایل-IMDB Movie-Data.csv شامل داده هایی مربوط به فیلم ها در سایت imdb است. پردازش اولیه در فایل CSVParser.cs که ضمیمه شده است، پیاده سازی شده است. کاری که شما باید انجام دهید این است که Extension Method های مناسبی را پیاده سازی کنید و از آنها برای جوابگویی به سوالاتی که در زیر مطرح شده است، استفاده کنید .

۱) ژانر فیلم هایی که مدت زمان آنها زیر ۱۰۰ دقیقه است را چاپ کند.

۲) نام کارگردان فیلم هایی که وین دیزل در آن بازی کرده است.

۳) مشخصات فیلمی با بیشترین رای در سال ۲۰۱۶ را برگرداند.

۴) نام فیلم ها به کارگردانی Bryan Singer را همراه با میزان فروش آنها به ترتیب نزولی چاپ کند.

۵) مجموع فروش تمامی فیلم های اکران شده در سال۲۰۱۱ را چاپ کند.

۶) ۱۰ تا از پرفروش ترین فیلم های ژانر اکشن که طولانی تر از دو ساعت هستند.

۷) فیلم هایی که در نامشان عدد وجود دارد.

۸) فیلم های Anne Hathaway, Jennifer Lawrence به ترتیب قدیم به جدید و بیشترین rating به کمترین.(فیلم های هر بازیگر نه اینکه هر دو با هم)

۹) مقایسه تعداد فیلم های درام و کمدی دارای rating بیشتر از ۸.

۱۰) نام بازیگری را بیابید که تعداد فیلم های بدی که بازی کرده است از بقیه بازیگر ها بیشتر است.(فیلم بد را به معنی rating کمتر از ۷ در نظر بگیرید).