Planning

Contents

1. Defining the problem

Mr Christos, my teacher in further maths, very often wants to draw graphs in one of his classes. He either does that by hand or with the aid of a GDC software that he has installed on the smartboard. However, in order for his students to have these graphs at home, they have to copy them on paper, along with any information calculated e.g. the maximum and minimum points etc[1]. This takes up valuable time of the lesson and distracts the students' attention. Moreover, it is possible that mistakes are made during the copy process. Therefore, he wants a solution to this problem, a way to easily draw graphs, find relative data, such as turning points, the area under the graph, the volume of revolution, numeric derivatives and other similar information for the function, and have it sent over to his students neatly.

2. Rationale for the proposed solution

The solution that I aim to implement is a program that graphs functions and allows any kind of calculations similar to a GDC, but also saves the graphs together with all this information about them in pdfs, which are well formatted and can then, either automatically or manually, be sent via email to the students of the correct class.

I will be using java for this task, because it is an object-oriented, platform-independent programming language, allowing me to create "Class" objects, representing each class that Mr. Christos teaches, as well as "Student" objects, holding the name and email addresses of each class's students. This is much harder in python, a programming language that I also like coding in, but was thus force to reject for this project. Moreover, through the javafx library, java makes it very easy both to create a GUI and to draw graphs. In java, it is also very easy to save these graphs in pdf and send them through email. Finally, the external library mXParser allows user-defined functions to be implemented. The above functionalities are harder to implement in other OOP languages such as C++. Thus, all in all, I found it reasonable to implement the solution in java, in the bluej IDE, with which I am familiar.

Word count: 356

[1] See Appendix A

3. Criteria of success

1. Program graphs any continuous function within some range
2. Program calculates local minimum points
3. Program calculates local maximum points
4. Program calculates the y-coordinate for any given x-value
5. Program finds roots to the function (x-intercepts)
6. Program finds y-intercept
7. Program finds numerical derivative at a given value of x
8. Program calculates the area under the graph
9. Program calculates the area between the graph and the x-axis
10. Program calculates the volume of revolution
11. Program allows graphing multiple functions
12. Program finds intersection of functions
13. Program saves graphs, along with any elements found in pdf
14. Program emails pdf to the correct students if there is internet connection, otherwise it does not crash
15. Program allows user to edit classes so as to always be up to date with new students, students who have dropped the class or new classes in the beginning of every year.
16. Program runs on smartboard
17. User-friendly GUI that does not crash