

تمرین ۱ (Rational.java)

کلاسی با نام Rational بصورت زیر پیاده سازی نمایید

```
class Rational
{
    public Rational(int num, int denum);
    public Rational(int num);
    public Rational(Rational base);
    public Rational();
    public Rational add(Rational num);
    public Rational sub(Rational num);
    public Rational mul(Rational num);
    public Rational div(Rational num);
    public double toFloatingPoint();
    public String toString();
}
```

این کلاس باید محاسبات ریاضی را بر روی کسرها انجام دهد. توجه کنید که همیشه باید نتایجی که در متغیرهای درونی خود ذخیره می کنید، در حالت ساده شده، باشند. (برای مثال، کسر $2/4$ بصورت یک شیء ذخیره خواهد شد که صورت آن ۱ و مخرج آن ۲ است).

یک سازنده فراهم کنید تا بتوان یک نمونه از این شیء را هنگام ساخته شدن، مقداردهی اولیه نیز کرد. (توسط پارامترهای ورودی که می گیرید)

یک سازنده بدون آرگومان نیز فراهم کنید تا در حالتی که در ابتدا مقداری داده نمی شود، کسر بصورت پیش فرض، مقدار صفر را بگیرد.

یک سازنده نیز فراهم کنید که از روی یک شیء Rational دیگر بتواند خود را بسازد. (یک Rational را بعنوان آرگومان بگیرد)

یک سری متدهای `public` برای هر کدام از اعمال زیر قرار دهید:

۱. جمع کسر فعلی با یک کسر دیگر بوسیله متد `add`.

`public Rational add (Rational num)`

۲. تفریق کسر فعلی از یک کسر دیگر بوسیله متد `sub`.

`public Rational sub (Rational num)`

۳. ضرب کسر فعلی با یک کسر دیگر بوسیله متد `mul`.

`public Rational mul (Rational num)`

۴. تقسیم کسر فعلی با یک کسر دیگر بوسیله متد `div`.

`public Rational div (Rational num)`

توجه کنید که در تمامی متدهای بالا نتایج در شیء که متد بر روی آن صدا زده شده است، ذخیره نمی شوند

بلکه باید در یک شیء کسر جدید ذخیره شوند که توسط این متدها برگردانده می شود.

برای مثال اگر اعداد کسری X و Y را داشته باشیم و مقدار اولیه هر کدام از آنها $1/2$ باشد، عمل `x.add(y)` بر

روی مقادیر X و Y تأثیری نخواهد داشت و مقدار هر دوی آنها $1/2$ باقی می ماند.

۵. یک عدد کسری را به فرمت اعشاری تبدیل کنید، بوسیله متد `toFloat`.

`public double toFloatingPoint (Rational num)`

۶. یک عدد کسری را به رشته متنی با قالب "`a/b`" تبدیل کنید که a صورت کسر و b مخرج کسر است.

`public String toString ()`

تمرین ۲ (Polynomial.java)

در این سوال شما باید کلاس Polynomial را پیاده سازی کنید که به صورت زیر می باشد:

```
class Polynomial {  
    Polynomial(double[] coefficients);  
    Polynomial(String str);  
    double evaluate(double x);  
    int degree();  
    Polynomial add(Polynomial p);  
    Polynomial subtract(Polynomial p);  
    Polynomial multiply(Polynomial p);  
    Polynomial divide(Polynomial p);  
    Polynomial differentiate();  
    Polynomial integral(double constant);  
    double findRoot(double start, double end, double accuracy)  
    double max(double start, double end, double accuracy)  
    boolean equals(Polynomial p);  
    String toString();  
}
```

Constructor اول آرایه ای از ضرایب می گیرد و چندجمله ای را می سازد.

Constructor دوم یک رشته می گیرد و چندجمله ای را می سازد. (مثلا $x^2+13x+1388$) (اختیاری)

متد `evaluate` مقدار داده شده را به متغیر چندجمله ای منصوب می کند و سپس با توجه به آن مقدار چندجمله ای را بدست می آورد.

متد `degree` درجه ی چندجمله ای را برمی گرداند.

متد `add` چندجمله ای داده شده را با چندجمله ای ما جمع می کند و حاصل را بعنوان خروجی برمی گرداند.

متد `subtract`. چندجمله ای داده شده از چندجمله ای ما کم می کند و حاصل را بعنوان خروجی برمی گرداند.
متد `multiply`. چندجمله ای داده شده را در چندجمله ای ما ضرب می کند و حاصل را بعنوان خروجی برمی گرداند.
متد `divide`. چندجمله ای ما را بر چندجمله ای داده شده تقسیم می کند و خارج قسمت را بعنوان خروجی برمی گرداند.

متد `differentiate`. مشتق چندجمله ای را بعنوان خروجی برمی گرداند.

متد `integral`. انتگرال چندجمله ای را با جمله ی ثابت `constant` برمی گرداند.

متد `findRoot`، ریشه ی چندجمله ای را از `start` تا `end` با دقت `accuracy` برمی گرداند. در صورتیکه ریشه ای پیدا نشد، مقدار `end+1` را برمی گرداند.

متد `max`. مقداری از متغیر (از `start` تا `end` با دقت `accuracy`) که در آن چندجمله ای مقدار بیشینه ی خود را دارد، برمی گرداند.

متد `equals`. مشخص می کند که چندجمله ای ما با چندجمله ای داده شده برابر است یا خیر.

متد `toString`. چندجمله ای را بصورت یک رشته برمی گرداند. (مثل $x^2+13x+1388$)

تمرین ۳ (Matrix.java)

در این سوال شما باید کلاس `Matrix` را پیاده سازی کنید که به صورت زیر می باشد.

```
class Matrix
{
    Matrix(int rows, int cols);
    Matrix(double[][] data);
    Matrix add(Matrix m);
    Matrix subtract(Matrix m);
    Matrix multiply(Matrix m);
    Matrix transpose();
    boolean equals(Matrix m);
    String toString();
}
```

Constructor اول، ماتریسی با تعداد سطر `rows` و تعداد ستون `cols` می سازد.

Constructor دوم، ماتریسی با درایه های داده شده در آرایه ی دوبعدی `data` می سازد.

متد `add`، ماتریس داده شده را با ماتریس ما جمع می کند و حاصل را برمی گرداند.

متد `subtract`، ماتریس داده شده را از ماتریس ما کم می کند و حاصل را برمی گرداند.

متد `multiply`، ماتریس داده شده را در ماتریس ما ضرب می کند و حاصل را برمی گرداند.

متد `transpose`، ترانهادی ماتریس ما را برمی گرداند.

متد `equals`، مشخص می کند ماتریس داده شده با ماتریس ما برابر است یا خیر.

متد `toString`، ماتریس ما را به یک رشته تبدیل می کند. مثلا ماتریس $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ را به رشته ی "a b\nc d" تبدیل نماید. بطور دقیق تر، برای یک ماتریس $A_{m \times n}$ باید ابتدا درایه های A_{11} و A_{12} و ... و A_{1n} را چاپ کند و سپس یک

کاراکتر خط جدید ($\backslash n$) را نوشته و سپس به سراغ سطر بعدی برود و به همین ترتیب m سطر را چاپ کند. (در انتهای چاپ سطر m هم باید یک $\backslash n$ نوشته شود).

اختیاری:

```
class SquareMatrix extends Matrix
{
    SquareMatrix(int n);
    SquareMatrix inverse();
    double determinant();
    double eigenValues(double start, double end, double accuracy);
}
```

Constructor آن، مقدار n را می‌گیرد و ماتریسی مربعی به اندازه‌ی سطر و ستون n تولید می‌کند.

متد `inverse`، معکوس ماتریس مربعی ما را برمی‌گرداند.

متد `determinant`، دترمینان ماتریس مربعی ما را برمی‌گرداند.

متد `eigenValue`، اولین مقدار ویژه^۲ی ماتریس مربعی ما را از `start` تا `end` با دقت `accuracy` برمی‌گرداند. در

صورتیکه مقدار ویژه‌ای نداشت، مقدار `end+1` را برمی‌گرداند. (می‌توانید از کلاس `Polynomial` استفاده کنید)

^۲ به صفحه‌ی ۴۵۴ کتاب ریاضی عمومی (۲) خود مراجعه فرمائید.