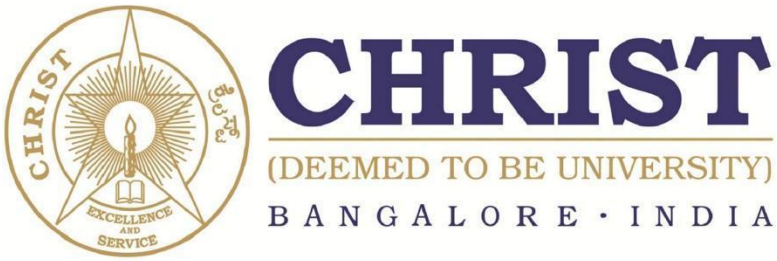


GRAPH MASTER

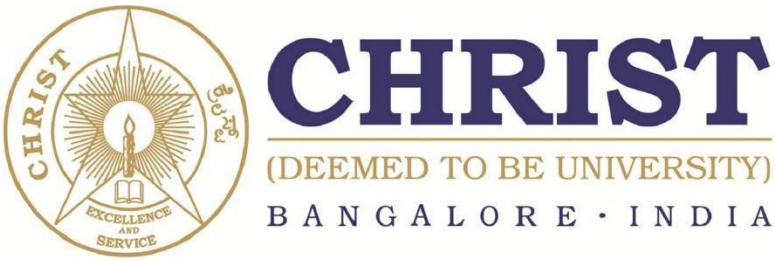
A Java Application for Graphing Mathematical Functions

Submitted by

ASHVATH S.P - 2162014

YAMINI.K - 2162054

D.SAI PRANATHI - 2162071



CHRIST (DEEMED TO-BE UNIVERSITY)

FACULTY OF ENGINEERING

VISION

Excellence and Service

MISSION

CHRIST is a nurturing ground for an individual's holistic development to make an effective contribution to the society in a dynamic environment.

CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

To fortify Ethical Computational Excellence

MISSION

- ✓ Imparts core and contemporary knowledge in the areas of Computation and Information Technology
- ✓ Promotes the culture of research and facilitates higher studies
- ✓ Acquaints the students with the latest industrial practices, team building and entrepreneurship
- ✓ Sensitizes the students to serve for environmental, social & ethical needs of society through lifelong learning.

ABSTRACT

Graph Master is a computer program designed to help users create, manipulate, and analyse graphs. The software provides a user-friendly interface that allows users to export data in various formats, visualise the data, and customise the charts to their specific needs. With its intuitive interface and powerful features, Graph Master is essential for anyone working with charts and data visualisation.

The GraphMaster project aims to develop a Java application for graphing mathematical functions. The application can plot 2D graphs for linear, quadratic, trigonometric, and exponential functions. The user-friendly interface allows users to enter the function and range of values to be plotted. The application uses Java concepts such as Exception Handling, Event Handling, and Swings to implement the calculator functionality and provide a smooth user experience.

Graph-oriented programming is a new programming paradigm that defines a graph-oriented way to build enterprise software using directed attributed graph databases on the backend side.

REQUIREMENT SPECIFICATION

INTRODUCTION

The graphical calculator is a software application providing a visual and user-friendly interface for mathematical calculations. It emulates a calculator with a graphical user interface (GUI) using Java programming language, allowing users to input mathematical expressions and obtain results in a visually appealing and intuitive manner. The graphical calculator is a popular project for learning GUI programming in Java and developing desktop applications.

The graphical calculator project typically includes basic arithmetic operations (addition, subtraction, multiplication, and division), error-handling mechanisms and a user-friendly interface with buttons for input and display panels for results. Depending on the project's requirements, the calculator may support other standard mathematical functions, such as square root, exponentiation, and trigonometric operations.

The primary objective of the graphical calculator is to provide a convenient and efficient tool for performing mathematical calculations, emphasising ease of use, accuracy, and error-free operation. The graphic calculator can be used for various purposes, including simple calculations in everyday life, educational purposes, or as a tool for engineering, finance, or science professionals.

SCOPE

The scope of the Graphical Calculator project can vary depending on the requirements and objectives of the project. However, here are some common aspects that can be considered within the scope of a typical graphical calculator project:

1. Graphical User Interface (GUI): The project would include designing a GUI using Java's Swing toolkit to create a visually appealing and user-friendly interface for the calculator. This would involve creating buttons, text fields, labels, and other components for user input and displaying results.
2. Basic Arithmetic Operations: The calculator should support standard arithmetic operations such as addition, subtraction, multiplication, and division, as well as other normal mathematical functions like square root, exponentiation, and trigonometric operations.
3. Error Handling: The calculator should include error handling mechanisms to handle invalid inputs, division by zero, and other error conditions, providing informative error messages to the user.
4. User-friendly Interface: The calculator should have a simple and intuitive interface with large buttons, clear display panels, and efficient navigation to ensure ease of use for the end-users.
5. Testing: The project should include thorough testing to ensure the calculations' accuracy and reliability and the calculator's functionality in different scenarios and edge cases.
6. Documentation: The project should be well-documented, including documentation that provides an overview, implementation details, screenshots of the GUI, instructions for using the calculator, and other relevant information.
7. Extensibility: The project may be designed with extensibility in mind, allowing for the easy addition of new features or enhancements in the future,

such as adding more advanced mathematical operations, scientific functions, or customisation options.

FUNCTIONAL REQUIREMENTS

1. Graph creation: The software should allow users to create various graph types, including bar charts, line graphs, scatter plots, and more. Users should be able to customise the graphs with different colours, labels, and other design elements.
2. Graph manipulation: Users can manipulate graphs by resizing and zooming.
3. Data export: The program should allow users to export their graphs in various formats, including PNG.
4. User interface: Graph Master should provide a user-friendly interface that makes it easy for users to create and manipulate graphs.
5. Compatibility: Graph Master should be compatible with various operating systems, including Windows, Mac, and Linux.

NON-FUNCTIONAL REQUIREMENTS

1. Performance: The graphical calculator should provide fast calculations and graphing capabilities with minimal lag or delays.
2. Accuracy: The calculator should provide accurate results for calculations and graphs with minimal rounding errors or inaccuracies.
3. Reliability: The calculator should be reliable, with minimal crashes, errors, or unexpected behaviours.
4. Usability: The calculator should have a user-friendly interface that is easy to use, intuitive navigation and clear instructions.
5. Portability: The calculator should be able to run on different platforms or devices with consistent functionality and performance.
6. Maintainability: The calculator should be designed and implemented for easy maintenance and updates.
7. Scalability: The calculator should handle varying levels of data or calculations without significant performance degradation.
8. Interoperability: The calculator should be able to interact with other software or systems using standard formats or protocols.

ASSUMPTIONS AND DEPENDENCIES

ASSUMPTIONS

1. Data quality: Graph Master assumes that the data the user provides is accurate and quality. Inaccurate or incomplete data can lead to incorrect or misleading graphs.
2. User knowledge: Graph Master assumes users understand graphing and data analysis.
3. Operating systems: Graph Master assumes that users have a compatible operating system, such as Windows, Mac, or Linux.

DEPENDENCIES

1. Hardware: Graph Master depends on the hardware specifications of the user's computer or device. The program may need to run more efficiently and effectively on older or lower-spec devices.
2. Third-party software: Graph Master may depend on software like Java to function correctly. Users may need to install these dependencies before using Graph Master.
3. Data sources: Graph Master depends on the availability and compatibility of data sources.

WORKING DESCRIPTION

GraphMaster is a Java-based application with a user-friendly interface for graphing various mathematical functions in 2D space. Its purpose is to provide a powerful tool for students, researchers, and professionals in multiple fields of science and engineering.

The application provides a graphical user interface for creating and displaying graphs of mathematical functions. Upon launching the application, users are presented with a window containing fields to enter the mathematical function and the range of values to be plotted. Functions can be entered using standard mathematical notation, such as y = sin(x) or y = 2x + 3. GraphMaster can plot various types of mathematical functions, including:

1. Linear functions, such as y = mx + b
2. Quadratic functions, such as y = ax^2 + bx + c

3. Trigonometric functions, such as y = sin(x) or y = cos(x)
4. Exponential functions, such as y = e^x or y = a^x

The application implements its functionality with Java concepts such as Exception Handling, Event Handling, and Swings, making it reasonably efficient and capable of graphing functions. The interface is designed to be user-friendly, with options to save the plotted graphs in standard image formats such as PNG.

GraphMaster consists of three packages:

1. The grapher package provides classes for visualising mathematical expressions and functions.
2. The parser package provides classes for parsing mathematical expressions from text input.
3. The main package, called graphmaster, provides the main entry point for the application and initialises the user interface.

The parser and expressions packages create expression objects from text input, which are evaluated and displayed in the graph window. The window provides controls for adjusting the graph's appearance, such as the range of the x and y axes, the style of the plot, and the type of function being graphed.

GraphMaster is platform-independent and developed using Java, allowing users to run it on different operating systems. The application provides appropriate error messages to the users in case of errors or exceptions.

GraphMaster is a valuable tool that simplifies graphing various mathematical functions, making it useful for students, researchers, and professionals in multiple fields such as physics, engineering, and economics. Whether you need to graph a simple linear function or a more complex exponential function, GraphMaster provides a quick and easy way to visualise your data.

SNAPSHOTS

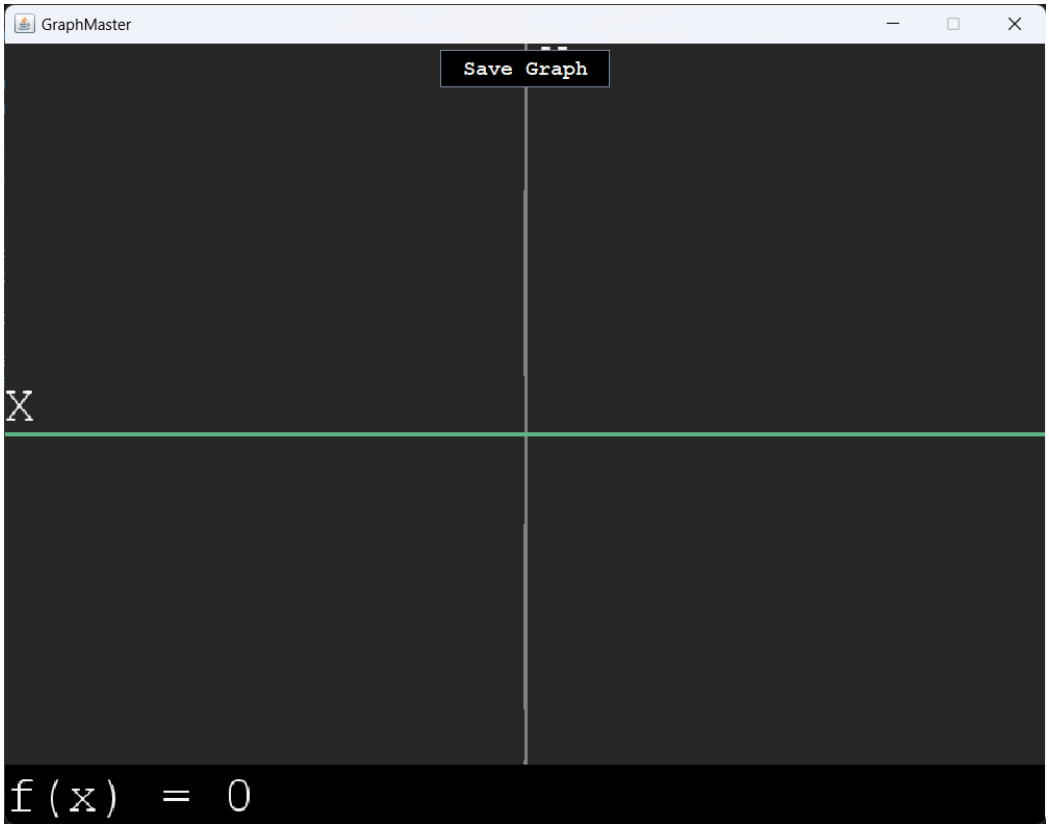


Figure 1: GraphMaster Home Screen Window.

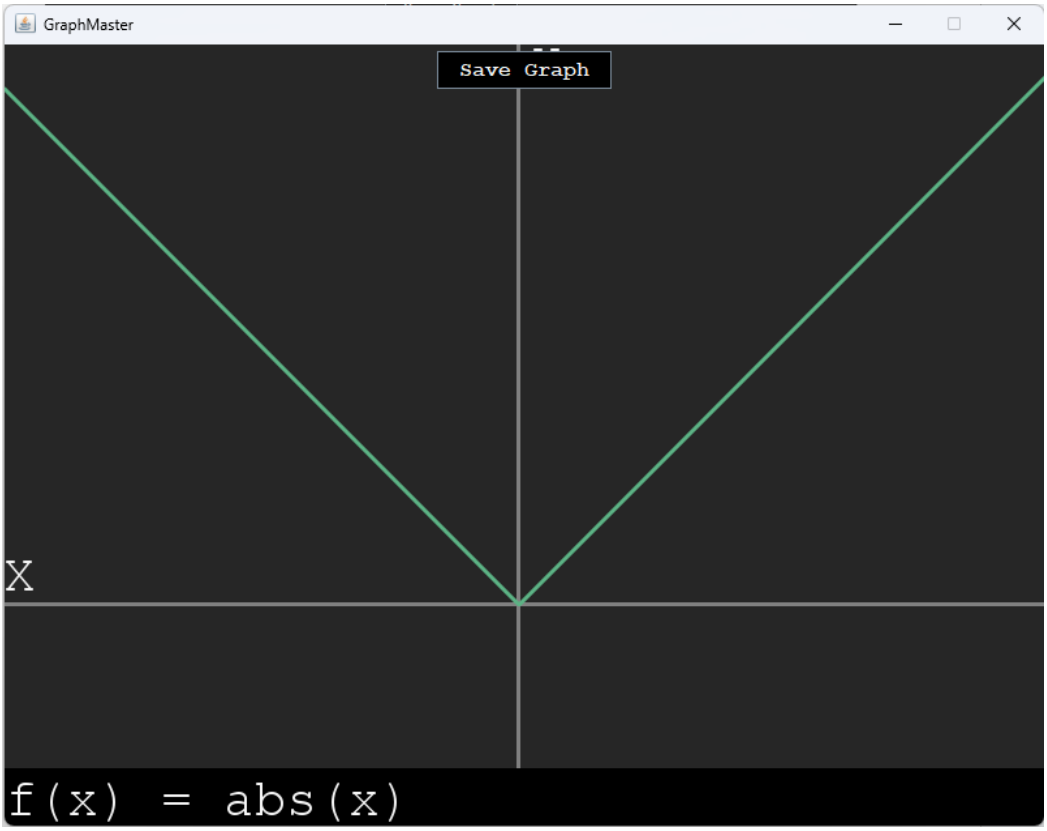


Figure 2: Graph of the absolute value function (|x|) plotted using the application.

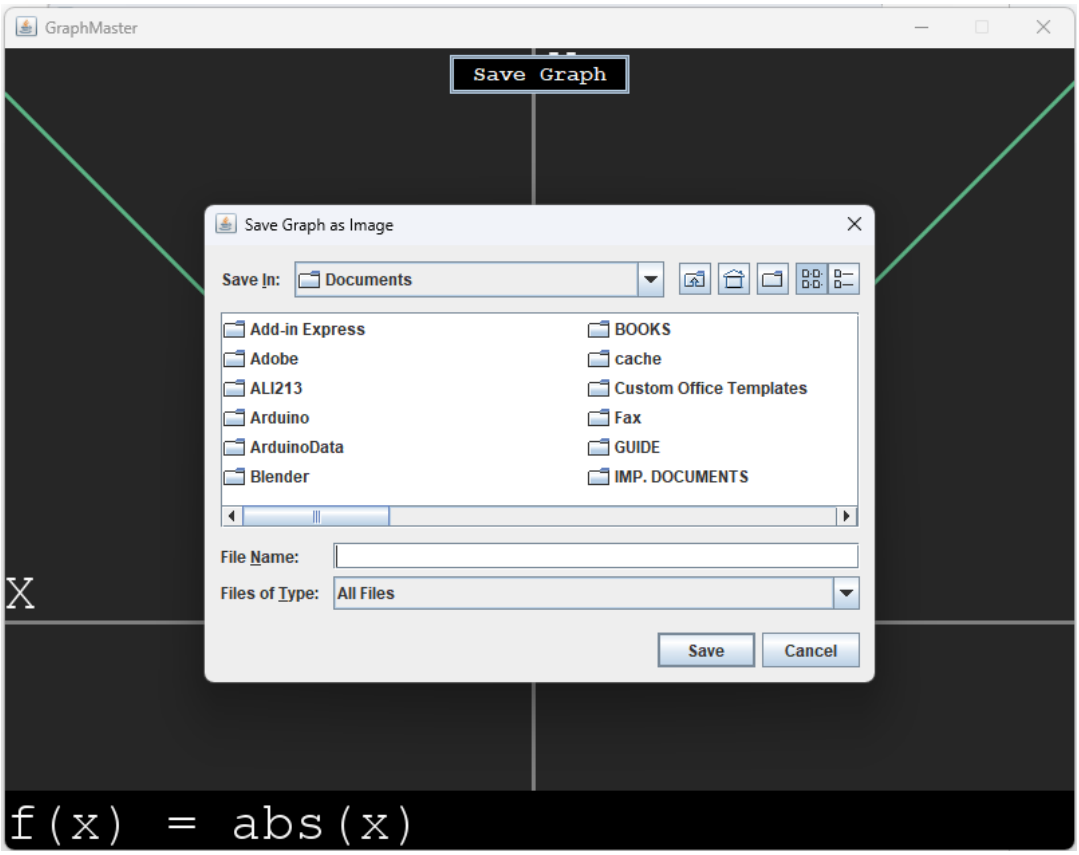


Figure 3: "Save Graph as Image" Window

This image shows the "Save Graph as Image" window that appears after clicking the "Save Graph" button in the GraphMaster application.

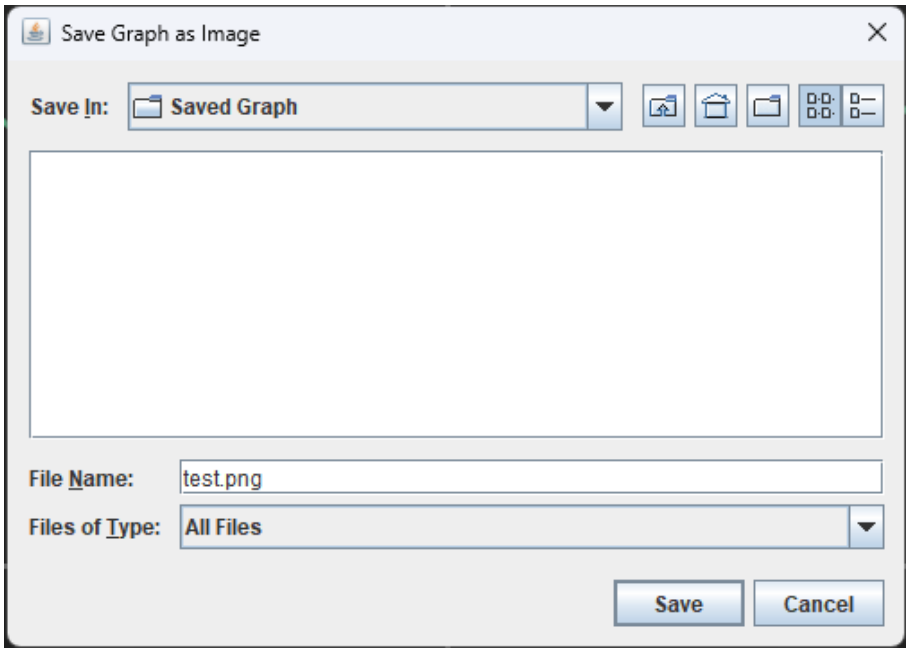


Figure 4: The "Save Graph as Image" window after selecting the desired directory and specifying the file name as "test".

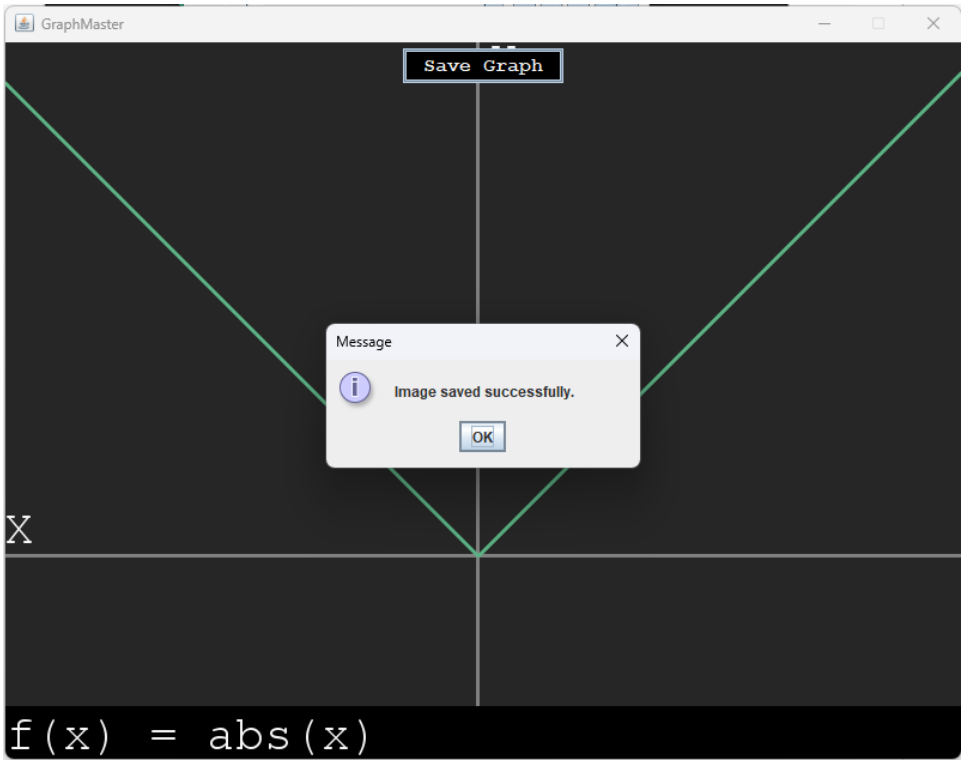


Figure 5: The "Message" window displayed after saving the graph in the system.

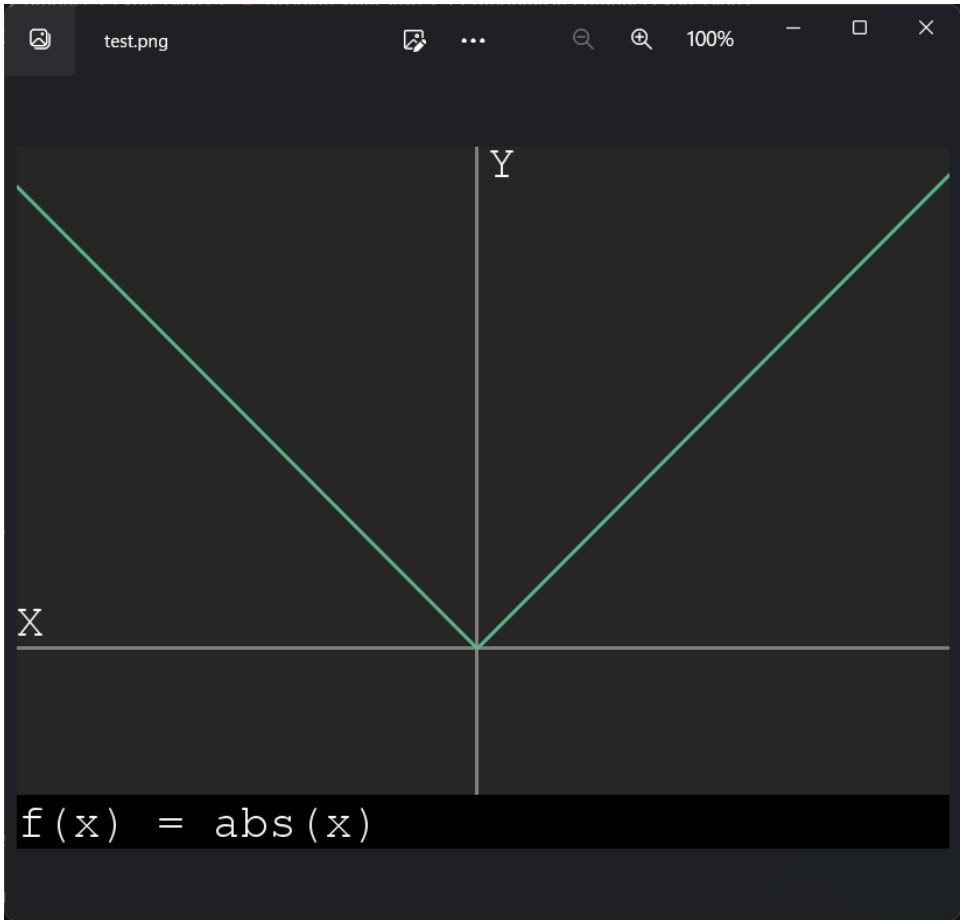


Figure 6: The "test.png" window in the system.

Figure 7: Graph of Nested Functions Displayed in Application Window

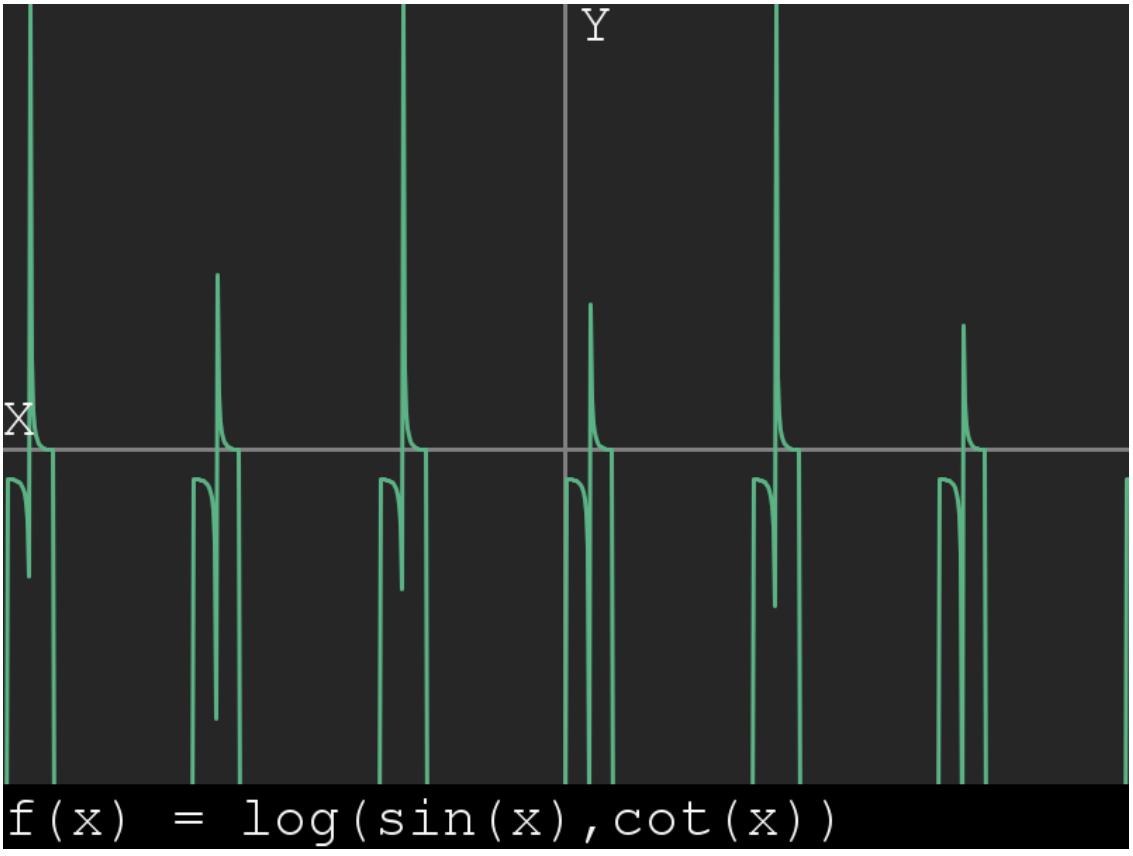


Figure 8: Graph of the Nested Log Function

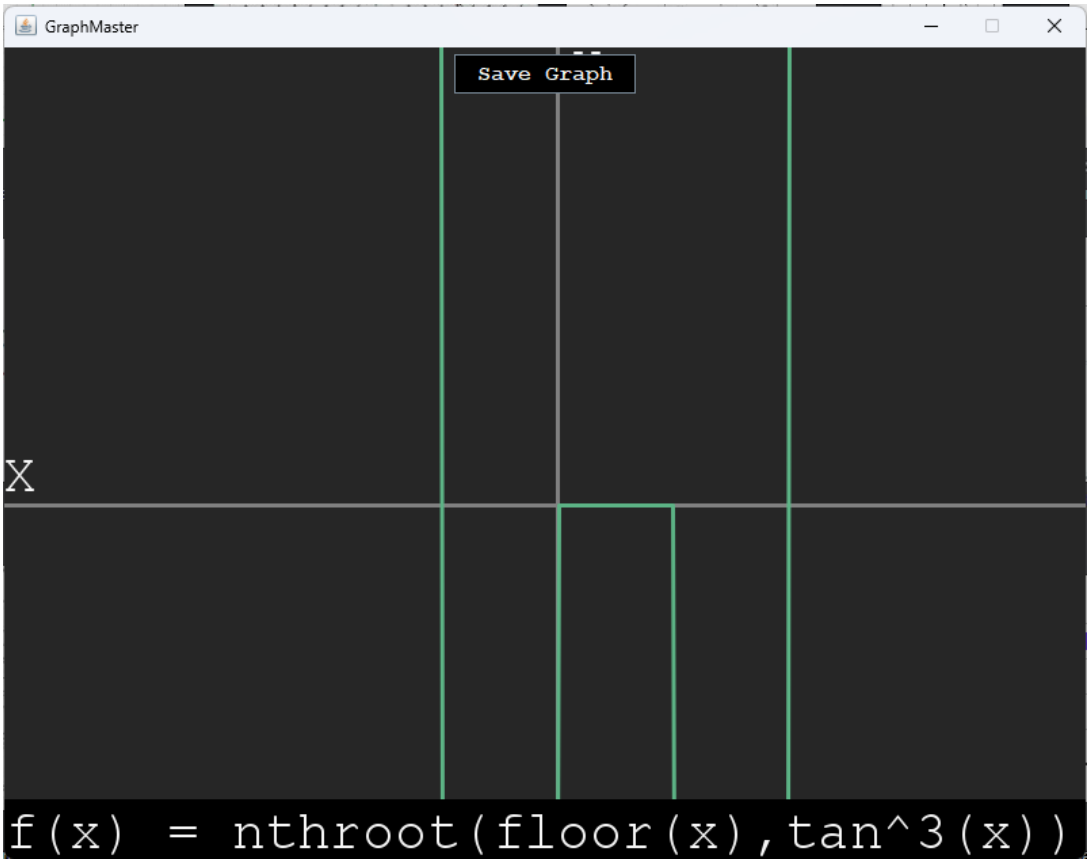


Figure 9: Application window displaying a Graph of Nested nthroot Function

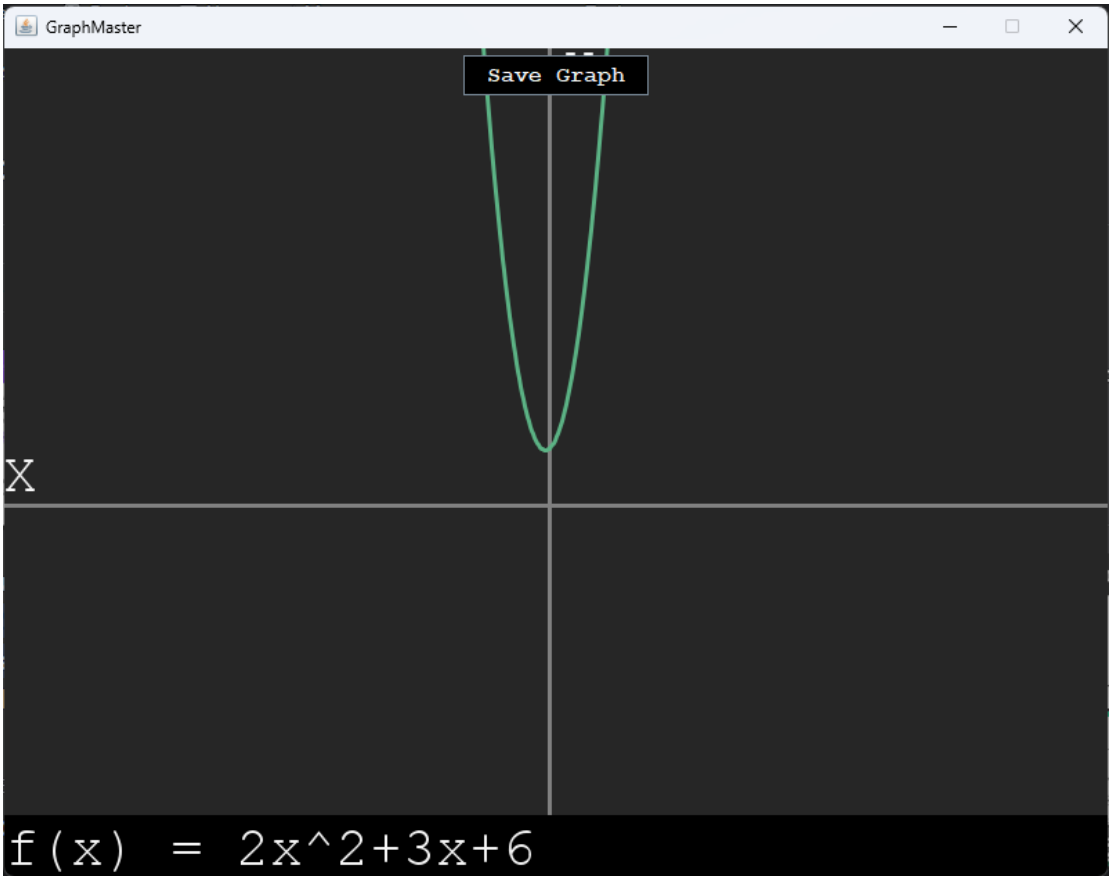


Figure 10: Application window displaying a graph of the quadratic equation.

CONCLUSION

In conclusion, a graphical calculator is a powerful and versatile tool that provides advanced mathematical functionalities in a graphical format. It offers a user-friendly interface, extensive mathematical capabilities, and usability features that make it useful in various real-time scenarios such as education, science and engineering, business and finance, professional use, and everyday tasks. Graphical calculators enable users to perform complex calculations, graph functions, analyse data, and make informed decisions efficiently and accurately. With their ease of use and real-time capabilities, graphical calculators are valuable tools for students, educators, scientists, engineers, and other professionals who require advanced mathematical functionalities in their day-to-day activities.

REFERENCES

1. Oracle. (2011). javax.swing Package Summary. Retrieved from https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html
2. Oracle. (n.d.). java.awt package summary. Retrieved from https://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html
3. Oracle. (n.d.). ImageIO (Java SE 7 & JDK 7). Retrieved April 14, 2023, from https://docs.oracle.com/javase/7/docs/api/javax/imageio/ImageIO.html
4. FuzzyCat - [Java/Swing] Function Grapher Program Part 1. (2018, May 9th).
5. Oracle. (n.d.). Math (Java SE 8). Retrieved April 14, 2023, from https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html
6. Oracle. (n.d.). Image (Java Platform SE 7). Retrieved from https://docs.oracle.com/javase/7/docs/api/java/awt/Image.html
7. Oracle. (n.d.). Graphics2D (Java Platform SE 7). Oracle Help Center. Retrieved April 14, 2023, from https://docs.oracle.com/javase/7/docs/api/java/awt/Graphics2D.html