Principles of Object Oriented Programming
Spring 2018
Assignment 2

The assignment should be submitted in groups of two students.

1. General Description

The assignment goal is to practice the following concepts

- Object-oriented design
- Collections
- Inheritance and substitution

In this assignment you will implement a Polynomial Calculator. The calculator will allow the user to compute polynomial addition, polynomial evaluation, and other services. The design should be general, and allow easy modification for any type of scalar. You need to implement the following classes (It may be abstract or interface. You can provide additional methods, if you wish.

Scalar

This class should support the following services:

- **Scalar add(Scalar s)**: accepts a scalar argument and returns a scalar which is the sum of the current scalar and the argument.
- **Scalar mul(Scalar s)**: accepts a scalar argument and returns a scalar which is the multiplication of the current scalar and the argument.
- **Scalar neg()**: returns a scalar which is the result of multiplying the current scalar by (-1).
- **Scalar inv()**: returns (1/scalar)
- **boolean equals(Scalar s)**: returns **true** if the argument Scalar and the current Scalar have the numeric value.

Two classes that extends/implements Scalar:

1. **RationalScalar**: For Rational numbers. A rational number is a number a/b, where a and b are integers,$b \neq 0$. It should be represented using two integer fields.
2. **RealScalar**: For Real numbers

PolyTerm

This class represents a polynomial term. A term is represented by a **coefficient**(Scalar) and an exponent(nonnegative integer). The polynomial:$4x^2 + 3x - 7$, has 3 PolyTerms:

- $4x^2 - coefficient: 4, \text{exponent: } 2$
- $3x - coefficient: 3, \text{exponent: } 1$
- $-7 - coefficient: -7, \text{exponent: } 0$

This class should support the following services:

- **boolean canAdd(PolyTerm pt)**: receives a PolyTerm and returns **true** if the argument PolyTerm can be added to the current PolyTerm (same **power**). Otherwise, returns false.
- **PolyTerm add(PolyTerm pt)**: receives a PolyTerm and returns a new PolyTerm which is the result of adding the current PolyTerm and the argument.
- **PolyTerm mul(PolyTerm pt)**: receives a PolyTerm and returns a new PolyTerm which is the result of multiplying the current PolyTerm and the argument.
- **Scalar evaluate(Scalar scalar):** evaluates the current term using the scalar. For example, $2x^2$ with scalar 3, result is $2 * 3^2 = 18$.
- **PolyTerm derivate()**: returns the PolyTerm which is the result of the derivation on the current PolyTerm ($2x^2$ will return $4x$).
- **boolean equals(PolyTerm pt)**: returns **true** if the argument PolyTerm is equal to the current PolyTerm (same coefficient and power)

Polynomial

This class represents a polynomial. A polynomial is represented by its terms (i.e. a sequence of PolyTerms). **Note, a polynomial does not include two terms with same exponent.**

This class should support the following services:

- **Polynomial add(Polynomial poly)**: receives a Polynomial and returns a Polynomial which is the sum of the current Polynomial with the argument.

- **Polynomial mul(Polynomial poly)**: receives a Polynomial and returns a Polynomial is the multiplication of the current Polynomial with the argument.
- **Scalar evaluate(Scalar scalar)**: evaluates the polynomial using the argument scalar.
- **Polynomial derivate()**: return the Polynomial which is the result of applying first order derivation ($P'(x)$) on the current Polynomial.
- **String toString()**: returns a friendly representation sorted by increasing **power** of PolyTerms for example $1 + x + x^2 + \cdots$.

  Hint: The **Collections** class has a **sort** function for objects that implements **Comparable**. You can make the PolyTerm class Comparable and sort them.
- **boolean equals(Polynomial poly)**: returns true if the argument polynomial is equal to the current polynomial

Calculator

This class will hold the main method. Inside the main method the user will be asked for input.

In addition, every class should include **getters and setters and toString.** The **toString** method should return a friendly representation of the object. In addition, you are free to add more methods or classes.

2. Input and Output Formats

A polynomial $a_0 + a_1x + a_2x^2 + a_3x^3 +$.. is represented as follows:

(coefficient)+(coefficient)x^1+(coefficient)x^2+…

Without any spaces

Some examples:

- 3+5x^1-6x^2= $3 + 5x - 6x^2$
- 4x^3+7x^8+x^11= $4x^3 + 7x^8 + x^{11}$
- 2/5x^1-11/200x^20= $\frac{2}{5}x - \frac{11}{200}x^{20}$

Rational numbers will always appear in the form A/B.Real numbers will be printed up to 3 digits after the decimal point.

You can assume valid input! (without two terms with the same exponent, valid scalars,..)

The coefficient is a **scalar** (either Rational or Real, depending on the user input) and can also have a negative value! In this case, instead of a "+" sign between the terms, there will be a "-" sign (as you can see in the examples above).

אין צורך לצמצם מספרים רציונאליים בפלט, ניתן גם להשאיר במכנה את המספר 1 . ניתן
לקבל גם מספרים רציונאליים בקלט ללא קו שבר (מספרים שלמים, לדוגמא: 2x^1-4x^2)

3. Running the program

The program starts with showing the message:

```
Please select the scalar field
```

The user will either enter "R" for Reals or "Q" for Rationals. Then, the next menu will show:

```
Please select an operation:

   1. Addition
   2. Multiplication
   3. Evaluation
   4. Derivate
   5. Exit
```

The next step, the user will be asked to insert polynomials (two or one) in separate lines according to the above format and returns a result shown in a valid form (no exponent will be shown more than once, exponents appear in ascending order, and terms with coefficient zero do not appear).

If option 5 is selected, the program exits. Otherwise, the menu is shown again and the user can continue in a similar way.

4. Requirements

1. Design:
    a. Define the components that take part in the system and their responsibilities.
    b. Design an appropriate **UML Class Diagram**.

    To be submitted in a file named hw2.pdf

2. Implementation:
    a. Implement the program according to your design (class diagram) and the instructions.
    b. Use package/s for your classes, do not use the default package.

    To be submitted in a file named hw2.jar

5. Submission Instructions

Submit to the CS Submission System **a single archive file (.zip or .tar.gz)**with the following contents:

1. hw2.pdf
2. hw2.jar – should contain the **source code and compiled class files** (check that you can correctly run the file hw2.jar from the command line)

6. Examples

Example 1

➢ java – jar hw2.jar

```
Welcome to the polynomial calculator.
Please select the scalar field
Rational (Q) or Real (R)
Q
Please select an operation:
1.Addition
2.Multiplication
3.Evaluation
4.Derivate
5.Exit "
1
You have selected: Addition
Please insert the first polynomial
4+3x^1
Please insert the second polynomial
5+2x^1+4x^2
```

```
The solution is:
9+5x^1+4x^2
```

Example 2

- ➢ java – jar hw2.jar

```
Welcome to the polynomial calculator.
Please select the scalar field
Rational (Q) or Real (R)
Q
Please select an operation:
1.Addition
2.Multiplication
3.Evaluation
4.Derivate
5.Exit
2
You have selected: Multiplication
Please insert the first polynomial
1+3x^1
Please insert the second polynomial
5+x^2
The solution is:
5+15x^1+x^2+3x^3
```

Example 3

- ➢ java – jar hw2.jar

```
Welcome to the polynomial calculator.
Please select the scalar field
Rational (Q) or Real (R)
Q
Please select an operation:
1.Addition
2.Multiplication
3.Evaluation
4.Derivate
5.Exit
3
You have selected: Evaluation
Please insert the polynomial
5+15x^1+x^2
Please insert the scalar
2
The solution is:
39
```

Example 4

- ➢ java – jar hw2.jar

```
Welcome to the polynomial calculator.
Please select the scalar field
1) Rational Or 2) Real
1
"Please select an operation:
1.Addition
2.Multiplication
3.Evaluation
4.Derivate
5.Exit
4
You  have  selected:  Derivate
Please  insert  the  polynomial
5+15x^1+1/3x^2
The derivative polynomial is: 15+2/3x^1
```

Example 5

- ➢ java – jar hw2.jar

```
"Welcome to the polynomial calculator.
Please select the scalar field
Rational (Q) or Real (R)
R
"Please select an operation:
1.Addition
2.Multiplication
3.Evaluation
4.Derivate
5.Exit
4
You  have  selected:  Derivate
Please  insert  the  polynomial
5+15x^1+0.75x^2
The derivative polynomial is: 15+1.5x^1
```

בהצלחה