

דגשים חשובים לפני התיעוד:

1. הבנאי שמקבל מחרוזת מפריד מונומים אך ורק על ידי הסימן "+".
במידה ורוצים להכניס מונום עם מקדם שלילי צריך לרשום "+" ואז את הסימן "-".
לדוגמא: `"2x+-3x^2+x^3"`.
2. התיעוד המלא בתקייה doc.

Polynom

בנאים:

Constructor	Description
<code><u>Polynom</u>()</code>	make new polynom
<code><u>Polynom</u>(java.lang.String string)</code>	<p>make new polynom by string the format of string is "$a_1x^{b_1}+a_2x^{b_2}+...+a_nx^{b_n}$" when $a_1,a_2,...,a_n$ is the coefficient for every monoms.</p> <p>$b_1,b_2,...,b_n$ is the power for every monoms.</p> <p>the "+" sign is requirement between every single monom, even if is negative. possible add negative monom by use "+-" sign.</p> <p>if monom power is 1 not necessary to write power sign("x^1").</p> <p>if monom power is 0 not necessary to write x sign("x^0").</p> <p>to enter zero is need get exactly the string "is zero".</p> <p>i ignore from space and charter case.</p> <p>for example: <code>1+5x+-2.1x^12+3.8x^4</code></p>
<code><u>Polynom</u>(<u>Polynom</u> p1)</code>	deep copy constructor

Modifier and Type	Method	Description
void	<u>add</u> (myMath.Monom m1)	add monom m1 to this polynom.
void	<u>add</u> (myMath.Polynom_able p1)	add polynom p1 to this polynom.
double	<u>area</u> (double x0, double x1, double eps)	Compute Riemann's Integral.
myMath.Polynom_able	<u>copy</u> ()	create a deep copy of this Polynom.
myMath.Polynom_able	<u>derivative</u> ()	Compute a new Polynom which is the derivative of this Polynom.
boolean	<u>equals</u> (myMath.Polynom_able p1)	Indicates whether some other polynom is "equal to" this one.
double	<u>f</u> (double x)	
private myMath.Monom	<u>find</u> (int power)	find monom with the same power.
private void	<u>insert</u> (myMath.Monom m1)	insert monom such that the list stay organize
boolean	<u>isZero</u> ()	Indicates whether this polynom is zero.
Iterator<myMath.Monom>	<u>iteretor</u> ()	get Iterator.
void	<u>multiply</u> (myMath.Polynom_able p2)	multiply this Polynom by p1

double	<u>root</u> (double x0, double x1, double eps)	find root. more details in root method in Polynom_able.
void	<u>subtract</u> (myMath.Polynom_able p1)	Subtract p1 from this Polynom
java.lang.String	<u>toString</u> ()	make string to print or to use in constructor.

Monom

בנאים:

Constructor	Description
<u>Monom</u> (double a, int b)	constructor
<u>Monom</u> (<u>Monom</u> ot)	copy constructor

Modifier and Type	Method	Description
void	<u>add</u> (<u>Monom</u> m1)	Add m1 to this monom. just for monom with the same power.
void	<u>derivative</u> ()	Compute a monom which is the derivative("nigzeret") of this monom.
boolean	<u>equals</u> (<u>Monom</u> another)	Indicates whether some other monom is "equal to" this one.
double	<u>f</u> (double x)	give x and return the result of a multiple x power b.
double	<u>get coefficient</u> ()	get coefficient
int	<u>get power</u> ()	get power
void	<u>multiply</u> (<u>Monom</u> m1)	multiply this monom by another monom.
private void	<u>set coefficient</u> (double a)	
private void	<u>set power</u> (int p)	
java.lang.String	<u>toString</u> ()	return in ths format: "ax^b" when a is coefficient and b is power.

תקציר:

בפרויקט זה אני בניתי פולינום המממש מגוון פונקציות מתמטיות.

אני השתדלתי לבנות אותו יציב ככול יכולתי כך שהקוד יעבוד עם כול קלט חוקי.

את הקוד בניתי בעצמי. במהלך העבודה נעזרתי בהסברים מיוטיוב על הפתרון המתמטי של הפונקציות areal root, בספרייה Math המובנת ב-java, והתייעצתי עם חברים לגבי איך לעשות test כמה שיותר טוב.

את הפורמט של ה-sting המצאתי בעצמי והוא מוסבר בתיעוד של הבנאי של המחלקה Polynom.

מקורות:

- <https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html> - הספרייה Math
- <https://youtu.be/Y2AUhxoQ-OQ> - סרטון שהסביר יפה על אלגוריתם Bisection
- <https://youtu.be/fttY0QQsACg?t=1326> - סרטון שהסביר קצת על סכומי רימן