

# تکلیف برنامه نویسی انتگرال

## مقدمه

کُد هایی که در اختیار دارید مشتمل بر 5 کلاس می باشد (که با استفاده از eclipse نوشته شده اند):

- 1. Expression
- 2. Polynomial
- 3. IntegralTest
- 4. IntegralCoordinator\_Part\_I
- 5. IntegralCoordinator\_Part\_II

شما می بایست به ترتیبی که در این مستند ذکر شده کلاس های فوق را (بسته به نوع آن) بنویسید، یا تکمیل کنید و یا تغییر دهید و یا تنها اجرا کنید.

## کلاس Expression

این کلاس، کلاسی است که در آن هر یک از جمله های یک چند جمله ای ذخیره می شود. هر جمله به شکل زیر است:

$$a x^b$$

که  $a$  را ضریب و  $b$  را توان می نامیم.

شما می بایست دو تابع از این کلاس را تکمیل نمایید:

### تابع ComputeValue

این تابع، می بایست عبارت  $a x^b$  را به ازای یک  $x$  خاص محاسبه نماید که به عبارتی محاسبه عبارت زیر خواهد بود:

$$a \times x \times x \times \dots \times x$$

که تعداد  $x$  ها به اندازه  $b$  است.

**\*\*نکته:** بعد از انجام این قسمت، تست شماره یک را از کلاس IntegralTest انجام دهید تا از جواب خود مطمئن شوید.

### تابع FindIntegralExpression

این تابع می بایست انتگرال عبارت کنونی را بدست آورد و به عنوان خروجی بیرون دهد. به عبارت دیگر خروجی جمله ی کنونی یا همان  $a x^b$  باید به صورت  $\frac{a}{b+1} x^{b+1}$  باشد.

**\*\*نکته:** بعد از انجام این قسمت، تست شماره دو را از کلاس IntegralTest انجام دهید تا از جواب خود مطمئن شوید.

## کلاس Polynomial

این کلاس کلاسی برای ذخیره ساختار داده ی یک چندجمله ای می باشد. ماهیت اصلی آن یک ArrayList از جمله ها (Expression) ها می باشد. به عبارت دیگر، هر چند جمله ای به صورت زیر است:

$$P(x) = \sum_{i=0}^n a_i x^{b_i}$$

شما می بایست شش تابع از این کلاس را تکمیل نمایید:

### تابع *ComputeValue*

این تابع، می بایست عبارت  $\sum_{i=0}^n a_i x^{b_i}$  را به ازای یک  $x$  خاص محاسبه نماید. در این راستا حتماً می بایست از تابع `computeValue` که در کلاس `Expression` نوشته اید استفاده نمایید و حاصل جمله های مختلف را با یکدیگر جمع نمایید. دقت داشته باشید که برخلاف آنچه در فیلم آموزشی جهت محاسبه چندجمله ای گفته شده است، ما برای راحتی پیاده سازی، از قاعده هرنر استفاده نمی کنیم.

**\*\*نکته:** بعد از انجام این قسمت، تست شماره سه را از کلاس `IntegralTest` انجام دهید تا از جواب خود مطمئن شوید.

### تابع *FindIntegralPolynomial*

این تابع می بایست انتگرال چندجمله ای کنونی را بدست آورد و به عنوان خروجی بیرون دهد. به عبارت دیگر خروجی چندجمله ای کنونی یا همان  $\sum_{i=0}^n a_i x^{b_i}$  باید به صورت  $\sum_{i=0}^n \frac{a_i}{b_i+1} x^{b_i+1}$  باشد. شما حتماً می بایست از تابع `FindIntegralExpression` که در کلاس `Expression` نوشته اید استفاده نمایید و حاصل انتگرال جمله های مختلف را با یکدیگر در یک چندجمله ای (به عنوان خروجی) ذخیره نمایید.

**\*\*نکته:** بعد از انجام این قسمت، تست شماره چهار را از کلاس `IntegralTest` انجام دهید تا از جواب خود مطمئن شوید.

### تابع *computeArea\_DefiniteIntegral*

این تابع تابعی است که می بایست مساحت زیر نمودار منحنی را در بازه داده شده، با استفاده از فرمول انتگرال معین برای چند جمله ای ها، محاسبه کند. شما حتماً باید از توابع `computeValue` و `findIntegralPolynomial` از کلاس `Polynomial` در اینجا استفاده کنید.

به عنوان یادآوری توجه کنید که:

$$F(x) = \int f(x) dx \rightarrow \int_a^b f(x) dx = F(b) - F(a)$$

**\*\*نکته:** بعد از انجام این قسمت، تست شماره پنج را از کلاس `IntegralTest` انجام دهید تا از جواب خود مطمئن شوید.

### تابع *simpsonFormula*

همانطور که می دانید، فرمول سیمسون، مساحت زیر نمودار یک تابع در یک بازه معین را با نظیر کردن نزدیک ترین سهمی روی آن تخمین می زند. این تابع، یک تابع ساده و کمی برای محاسبه فرمول سیمسون با استفاده از ورودی های داده شده است. به عبارتی، برای ورودی های  $a$  و  $b$ ، این تابع فرمول زیر را محاسبه می کند (به سمت راست تخمین نگاه کنید):

$$\int_a^b f(x) dx \approx \frac{b-a}{6} \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right].$$

شما حتما" باید از تابع `computeValue` از کلاس `Polynomial` برای این قسمت استفاده نمایید.

**\*\* نکته:** بعد از انجام این قسمت، تست شماره شش را از کلاس `IntegralTest` انجام دهید تا از جواب خود مطمئن شوید.

### تابع `computeArea_Simpson`

در این تابع مساحت زیر نمودار چندجمله ای با استفاده از روش سیمسون و پارامتر  $h$  تخمین زده می شود. شما می بایست بازه ی داده شده را به زیربازه های به طول حداکثر  $h$  تقسیم بندی نمایید و برای هر زیربازه از فرمول سیمسون (تابع `simpsonFormula`) استفاده نمایید و تمامی موارد را با یکدیگر جمع کنید.

**\*\* نکته:** بعد از انجام این قسمت، تست شماره هفت را از کلاس `IntegralTest` انجام دهید تا از جواب خود مطمئن شوید.

### تابع `computeArea_MonteCarlo`

در این تابع مساحت زیر نمودار چندجمله ای با استفاده از روش مونته کارلو و پارامتر  $n$  تخمین زده می شود. شما می بایست ابتدا مستطیل محیط نمودار را در بازه مذکور بیابید. سپس  $n$  نقطه به صورت تصادفی در آن برگزینید و ببینید که چه کسری از ایشان در زیر نمودار جای می گیرند. با ضرب این کسر در مساحت مستطیل، تخمینی از مساحت زیر نمودار بدست می آید. توجه داشته باشید که طول مستطیل محیطی، همان طول بازه است و عرض آن – با توجه به مثبت بودن ضرایب و در نتیجه صعودی بودن چندجمله ای – ارتفاع نقطه ی منتظر انتهای بازه روی چندجمله ای است.

شما حتما" باید از تابع `computeValue` از کلاس `Polynomial` در اینجا استفاده کنید.

**\*\* نکته:** بعد از انجام این قسمت، تست شماره هشت را از کلاس `IntegralTest` انجام دهید تا از جواب خود مطمئن شوید.

## کلاس `IntegralCoordinator_Part_I`

این کلاس تنها برای تولید داده ها برای کشیدن نمودار های مختلف است. با این کلاس می توان داده های 24 نمودار خواسته شده از شما در تکلیف انتگرال را تولید کرد و متعاقبا" رسم نمود. برای این منظور کافی است شما پارامتر های `n`, `h`, `begin`, `end`, `isSimpson` را تنظیم کنید. بدین ترتیب، داده های لازم برای ترسیم دو نمودار با پارامتر های مذکور تولید خواهد شد: داده های (درجه چندجمله ای- زمان اجرا) و داده های (درجه چندجمله ای- خطای نسبی).

اکنون شما می بایست 24 نمودار مذکور را با 12 بار اجرای این کلاس رسم نمایید. دقت کنید بهتر است خودتان با سلیقه خود، متغیر های `MAX_DEGREE`, `MAX_CONFIG`, `MAX_REPEAT` را تنظیم کنید.

## کلاس `IntegralCoordinator_Part_II`

این کلاس خالی است و شما می بایست آن را بنویسید. شما باید به طریقی آن را بنویسید که داده های 8 نمودار انتهایی خواسته شده از شما در تکلیف انتگرال را بتوان با آن رسم کرد. بدون شک، کلاس `IntegralCoordinator_Part_I` می توان برای شما بسیار مفید باشد.