מבוא לחישוב – סמסטר ב' תש"פ מטלה מס' 4

**הנחיות כלליות:**

1. מטלה זאת נעשית **ביחידים** בלבד.
2. יש **לכתוב הערות** במקומות המתאימים ולהשתמש בשמות משתנים משמעותיים.
3. יש לכתוב טסטים **JUNIT** לבדיקת התכנה.
4. אחרי סיום המטלה, יש לדחוס את כל הקבצים עם סיומת **java** לתוך קובץ מסוג **zip** ששמו הוא **מס' תעודת הזהות** של הסטודנט.
5. המטלה תיבדק באמצעות בדיקות אוטומטיות. לכן, יש להקפיד על הפלטים בדיוק כפי שהוגדר.
6. הקפידו לבדוק ולטפל בכל סוגי הקלטים, מצורפת דוגמת הרצה של התרגיל, אשר מדגימה קלטים \ פלטים שונים.
7. **לא יתקבלו עבודות שנשלחו בדואר אלקטרוני!**
8. **כל הפונקציות יהיו תחת מחלקה אחת בשם M\_4 package בשם : M\_4**

חלק ראשון:

במטלה זו נעסוק במימוש פולינומים: כולל יכולת של יצירה, אתחול, חיבור, כפל, חישוב נגזרות ואחרות. יש לממש את המחלקות: מונום, פולינום, ומחלקת בדיקה כדי לוודא שהקוד שלכם עובד.

1. ) - Monomמונום) – מחלקה שמייצגת פונקציה מהצורה  
    ,   
   כאשר הוא מספר ממשי, הוא מספר שלם אי-שלילי ().

המחלקה מכילה מתודות ליצירת מונום, העתקה עמוקה, חישוב ערך של מונום בנקודה x, חיבור וכפל של שני מונומים, גזירת מונום.

1. Polynom – היא מחלקה שמייצגת אוסף סדור (לפי חזקות) של מונומים:   
   המחלקה מכילה מתודות ליצירת פולינום, אתחול, העתקה עמוקה, הוספת מונום לפולינום , חיבור וכפל של שני פולינומים, גזירת פולינום.

תיאור מחלקות :

/\*\*

\* This class represents a simple "Monom" of shape a\*x^b,

\* where a is a real number and b none negative integer

\* see: https://en.wikipedia.org/wiki/Monomial

\* The class support simple operations as:

\* construction, value at x, derivative, add and multiply.

\* When constructor gets Monom as string, number of possible  
 \* shapes: String[] monoms = {"2", "-x","-3.2x^2","0"};  
 \* **@author** Boaz

\*

\*/

|  |
| --- |
| Constructors |
| Constructors and Description |
| [Monom](https://docs.oracle.com/javase/7/docs/api/java/util/Stack.html#Stack())(double coeff, int power)  Creates a Monom. |
|  |
| [Monom](https://docs.oracle.com/javase/7/docs/api/java/util/Stack.html#Stack())(Monom m)  Copy constructor. |
|  |
| [Monom](https://docs.oracle.com/javase/7/docs/api/java/util/Stack.html#Stack())(String s)  Creates Monom from String |
| **Example:** {"2", "-x","-3.2x^2","0"}; |

|  |  |
| --- | --- |
| Modifier and Type | Method and Description |
| boolean | isZero()  Tests if this is a zero monom (the coefficient is zero) |
|  |  |
| boolean | equals(Monom m)  Tests if this is equal to m |
|  |  |
| void | add(Monom m)  Add m to this Monom, (when m.power==this.power) |
|  |  |
| void | void multiply(Monom m)  Multiply this Monom by m |
|  |  |
| double | f(double x)  Calculate the value of Monom in point x |
|  |  |
| int | compareTo(Monom m)  compare two monoms:  returns -1 if this.power < m.power,  +1 if this.power > m.power 0 if this.power = m.power **and** this.coeff = m.coeff |
|  |  |
| Monom | derivative()  Compute a new Monom which is the derivative of this Monom |
|  |  |
| String | toString()  Returns String representing the value of Monom |
|  |  |

המחלקה העיקרית שעליכם לממש היא הפולינום, להלן תיאור המתודות:

/\*\*This class represents a Polynom with add, multiply  
 \* functionality,

\* it also should support the following:

\* 1. Riemann's Integral:   
 \* https://en.wikipedia.org/wiki/Riemann\_integral

\* 2. Finding a numerical value between two values   
 \* (currently support root only f(x)=0).

\* 3. Derivative

\* **@author** Boaz \*/

|  |
| --- |
| Constructors |
| Constructors and Description |
| Polynom ()  Creates a Polynom |
|  |
| Polynom (Polynom p)  Copy constructor. |
|  |

|  |  |
| --- | --- |
| Modifier and Type | Method and Description |
| void | add(Monom m)  Add m to this Polynom |
|  |  |
| void | add(Polynom p)  Add p to this 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, see  https://en/wikipedia.org/wiki/Riemann\_integral |
|  |  |
| Polynom | derivative()  Compute a new Polynom which is the derivative of this Polynom |
|  |  |
| boolean | equals(Polynom p)  Test if this logically equals to p |
|  |  |
| boolean | isZero()  Tests if this is a zero Polynom (all monoms are zero) |
|  |  |
| void | multiply(Polynom p)  Multiply this Polynom by p |
|  |  |
| double | root(double x0, double x1, double eps)  Compute a value x: x0 ≤ x ≤ x1 for whitch |f(x)|<eps,  assuming f(x0)\*f(x1) ≤ 0, returns x2: : x0 ≤ x2 ≤ x1 && |f(x2)| ≤ eps |
|  |  |
| void | substruct(Polynom p)  Substruct p from this Polynom |
|  |  |

חלק שני:

יש לכתוב בדיקות **JUNIT** לכל מתודות של המחלקות Monom ו- Polynom

המופיעות בחלק ראשון.

עבודה מהנה ומועילה!

**נספח דוגמה לבדיקות:**

**private** **static** **void** PolynomTest(){

System.***out***.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\* Polynom Check");

Polynom p1 = **new** Polynom();

p1.add(**new** Monom(3,2));

p1.add(**new** Monom(4,3));

System.***out***.println("p1.size(): "+p1.size());

System.***out***.println("Polynom p1: "+p1.toString());

//

Polynom p2 = **new** Polynom();

p2.add(**new** Monom(8,5));

p2.add(**new** Monom(4,4));

System.***out***.println("p2.size(): "+p2.size());

System.***out***.println("Polynom p2: "+p2.toString());

//

**double** x = 1.6;

Polynom p3 = p1.multy(p2);

System.***out***.println("p3.size(): "+p3.size());

System.***out***.println("Polynom p3: "+p3.toString());

System.***out***.println("p3.valueAt(x): "+p3.valueAt(x));

//

System.***out***.println("p2.valueAt(x): "+p2.valueAt(x));

Polynom p4 = p3.multy(p2);

System.***out***.println("Polynom p4: "+p4.toString());

System.***out***.println("p4.valueAt(x): "+p4.valueAt(x));

}

OUTPUT

\*\*\*\*\*\*\*\*\*\*\*\*\*\* Polynom Check

p1.size(): 2

Polynom p1: (3.0\*x^2)+(4.0\*x^3)+

p2.size(): 2

Polynom p2: (4.0\*x^4)+(8.0\*x^5)+

p3.size(): 3