Group 15: MIS

Generated by Doxygen 1.8.13

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

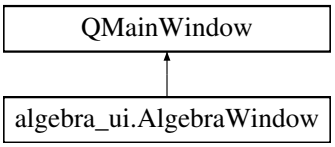
Chapter 4

Class Documentation

4.1 algebra_ui.AlgebraWindow Class Reference

AlgebraWindow is a class that implements the GUI components for the Algebra operation menu.

Inheritance diagram for algebra_ui.AlgebraWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the Algebra window.

Public Attributes

- **path**
- **xpath**
- **slope1**
- **slope2**
- **pytha**

4.1.1 Detailed Description

AlgebraWindow is a class that implements the GUI components for the Algebra operation menu.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 __init__()

```
def algebra_ui.AlgebraWindow.__init__ (
              self,
              path = "" )
```

The constructor of the Algebra window.

Creates a pop up window that displays and sets up the buttons that are necessary to navigate from the Algebra window to other parts of the application. Also sets up the Algebra window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

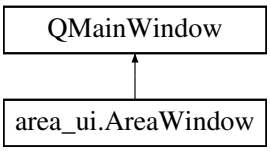
The documentation for this class was generated from the following file:

- src/uis/algebra_ui.py

4.2 area_ui.AreaWindow Class Reference

AreaWindow is a class that implements the GUI components for the Area operation.

Inheritance diagram for area_ui.AreaWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the Area window.
- def area (self)

  Displays the area of selected shape given appropriate side lengths/radius.

Public Attributes

- **path**

4.2.1 Detailed Description

AreaWindow is a class that implements the GUI components for the Area operation.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 __init__()

```
def area_ui.AreaWindow.__init__ (
              self,
              path = "" )
```

The constructor of the Area window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.2.3 Member Function Documentation

4.2.3.1 area()

```
def area_ui.AreaWindow.area (
              self )
```

Displays the area of selected shape given appropriate side lengths/radius.

Takes in up to 3 side lengths and a radius as input from the user through input fields, and shows the user the result on the window

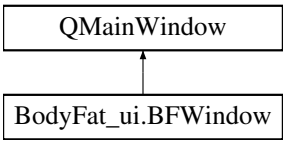
The documentation for this class was generated from the following file:

- src/uis/area_ui.py

4.3 BodyFat_ui.BFWindow Class Reference

BFWindow is a class that implements the GUI components for the Body Fat operation.

Inheritance diagram for BodyFat_ui.BFWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the Body Fat window.
- def bf (self)

  Displays the Body Fat rating and its meaning based on the metrics the user provides.

Public Attributes

- **path**

4.3.1 Detailed Description

BFWindow is a class that implements the GUI components for the Body Fat operation.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 __init__()

```
def BodyFat_ui.BFWindow.__init__ (
              self,
              path = "" )
```

The constructor of the Body Fat window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.3.3 Member Function Documentation

4.3.3.1 bf()

```
def BodyFat_ui.BFWindow.bf (
              self )
```

Displays the Body Fat rating and its meaning based on the metrics the user provides.

Takes in age, gender, height, weight, neck size and waist size from the user through input fields, and shows the user the result on the window

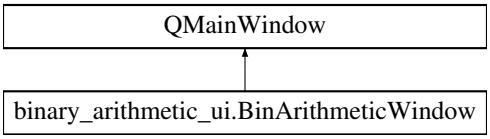
The documentation for this class was generated from the following file:

- src/uis/BodyFat_ui.py

4.4 binary_arithmetic_ui.BinArithmeticWindow Class Reference

BinArithmeticWindow is a class that implements the GUI components for the Binary Arithmetic operations.

Inheritance diagram for binary_arithmetic_ui.BinArithmeticWindow:



Public Member Functions

- def __init__ (self, path="")
  The constructor of the Binary Arithmetic window.
- def binArithmetic (self)
  Displays the arithmetic output of two binary numbers using various operators.

Public Attributes

- **path**

4.4.1 Detailed Description

BinArithmeticWindow is a class that implements the GUI components for the Binary Arithmetic operations.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 __init__()

```
def binary_arithmetic_ui.BinArithmeticWindow.__init__ (
              self,
              path = "" )
```

The constructor of the Binary Arithmetic window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.4.3 Member Function Documentation

4.4.3.1 binArithmetic()

```
def binary_arithmetic_ui.BinArithmeticWindow.binArithmetic (
            self )
```

Displays the arithmetic output of two binary numbers using various operators.

Takes in two binary numbers and the operator from the user through input fields, and shows the user the result on the window

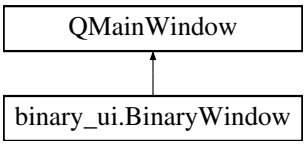
The documentation for this class was generated from the following file:

- src/uis/binary_arithmetic_ui.py

4.5 binary_ui.BinaryWindow Class Reference

BinaryWindow is a class that implements the GUI components for the Binary operation menu.

Inheritance diagram for binary_ui.BinaryWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the Binary window.

Public Attributes

- **path**

4.5.1 Detailed Description

BinaryWindow is a class that implements the GUI components for the Binary operation menu.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 __init__()

```
def binary_ui.BinaryWindow.__init__ (
              self,
              path = "" )
```

The constructor of the Binary window.

Creates a pop up window that displays and sets up the buttons that are necessary to navigate from the Binary window to other parts of the application. Also sets up the Binary window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

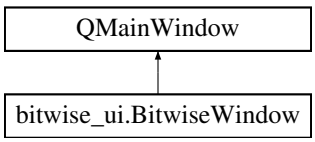
The documentation for this class was generated from the following file:

- src/uis/binary_ui.py

4.6 bitwise_ui.BitwiseWindow Class Reference

BitwiseWindow is a class that implements the GUI components for the Bitwise operations.

Inheritance diagram for bitwise_ui.BitwiseWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the Bitwise window.
- def bitwise (self)

  Displays the output of bitwise operations on one or two binary numbers.

Public Attributes

- **path**

4.6.1 Detailed Description

BitwiseWindow is a class that implements the GUI components for the Bitwise operations.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 __init__()

```
def bitwise_ui.BitwiseWindow.__init__ (
              self,
              path = "" )
```

The constructor of the Bitwise window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.6.3 Member Function Documentation

4.6.3.1 bitwise()

```
def bitwise_ui.BitwiseWindow.bitwise (
              self )
```

Displays the output of bitwise operations on one or two binary numbers.

Takes in one or two binary numbers and the operator from the user through input fields, and shows the user the result on the window

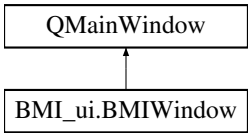
The documentation for this class was generated from the following file:

- src/uis/bitwise_ui.py

4.7 BMI_ui.BMIWindow Class Reference

BMIWindow is a class that implements the GUI components for the BMI operation.

Inheritance diagram for BMI_ui.BMIWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the BMI window.
- def bmi (self)

  Displays the BMI and its meaning based on the metrics the user provides.

Public Attributes

- **path**

4.7.1 Detailed Description

BMIWindow is a class that implements the GUI components for the BMI operation.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 __init__()

```
def BMI_ui.BMIWindow.__init__ (
              self,
              path = "" )
```

The constructor of the BMI window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.7.3 Member Function Documentation

4.7.3.1 bmi()

```
def BMI_ui.BMIWindow.bmi (
              self )
```

Displays the BMI and its meaning based on the metrics the user provides.

Takes in height and weight from the user through input fields, and shows the user the result on the window

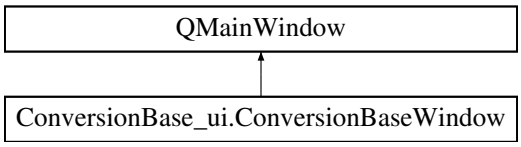
The documentation for this class was generated from the following file:

- src/uis/BMI_ui.py

4.8 ConversionBase_ui.ConversionBaseWindow Class Reference

ConversionBaseWindow is a class that implements the GUI components for the base conversion operation.

Inheritance diagram for ConversionBase_ui.ConversionBaseWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the base conversion window.
- def baseconvert (self)

  Displays the conversion a value of a base type 1 to a value of base type 2.

Public Attributes

- **path**

4.8.1 Detailed Description

ConversionBaseWindow is a class that implements the GUI components for the base conversion operation.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 __init__()

```
def ConversionBase_ui.ConversionBaseWindow.__init__ (
            self,
            path = "" )
```

The constructor of the base conversion window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.8.3 Member Function Documentation

4.8.3.1 baseconvert()

```
def ConversionBase_ui.ConversionBaseWindow.baseconvert (
              self )
```

Displays the conversion a value of a base type 1 to a value of base type 2.

Takes in 1 value and the convert to and convert from type as input from the user through input fields and shows the user the result on the window

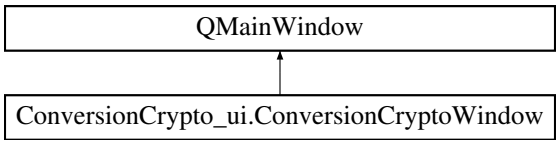
The documentation for this class was generated from the following file:

- src/uis/ConversionBase_ui.py

4.9 ConversionCrypto_ui.ConversionCryptoWindow Class Reference

ConversionCryptoWindow is a class that implements the GUI components for the crypto conversion operation.

Inheritance diagram for ConversionCrypto_ui.ConversionCryptoWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the crypto conversion window.
- def cryptoconvert (self)

  Displays the conversion a value of a base type 1 to a value of base type 2.

Public Attributes

- **path**

4.9.1 Detailed Description

ConversionCryptoWindow is a class that implements the GUI components for the crypto conversion operation.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 __init__()

```
def ConversionCrypto_ui.ConversionCryptoWindow.__init__ (
              self,
              path = "" )
```

The constructor of the crypto conversion window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.9.3 Member Function Documentation

4.9.3.1 cryptoconvert()

```
def ConversionCrypto_ui.ConversionCryptoWindow.cryptoconvert (
              self )
```

Displays the conversion a value of a base type 1 to a value of base type 2.

Takes in 1 value and the convert to and convert from type as input from the user through input fields and shows the user the result on the window

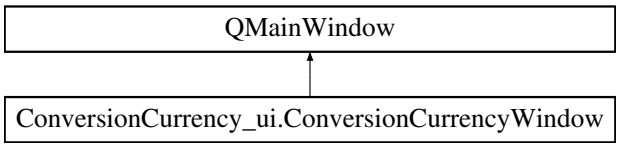
The documentation for this class was generated from the following file:

- src/uis/ConversionCrypto_ui.py

4.10 ConversionCurrency_ui.ConversionCurrencyWindow Class Reference

ConversionBaseWindow is a class that implements the GUI components for the currency conversion operation.

Inheritance diagram for ConversionCurrency_ui.ConversionCurrencyWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the currency conversion window.
- def currconvert (self)

  Displays the conversion a value of a base type 1 to a value of base type 2.

Public Attributes

- **path**

4.10.1 Detailed Description

ConversionBaseWindow is a class that implements the GUI components for the currency conversion operation.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 __init__()

```
def ConversionCurrency_ui.ConversionCurrencyWindow.__init__ (
              self,
              path = "" )
```

The constructor of the currency conversion window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.10.3 Member Function Documentation

4.10.3.1 currconvert()

```
def ConversionCurrency_ui.ConversionCurrencyWindow.currconvert (
              self )
```

Displays the conversion a value of a base type 1 to a value of base type 2.

Takes in 1 value and the convert to and convert from type as input from the user through input fields and shows the user the result on the window

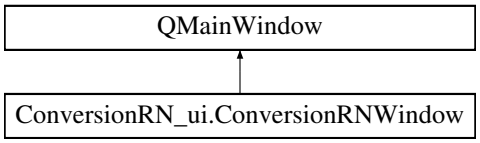
The documentation for this class was generated from the following file:

- src/uis/ConversionCurrency_ui.py

4.11 ConversionRN_ui.ConversionRNWindow Class Reference

ConversionBaseWindow is a class that implements the GUI components for the roman numeral conversion operation.

Inheritance diagram for ConversionRN_ui.ConversionRNWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the roman numeral conversion window.
- def RNconvert (self)

  Displays the conversion a value of a base type 1 to a value of base type 2.

Public Attributes

- **path**

4.11.1 Detailed Description

ConversionBaseWindow is a class that implements the GUI components for the roman numeral conversion operation.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 __init__()

```
def ConversionRN_ui.ConversionRNWindow.__init__ (
            self,
            path = "" )
```

The constructor of the roman numeral conversion window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.11.3 Member Function Documentation

4.11.3.1 RNconvert()

```
def ConversionRN_ui.ConversionRNWindow.RNconvert (
              self )
```

Displays the conversion a value of a base type 1 to a value of base type 2.

Takes in 1 value and the convert to and convert from type as input from the user through input fields and shows the user the result on the window

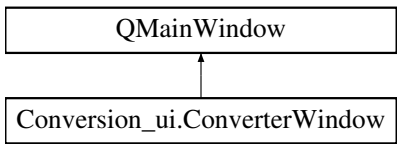
The documentation for this class was generated from the following file:

- src/uis/ConversionRN_ui.py

4.12 Conversion_ui.ConverterWindow Class Reference

ConverterWindow is a class that implements the GUI components for the Conversion operation menu.

Inheritance diagram for Conversion_ui.ConverterWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the Conversion window.

Public Attributes

- **path**
- **xpath**
- **currency**
- **base**
- **crypto**
- **RN**

4.12.1 Detailed Description

ConverterWindow is a class that implements the GUI components for the Conversion operation menu.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 __init__()

```
def Conversion_ui.ConverterWindow.__init__ (
              self,
              path = "" )
```

The constructor of the Conversion window.

Creates a pop up window that displays and sets up the buttons that are necessary to navigate from the Conversion window to other parts of the application. Also sets up the Conversion window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

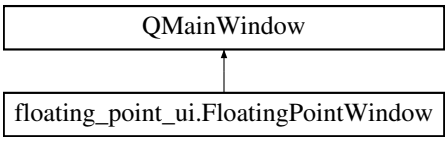
The documentation for this class was generated from the following file:

- src/uis/Conversion_ui.py

4.13 floating_point_ui.FloatingPointWindow Class Reference

FloatingPointWindow is a class that implements the GUI components for the Floating Point operation.

Inheritance diagram for floating_point_ui.FloatingPointWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the Floating Point window.
- def floating_point (self)

  Displays the conversion of a decimal number to IEEE 754 floating point representation and vice versa.

Public Attributes

- **path**

4.13.1 Detailed Description

FloatingPointWindow is a class that implements the GUI components for the Floating Point operation.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 __init__()

```
def floating_point_ui.FloatingPointWindow.__init__ (
              self,
              path = "" )
```

The constructor of the Floating Point window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.13.3 Member Function Documentation

4.13.3.1 floating_point()

```
def floating_point_ui.FloatingPointWindow.floating_point (
              self )
```

Displays the conversion of a decimal number to IEEE 754 floating point representation and vice versa.

Takes in decimal number or floating point number from the user through input fields, and shows the user the result on the window

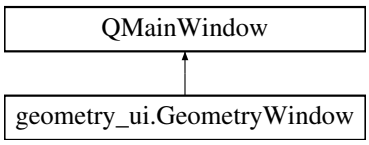
The documentation for this class was generated from the following file:

- src/uis/floating_point_ui.py

4.14 geometry_ui.GeometryWindow Class Reference

GeometryWindow is a class that implements the GUI components for the Geometry operation menu.

Inheritance diagram for geometry_ui.GeometryWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the Geometry window.

Public Attributes

- **path**

4.14.1 Detailed Description

GeometryWindow is a class that implements the GUI components for the Geometry operation menu.

4.14.2 Constructor & Destructor Documentation

4.14.2.1 __init__()

```
def geometry_ui.GeometryWindow.__init__ (
              self,
              path = "" )
```

The constructor of the Geometry window.

Creates a pop up window that displays and sets up the buttons that are necessary to navigate from the Geometry window to other parts of the application. Also sets up the Geometry window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

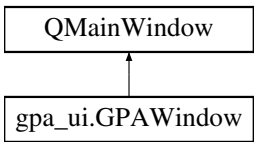
The documentation for this class was generated from the following file:

- src/uis/geometry_ui.py

4.15 gpa_ui.GPAWindow Class Reference

GPAWindow is a class that implements the GUI components for the GPA operation menu.

Inheritance diagram for gpa_ui.GPAWindow:



Public Member Functions

- def __init__ (slef, path="")

  The constructor of the GPA window.
- def gpa (self)

  Displays the 12.0 gpa from the metrics the user provides.

Public Attributes

- **path**

4.15.1 Detailed Description

GPAWindow is a class that implements the GUI components for the GPA operation menu.

4.15.2 Constructor & Destructor Documentation

4.15.2.1 __init__()

```
def gpa_ui.GPAWindow.__init__ (
              slef,
              path = "" )
```

The constructor of the GPA window.

Creates a pop up window that displays and sets up the buttons that are necessary to navigate from the GPA window to other parts of the application. Also sets up the GPA window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.15.3 Member Function Documentation

4.15.3.1 gpa()

```
def gpa_ui.GPAWindow.gpa (
              self )
```

Displays the 12.0 gpa from the metrics the user provides.

Takes in the grades and their weights through input fields and shows the users GPA result on the window

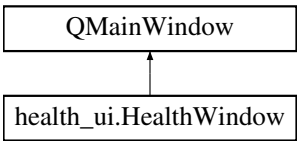
The documentation for this class was generated from the following file:

- src/uis/gpa_ui.py

4.16 health_ui.HealthWindow Class Reference

HealthWindow is a class that implements the GUI components for the Health operation menu.

Inheritance diagram for health_ui.HealthWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the Health window.

Public Attributes

- **path**
- **xpath**
- **bmi**
- **bf**

4.16.1 Detailed Description

HealthWindow is a class that implements the GUI components for the Health operation menu.

4.16.2 Constructor & Destructor Documentation

4.16.2.1 __init__()

```
def health_ui.HealthWindow.__init__ (
              self,
              path = "" )
```

The constructor of the Health window.

Creates a pop up window that displays and sets up the buttons that are necessary to navigate from the Health window to other parts of the application. Also sets up the Health window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

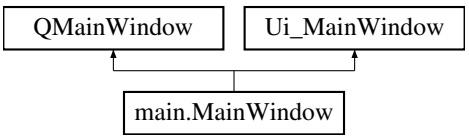
The documentation for this class was generated from the following file:

- src/uis/health_ui.py

4.17 main.MainWindow Class Reference

MainWindow is a class that implements the GUI components for the Main menu.

Inheritance diagram for main.MainWindow:



Public Member Functions

- def __init__ (self, args, kwargs)

  The constructor of the Main window.
- def storeMem (self)

  Stores the current number.
- def getMem (self)

Displays current stored number.

- def display (self)

  Displays number to user.
- def valueInput (self, v)

  Display value of input.
- def reset (self)

  Empty the line and current number stored.
- def addition (self)

  Conducts calculator addition operation.
- def subtraction (self)

  Conducts calculator subtraction operation.
- def multiplication (self)

  Conducts calculator multiplication operation.
- def power (self)

  Conducts calculator power operation.
- def division (self)

  Conducts calculator division operation.
- def left_bracket (self)

  Adds left bracket operation.
- def right_bracket (self)

  Adds right bracket operation.
- def equals (self)

  Evaluates operation.
- def keyPressEvent (self, event)

  Runs functionality for each button click in the calculator.

Static Public Member Functions

- def **credits** ()

Public Attributes

- **converters**
- **time**
- **algebra**
- **stock**
- **lineEdit**
- **currNum**
- **mem**

4.17.1 Detailed Description

MainWindow is a class that implements the GUI components for the Main menu.

4.17.2 Constructor & Destructor Documentation

4.17.2.1 __init__()

```
def main.MainWindow.__init__ (
              self,
              args,
              kwargs )
```

The constructor of the Main window.

Creates a pop up window that displays and sets up the buttons that are necessary to navigate from the Main window to other parts of the application. Also sets up the Main window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.17.3 Member Function Documentation

4.17.3.1 addition()

```
def main.MainWindow.addition (
              self )
```

Conducts calculator addition operation.

Check if line input prior is not another operation and if it is not, display an empty string and adds an addition operation

4.17.3.2 display()

```
def main.MainWindow.display (
              self )
```

Displays number to user.

displays the number that is currently stored onto the calculator display

4.17.3.3 division()

```
def main.MainWindow.division (
              self )
```

Conducts calculator division operation.

Check if line input prior is not another operation and if it is not, display an empty string and adds a division operation

4.17.3.4 equals()

```
def main.MainWindow.equals (
              self )
```

Evaluates operation.

Evaluates operation and displays answer

4.17.3.5 getMem()

```
def main.MainWindow.getMem (
              self )
```

Displays current stored number.

Checks if current stored number is empty and adds new number number to store and display

4.17.3.6 keyPressEvent()

```
def main.MainWindow.keyPressEvent (
              self,
              event )
```

Runs functionality for each button click in the calculator.

Hooks up the calculator button presses to the functions adding them to the operation

4.17.3.7 left_bracket()

```
def main.MainWindow.left_bracket (
              self )
```

Adds left bracket operation.

Clears the display and adds a left bracket to the operation

4.17.3.8 multiplication()

```
def main.MainWindow.multiplication (
              self )
```

Conducts calculator multiplication operation.

Check if line input prior is not another operation and if it is not, display an empty string and adds a multiplication operation

4.17.3.9 power()

```
def main.MainWindow.power (
              self )
```

Conducts calculator power operation.

Check if line input prior is not another operation and if it is not, display an empty string and adds a power operation

4.17.3.10 reset()

```
def main.MainWindow.reset (
              self )
```

Empty the line and current number stored.

Clears the line value and the current number value to an empty string value and display new blank value

4.17.3.11 right_bracket()

```
def main.MainWindow.right_bracket (
              self )
```

Adds right bracket operation.

Clears the display and adds a right bracket to the operation

4.17.3.12 storeMem()

```
def main.MainWindow.storeMem (
              self )
```

Stores the current number.

stores number for future use

4.17.3.13 subtraction()

```
def main.MainWindow.subtraction (
              self )
```

Conducts calculator subtraction operation.

Check if line input prior is not another operation and if it is not, display an empty string and adds a subtraction operation

4.17.3.14 valueInput()

```
def main.MainWindow.valueInput (
              self,
              v )
```

Display value of input.

Adds the input v to the value of current number and displays it

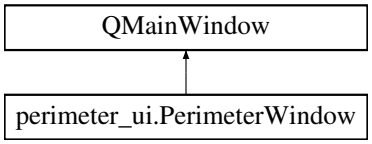
The documentation for this class was generated from the following file:

- src/main.py

4.18 perimeter_ui.PerimeterWindow Class Reference

PerimeterWindow is a class that implements the GUI components for the Perimeter operation.

Inheritance diagram for perimeter_ui.PerimeterWindow:



Public Member Functions

- def __init__ (self, path="")
  The constructor of the Perimeter window.
- def perimeter (self)
  Displays the perimeter of selected shape given appropriate side lengths/radius.

Public Attributes

- **path**

4.18.1 Detailed Description

PerimeterWindow is a class that implements the GUI components for the Perimeter operation.

4.18.2 Constructor & Destructor Documentation

4.18.2.1 __init__()

```
def perimeter_ui.PerimeterWindow.__init__ (
              self,
              path = "" )
```

The constructor of the Perimeter window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.18.3 Member Function Documentation

4.18.3.1 perimeter()

```
def perimeter_ui.PerimeterWindow.perimeter (
              self )
```

Displays the perimeter of selected shape given appropriate side lengths/radius.

Takes in up to 3 side lengths and a radius as input from the user through input fields, and shows the user the result on the window

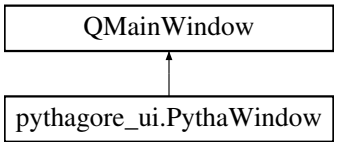
The documentation for this class was generated from the following file:

- src/uis/perimeter_ui.py

4.19 pythagore_ui.PythaWindow Class Reference

PythaWindow is a class that implements the GUI components for the Pythagorean Theorem operation.

Inheritance diagram for pythagore_ui.PythaWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the Pythagorean Theorem window.
- def pytha (self)

  Displays the length of the missing side of a right angle triangle.

Public Attributes

- **path**

4.19.1 Detailed Description

PythaWindow is a class that implements the GUI components for the Pythagorean Theorem operation.

4.19.2 Constructor & Destructor Documentation

4.19.2.1 __init__()

```
def pythagore_ui.PythaWindow.__init__ (
              self,
              path = "" )
```

The constructor of the Pythagorean Theorem window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.19.3 Member Function Documentation

4.19.3.1 pytha()

```
def pythagore_ui.PythaWindow.pytha (
              self )
```

Displays the length of the missing side of a right angle triangle.

Takes the inputs of two sides from the user through input fields, and shows the user the length of the missing side on the window

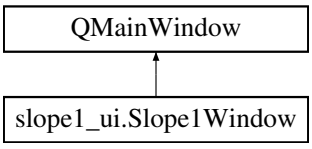
The documentation for this class was generated from the following file:

- src/uis/pythagore_ui.py

4.20 slope1_ui.Slope1Window Class Reference

Slope1Window is a class that implements the GUI components for the Slope operation.

Inheritance diagram for slope1_ui.Slope1Window:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the Slope window.
- def slope (self)

  Displays the slope of a line given two coordinates.

Public Attributes

- **path**

4.20.1 Detailed Description

Slope1Window is a class that implements the GUI components for the Slope operation.

4.20.2 Constructor & Destructor Documentation

4.20.2.1 __init__()

```
def slope1_ui.Slope1Window.__init__ (
              self,
              path = "" )
```

The constructor of the Slope window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.20.3 Member Function Documentation

4.20.3.1 slope()

```
def slope1_ui.Slope1Window.slope (
              self )
```

Displays the slope of a line given two coordinates.

Takes in two coordinates as input from the user through input fields, and shows the user the result on the window

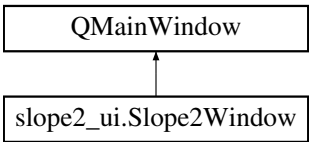
The documentation for this class was generated from the following file:

- src/uis/slope1_ui.py

4.21 slope2_ui.Slope2Window Class Reference

Slope2Window is a class that implements the GUI components for the Y-intercept operation.

Inheritance diagram for slope2_ui.Slope2Window:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the Y-intercept window.
- def yInt (self)

  Displays the y-intercept of the given slope and coordinate.

Public Attributes

- **path**

4.21.1 Detailed Description

Slope2Window is a class that implements the GUI components for the Y-intercept operation.

4.21.2 Constructor & Destructor Documentation

4.21.2.1 __init__()

```
def slope2_ui.Slope2Window.__init__ (
              self,
              path = "" )
```

The constructor of the Y-intercept window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.21.3 Member Function Documentation

4.21.3.1 yInt()

```
def slope2_ui.Slope2Window.yInt (
              self )
```

Displays the y-intercept of the given slope and coordinate.

Takes the inputs of a slope and a coordinate from the user through input fields, and shows the user the result on the window

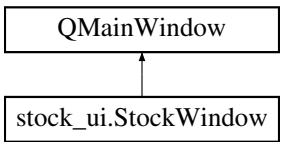
The documentation for this class was generated from the following file:

- src/uis/slope2_ui.py

4.22 stock_ui.StockWindow Class Reference

StockWindow is a class that implements the GUI components for the Stock operation menu.

Inheritance diagram for stock_ui.StockWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the Stock window.
- def stock (self)

  Displays the loss or gain on the stock from the metrics the user provides.

Public Attributes

- **path**

4.22.1 Detailed Description

StockWindow is a class that implements the GUI components for the Stock operation menu.

4.22.2 Constructor & Destructor Documentation

4.22.2.1 __init__()

```
def stock_ui.StockWindow.__init__ (
              self,
              path = "" )
```

The constructor of the Stock window.

Creates a pop up window that displays and sets up the buttons that are necessary to navigate from the Stocks window to other parts of the application. Also sets up the Stocks window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.22.3 Member Function Documentation

4.22.3.1 stock()

```
def stock_ui.StockWindow.stock (
              self )
```

Displays the loss or gain on the stock from the metrics the user provides.

Takes in the number of shares, purchase price, sell price, purchase commission and sell commission, through input fields, and shows the user the result on the window

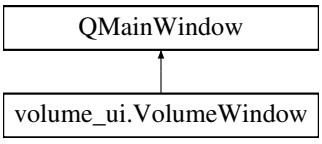
The documentation for this class was generated from the following file:

- src/uis/stock_ui.py

4.23 volume_ui.VolumeWindow Class Reference

VolumeWindow is a class that implements the GUI components for the Volume operation.

Inheritance diagram for volume_ui.VolumeWindow:



Public Member Functions

- def __init__ (self, path="")

  The constructor of the Volume window.
- def volume (self)

  Displays the volume of selected 3D shape given appropriate dimensions.

Public Attributes

- **path**

4.23.1 Detailed Description

VolumeWindow is a class that implements the GUI components for the Volume operation.

4.23.2 Constructor & Destructor Documentation

4.23.2.1 __init__()

```
def volume_ui.VolumeWindow.__init__ (
              self,
              path = "" )
```

The constructor of the Volume window.

Creates a pop up window that displays and sets up the buttons and input fields that are necessary to obtain input from the user and calculate the appropriate answer. Also sets up the window according to the created style sheet.

Parameters

path	The current path on which the file is found. Default value is an empty path.

4.23.3 Member Function Documentation

4.23.3.1 volume()

```
def volume_ui.VolumeWindow.volume (
              self )
```

Displays the volume of selected 3D shape given appropriate dimensions.

Takes in up to 3 dimensions and/or radius as input from the user through input fields, and shows the user the result on the window

The documentation for this class was generated from the following file:

- src/uis/volume_ui.py

Chapter 5

File Documentation

5.1 src/main.py File Reference

Provides a class to display the Main window.

Classes

- class main.MainWindow

  MainWindow is a class that implements the GUI components for the Main menu.

Functions

- def **main.start_gui** ()

5.1.1 Detailed Description

Provides a class to display the Main window.

Date

March 18, 2022

5.2 src/uis/algebra_ui.py File Reference

Provides a class to display the Algebra window.

Classes

- class algebra_ui.AlgebraWindow

  AlgebraWindow is a class that implements the GUI components for the Algebra operation menu.

Variables

- **algebra_ui.app** = QApplication(sys.argv)
- **algebra_ui.window** = AlgebraWindow()

5.2.1 Detailed Description

Provides a class to display the Algebra window.

Date

March 17, 2022

5.3 src/uis/area_ui.py File Reference

Provides a class to display the Area window.

Classes

- class area_ui.AreaWindow

  AreaWindow is a class that implements the GUI components for the Area operation.

Variables

- **area_ui.app** = QApplication(sys.argv)
- **area_ui.window** = AreaWindow()

5.3.1 Detailed Description

Provides a class to display the Area window.

Date

March 18, 2022

5.4 src/uis/binary_arithmetic_ui.py File Reference

Provides a class to display the Binary Arithmetic window.

Classes

- class binary_arithmetic_ui.BinArithmeticWindow

  BinArithmeticWindow is a class that implements the GUI components for the Binary Arithmetic operations.

Variables

- **binary_arithmetic_ui.app** = QApplication(sys.argv)
- **binary_arithmetic_ui.window** = BinArithmeticWindow()

5.4.1 Detailed Description

Provides a class to display the Binary Arithmetic window.

Date

March 18, 2022

5.5 src/uis/binary_ui.py File Reference

Provides a class to display the Binary window.

Classes

- class binary_ui.BinaryWindow

  BinaryWindow is a class that implements the GUI components for the Binary operation menu.

Variables

- **binary_ui.app** = QApplication(sys.argv)
- **binary_ui.window** = BinaryWindow()

5.5.1 Detailed Description

Provides a class to display the Binary window.

Date

March 18, 2022

5.6 src/uis/bitwise_ui.py File Reference

Provides a class to display the Bitwise Operation window.

Classes

- class bitwise_ui.BitwiseWindow

  BitwiseWindow is a class that implements the GUI components for the Bitwise operations.

Variables

- **bitwise_ui.app** = QApplication(sys.argv)
- **bitwise_ui.window** = BitwiseWindow()

5.6.1 Detailed Description

Provides a class to display the Bitwise Operation window.

Date

March 18, 2022

5.7 src/uis/BMI_ui.py File Reference

Provides a class to display the BMI window.

Classes

- class BMI_ui.BMIWindow

  BMIWindow is a class that implements the GUI components for the BMI operation.

Variables

- **BMI_ui.app** = QApplication(sys.argv)
- **BMI_ui.window** = BMIWindow()

5.7.1 Detailed Description

Provides a class to display the BMI window.

Date

March 17, 2022

5.8 src/uis/BodyFat_ui.py File Reference

Provides a class to display the Body Fat window.

Classes

- class BodyFat_ui.BFWindow

  BFWindow is a class that implements the GUI components for the Body Fat operation.

Variables

- **BodyFat_ui.app** = QApplication(sys.argv)
- **BodyFat_ui.window** = BFWindow()

5.8.1 Detailed Description

Provides a class to display the Body Fat window.

Date

March 17, 2022

5.9 src/uis/Calculators/algebra_calculator.py File Reference

Alegbraic algorithms.

Functions

- def algebra_calculator.slopeOfLine (x1, y1, x2, y2)
  Calculates slope of a line given 2 points.
- def algebra_calculator.yIntercept (m, x, y)
  Calculates y-intercept of a line given a point and the slope.
- def algebra_calculator.pyTheorem (solve, a, b, c)
  Calculates pythagorean theorem of a right triangle given two sides and the side to solve for.

5.9.1 Detailed Description

Alegbraic algorithms.

Date

March 17, 2022

5.9.2 Function Documentation

5.9.2.1 pyTheorem()

```
def algebra_calculator.pyTheorem (
              solve,
              a,
              b,
              c )
```

Calculates pythagorean theorem of a right triangle given two sides and the side to solve for.

Parameters

solve	A string that represents the missing side
a	A real number that represents a side the is not the hypotenuse
b	A real number that represents a side the is not the hypotenuse
c	A real number that represents the hypotenuse

Returns

The length of the missing side

Exceptions

ValueError	Throws an exception if hypotenuse is not the longest side

5.9.2.2 slopeOfLine()

```
def algebra_calculator.slopeOfLine (
              x1,
              y1,
              x2,
              y2 )
```

Calculates slope of a line given 2 points.

Parameters

x1	A real number that represents the X-coordinate of the first point
y1	A real number that represents the Y-coordinate of the first point
x2	A real number that represents the X-coordinate of the second point
y2	A real number that represents the Y-coordinate of the second point

Returns

The slope of the line

Exceptions

ValueError	Throws an exception if x2 and x1 are equal

5.9.2.3 yIntercept()

```
def algebra_calculator.yIntercept (
              m,
```

```
x,
y )
```

Calculates y-intercept of a line given a point and the slope.

Parameters

m	A real number that represents the slope of the line
x	A real number that represents the X-coordinate of the point
y	A real number that represents the Y-coordinate of the point

Returns

The y-intercept of the line

Exceptions

ValueError	Throws an exception if x2 and x1 are equal

5.10 src/uis/Calculators/binary_calculator.py File Reference

Binary algorithms.

Functions

- def binary_calculator.toFloatingPoint (n)
  Calculates IEEE 754 representation from decimal.
- def binary_calculator.toDecimal (n)
  Calculates decimal number from IEEE 754 representation.
- def binary_calculator.binAdd (n, m)
  Calculates sum of two binary numbers.
- def binary_calculator.binSub (n, m)
  Calculates difference of two binary numbers.
- def binary_calculator.binMult (n, m)
  Calculates product of two binary numbers.
- def binary_calculator.binDiv (n, m)
  Calculates quotient of two binary numbers.
- def binary_calculator.binPow (n, m)
  Calculates power of two binary numbers.
- def binary_calculator.bitwiseAND (n, m)
  Calculates bitwise AND of two binary numbers.
- def binary_calculator.bitwiseOR (n, m)
  Calculates bitwise OR of two binary numbers.
- def binary_calculator.bitwiseNOT (n)
  Calculates bitwise NOT of binary number.
- def binary_calculator.bitwiseXOR (n, m)
  Calculates bitwise XOR of two binary numbers.
- def binary_calculator.rshift (n, shiftNum, length)
  Calculates rightward bit shift of binary number using given shift number and length.
- def binary_calculator.lshift (n, shiftNum, length)
  Calculates leftward bit shift of binary number using given shift number and length.

5.10.1 Detailed Description

Binary algorithms.

Date

March 18, 2022

5.10.2 Function Documentation

5.10.2.1 binAdd()

```
def binary_calculator.binAdd (
              n,
              m )
```

Calculates sum of two binary numbers.

Parameters

n	Binary number
m	Binary number

Returns

Sum of n and m

Exceptions

ValueError	Throws an exception if n or m are invalid

5.10.2.2 binDiv()

```
def binary_calculator.binDiv (
              n,
              m )
```

Calculates quotient of two binary numbers.

Parameters

n	Binary number
m	Binary number

Returns

Quotient of n and m

Exceptions

ValueError	Throws an exception if n or m are invalid or m equals zero

5.10.2.3 binMult()

```
def binary_calculator.binMult (
              n,
              m )
```

Calculates product of two binary numbers.

Parameters

n	Binary number
m	Binary number

Returns

Product of n and m

Exceptions

ValueError	Throws an exception if n or m are invalid

5.10.2.4 binPow()

```
def binary_calculator.binPow (
              n,
              m )
```

Calculates power of two binary numbers.

Parameters

n	Binary number
m	Binary number

Returns

Power of n to the m

Exceptions

ValueError	Throws an exception if n or m are invalid or n and m are both zero

5.10.2.5 binSub()

```
def binary_calculator.binSub (
              n,
              m )
```

Calculates difference of two binary numbers.

Parameters

n	Binary number
m	Binary number

Returns

Difference of n and m

Exceptions

ValueError	Throws an exception if n or m are invalid

5.10.2.6 bitwiseAND()

```
def binary_calculator.bitwiseAND (
              n,
              m )
```

Calculates bitwise AND of two binary numbers.

Parameters

n	Binary number
m	Binary number

Returns

Bitwise AND of n and m

Exceptions

ValueError	Throws an exception if n or m are invalid

5.10.2.7 bitwiseNOT()

```
def binary_calculator.bitwiseNOT (
              n )
```

Calculates bitwise NOT of binary number.

Parameters

n	Binary number

Returns

Bitwise NOT of n

Exceptions

ValueError	Throws an exception if n is invalid

5.10.2.8 bitwiseOR()

```
def binary_calculator.bitwiseOR (
              n,
              m )
```

Calculates bitwise OR of two binary numbers.

Parameters

n	Binary number
m	Binary number

Returns

Bitwise OR of n and m

Exceptions

ValueError	Throws an exception if n or m are invalid

5.10.2.9 bitwiseXOR()

```
def binary_calculator.bitwiseXOR (
              n,
              m )
```

Calculates bitwise XOR of two binary numbers.

Parameters

n	Binary number
m	Binary number

Returns

Bitwise XOR of n and m

Exceptions

ValueError	Throws an exception if n or m are invalid

5.10.2.10 lshift()

```
def binary_calculator.lshift (
              n,
              shiftNum,
              length )
```

Calculates leftward bit shift of binary number using given shift number and length.

Parameters

n	Binary number
shiftNum	Number of shifts
length	Length of binary number

Returns

n bit shifted leftward shiftNum times

Exceptions

ValueError	Throws an exception if n larger than length or n in invalid

5.10.2.11 rshift()

```
def binary_calculator.rshift (
              n,
              shiftNum,
              length )
```

Calculates rightward bit shift of binary number using given shift number and length.

Parameters

n	Binary number
shiftNum	Number of shifts
length	Length of binary number

Returns

n bit shifted rightward shiftNum times

Exceptions

ValueError	Throws an exception if n larger than length or n in invalid

5.10.2.12 toDecimal()

```
def binary_calculator.toDecimal (
              n )
```

Calculates decimal number from IEEE 754 representation.

Parameters

n	IEEE 754 binary number

Returns

Decimal representation

Exceptions

ValueError	Throws an exception if n is invalid

5.10.2.13 toFloatingPoint()

```
def binary_calculator.toFloatingPoint (
              n )
```

Calculates IEEE 754 representation from decimal.

Parameters

n	Decimal number

Returns

IEEE 754 floating point representation

Exceptions

ValueError	Throws an exception if n is too large

5.11 src/uis/Calculators/conversion_calculator.py File Reference

Conversion Algorithms.

Functions

- def conversion_calculator.convertCurrency (initialVal, currFrom, currTo)

  Converts from selected currency to another selected currency.
- def conversion_calculator.convertCrypto (initialVal, currFrom, currTo)

  Converts from selected cryptocurrency to another selected cryptocurrency.
- def conversion_calculator.convertBase (initialVal, baseFrom, baseTo)

  Converts from a selected numerical value of a base to another base value.
- def conversion_calculator.convertRN (initialVal, RNFrom, RNTo)

  Converts from a decimal value to a roman numeral value and from a roman numeral value to a decimal value.

5.11.1 Detailed Description

Conversion Algorithms.

Date

March 18, 2022

5.11.2 Function Documentation

5.11.2.1 convertBase()

```
def conversion_calculator.convertBase (
              initialVal,
              baseFrom,
              baseTo )
```

Converts from a selected numerical value of a base to another base value.

Parameters

initialVal	A real number that represents the initial numerical value
baseFrom	A string value that represents the base of the initialVal
baseTo	A string value that represents which base to convert to

Returns

the final value after conversion

5.11.2.2 convertCrypto()

```
def conversion_calculator.convertCrypto (
              initialVal,
              currFrom,
              currTo )
```

Converts from selected cryptocurrency to another selected cryptocurrency.

Parameters

initialVal	A real number that represents the cryptocurrency value
currFrom	A string value that represents the cryptocurrency of the initialVal
currTo	A string value that represents which cryptocurrency to convert to

Returns

the final value after conversion

5.11.2.3 convertCurrency()

```
def conversion_calculator.convertCurrency (
              initialVal,
```

```
                currFrom,
                currTo )
```

Converts from selected currency to another selected currency.

Parameters

initialVal	A real number that represents the currency value
currFrom	A string value that represents the currency of the initialVal
currTo	A string value that represents which currency to convert to

Returns

the final value after conversion

5.11.2.4 convertRN()

```
def conversion_calculator.convertRN (
                initialVal,
                RNFrom,
                RNTo )
```

Converts from a decimal value to a roman numeral value and from a roman numeral value to a decimal value.

Parameters

initialVal	A string that represents the initial value
RNFrom	A string value that represents the type of the initialVal
RNTo	A string value that represents which type to convert to

Returns

the final value after conversion

5.12 src/uis/Calculators/geometry_calculator.py File Reference

Geometry algorithms.

Functions

- def geometry_calculator.getArea (shape, a, b, c, r)

  Calculates area of given shape with given side lengths or radius.
- def geometry_calculator.getPerimeter (shape, a, b, c, r)

  Calculates perimeter of given shape with given side lengths or radius.
- def geometry_calculator.getVolume (shape, l, w, h, r)

  Calculates volume of given shape with given dimensions.

5.12.1 Detailed Description

Geometry algorithms.

Date

March 18, 2022

5.12.2 Function Documentation

5.12.2.1 getArea()

```
def geometry_calculator.getArea (
              shape,
              a,
              b,
              c,
              r )
```

Calculates area of given shape with given side lengths or radius.

Parameters

shape	An integer that represents the shape
a	Side length a
b	Side length b
c	Side length c
r	Radius

Returns

Area

Exceptions

ValueError	Throws an exception if required side lengths or radius are invalid

5.12.2.2 getPerimeter()

```
def geometry_calculator.getPerimeter (
              shape,
              a,
              b,
```

```
                c,
                r )
```

Calculates perimeter of given shape with given side lengths or radius.

Parameters

shape	An integer that represents the shape
a	Side length a
b	Side length b
c	Side length c
r	Radius

Returns

Perimeter

Exceptions

ValueError	Throws an exception if required side lengths or radius are invalid

5.12.2.3 getVolume()

```
def geometry_calculator.getVolume (
                shape,
                l,
                w,
                h,
                r )
```

Calculates volume of given shape with given dimensions.

Parameters

shape	An integer that represents the shape
l	Length
w	Width
h	Height
r	Radius

Returns

Volume

Exceptions

ValueError	Throws an exception if required dimensions invalid

5.13 src/uis/Calculators/gpa_calculator.py File Reference

gpa algorithms

Functions

- def gpa_calculator.gpaCalculate (gradeList)
  Calculates the GPA of the user.

5.13.1 Detailed Description

gpa algorithms

Date

March 17, 2022

5.13.2 Function Documentation

5.13.2.1 gpaCalculate()

```
def gpa_calculator.gpaCalculate (
            gradeList )
```

Calculates the GPA of the user.

Parameters

gradeList	A list with floats that carries the grades of each class multipled by the weight of the class.

Returns

The average GPA of the student

5.14 src/uis/Calculators/health_calculator.py File Reference

Health algorithms.

Functions

- def health_calculator.bodyMassIndex (weight, height)
  Calculates the body mass index of a person.
- def health_calculator.bodyFat (gender, height, neck, waist, hip=None)
  Calculates the body fat percentage of a person.

5.14.1 Detailed Description

Health algorithms.

Date

March 17, 2022

5.14.2 Function Documentation

5.14.2.1 bodyFat()

```
def health_calculator.bodyFat (
              gender,
              height,
              neck,
              waist,
              hip = None )
```

Calculates the body fat percentage of a person.

Parameters

gender	A string that represents the gender of the person
height	A real number that represents the height of the user in centimeters
neck	A real number that represents the size of a person's neck in centimeters
waist	A real number that represents the size of a person's waist in centimeters
hip	A real number that represents the size of a person's hip in centimeters

Returns

A string displaying the body fat percentage and its meaning

Exceptions

ValueError	Throws an exception if a measurement is 0

5.14.2.2 bodyMassIndex()

```
def health_calculator.bodyMassIndex (
              weight,
              height )
```

Calculates the body mass index of a person.

Parameters

weight	A real number that represents the weight of the user in pounds
height	A real number that represents the height of the user in centimeters

Returns

A string displaying the BMI coefficient and its meaning

Exceptions

ValueError	Throws an exception if height is 0

5.15 src/uis/Calculators/main_calculator.py File Reference

main calculator algorithms

Functions

- def main_calculator.evaluate ()

  Evaluates operation.
- def main_calculator.addition ()

  Conducts calculator addition operation.
- def main_calculator.subtraction ()

  Conducts calculator subtraction operation.
- def main_calculator.multiplication (self)

  Conducts calculator multiplication operation.
- def main_calculator.power ()

  Conducts calculator power operation.
- def main_calculator.division ()

  Conducts calculator division operation.
- def main_calculator.left_bracket ()

  Adds left bracket operation.
- def main_calculator.right_bracket ()

  Adds right bracket operation.

Variables

- string **main_calculator.lineEdit** = ""

5.15.1 Detailed Description

main calculator algorithms

Date

March 18, 2022

5.15.2 Function Documentation

5.15.2.1 addition()

```
def main_calculator.addition ( )
```

Conducts calculator addition operation.

Check if line input prior is not another operation and if it is not,adds an addition operation

5.15.2.2 division()

```
def main_calculator.division ( )
```

Conducts calculator division operation.

Check if line input prior is not another operation and if it is not, adds a division operation

5.15.2.3 evaluate()

```
def main_calculator.evaluate ( )
```

Evaluates operation.

Evaluates operation

5.15.2.4 left_bracket()

```
def main_calculator.left_bracket ( )
```

Adds left bracket operation.

adds a left bracket to the operation

5.15.2.5 multiplication()

```
def main_calculator.multiplication (
              self )
```

Conducts calculator multiplication operation.

Check if line input prior is not another operation and if it is not,adds a multiplication operation

5.15.2.6 power()

```
def main_calculator.power ( )
```

Conducts calculator power operation.

Adds a power operation

5.15.2.7 right_bracket()

```
def main_calculator.right_bracket ( )
```

Adds right bracket operation.

adds a right bracket to the operation

5.15.2.8 subtraction()

```
def main_calculator.subtraction ( )
```

Conducts calculator subtraction operation.

Check if line input prior is not another operation and if it is not, adds a subtraction operation

5.16 src/uis/Calculators/stocks_calculator.py File Reference

stock algorithms

Functions

- def stocks_calculator.calcUserGainLossCase1 (shares, purchasePrice, sellPrice, buyCommission, sell↩ Commission)

  Calculates the profit gain or loss when a broker is used.
- def stocks_calculator.calcUserGainLossCase2 (shares, purchasePrice, sellPrice)

  Calculates the profit gain or loss when a broker is not used.

5.16.1 Detailed Description

stock algorithms

Date

March 17, 2022

5.16.2 Function Documentation

5.16.2.1 calcUserGainLossCase1()

```
def stocks_calculator.calcUserGainLossCase1 (
              shares,
              purchasePrice,
              sellPrice,
              buyCommission,
              sellCommission )
```

Calculates the profit gain or loss when a broker is used.

Parameters

shares	A float that represents the amount of shares of a stock
purchasePrice	A float that represents the purchase price of the stock
sellPrice	A float that represents the price the stock was sold at
buyCommission	A float that represents the price of commission the broker charged at purchase
sellCommission	A float that represents the price of commission the broker charged when sold

Returns

The gain or loss on the stock

5.16.2.2 calcUserGainLossCase2()

```
def stocks_calculator.calcUserGainLossCase2 (
              shares,
              purchasePrice,
              sellPrice )
```

Calculates the profit gain or loss when a broker is not used.

Parameters

shares	A float that represents the amount of shares of a stock
purchasePrice	A float that represents the purchase price of the stock
sellPrice	A float that represents the price the stock was sold at

Returns

The gain or loss on the stock

5.17 src/uis/Conversion_ui.py File Reference

Provides a class to display the Conversion window.

Classes

- class Conversion_ui.ConverterWindow

  ConverterWindow is a class that implements the GUI components for the Conversion operation menu.

Variables

- **Conversion_ui.app** = QApplication(sys.argv)
- **Conversion_ui.window** = ConverterWindow()

5.17.1 Detailed Description

Provides a class to display the Conversion window.

Date

March 17, 2022

5.18 src/uis/ConversionBase_ui.py File Reference

Provides a class to display the base conversion window.

Classes

- class ConversionBase_ui.ConversionBaseWindow

  ConversionBaseWindow is a class that implements the GUI components for the base conversion operation.

Variables

- **ConversionBase_ui.app** = QApplication(sys.argv)
- **ConversionBase_ui.window** = ConversionBaseWindow()

5.18.1 Detailed Description

Provides a class to display the base conversion window.

Date

March 18, 2022

5.19 src/uis/ConversionCrypto_ui.py File Reference

Provides a class to display the crypto conversion window.

Classes

- class ConversionCrypto_ui.ConversionCryptoWindow

  ConversionCryptoWindow is a class that implements the GUI components for the crypto conversion operation.

Variables

- **ConversionCrypto_ui.app** = QApplication(sys.argv)
- **ConversionCrypto_ui.window** = ConversionCryptoWindow()

5.19.1 Detailed Description

Provides a class to display the crypto conversion window.

Date

March 18, 2022

5.20 src/uis/ConversionCurrency_ui.py File Reference

Provides a class to display the currency conversion window.

Classes

- class ConversionCurrency_ui.ConversionCurrencyWindow

  ConversionBaseWindow is a class that implements the GUI components for the currency conversion operation.

Variables

- **ConversionCurrency_ui.app** = QApplication(sys.argv)
- **ConversionCurrency_ui.window** = ConversionCurrencyWindow()

5.20.1 Detailed Description

Provides a class to display the currency conversion window.

Date

March 18, 2022

5.21 src/uis/ConversionRN_ui.py File Reference

Provides a class to display the roman numeral conversion window.

Classes

- class ConversionRN_ui.ConversionRNWindow

  ConversionBaseWindow is a class that implements the GUI components for the roman numeral conversion operation.

Variables

- **ConversionRN_ui.app** = QApplication(sys.argv)
- **ConversionRN_ui.window** = ConversionRNWindow()

5.21.1 Detailed Description

Provides a class to display the roman numeral conversion window.

Date

March 18, 2022

5.22 src/uis/floating_point_ui.py File Reference

Provides a class to display the Floating Point window.

Classes

- class floating_point_ui.FloatingPointWindow

  FloatingPointWindow is a class that implements the GUI components for the Floating Point operation.

Variables

- **floating_point_ui.app** = QApplication(sys.argv)
- **floating_point_ui.window** = FloatingPointWindow()

5.22.1 Detailed Description

Provides a class to display the Floating Point window.

Date

March 18, 2022

5.23 src/uis/geometry_ui.py File Reference

Provides a class to display the Geometry window.

Classes

- class geometry_ui.GeometryWindow

  GeometryWindow is a class that implements the GUI components for the Geometry operation menu.

Variables

- **geometry_ui.app** = QApplication(sys.argv)
- **geometry_ui.window** = GeometryWindow()

5.23.1 Detailed Description

Provides a class to display the Geometry window.

Date

March 18, 2022

5.24 src/uis/gpa_ui.py File Reference

Provides a class to display the GPA window.

Classes

- class gpa_ui.GPAWindow

  GPAWindow is a class that implements the GUI components for the GPA operation menu.

Variables

- **gpa_ui.app** = QApplication(sys.argv)
- **gpa_ui.window** = GPAWindow()

5.24.1 Detailed Description

Provides a class to display the GPA window.

Date

March 17, 2022

5.25 src/uis/health_ui.py File Reference

Provides a class to display the Health window.

Classes

- class health_ui.HealthWindow

  HealthWindow is a class that implements the GUI components for the Health operation menu.

Variables

- **health_ui.app** = QApplication(sys.argv)
- **health_ui.window** = HealthWindow()

5.25.1 Detailed Description

Provides a class to display the Health window.

Date

March 17, 2022

5.26 src/uis/perimeter_ui.py File Reference

Provides a class to display the Perimeter window.

Classes

- class perimeter_ui.PerimeterWindow

  PerimeterWindow is a class that implements the GUI components for the Perimeter operation.

Variables

- **perimeter_ui.app** = QApplication(sys.argv)
- **perimeter_ui.window** = PerimeterWindow()

5.26.1 Detailed Description

Provides a class to display the Perimeter window.

Date

March 18, 2022

5.27 src/uis/pythagore_ui.py File Reference

Provides a class to display the Pythagorean Theorem window.

Classes

- class pythagore_ui.PythaWindow

  PythaWindow is a class that implements the GUI components for the Pythagorean Theorem operation.

Variables

- **pythagore_ui.app** = QApplication(sys.argv)
- **pythagore_ui.window** = PythaWindow()

5.27.1 Detailed Description

Provides a class to display the Pythagorean Theorem window.

Date

March 17, 2022

5.28 src/uis/slope1_ui.py File Reference

Provides a class to display the Slope window.

Classes

- class slope1_ui.Slope1Window

  Slope1Window is a class that implements the GUI components for the Slope operation.

Variables

- **slope1_ui.app** = QApplication(sys.argv)
- **slope1_ui.window** = Slope1Window()

5.28.1 Detailed Description

Provides a class to display the Slope window.

Date

March 17, 2022

5.29 src/uis/slope2_ui.py File Reference

Provides a class to display the Y-intercept window.

Classes

- class slope2_ui.Slope2Window

  Slope2Window is a class that implements the GUI components for the Y-intercept operation.

Variables

- **slope2_ui.app** = QApplication(sys.argv)
- **slope2_ui.window** = Slope2Window()

5.29.1 Detailed Description

Provides a class to display the Y-intercept window.

Date

March 17, 2022

5.30 src/uis/stock_ui.py File Reference

Provides a class to display the Stocks window.

Classes

- class stock_ui.StockWindow

  StockWindow is a class that implements the GUI components for the Stock operation menu.

Variables

- **stock_ui.app** = QApplication(sys.argv)
- **stock_ui.window** = StockWindow()

5.30.1 Detailed Description

Provides a class to display the Stocks window.

Date

March 17, 2022

5.31 src/uis/volume_ui.py File Reference

Provides a class to display the Volume window.

Classes

- class volume_ui.VolumeWindow

  VolumeWindow is a class that implements the GUI components for the Volume operation.

Variables

- **volume_ui.app** = QApplication(sys.argv)
- **volume_ui.window** = VolumeWindow()

5.31.1 Detailed Description

Provides a class to display the Volume window.

Date

March 18, 2022

Index