

Pacakge MyMath

חבילה זו מייצגת את הביטוי המתמטי פולינום המורכב ממונומים

Writers

Gal Hadida(205516321) , Roi Abramovitch(311505481).

Description

- **Include 3 interface :**
 - **Function.**
 - **Count_function.**
 - **Polynom_able .**
- **Main Class :**
 - **Test.**
- **Other Class :**
 - **Polynom.**
 - **Monom.**
 - **Monom_Comperator.**
 - **LinePlotTest.**

Monom ax^b

המחלקה מממשת ביטוי מהצורה ax^b כאשר a הוא מקדם מסוג `double` ו b הוא חזקה מסוג `INT`

המחלקה כוללת בנאים ופונקציות המאפשרות לבצע חיבור, חיסור, כפל בין מונומים, חישוב נגזרת של מונום, פונקציה שהופכת מחרוזת של מונום לאובייקט מונום, והדפסת מחרוזת של המונום.

Constructors		
Constructor	Description	
Monom(double a, int b)	constructor	
Monom(Monom ot)		

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	add(Monom m1)	this function add a different monom to this monom by the same power
void	derivative(Monom ot)	this function get a monom and derivative him
double	f(double x)	This interface represents a simple function of type $y=f(x)$, where both y and x are real numbers.
double	get_coefficient()	
int	get_power()	
void	multiply(Monom ot)	this function get a monom and multiply by this monom
Monom	StringMonom(java.lang.String s)	this function get a monom by string and change him to object monom
java.lang.String	toString()	this function print a string with a monom

Methods inherited from class java.lang.Object

Method Monom

1. `public double f(double x)` = הפונקציה מקבלת פרמטר x ומוצאת את ערך y כאשר הפונקציה f היא המונום שעליו מפעילים את x לכן בפונקציה קראנו למונום שלנו והפעלנו את המקדם כפול ה- x שקיבלנו בחזקת המספר של החזקה.
2. `public void derivative(Monom ot)` = הפונקציה מקבלת מונום כלשהו ומבצעת עליו נגזרת של מונום. כמו כן בדקנו שאם החזקה של המספר היא 0 הרי שנגזרת של מספר בחזקת 0 הוא 0.
3. `public void multiply(Monom ot)` = הפונקציה מקבלת מונום נוסף ומכפילה את המונום שיש במחלקה עם המונום הנוסף
4. `public void add(Monom m1)` = הפונקציה מקבלת מונום נוסף ובודקת אם אפשר לחבר עם המונום של המחלקה, התנאי שזה יתקיים זה שהחזקות שלהם שוות וכמובן שהמונום לא ריק.
5. `public Monom StringMonom(String s)` = הפונקציה מקבלת מחרוזת שמכיל מונום כלשהו ובודקת אם הוא מהצורה $3*x^2$ או מהצורה $3x^2$ ומטפלת בהתאם הפונקציה הופכת את מחרוזת מונום הנ'ל לאובייקט מונום חדש כמו כן בודקת מצבים בהם יש מינוסים לפני המונום ומטפלת בהתאם לזה.
6. `public String toString()` = הפונקציה מקבלת אובייקט מונום ומחזירה אותו מחרוזת בצורה של $3*x^2$ כלומר עם סימון של $*$ וגם סימון של X עם המקדם והחזקה
7. `public int compare(Monom o1, Monom o2)` = הפונקציה מקבלת שני מונומים ובודקת מי יותר גדול ממי. הפונקציה לוקחת את שני המונומים ובודקת את החזקות שלהם אם אחד גדול מהשני תחזיר 1 או מינוס 1 אם הם שווים תחזיר 0.

Polynom $ax^b + cx^n$

מחלקה זאת מממשת פולינום מהצורה $a*x^b+....+c*x^n$ הפולינום יוצג על ידי `arraylist` מסוג `Monoms`.

המחלקה כוללת בנאים , בנאי המקבל פולינום בצורת מחרוזת והופך אותה לאובייקט פולינום .

פונקציות המבצעות חיבור חיסור וכפל של שני הפולינומים ,חישוב נגזרת של פולינום , חישוב שטח לפי שיטת ריימן לחישוב אינטגרל , חישוב נק' חיתוך של הפונקציה עם ציר ה x , פונקציית הדפסה שתדפיס את הפולינום.

Constructors		
Constructor	Description	
Polynom()		
Polynom(java.lang.String string)	the function Initializing string to object from the type polynom	

Method Summary		
All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	add(Monom m1)	Checks whether there is a Monom with the same __power , If exists , It add the coefficient to Monom that exists.
void	add(Polynom_able p1)	this function add a diffrent Polynom to our object Polynom
double	area(double x0, double x1, double eps)	Compute Riemann's Integral over this Polynom starting from x0, till x1 using eps size steps,
Polynom_able	copy()	Makes a precise copy of the polynom and saves it in a new loction in the memory
Polynom_able	derivative()	Compute a new Polynom which is the derivative of this Polynom
boolean	equals(Polynom_able p1)	Test if this Polynom is logically equals to p1.
double	f(double x)	the method calculator the result of polynom for x
boolean	isZero()	test if the polynom contain Monom that his coefficient = 0
java.util.Iterator<Monom>	iteretor()	this function run over each object in the arryList
void	multiply(Polynom_able p1)	Multiply this Polynom by a different Polynom
double	root(double x0, double x1, double eps)	Calculates the function cuts with the axes until approximation of epsilon
void	subtract(Polynom_able p1)	this function subtract two Polynom Multiply p1 with Monom =(-1)' and adds it to the Polynom of the function
java.lang.String	toString()	this function print the Polynom by string
Methods inherited from class java.lang.Object		
equals, getClass, hashCode, notify, notifyAll, wait, wait, wait		

Method Polynom

1. `public Polynom(String string)` = הפונקציה מקבלת מחרוזת של פולינום הופכת אותו בעזרת הפונקציה של `monom(String string)` כך שכל חלק מחרוזת שהוא מונם יהפוך לאובייקט מסוג מונם ואז מכניסה אותו לתוך פולינום של המחלקה בעזרת פונקציית `add` של פולינום שמקבלת מונם ומכניסה לתוך `arrayList` של פולינום

2. `public double f(double x)` = הפונקציה עוברת במצביע על הפולינום של המחלקה ומפעילה את פונקציה `f` של מונם (מספר 4 בפונקציות מונם) על כל מונם בפולינום על ידי המשתנה `X` שמקבלת ומכניסה את הערך שחזר למשתנה `sum` ובסוף הריצה על כל הפולינום מחזירה את המשתנה `sum` שמכיל את ערך `Y` של הפולינום.

3. `public void add(Monom m1)` = הפונקציה מקבלת מונם חדש ובודקת על ידי מצביע שרץ על כל הפולינום, בכל פולינום בודקת אם החזקה שלהם שווה ואז מפעילה את `add` של מונם. אם החזקות לא שוות אז פשוט מכניסה תוך `arrayList` את המונם החדש בסוף הפעולה עושה `sort` (עובד על ידי הפונקציה ב `Monom_Comperator`) על הפולינום כדי לסדר אותו מהגדול לקטן (לפי חזקות).

4. `public void multiply(Polynom_able p1)` = הפונקציה מקבלת פולינום חדש ומכפילה את הפולינום של המחלקה בפולינום החדש כאשר יש מצביע שרץ על כל פולינום ובפעם הראשונה מכפיל את המונם הראשון של פולינום החדש בכל מונם בפולינום של המחלקה. כאשר המצביע עובר למונם השני בפולינום החדש ההכפלה שלו תהיה על פולינום מועתק של הפולינום של המחלקה ובכך לא מאבדים את האינפורמציה של הכפלה הראשונה. כמו כן הפונקציה השתמשה במונם חדש כאשר נכסים להכפיל את הפולינום העתקה כדי שגם בפעם הבאה אותו מונם יישאר כמו בפעם הקודמת. לאחר כל הכפלה הפונקציה קראה לפונקציית `add` של מונם כדי להכניס את המונם המוכפל לתוך הפולינום החדש. בסוף כל סיבוב חזר המצביע של הפולינום העתקה ובסיבוב הראשון חזר המצביע של הפולינום של המחלקה, זה בשביל להכפיל כל פעם את על המונומים שוב.

5. `public void subtract(Polynom_able p1)` = הפונקציה מקבלת פולינום חדש ומחסרת את הפולינום של המחלקה פחות הפולינום החדש. הפונקציה משתמשת בפונקציית ההכפלה של הפולינום כאשר מכפילה את כל הפולינום החדש במינוס אחד ואז משתמשת במשתמשת בפונקציית `add` של פולינום

6. `public boolean equals(Polynom_able p1)` = הפונקציה מקבלת פולינום חדש ובודקת אם הוא שווה לפולינום של המחלקה. הפונקציה בודקת בעזרת פונקציית `size` שיש לה במחלקה את גדלי הפולינומים אם הם לא שווים תחזיר שקר אם שווים תבדוק במצביע על כל פולינום אם החזקות שוות אם

לא תחזיר שקר אם כן תבדוק אם המקדמים של כל מונום שווים (הערה):
הפולינום מסודרים המגדול לקטן).

7. `public boolean isZero()` = הפונקציה בודקת אם הפולינום ריק משום שכל המקדמים הם אפס והפונקציה בודקת אם הפולינום ריק כי אם המקדם הוא 0 אז הוא מוסר מהפולינום ולכן הפונקציה רצה במצביע על הפולינום ובודקת אם יש לו מונום אם יש יחזיר שקר אם אין יחזיר אמת.

8. `public Polynom_able copy()` = הפונקציה מעתיקה את הפולינום לפי ערכים ומכניסה אותו לתוך פולינום חדש. הפונקציה רצה במצביע על הפולינום ומכניסה לתוך פולינום חדש באמצעות פונקציית `add` של מונום.

9. `public Polynom_able derivative()` = הפונקציה לוקחת פולינום וגוזרת אותו ומכניסה לתוך פולינום חדש את הנגזרת. הפונקציה רצה במצביע על הפולינום ומשתמשת בפונקציית נגזרת של מונום על כל מונום ואחרי הגזירה מכניסה באמצעות פונקציית `add` של מונום לתוך הפולינום החדש.

10. `public Iterator<Monom> iteretor()` = הפונקציה יוצרת את המצביע על הפולינום כאשר מפעילים את הפונקציה היא יוצרת מצביע על `arrayList` שמכיל פולינום וככה יכולים לעבור על כל המונומים בפולינום.

11. `private int size(Polynom_able p1)` = זוהי פונקציית עזר רק למחלקת פולינום שבודקת על ידי מצביע כמה מונומים יש בפולינום – השימוש של הפונקציה היא בשביל פונקציית `equel` של פולינום.

12. `public String toString()` = הפונקציה מחזירה מחרוזת של פולינום. הפונקציה מקבלת את הפולינום ומדפיסה את המונום הראשון ואז רצה עליו במצביע שעובר על כל מונום בפולינום. נבדק אם המקדם של המונום שווה ל0 ואז ממשיכים למונום הבא ואם הוא לא שווה ל0 אז ייבדק אם גדול מ0 או קטן מ0 כדי להדפיס לפי מינוס או פלוס.

13. `public double root(double x0, double x1, double eps)` = הפונקציה מקבלת `x1`, `x0` ו `eps` מהמשתמש, ועליה לחשב את נקודת החיתוך של הפונקציה בין ה-`x`ים עד לידי דיוק של קטן שווה מ `eps` שהמשתמש הכניס. בכל שלב הפונקציה בודקת את ערכי ה `x` עבור אותה הפונקציה ועל פי כך קובעת איזה `x` לקדם כלפי ה `x` השני, עד אשר המרחק בין ה-`X`ים קטן מ `eps`.

14. `public double area(double x0, double x1, double eps)` = הפונקציה מקבלת `x1`, `x0` ו `eps` מהמשתמש, עליה לחשב את שטח הפונקציה בין שני ה `X`ים שקיבלנו הפונקציה מחשבת לפי אינטגרל ריימן, היא מחשבת את השטח של מלבנים מהנק' `x0` על ציר ה `X` עד לנק' `x0+eps` כפול הגובה של `x0` באותה נקודה. בכל שלב היא מקדמת את הנק' על ציר ה `X` בעוד `eps` עד אשר מגיעה ל `x1`. ובסוף סוכמת את שטח כל המלבנים.

15. `public double areaUnderAxisX(double x0, double x1, double eps)`

הפונקציה מקבלת x_0 , x_1 ו- eps מהמשתמש, עליה לחשב את שטח הפונקציה בין שני הנקודות שקיבלנו. הפונקציה מחשבת לפי אינטגרל רימן, אך ורק את השטח הכלוא ממתחת לציר ה- x ולפונקציה עצמה.

היא מחשבת את השטח של מלבנים מהנקודה x_0 על ציר ה- x עד לנקודה x_0+eps כפול הגובה של x_0 באותה נקודה. בכל שלב היא מקדמת את הנקודה על ציר ה- x בעוד eps עד אשר מגיעה ל- x_1 . ובסוף סוכמת את שטח כל המלבנים.

LinePlotTest

מחלקה זאת מציירת פולינום מהצורה $a*x^b+...+c*x^n$ הפולינום יוצג על ידי גרף המשוורטט בעזרת הצירים x ו- y .
המחלקה קובעת את גודל הגרף, הצבעים, המסגרת.
המחלקה גוזרת את הפולינום ומוצאת נק' קיצון במקרה ויש על ידי גזירת הפונקציה והשוואתה לאפס.
במקרה ויש נק' קיצון המחלקה מדגישה אותם בגרף