מבוא לתכנות מונחה עצמים – מטלה 0

במטלה זו "נחזור לכושר" תכנותי בדגש על תכנות של מחלקות לפי ממשקים, המטלה עוסקת במימוש בפולינומים: כולל יכולות של אתחול, יצירה, חיבור, כפל, חישוב נגזרת, אינטגרל מסוים ועוד. כדי לממש את המטלה עליכם למעשה לעבוד לפי הממשקים המוגדרים של פולינום, פונקציה ופונקציה רציפה ולממש את המחלקות: פולינום, מונום, ומחלקת בדיקה לוודא שהקוד שלכם נכון ועובד היטב.

המטלה עצמה:

1. הורידו את התיעוד, והקוד הקיים, פתחו פרויקט java.
2. הכירו את הממשק Polynom_able, תכננו כיצד אתם הולכים לממש את המחלקה Polynom, הקפידו לחשוב גם פונקציות בדיקה במחלקה PolynomTest
3. התחילו במימוש והשלמה של המחלקה Monom, בדקו אותה בנפרד (תוכלו לעשות שימוש ראשוני במחלקה MonomTest) אבל עליכם להרחיב אותה משמעותית ולבדוק את כל הפונקציונליות שלה.
4. ממשו את המחלקה Polynom שממשת את הממשק Polynom_able, כאשר אתם מקפידים לבדוק אותה היטב במחלקה PolynomTest.
5. כתבו מסמך תיעוד נפרד בקובץ Readme.pdf שכולל את ההסברים המלאים למטלה שלכם.

Package myMath

Interface Summary

Interface	Description
cont_function	The interface represents a continuance function
function	This interface represents a simple function of type y=f(x), where both y and x are real numbers.
Polynom_able	This interface represents a general Polynom: f(x) = a_1X^b_1 + a_2*X^b_2 ...

Class Summary

Class	Description
Monom	This class represents a simple "Monom" of shape a*x^b, where a is a real number and a is an integer (summed a none negative), see: https://en.wikipedia.org/wiki/Monomial The class implements function and support simple operations as: construction, value at x, derivative, add and multiply.
MonomTest	This class represents a simple (naive) tester for the Monom class, Note: (i) The class is NOT a JUNIT - (i.e., educational reasons) - should be changed to a proper JUnit in Ex1.
Polynom	This class represents a Polynom with add, multiply functionality, it also should support the following: 1.
PolynomTest	

איור 1: רשימת המחלקות במטלה 0.

Method Summary

All Methods | Instance Methods | Abstract Methods

Modifier and Type	Method	Description
void	add(Monom m1)	Add m1 to this Polynom
void	add(Polynom_able p1)	Add p1 to this Polynom
Polynom_able	copy()	create a deep copy of this Polynom
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.
boolean	isZero()	Test if this is the Zero Polynom
java.util.Iterator<Monom>	iteretor()	
void	multiply(Monom m1)	Multiply this Polynom by Monom m1
void	multiply(Polynom_able p1)	Multiply this Polynom by p1
void	substract(Polynom_able p1)	Subtract p1 from this Polynom

Methods inherited from interface myMath.cont_function

area, root

איור 2: רשימת השיטות שיש לממש (מתוך ממשק Polynom_able והממשקים שהוא יורש מהם).

הנחיות טכניות חשובות להכנה והגשה של מטלה 0:

1. את מטלה 0 אתם מגישים למודל כקובץ zip, שם הקובץ יהיה תעודות הזהות של הסטודנטים המגישים מופרדים בקו תחתון. לדוגמא 987654321_123456789,
(שימו לב, שאת כל שאר המטלות יש להגיש אך רק כקישור לגרסת commit מסויימת).
2. קובץ הZIP יכיל בתוכו את קובץ ה readme ותיקיה בשם myMath שתכיל את כל קבצי הג'אווה. שימו לב לאותיות גדולות\קטנות. אין לעטוף את הקוד בתיקיות נוספות.
3. לשים לב בפונקציית הequals שבגלל שימוש בfloat and double יש לפעמים בעיות נומריות לדוגמא המספר 3 עלול להיות מיוצג 3.0000001 וגם 2.99999999 יש להתחשב בסטייה זו כאשר משווים. שוליים של 0.000001 לכל כיוון – ראו מימוש במחלקה Monom.
4. את המטלה יש לעשות ביחידים או זוגות (לא שלשות).
5. אסור להעתיק קוד משום מקור, ובכל מקרה חובה להביא את המקורות בהם נעזרתם.
6. את המטלה יש להגיש למודל לפני המועד האחרון (מטלות שתוגשנה באיחור לא תזכנה לציון מלא).

בהצלחה.