Ariel University, School of Computer Science, 2023

Introduction to Computer Science

Ex2 - Basic programming: static functions, arrays, and JUnit

Abstract:

This assignment introduces the use of functional programming and testing. The main task in this assignment is to implement a set of static functions over 1D arrays representing polynomials. Moreover, in this assignment, you are required to implement a JUnit class - to test all your functions.

To Do

1. Download Ex2 and run the classes Ex2_GUI.java, this class calls the Ex2.java class which you should implement (a basic skeleton with few implemented functions is given to you). Here is a Link to “how to start working” on Ex2.
2. Implement your solution in the given Ex2.java. Make sure your program creates the same GUI as shown in Figure 1. Here is a link to a detailed explanation regarding what is needed in each function.
3. Implement additional JUnit tests to Ex2Test. Several JUnit examples are given in class Ex2Test - make sure you add additional tests. Make sure **each** function in Ex2.java is being tested by Ex2Test.java (most probably you will need more than a single Testing function for each function in Ex2). Here is a Link to “how to start working with JUnit”
4. Add detailed documentation (in English) to Ex2.java and Ex2Test.java files with the related description for each function.

Notes:

1. Work alone! - you can talk about this assignment with anyone in class - but when writing your solution DIY!. Please go over this document which covers the School’s honesty policy.
2. Ex2 contains four classes: Ex2.java, Ex2Test, Ex2_GUI.java, StdDraw.java. The first two should be implemented the last two - should be kept “as is”.

3. The implementation of the function should be as efficient and elegant as possible.
4. Submission guidelines: your solution should be written as a java project the following two files are required: Ex2.java, Ex2Test.java, make sure to submit these two files and no other files.
5. Your solution should be submitted to Moodle according to the instructions - as presented to you in the TA sessions.

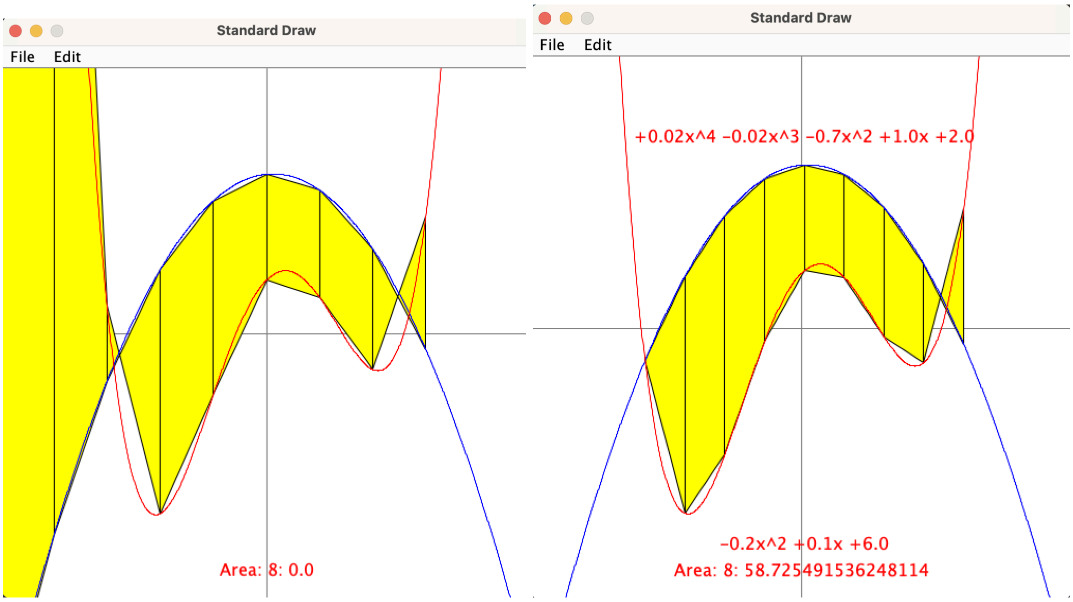
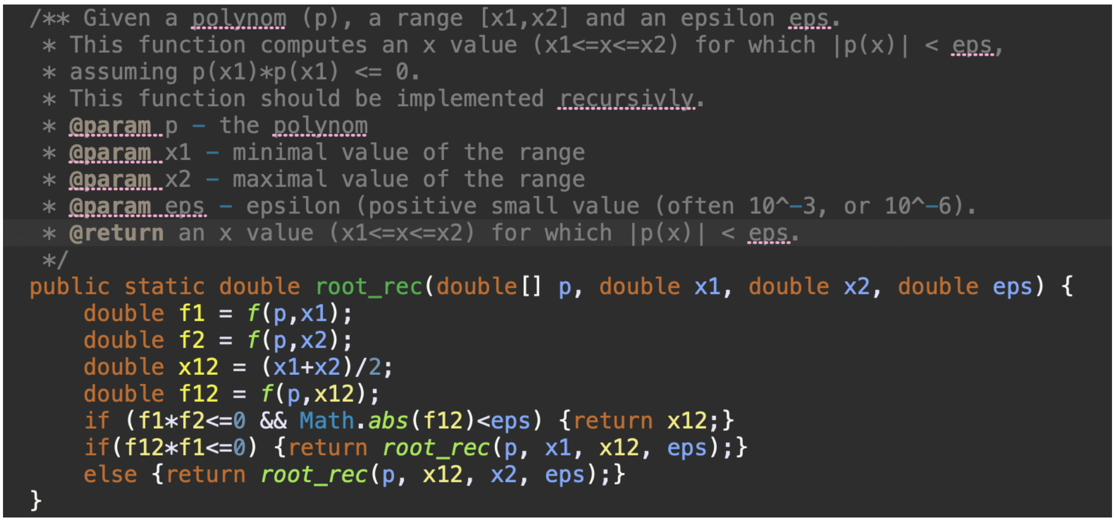


Figure 1: an example of running the attached Ex2_GUI.java file on Ex2.java, Left: without any additional implementation. Right: after implementing the needed functions.



```
/** Given a polynom (p), a range [x1,x2] and an epsilon eps.
 * This function computes an x value (x1<=x<=x2) for which |p(x)| < eps,
 * assuming p(x1)*p(x1) <= 0.
 * This function should be implemented recursivly.
 * @param p - the polynom
 * @param x1 - minimal value of the range
 * @param x2 - maximal value of the range
 * @param eps - epsilon (positive small value (often 10^-3, or 10^-6).
 * @return an x value (x1<=x<=x2) for which |p(x)| < eps.
 */
public static double root_rec(double[] p, double x1, double x2, double eps) {
    double f1 = f(p,x1);
    double f2 = f(p,x2);
    double x12 = (x1+x2)/2;
    double f12 = f(p,x12);
    if (f1*f2<=0 && Math.abs(f12)<eps) {return x12;}
    if(f12*f1<=0) {return root_rec(p, x1, x12, eps);}
    else {return root_rec(p, x12, x2, eps);}
}
```

Figure 2: the basic implementation of a recursive root computation (not required for Ex2).