ICS4U Assignment

Inheritance

The goal of this assignment question is to familiarize yourself with inheritance, polymorphism and abstract classes and interfaces.

General Description of the Problem

You are to classify a certain subset of mathematical functions and implement them in Java using inheritance. The base class of all functions will be the **Function** class.

The Function Class

The base class Function defines the following:

1) the domain of the function.
2) the name of the function.
3) the colour that the function is to be drawn in (as an RGB triplet, each colour being one byte)

Inheriting from this class, you will create the following additional mathematical functions:

- Line of the form $y = m(x - x_1) + b$
- Arc of the form $y = \sqrt{r^2 - (x - x_1)^2 + y_1}$
- Natural Logarithm of the form $y = a\ ln(x - x_1) + b$
- Quadratic of the form $y = a(x - x_1)^2 + b(x - x_1) + c$
- Cubic of the form $y = a(x - x_1)^3 + b(x - x_1)^2 + c(x - x_1) + d$
- Parabola of the form $y = a(x - x_1)^2 + b$

where a, b, c, d, x_1, y_1 are constants and x is the variable of the function $y = f(x)$.

There are seven classes to implement, one for each class of functions defined above. The names of the function classes along with their constructor signatures are given as:

1) Linear class	**Linear**(double m, double b, double x1)
2) Arc class	**Arc**(double r, double xcenter, double ycenter)
3) Logarithm class	**Logarithm**(double a, double b, double x1)
4) Quadratic class	**Quadratic**(double a, double b, double c, double x1)
5) Cubic class	**Cubic**(double a, double b, double c, double d, double x1)
6) Parabola class	**Parabola**(double a, double b, double x1)

The provided Function class also has the following method signatures which need to be completed:

public abstract Function(double x1, double x2)
Constructor method for the Function class. The parameters represent the domain of the function.

public abstract String toString();
Returns a String containing the actual function. For example, if the function was an object of the Parabola class, this method might return 2.0*(x - 4.0)^2 + 5.0. There are spaces before and after the +, - operators **only**.

public void setDomain(double x1, double x2)
Sets the domain of this function to be between $[x1, x2]$ where $x2 > x1$.

public double getStartDomain()
Returns the starting value of the domain.

public double getEndDomain()
Returns the ending value of the domain.

public void setColour(Color col)
Sets the drawing colour for this function. The Color class is from JavaFX.

public int getColour()
Returns the colour of this function.

public void setName(String name)

Sets the name of this function type.

public String getName()
Returns the name of this function type as a String.

When writing your subclassed functions, use inheritance and polymorphism **whenever possible.**

In addition to the constructors, for each of the above classes you will implement the following Interfaces:

Drawable interface:

```
public void draw(Canvas c);
```

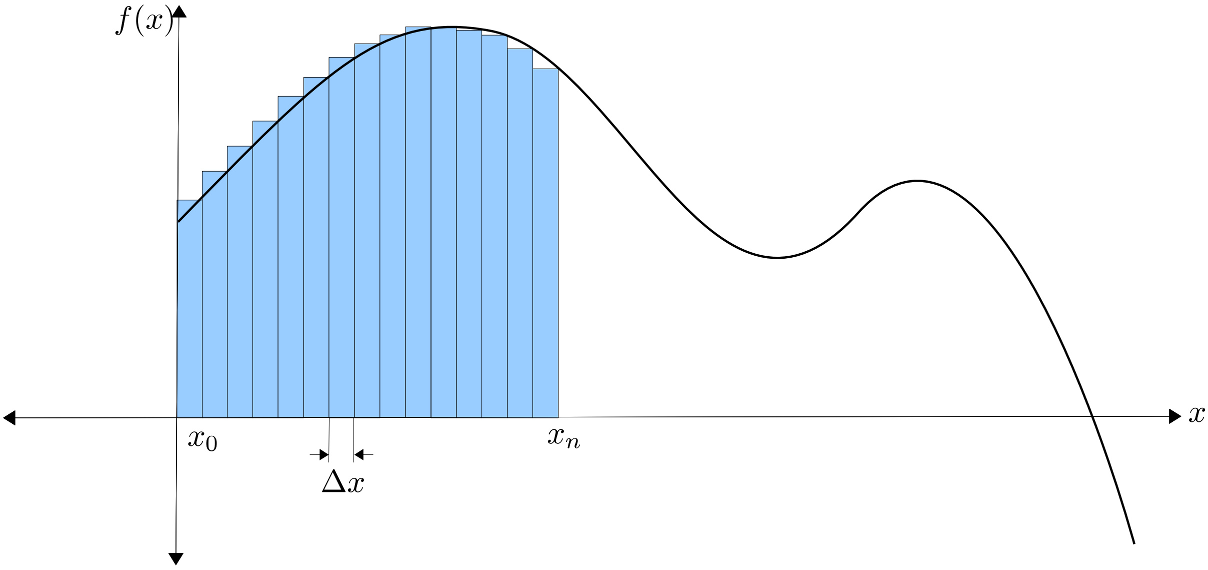
Calculations interface:

```
public double val(double x);
public boolean undefined(double x);
public double getArea(double x_start, double x_end);
public double getSlope(double x);
```

These files are posted on the Google classroom.

Implemention Notes

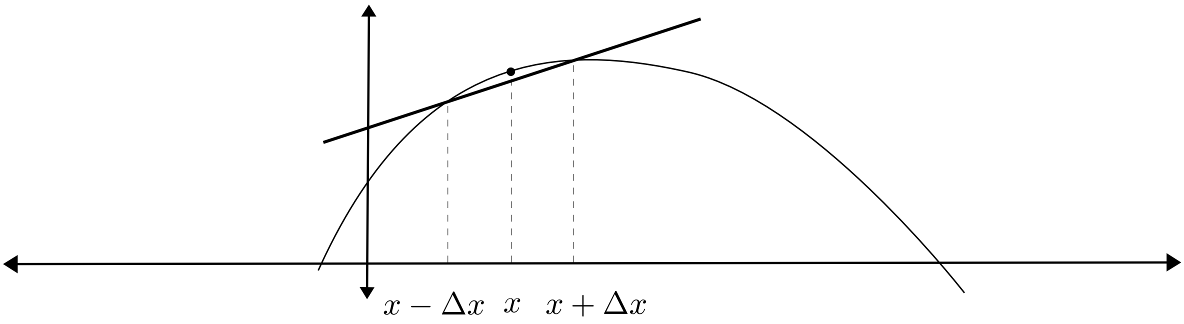
Calculation of area under graph:
For the getArea() method, you will calculate the area under the function by splitting up the graph into very thin rectangles and adding them up to approximate the total area. This approach is shown below:



Given a starting point x_0, and an ending point x_n, we divide up the area under the curve into small rectangles each of width Δx and a height of $f(x_i)$. Adding up all of these rectangle approximates the area under the curve:

$$Area \approx \sum_{i=x_0}^{x_n} f(x_i)\Delta x$$

Calculation of the slope at a point on the graph:
To calculate the slope at a point x on the graph we will approximate it by calculating the slope of the secant line whose ends are to the left and right of the point x:



The slope can then be calculated as:

$$Slope = \frac{\Delta y}{\Delta x} = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}$$

ICS4U

The FunctionTester Class

This is the class that contains the main method. Create the following functions:

1) $f(x) = a(x - x_1)^2 + b$ where $a = 1.0, b = 0.0, x_1 = 0.0$
2) $f(x) = m(x - x_1) + b$ where $m = 1.0, b = 1.0, x_1 = 0.0$
3) $f(x) = \sqrt{r^2 - (x - x_{centre})^2} + y_{centre}$ where $r = 4.0, x_{centre} = 0.0, y = -2.0$
4) $f(x) = a(x - x_1)^2 + b(x - x_1) + c$ where $a = 0.25, b = -0.5, c = -1.0, x_1 = 0.0$
5) $f(x) = a(x - x_1)^3 + b(x - x_1)^2 + c(x - x_1) + d$ where $a = 0.35, b = 0.25, c = -0.5, d = -1.0, x_1 = 2.0$
6) $f(x) = a\ \log(x - x_1) + b$ where $a = 1.0, b = 0.0, x_1 = 0.0$

Testing

Make sure to test your program with the six functions provided. Verify that the computed results match the actual values. You can try other test cases if you like. Your program should also display your functions using JavaFX. There is a JavaFX template provided to illustrate how to use the Canvas and GraphicsContext classes.

What to hand In

1. A diagram visually describing your inheritance structure as shown in class.
2. A archived version of your Eclipse project zipped up, submitted to the course web site.

Your project should contain the Java source code for the six functions:

Linear.java
Quadratic.java
Parabola.java
Logarithm.java
Arc.java
Cubic.java

along with the Function.java and FunctionTester.java file that will test your Function class. For completeness, you archived file should also contain copies of the Drawable and Calculations interface files.

- State any assumptions that you made when writing your program.
- State any conditions under which your program may fail and provide an explanation of how and why it may fail.