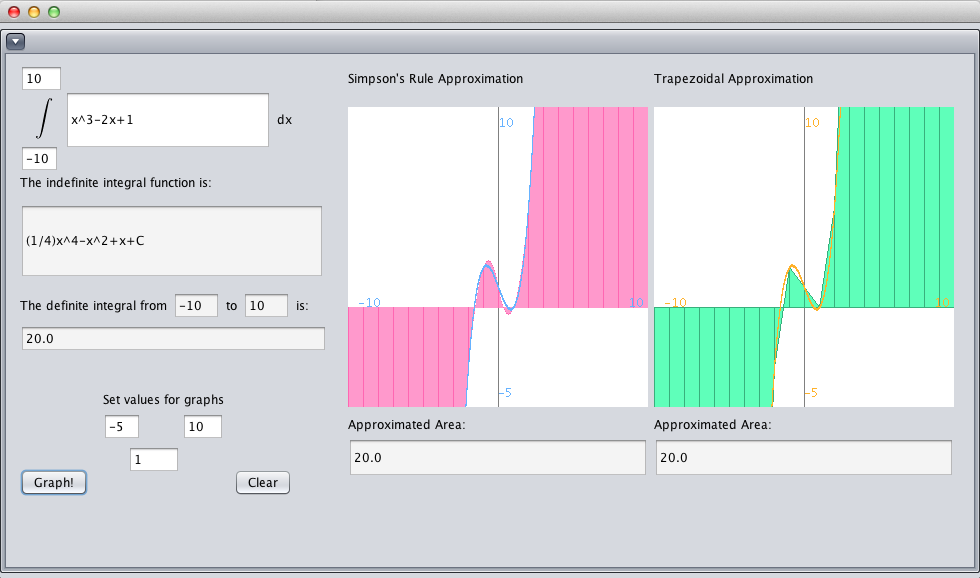
# Design Document: Area under Curve

## Description of the program

Area under Curve is a computer program that helps you investigate three ways of finding the area under a polynomial curve: finding the exact area by calculating the definite integral, and approximate the area with Simpson’s Rule and Trapezoidal Rule. By comparing the areas approximated by the two different rules with the exact area (found by using definite integral) under different conditions, one can see the difference in efficiency of the methods. The user can input a polynomial function and have the program calculate the integral function of the input function, as well as the exact area under the curve between the user-defined intervals. The function will also investigate the area under the curve with two approximation methods, Simpson’s Rule and Trapezoidal Rule. The user is able to define his own ∆x – the interval of approximation. The program will generate two graphs with the original input function and the approximated areas calculated with the two Rules respectively.



## Inputs to the program

* A polynomial function (with integer coefficients and exponents)
* An integer x-minimum value
* An integer x-maximum value
* A integer y-minimum value
* A integer y-maximum value
* A decimal or integer delta x value

Note: due to limitations of Simpson’s Rule, (xMax-xMin)/∆x must produce a even number, as Simpson’s Rule approximate one parabola over two ∆x’s. Please see: [Simpson's Rule](simpson.pdf). If the ∆x entered does not produce an even number of intervals between x minimum and x maximum, the program will return “Invalid ∆x” under “Approximated Area”

## Class

#### AreaUnderCurve Class

|  |  |
| --- | --- |
| Fields | Description |
| Simpson s; | Simpson object |
| Trapezoid t; | Trapezoid object |
| private javax.swing.JPanel SimpsonPanel;  private javax.swing.JPanel TrapezoidalPanel;  private java.awt.Canvas canvas1;  private java.awt.Canvas canvas2;  private javax.swing.JButton jButton1;  private javax.swing.JButton jButton2;  private javax.swing.JInternalFrame jInternalFrame1;  private javax.swing.JLabel jLabel1;  private javax.swing.JLabel jLabel10;  private javax.swing.JLabel jLabel11;  private javax.swing.JLabel jLabel12;  private javax.swing.JLabel jLabel2;  private javax.swing.JLabel jLabel3;  private javax.swing.JLabel jLabel4;  private javax.swing.JLabel jLabel5;  private javax.swing.JLabel jLabel6;  private javax.swing.JLabel jLabel7;  private javax.swing.JLabel jLabel8;  private javax.swing.JLabel jLabel9;  private javax.swing.JTabbedPane jTabbedPane1;  private javax.swing.JTextField jTextField1;  private javax.swing.JTextField jTextField10;  private javax.swing.JTextField jTextField11;  private javax.swing.JTextField jTextField12;  private javax.swing.JTextField jTextField2;  private javax.swing.JTextField jTextField3;  private javax.swing.JTextField jTextField4;  private javax.swing.JTextField jTextField5;  private javax.swing.JTextField jTextField6;  private javax.swing.JTextField jTextField7;  private javax.swing.JTextField jTextField8;  private javax.swing.JTextField jTextField9; | Variables declaration (automatically generated by netbeans) |
| Methods | Description |
| public AreaUnderCurve() | Constructor: initializes components |
| private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) | Creates a text field on GUI |
| private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) |
| private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) |
| private void jTextField4ActionPerformed(java.awt.event.ActionEvent evt) |
| private void jTextField5ActionPerformed(java.awt.event.ActionEvent evt) |
| private void jTextField6ActionPerformed(java.awt.event.ActionEvent evt) |
| private void jTextField7ActionPerformed(java.awt.event.ActionEvent evt) |
| private void jTextField8ActionPerformed(java.awt.event.ActionEvent evt) |
| private void jTextField9ActionPerformed(java.awt.event.ActionEvent evt) |
| private void jTextField10ActionPerformed(java.awt.event.ActionEvent evt) |
| private void jTextField11ActionPerformed(java.awt.event.ActionEvent evt) |
| void jButton1ActionPerformed(java.awt.event.ActionEvent evt) | Clears all text fields |
| void jButton2ActionPerformed(java.awt.event.ActionEvent evt) | Reads the user input in text fields to set the canvas dimensions and creates a Simpson and a Trapezoid object to perform calculations and graphing |
| public static void main(String args[]) | Runs the program |

#### Polynomial Class

|  |  |
| --- | --- |
| Fields | Description |
| ArrayList<Integer> exponents | An array list of integers that are exponents of the terms in a function |
| ArrayList<Fraction> coefficients | An array list of Fractions that are coefficients of the terms in a function |
| yval[] | Array for y-coordinates of points on function |
| Methods | Description |
| public Polynomial (String p) | Constructor: reads a string into polynomial, sets the exponents array list and coefficients array list. yval[] is only set when plotPoints function is called |
| public Polynomial (ArrayList<Fraction> coef, ArrayList<Integer> expo) | Constructor: takes an array list of Fractions and append it to the coefficients and an array list of integers and append it to the exponents |
| public void fillArrayList(String p) | Fills array lists for coefficients and exponents if the constructor that takes a string is called |
| public String displayPolynomial() | Displays polynomial in expanded form |
| public double calcY(double x) | Calculates the y value of a polynomial function at a given x value |
| public void plotPoints(int width, int xMin, double increX) | Appends y-coordinates calculated by calcY()of points on function into the yval[] array |
| public Polynomial findIndefiniteIntegral() | Calculates the indefinite integral of the function |
| public double calcDefiniteIntegral(double xMin, double xMax) | Uses the indefinite integral function to calculate the definite integral over a given interval |

#### Fraction Class

|  |  |
| --- | --- |
| Fields | Description |
| int numerator | Keeps the numerator of the fraction |
| int denominator | Keeps the denominator of the fraction |
| double doubleVal | Keeps the decimal value of the fraction |
| Methods | Description |
| public Fraction(int n, int d) | Constructor: Takes two integers and append them to the numerator and the denominator |
| public Fraction(double a) | Constructor: Allows the coefficients of function to be appended as a double value |
| public static int findGCD(int num1, int num2) | Finds greatest common denominator of two numbers using Euclid's method |
| public Fraction reduceFraction() | Reduces the fraction using the findGCD method |
| public String displayFraction() | Puts the values into fraction form |

#### Trapezoidal Class

|  |  |
| --- | --- |
| Fields | Description |
| Polynomial f | Polynomial function which the program performs approximation on |
| double deltaX | The interval which the program approximate one area over |
| JPanel | The panel for graphing |
| int width | Width of panel/screen |
| int length | Length of the panel/screen |
| Fields for Graphing | Description |
| int xMin, xMax, yMin, yMax | Dimensions of graph (set by user) |
| int yPos, xPos | Position on screen of y-axis/x-axis, calculated with length/width, yMin/xMin, yMax/xMax |
| double incrementY, incrementX | Calculates increments of x and y values with ength/width, yMin/xMin, yMax/xMax. Each pixel across the screen has an increment of x or y based on the values calculated here |
| Methods | Description |
| public Trapezoid(double delX, Polynomial function, JPanel jp) | Constructor: Sets all fields except the fields for graphing |
| public double calcTrapeArea(double x1) | Calculates area of one trapezoid at a given point |
| public double approxAreaTrape(double xMin, double xMax) | Approximate area under curve by adding up trapezoids at different points |
| public void paintImage(){ | Graphing function |

#### Simpson Class

|  |  |
| --- | --- |
| Fields | Description |
| Polynomial f | Polynomial function which the program performs approximation on |
| double deltaX | The interval which the program approximate one area over |
| JPanel | The panel for graphing |
| int width | Width of panel/screen |
| int length | Length of the panel/screen |
| Fields for Graphing | Description |
| int xMin, xMax, yMin, yMax | Dimensions of graph (set by user) |
| int yPos, xPos | Position on screen of y-axis/x-axis, calculated with length/width, yMin/xMin, yMax/xMax |
| double incrementY, incrementX | Calculates increments of x and y values with ength/width, yMin/xMin, yMax/xMax. Each pixel across the screen has an increment of x or y based on the values calculated here |
| Methods | Description |
| public Simpson(double delX, Polynomial function, JPanel jp | Constructor: Sets all fields except the fields for graphing |
| public static Polynomial solveMatrix (double[] q1, double[] q2, double[] q3) | Finds solution to a linear system with three equations |
| public Polynomial calcParabola(double startingX) | Calculates equation of parabola with three points, starting from a given point |
| public double calcAreaUnderParabola(Polynomial p, double startingX) | Takes the function of the parabola as an argument and calculates the area under a parabola over three points/two delta X's using the Simpson's Rule formula |
| public double approxAreaSimp(double xMin, double xMax) | Approximates area over the interval between x minimum and x maximum |
| public double[] drawParabola(Polynomial p, double startingX) | Takes function of estimated parabola and returns an array of values on the parabola |
| public void paintImage(){ | Graphing function |