

Group 15: MIS

Generated by Doxygen 1.8.13

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	algebra_ui.AlgebraWindow Class Reference	7
4.1.1	Detailed Description	7
4.1.2	Constructor & Destructor Documentation	8
4.1.2.1	__init__()	8
4.2	area_ui.AreaWindow Class Reference	8
4.2.1	Detailed Description	9
4.2.2	Constructor & Destructor Documentation	9
4.2.2.1	__init__()	9
4.2.3	Member Function Documentation	9
4.2.3.1	area()	9
4.3	BodyFat_ui.BFWindow Class Reference	10
4.3.1	Detailed Description	10
4.3.2	Constructor & Destructor Documentation	10
4.3.2.1	__init__()	10
4.3.3	Member Function Documentation	11

4.3.3.1	<code>bf()</code>	11
4.4	binary_arithmetic_ui.BinArithmeticWindow Class Reference	11
4.4.1	Detailed Description	12
4.4.2	Constructor & Destructor Documentation	12
4.4.2.1	<code>__init__()</code>	12
4.4.3	Member Function Documentation	12
4.4.3.1	<code>binArithmetic()</code>	12
4.5	binary_ui.BinaryWindow Class Reference	13
4.5.1	Detailed Description	13
4.5.2	Constructor & Destructor Documentation	13
4.5.2.1	<code>__init__()</code>	13
4.6	bitwise_ui.BitwiseWindow Class Reference	14
4.6.1	Detailed Description	14
4.6.2	Constructor & Destructor Documentation	14
4.6.2.1	<code>__init__()</code>	15
4.6.3	Member Function Documentation	16
4.6.3.1	<code>bitwise()</code>	16
4.7	BMI_ui.BMIWindow Class Reference	16
4.7.1	Detailed Description	17
4.7.2	Constructor & Destructor Documentation	17
4.7.2.1	<code>__init__()</code>	17
4.7.3	Member Function Documentation	17
4.7.3.1	<code>bmi()</code>	17
4.8	main_calculator.Calculator Class Reference	18
4.8.1	Detailed Description	18
4.8.2	Member Function Documentation	19
4.8.2.1	<code>addition()</code>	19
4.8.2.2	<code>division()</code>	19
4.8.2.3	<code>getMem()</code>	19
4.8.2.4	<code>left_bracket()</code>	19

4.8.2.5	multiplication()	19
4.8.2.6	power()	20
4.8.2.7	reset()	20
4.8.2.8	right_bracket()	20
4.8.2.9	storeMem()	20
4.8.2.10	subtraction()	20
4.8.2.11	valueInput()	21
4.9	ConversionBase_ui.ConversionBaseWindow Class Reference	21
4.9.1	Detailed Description	21
4.9.2	Constructor & Destructor Documentation	22
4.9.2.1	__init__()	22
4.9.3	Member Function Documentation	22
4.9.3.1	baseconvert()	22
4.10	ConversionCrypto_ui.ConversionCryptoWindow Class Reference	22
4.10.1	Detailed Description	23
4.10.2	Constructor & Destructor Documentation	23
4.10.2.1	__init__()	23
4.10.3	Member Function Documentation	23
4.10.3.1	cryptoconvert()	23
4.11	ConversionCurrency_ui.ConversionCurrencyWindow Class Reference	24
4.11.1	Detailed Description	24
4.11.2	Constructor & Destructor Documentation	24
4.11.2.1	__init__()	24
4.11.3	Member Function Documentation	25
4.11.3.1	currconvert()	25
4.12	ConversionRN_ui.ConversionRNWindow Class Reference	25
4.12.1	Detailed Description	26
4.12.2	Constructor & Destructor Documentation	26
4.12.2.1	__init__()	26
4.12.3	Member Function Documentation	26

4.12.3.1	RNconvert()	26
4.13	Conversion_ui.ConverterWindow Class Reference	27
4.13.1	Detailed Description	27
4.13.2	Constructor & Destructor Documentation	27
4.13.2.1	__init__()	27
4.14	floating_point_ui.FloatingPointWindow Class Reference	28
4.14.1	Detailed Description	28
4.14.2	Constructor & Destructor Documentation	28
4.14.2.1	__init__()	28
4.14.3	Member Function Documentation	29
4.14.3.1	floating_point()	29
4.15	geometry_ui.GeometryWindow Class Reference	29
4.15.1	Detailed Description	30
4.15.2	Constructor & Destructor Documentation	30
4.15.2.1	__init__()	30
4.16	gpa_ui.GPAWindow Class Reference	30
4.16.1	Detailed Description	31
4.16.2	Member Function Documentation	31
4.16.2.1	add()	31
4.16.2.2	gpa()	31
4.16.3	Member Data Documentation	31
4.16.3.1	gradeList	31
4.17	health_ui.HealthWindow Class Reference	32
4.17.1	Detailed Description	32
4.17.2	Constructor & Destructor Documentation	32
4.17.2.1	__init__()	32
4.18	main.MainWindow Class Reference	33
4.18.1	Detailed Description	34
4.18.2	Constructor & Destructor Documentation	34
4.18.2.1	__init__()	34

4.18.3	Member Function Documentation	35
4.18.3.1	addition()	35
4.18.3.2	display()	35
4.18.3.3	division()	35
4.18.3.4	equals()	35
4.18.3.5	getMem()	35
4.18.3.6	keyPressEvent()	36
4.18.3.7	left_bracket()	36
4.18.3.8	multiplication()	36
4.18.3.9	power()	36
4.18.3.10	reset()	36
4.18.3.11	right_bracket()	37
4.18.3.12	storeMem()	37
4.18.3.13	subtraction()	37
4.18.3.14	valueInput()	37
4.19	perimeter_ui.PerimeterWindow Class Reference	37
4.19.1	Detailed Description	38
4.19.2	Constructor & Destructor Documentation	38
4.19.2.1	__init__()	38
4.19.3	Member Function Documentation	38
4.19.3.1	perimeter()	39
4.20	pythagore_ui.PythaWindow Class Reference	39
4.20.1	Detailed Description	39
4.20.2	Constructor & Destructor Documentation	40
4.20.2.1	__init__()	40
4.20.3	Member Function Documentation	40
4.20.3.1	pytha()	40
4.21	stock_ui.StockWindow Class Reference	40
4.21.1	Detailed Description	41
4.21.2	Constructor & Destructor Documentation	41
4.21.2.1	__init__()	41
4.21.3	Member Function Documentation	41
4.21.3.1	stock()	41
4.22	volume_ui.VolumeWindow Class Reference	42
4.22.1	Detailed Description	42
4.22.2	Constructor & Destructor Documentation	42
4.22.2.1	__init__()	42
4.22.3	Member Function Documentation	43
4.22.3.1	volume()	43

5 File Documentation	45
5.1 src/main.py File Reference	45
5.1.1 Detailed Description	45
5.2 src/uis/algebra_ui.py File Reference	45
5.2.1 Detailed Description	46
5.3 src/uis/area_ui.py File Reference	46
5.3.1 Detailed Description	46
5.4 src/uis/binary_arithmetic_ui.py File Reference	46
5.4.1 Detailed Description	47
5.5 src/uis/binary_ui.py File Reference	47
5.5.1 Detailed Description	47
5.6 src/uis/bitwise_ui.py File Reference	47
5.6.1 Detailed Description	48
5.7 src/uis/BMI_ui.py File Reference	48
5.7.1 Detailed Description	48
5.8 src/uis/BodyFat_ui.py File Reference	48
5.8.1 Detailed Description	49
5.9 src/uis/Calculators/algebra_calculator.py File Reference	49
5.9.1 Detailed Description	49
5.9.2 Function Documentation	49
5.9.2.1 pyTheorem()	49
5.9.2.2 slopeOfLine()	50
5.9.2.3 yIntercept()	50
5.10 src/uis/Calculators/binary_calculator.py File Reference	51
5.10.1 Detailed Description	52
5.10.2 Function Documentation	52
5.10.2.1 binAdd()	52
5.10.2.2 binDiv()	52
5.10.2.3 binMult()	53
5.10.2.4 binPow()	53

5.10.2.5	binSub()	54
5.10.2.6	bitwiseAND()	54
5.10.2.7	bitwiseNOT()	55
5.10.2.8	bitwiseOR()	55
5.10.2.9	bitwiseXOR()	56
5.10.2.10	lshift()	56
5.10.2.11	rshift()	57
5.10.2.12	toDecimal()	57
5.10.2.13	toFloatingPoint()	58
5.11	src/uis/Calculators/conversion_calculator.py File Reference	58
5.11.1	Detailed Description	59
5.11.2	Function Documentation	59
5.11.2.1	convertBase()	59
5.11.2.2	convertCrypto()	59
5.11.2.3	convertCurrency()	60
5.11.2.4	convertRN()	60
5.11.3	Variable Documentation	61
5.11.3.1	conversion_table	61
5.11.3.2	cryptoCVals	61
5.11.3.3	currencyCVals	61
5.12	src/uis/Calculators/geometry_calculator.py File Reference	62
5.12.1	Detailed Description	62
5.12.2	Function Documentation	62
5.12.2.1	getPerimeter()	62
5.12.2.2	getVolume()	63
5.13	src/uis/Calculators/gpa_calculator.py File Reference	63
5.13.1	Detailed Description	64
5.13.2	Function Documentation	64
5.13.2.1	gpaCalculate()	64
5.14	src/uis/Calculators/health_calculator.py File Reference	64

5.14.1 Detailed Description	65
5.14.2 Function Documentation	65
5.14.2.1 bodyFat()	65
5.14.2.2 bodyMassIndex()	65
5.15 src/uis/Calculators/main_calculator.py File Reference	66
5.15.1 Detailed Description	66
5.16 src/uis/Calculators/stocks_calculator.py File Reference	66
5.16.1 Detailed Description	67
5.16.2 Function Documentation	67
5.16.2.1 calcUserGainLossCase1()	67
5.16.2.2 calcUserGainLossCase2()	67
5.17 src/uis/Conversion_ui.py File Reference	68
5.17.1 Detailed Description	68
5.18 src/uis/ConversionBase_ui.py File Reference	68
5.18.1 Detailed Description	69
5.19 src/uis/ConversionCrypto_ui.py File Reference	69
5.19.1 Detailed Description	69
5.20 src/uis/ConversionCurrency_ui.py File Reference	69
5.20.1 Detailed Description	70
5.21 src/uis/ConversionRN_ui.py File Reference	70
5.21.1 Detailed Description	70
5.22 src/uis/floating_point_ui.py File Reference	70
5.22.1 Detailed Description	71
5.23 src/uis/geometry_ui.py File Reference	71
5.23.1 Detailed Description	71
5.24 src/uis/gpa_ui.py File Reference	71
5.24.1 Detailed Description	72
5.25 src/uis/health_ui.py File Reference	72
5.25.1 Detailed Description	72
5.26 src/uis/perimeter_ui.py File Reference	72
5.26.1 Detailed Description	73
5.27 src/uis/pythagore_ui.py File Reference	73
5.27.1 Detailed Description	73
5.28 src/uis/stock_ui.py File Reference	73
5.28.1 Detailed Description	74
5.29 src/uis/volume_ui.py File Reference	74
5.29.1 Detailed Description	74

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

main_calculator.Calculator	18
QMainWindow	
algebra_ui.AlgebraWindow	7
area_ui.AreaWindow	8
binary_arithmetic_ui.BinArithmeticWindow	11
binary_ui.BinaryWindow	13
bitwise_ui.BitwiseWindow	14
BMI_ui.BMIWindow	16
BodyFat_ui.BFWindow	10
Conversion_ui.ConverterWindow	27
ConversionBase_ui.ConversionBaseWindow	21
ConversionCrypto_ui.ConversionCryptoWindow	22
ConversionCurrency_ui.ConversionCurrencyWindow	24
ConversionRN_ui.ConversionRNWindow	25
floating_point_ui.FloatingPointWindow	28
geometry_ui.GeometryWindow	29
gpa_ui.GPAWindow	30
health_ui.HealthWindow	32
main.MainWindow	33
perimeter_ui.PerimeterWindow	37
pythagore_ui.PythaWindow	39
stock_ui.StockWindow	40
volume_ui.VolumeWindow	42
Ui_MainWindow	
main.MainWindow	33

Chapter 2

Class Index

2.1 Class List

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

algebra_ui.AlgebraWindow	
AlgebraWindow is a class that implements the GUI components for the Algebra operation menu	7
area_ui.AreaWindow	
AreaWindow is a class that implements the GUI components for the Area operation	8
BodyFat_ui.BFWindow	
BFWindow is a class that implements the GUI components for the Body Fat operation	10
binary_arithmetic_ui.BinArithmeticWindow	
BinArithmeticWindow is a class that implements the GUI components for the Binary Arithmetic operations	11
binary_ui.BinaryWindow	
BinaryWindow is a class that implements the GUI components for the Binary operation menu	13
bitwise_ui.BitwiseWindow	
BitwiseWindow is a class that implements the GUI components for the Bitwise operations	14
BMI_ui.BMIWindow	
BMIWindow is a class that implements the GUI components for the BMI operation	16
main_calculator.Calculator	
Calculator is a class that implements the functionality of a basic calculator	18
ConversionBase_ui.ConversionBaseWindow	
ConversionBaseWindow is a class that implements the GUI components for the base conversion operation	21
ConversionCrypto_ui.ConversionCryptoWindow	
ConversionCryptoWindow is a class that implements the GUI components for the crypto conversion operation	22
ConversionCurrency_ui.ConversionCurrencyWindow	
ConversionBaseWindow is a class that implements the GUI components for the currency conversion operation	24
ConversionRN_ui.ConversionRNWindow	
ConversionBaseWindow is a class that implements the GUI components for the roman numeral conversion operation	25
Conversion_ui.ConverterWindow	
ConverterWindow is a class that implements the GUI components for the Conversion operation menu	27
floating_point_ui.FloatingPointWindow	
FloatingPointWindow is a class that implements the GUI components for the Floating Point operation	28

geometry_ui.GeometryWindow	
GeometryWindow is a class that implements the GUI components for the Geometry operation menu	29
gpa_ui.GPAWindow	
GPAWindow is a class that implements the GUI components for the GPA operation menu . . .	30
health_ui.HealthWindow	
HealthWindow is a class that implements the GUI components for the Health operation menu .	32
main.MainWindow	
MainWindow is a class that implements the GUI components for the Main menu	33
perimeter_ui.PerimeterWindow	
PerimeterWindow is a class that implements the GUI components for the Perimeter operation .	37
pythagore_ui.PythaWindow	
PythaWindow is a class that implements the GUI components for the Pythagorean Theorem operation	39
stock_ui.StockWindow	
StockWindow is a class that implements the GUI components for the Stock operation menu . .	40
volume_ui.VolumeWindow	
VolumeWindow is a class that implements the GUI components for the Volume operation . . .	42

Chapter 3

File Index

3.1 File List

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

src/main.py	Provides a class to display the Main window	45
src/uis/algebra_ui.py	Provides a class to display the Algebra window	45
src/uis/area_ui.py	Provides a class to display the Area window	46
src/uis/binary_arithmetic_ui.py	Provides a class to display the Binary Arithmetic window	46
src/uis/binary_ui.py	Provides a class to display the Binary window	47
src/uis/bitwise_ui.py	Provides a class to display the Bitwise Operation window	47
src/uis/BMI_ui.py	Provides a class to display the BMI window	48
src/uis/BodyFat_ui.py	Provides a class to display the Body Fat window	48
src/uis/Conversion_ui.py	Provides a class to display the Conversion window	68
src/uis/ConversionBase_ui.py	Provides a class to display the base conversion window	68
src/uis/ConversionCrypto_ui.py	Provides a class to display the crypto conversion window	69
src/uis/ConversionCurrency_ui.py	Provides a class to display the currency conversion window	69
src/uis/ConversionRN_ui.py	Provides a class to display the roman numeral conversion window	70
src/uis/floating_point_ui.py	Provides a class to display the Floating Point window	70
src/uis/geometry_ui.py	Provides a class to display the Geometry window	71
src/uis/gpa_ui.py	Provides a class to display the GPA window	71
src/uis/health_ui.py	Provides a class to display the Health window	72
src/uis/perimeter_ui.py	Provides a class to display the Perimeter window	72

src/uis/ pythagore_ui.py	
Provides a class to display the Pythagorean Theorem window	73
src/uis/ stock_ui.py	
Provides a class to display the Stocks window	73
src/uis/ volume_ui.py	
Provides a class to display the Volume window	74
src/uis/Calculators/ algebra_calculator.py	
Alegbraic algorithms	49
src/uis/Calculators/ binary_calculator.py	
Binary algorithms	51
src/uis/Calculators/ conversion_calculator.py	
Conversion Algorithms	58
src/uis/Calculators/ geometry_calculator.py	
Geometry algorithms	62
src/uis/Calculators/ gpa_calculator.py	
Gpa algorithms	63
src/uis/Calculators/ health_calculator.py	
Health algorithms	64
src/uis/Calculators/ main_calculator.py	
Main calculator algorithms	66
src/uis/Calculators/ stocks_calculator.py	
Stock algorithms	66

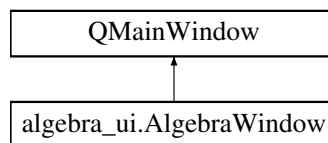
Chapter 4

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.
- `def closeEvent (self, event)`
Closes the window and any other algebra operation windows.

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

<code>path</code>	The current path on which the file is found. Default value is an empty path.
-------------------	--

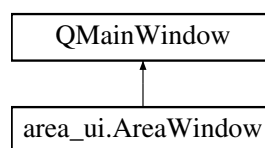
The documentation for this class was generated from the following file:

- [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.
- `def setFields (self)`
Changes and displays in text boxes corresponding to chosen shape.
- `def closeEvent (self, event)`
Closes window and clears inputs upon close.
- `def clearFields (self)`
Clears all input and output fields.

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

<code>path</code>	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

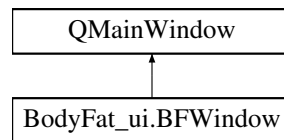
The documentation for this class was generated from the following file:

- `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.
- `def closeEvent (self, event)`
Resets fields and closes window.
- `def clearFields (self)`
Clears all input and output fields.

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

<i>path</i>	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, and weight from the user through input fields, and shows the user the result on the window

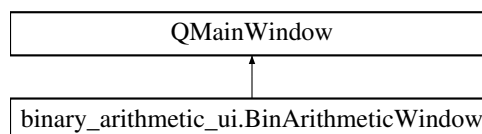
The documentation for this class was generated from the following file:

- [src/uis/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.
- `def closeEvent (self, event)`
Closes window and clears inputs upon close.
- `def clearFields (self)`
Clears all input and output fields.

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

<code>path</code>	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

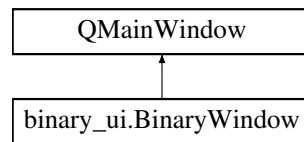
The documentation for this class was generated from the following file:

- [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.
- `def closeEvent (self, event)`
Closes the window and any other geometry operation windows.

Public Attributes

- `path`
- `xpath`
- `fp`
- `ba`
- `bw`

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

<code>path</code>	The current path on which the file is found. Default value is an empty path.
-------------------	--

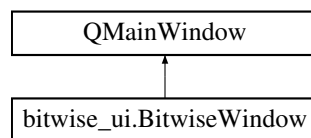
The documentation for this class was generated from the following file:

- [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.
- `def setFields(self)`
Changes and displays in text boxes corresponding to chosen operator.
- `def closeEvent(self, event)`
Closes window and clears inputs upon close.
- `def clearFields(self)`
Clears all input and output fields.

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

<i>path</i>	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

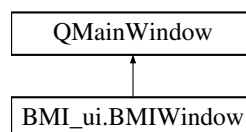
The documentation for this class was generated from the following file:

- [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.
- `def closeEvent (self, event)`
Resets fields and closes window.
- `def clearFields (self)`
Clears all input and output fields.

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

<code>path</code>	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 main_calculator.Calculator Class Reference

[Calculator](#) is a class that implements the functionality of a basic calculator.

Public Member Functions

- `def __init__ (self)`
The constructor of the [Calculator](#).
- `def getCurrNum (self)`
Retreives current displayed number.
- `def storeMem (self)`
Stores the current number.
- `def getMem (self)`
Displays current stored number.
- `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 evaluate (self)`
Evaluates operation.
- `def delete (self)`
Deletes entered input number by number.

Public Attributes

- **lineEdit**
- **currNum**
- **mem**

4.8.1 Detailed Description

[Calculator](#) is a class that implements the functionality of a basic calculator.

4.8.2 Member Function Documentation

4.8.2.1 addition()

```
def main_calculator.Calculator.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

4.8.2.2 division()

```
def main_calculator.Calculator.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

4.8.2.3 getMem()

```
def main_calculator.Calculator.getMem (
    self )
```

Displays current stored number.

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

4.8.2.4 left_bracket()

```
def main_calculator.Calculator.left_bracket (
    self )
```

Adds left bracket operation.

Clears the display and adds a left bracket to the operation

4.8.2.5 multiplication()

```
def main_calculator.Calculator.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

4.8.2.6 power()

```
def main_calculator.Calculator.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

4.8.2.7 reset()

```
def main_calculator.Calculator.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

4.8.2.8 right_bracket()

```
def main_calculator.Calculator.right_bracket (
    self )
```

Adds right bracket operation.

Clears the display and adds a right bracket to the operation

4.8.2.9 storeMem()

```
def main_calculator.Calculator.storeMem (
    self )
```

Stores the current number.

stores number for future use

4.8.2.10 subtraction()

```
def main_calculator.Calculator.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

4.8.2.11 valueInput()

```
def main_calculator.Calculator.valueInput (
    self,
    v )
```

Display value of input.

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

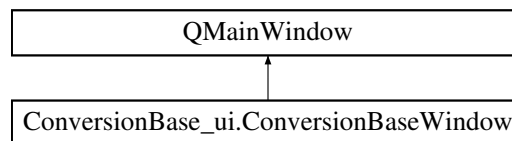
The documentation for this class was generated from the following file:

- [src/uis/Calculators/main_calculator.py](#)

4.9 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.
- `def closeEvent (self, event)`
Closes window and clears inputs upon close.
- `def clearFields (self)`
Clears all input and output fields.

Public Attributes

- `path`

4.9.1 Detailed Description

[ConversionBaseWindow](#) is a class that implements the GUI components for the base conversion operation.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 `__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

<i>path</i>	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 `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

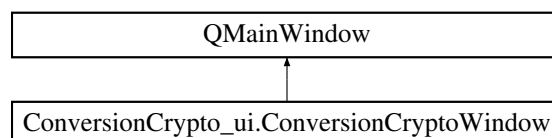
The documentation for this class was generated from the following file:

- [src/uis/ConversionBase_ui.py](#)

4.10 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.
- def `closeEvent` (self, event)
- def `clearFields` (self)
Clears all input and output fields.

Public Attributes

- `path`

4.10.1 Detailed Description

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

4.10.2 Constructor & Destructor Documentation

4.10.2.1 `__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

<code>path</code>	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 `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

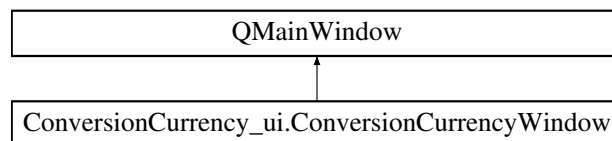
The documentation for this class was generated from the following file:

- [src/uis/ConversionCrypto_ui.py](#)

4.11 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.
- `def closeEvent (self, event)`
- `def clearFields (self)`
Clears all input and output fields.

Public Attributes

- **path**

4.11.1 Detailed Description

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

4.11.2 Constructor & Destructor Documentation

4.11.2.1 `__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

<i>path</i>	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 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

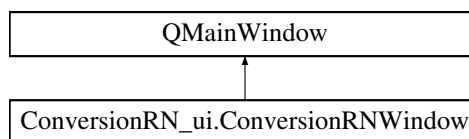
The documentation for this class was generated from the following file:

- [src/uis/ConversionCurrency_ui.py](#)

4.12 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.
- `def closeEvent (self, event)`
- `def clearFields (self)`
Clears all input and output fields.

Public Attributes

- **path**

4.12.1 Detailed Description

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

4.12.2 Constructor & Destructor Documentation

4.12.2.1 __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

<i>path</i>	The current path on which the file is found. Default value is an empty path.
-------------	--

4.12.3 Member Function Documentation

4.12.3.1 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

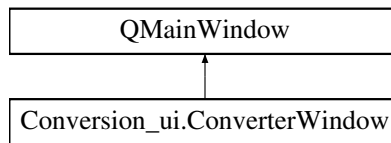
The documentation for this class was generated from the following file:

- [src/uis/ConversionRN_ui.py](#)

4.13 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.
- `def closeEvent (self, event)`
Closes the window and any other algebra operation windows.

Public Attributes

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

4.13.1 Detailed Description

[ConverterWindow](#) is a class that implements the GUI components for the Conversion operation menu.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 `__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

<code>path</code>	The current path on which the file is found. Default value is an empty path.
-------------------	--

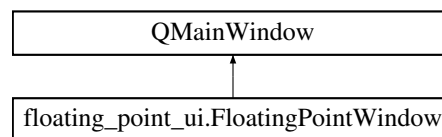
The documentation for this class was generated from the following file:

- `src/uis/Conversion_ui.py`

4.14 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.
- `def closeEvent(self, event)`
Closes window and clears inputs upon close.
- `def clearFields(self)`
Clears all input and output fields.

Public Attributes

- `path`

4.14.1 Detailed Description

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

4.14.2 Constructor & Destructor Documentation

4.14.2.1 __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

<code>path</code>	The current path on which the file is found. Default value is an empty path.
-------------------	--

4.14.3 Member Function Documentation

4.14.3.1 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

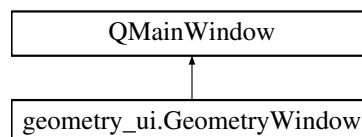
The documentation for this class was generated from the following file:

- [src/uis/floating_point_ui.py](#)

4.15 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.
- `def closeEvent (self, event)`
Closes the window and any other geometry operation windows.

Public Attributes

- `path`
- `xpath`
- `a`
- `p`
- `v`

4.15.1 Detailed Description

[GeometryWindow](#) is a class that implements the GUI components for the Geometry operation menu.

4.15.2 Constructor & Destructor Documentation

4.15.2.1 `__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

<i>path</i>	The current path on which the file is found. Default value is an empty path.
-------------	--

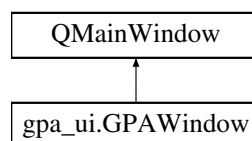
The documentation for this class was generated from the following file:

- [src/uis/geometry_ui.py](#)

4.16 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__ (self, path="")`
- `def gpa (self)`
Displays the 12.0 gpa from the metrics the user provides.
- `def add (self)`
Sets up the visual aspect of the calculator and sets up inputs.
- `def closeEvent (self, event)`
Closes window and clears inputs upon close.
- `def clearFields (self)`
Clears all input and output fields.

Public Attributes

- `path`

Static Public Attributes

- list `gradeList` = []
The constructor of the GPA window.
- int `index` = 0
- int `totalWeight` = 0

4.16.1 Detailed Description

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

4.16.2 Member Function Documentation

4.16.2.1 `add()`

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

Sets up the visual aspect of the calculator and sets up inputs.

Takes in the grades and their weights through input fields and shows the users grades in the window as a tuple.

4.16.2.2 `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

4.16.3 Member Data Documentation

4.16.3.1 `gradeList`

```
list gpa_ui.GPAWindow.gradeList = [] [static]
```

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

<i>path</i>	The current path on which the file is found. Default value is an empty path.
-------------	--

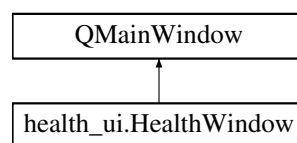
The documentation for this class was generated from the following file:

- [src/uis/gpa_ui.py](#)

4.17 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.
- `def closeEvent (self, event)`
Closes the window and any other health operation windows.

Public Attributes

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

4.17.1 Detailed Description

[HealthWindow](#) is a class that implements the GUI components for the Health operation menu.

4.17.2 Constructor & Destructor Documentation

4.17.2.1 __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

<code>path</code>	The current path on which the file is found. Default value is an empty path.
-------------------	--

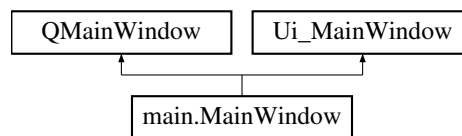
The documentation for this class was generated from the following file:

- [src/uis/health_ui.py](#)

4.18 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 [displayError](#) (self)
Displays error message.
- 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.
- def `closeEvent` (self, event)
Closes the window and any other open windows upon confirmation.

Static Public Member Functions

- def `credits` ()

Public Attributes

- `converters`
- `algebra`
- `stock`
- `health`
- `gpa`
- `binary`
- `geo`
- `calc`

4.18.1 Detailed Description

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

4.18.2 Constructor & Destructor Documentation

4.18.2.1 `__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

<code>path</code>	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 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

4.18.3.2 display()

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

Displays number to user.

displays the number that is currently stored onto the calculator display

4.18.3.3 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

4.18.3.4 equals()

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

Evaluates operation.

Evaluates operation and displays answer

4.18.3.5 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

4.18.3.6 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

4.18.3.7 left_bracket()

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

Adds left bracket operation.

Clears the display and adds a left bracket to the operation

4.18.3.8 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

4.18.3.9 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

4.18.3.10 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

4.18.3.11 right_bracket()

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

Adds right bracket operation.

Clears the display and adds a right bracket to the operation

4.18.3.12 storeMem()

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

Stores the current number.

stores number for future use

4.18.3.13 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

4.18.3.14 valueInput()

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

Display value of input.

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

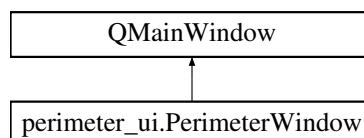
The documentation for this class was generated from the following file:

- src/[main.py](#)

4.19 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.
- `def setFields (self)`
Changes and displays in text boxes corresponding to chosen shape.
- `def closeEvent (self, event)`
Closes window and clears inputs upon close.
- `def clearFields (self)`
Clears all input and output fields.

Public Attributes

- `path`

4.19.1 Detailed Description

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

4.19.2 Constructor & Destructor Documentation

4.19.2.1 __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

<code>path</code>	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 `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

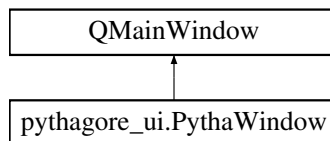
The documentation for this class was generated from the following file:

- `src/uis/perimeter_ui.py`

4.20 `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.
- `def closeEvent (self, event)`
Resets fields upon close of window.
- `def clearFields (self)`
Clears all input and output fields.

Public Attributes

- `path`

4.20.1 Detailed Description

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

4.20.2 Constructor & Destructor Documentation

4.20.2.1 `__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

<i>path</i>	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 `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

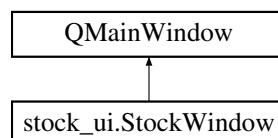
The documentation for this class was generated from the following file:

- [src/uis/pythagore_ui.py](#)

4.21 `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.
- def `closeEvent` (self, event)
Closes window and clears inputs upon close.
- def `clearFields` (self)
Clears all input and output fields.

Public Attributes

- `path`

4.21.1 Detailed Description

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

4.21.2 Constructor & Destructor Documentation

4.21.2.1 `__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

<code>path</code>	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 `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

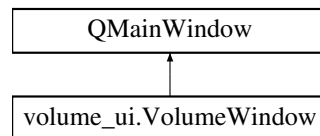
The documentation for this class was generated from the following file:

- [src/uis/stock_ui.py](#)

4.22 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.
- `def setFields(self)`
Changes and displays in text boxes corresponding to chosen shape.
- `def closeEvent(self, event)`
Closes window and clears inputs upon close.
- `def clearFields(self)`
Clears all input and output fields.

Public Attributes

- `path`

4.22.1 Detailed Description

[VolumeWindow](#) is a class that implements the GUI components for the Volume operation.

4.22.2 Constructor & Destructor Documentation

4.22.2.1 __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

<i>path</i>	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 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 30, 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 30, 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 30, 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 30, 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

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

Returns

The length of the missing side

Exceptions

<i>ValueError</i>	Throws an exception if hypotenuse is not the longest side or sides are not postive
-------------------	--

5.9.2.2 slopeOfLine()

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

Calculates slope of a line given 2 points.

Parameters

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

Returns

The slope of the line

Exceptions

<i>ZeroDivisionError</i>	Throws an exception if x2 and x1 are equal
--------------------------	--

5.9.2.3 yIntercept()

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

$$\begin{matrix} x, \\ y \end{matrix}$$

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

Parameters

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

Returns

The y-intercept of the line

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

<i>n</i>	Binary number
<i>m</i>	Binary number

Returns

Sum of n and m

Exceptions

<i>ValueError</i>	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

<i>n</i>	Binary number
<i>m</i>	Binary number

Returns

Quotient of n and m

Exceptions

<i>ZeroDivisionError</i>	Throws an exception if m equals zero
--------------------------	--------------------------------------

5.10.2.3 binMult()

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

Calculates product of two binary numbers.

Parameters

<i>n</i>	Binary number
<i>m</i>	Binary number

Returns

Product of n and m

Exceptions

<i>ValueError</i>	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

<i>n</i>	Binary number
<i>m</i>	Binary number

Returns

Power of n to the m

Exceptions

<i>ValueError</i>	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

<i>n</i>	Binary number
<i>m</i>	Binary number

Returns

Difference of n and m

Exceptions

<i>ValueError</i>	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

<i>n</i>	Binary number
<i>m</i>	Binary number

Returns

Bitwise AND of n and m

Exceptions

<i>ValueError</i>	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

<i>n</i>	Binary number
----------	---------------

Returns

Bitwise NOT of n

Exceptions

<i>ValueError</i>	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

<i>n</i>	Binary number
<i>m</i>	Binary number

Returns

Bitwise OR of n and m

Exceptions

<i>ValueError</i>	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

<i>n</i>	Binary number
<i>m</i>	Binary number

Returns

Bitwise XOR of n and m

Exceptions

<i>ValueError</i>	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

<i>n</i>	Binary number
<i>shiftNum</i>	Number of shifts
<i>length</i>	Length of binary number

Returns

n bit shifted leftward shiftNum times

Exceptions

<i>ValueError</i>	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

<i>n</i>	Binary number
<i>shiftNum</i>	Number of shifts
<i>length</i>	Length of binary number

Returns

n bit shifted rightward shiftNum times

Exceptions

<i>ValueError</i>	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

<i>n</i>	IEEE 754 binary number
----------	------------------------

Returns

Decimal representation

Exceptions

<i>ValueError</i>	Throws an exception if n is invalid
-------------------	-------------------------------------

5.10.2.13 toFloatingPoint()

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

Calculates IEEE 754 representation from decimal.

Parameters

<i>n</i>	Decimal number
----------	----------------

Returns

IEEE 754 floating point representation

Exceptions

<i>ValueError</i>	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.decToBin** (x, arr)
- def **conversion_calculator.decToOct** (x, arr)
- def **conversion_calculator.decToHex** (x, arr)
- 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.
- def **conversion_calculator.dectoRN** (x)
- def **conversion_calculator.romanToInt** (x)

Variables

- dictionary `conversion_calculator.currencyCVals`
- dictionary `conversion_calculator.cryptoCVals`
- dictionary `conversion_calculator.conversion_table`

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

<i>initialVal</i>	A real number that represents the initial numerical value
<i>baseFrom</i>	A string value that represents the base of the initialVal
<i>baseTo</i>	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

<i>initialVal</i>	A real number that represents the cryptocurrency value
<i>currFrom</i>	A string value that represents the cryptocurrency of the initialVal
<i>currTo</i>	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

<i>initialVal</i>	A real number that represents the currency value
<i>currFrom</i>	A string value that represents the currency of the initialVal
<i>currTo</i>	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,
    RNTTo )
```

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

Parameters

<i>initialVal</i>	A string that represents the initial value
<i>RNFrom</i>	A string value that represents the type of the initialVal
<i>RNTTo</i>	A string value that represents which type to convert to

Returns

the final value after conversion

5.11.3 Variable Documentation**5.11.3.1 conversion_table**

dictionary conversion_calculator.conversion_table

Initial value:

```
1 = {0: '0', 1: '1', 2: '2', 3: '3',
2      4: '4', 5: '5', 6: '6', 7: '7',
3      8: '8', 9: '9', 10: 'A', 11: 'B',
4      12: 'C', 13: 'D', 14: 'E', 15: 'F'}
```

5.11.3.2 cryptoCVals

dictionary conversion_calculator.cryptoCVals

Initial value:

```
1 = {
2     ("Bitcoin", "Ethereum"): 14.04,
3     ("Bitcoin", "Dogecoin"): 344342.42,
4     ("Ethereum", "Bitcoin"): 0.071,
5     ("Ethereum", "Dogecoin"): 24483.37,
6     ("Dogecoin", "Bitcoin"): 0.0000029,
7     ("Dogecoin", "Ethereum"): 0.000041,
8
9 }
```

5.11.3.3 currencyCVals

dictionary conversion_calculator.currencyCVals

Initial value:

```
1 = {
2     ("US Dollars", "Euros"): 0.91,
3     ("US Dollars", "Jap Yen"): 119.45,
4     ("US Dollars", "Pounds"): 0.76,
5     ("US Dollars", "CA Dollars"): 1.26,
6
7     ("Euros", "US Dollars"): 1.10,
8     ("Euros", "Jap Yen"): 131.70,
9     ("Euros", "Pounds"): 0.84,
10    ("Euros", "CA Dollars"): 1.39,
11
12    ("Jap Yen", "US Dollars"): 0.0084,
13    ("Jap Yen", "Euros"): 0.0076,
14    ("Jap Yen", "Pounds"): 0.0064,
15    ("Jap Yen", "CA Dollars"): 0.011,
16
17    ("Pounds", "US Dollars"): 1.32,
18    ("Pounds", "Euros"): 1.19,
19    ("Pounds", "Jap Yen"): 157.36,
20    ("Pounds", "CA Dollars"): 1.66,
21
22    ("CA Dollars", "US Dollars"): 0.79,
23    ("CA Dollars", "Euros"): 0.72,
24    ("CA Dollars", "Jap Yen"): 0.60,
25    ("CA Dollars", "Pounds"): 94.70,
26 }
```

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

Geometry algorithms.

Functions

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

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

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

Parameters

<i>shape</i>	An integer that represents the shape
<i>a</i>	Side length a
<i>b</i>	Side length b
<i>c</i>	Side length c
<i>r</i>	Radius

Returns

Perimeter

Exceptions

<i>ValueError</i>	Throws an exception if required side lengths or radius are invalid
-------------------	--

5.12.2.2 getVolume()

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

Calculates volume of given shape with given dimensions.

Parameters

<i>shape</i>	An integer that represents the shape
<i>l</i>	Length
<i>w</i>	Width
<i>h</i>	Height
<i>r</i>	Radius

Returns

Volume

Exceptions

<i>ValueError</i>	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, weightTotal)
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,
    weightTotal )
```

Calculates the GPA of the user.

Parameters

<i>gradeList</i>	A list with floats that carries the grades of each class multiplied by the weight of the class.
------------------	---

Returns

The average GPA of the student

Exceptions

<i>ZeroDivisionError</i>	Throws an exception if weightTotal is zero
--------------------------	--

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](#) (weight, height, gender, age)
Calculates the body fat percentage of a person.

5.14.1 Detailed Description

Health algorithms.

Date

March 31, 2022

5.14.2 Function Documentation

5.14.2.1 bodyFat()

```
def health_calculator.bodyFat (
    weight,
    height,
    gender,
    age )
```

Calculates the body fat percentage of a person.

Parameters

<i>gender</i>	A string that represents the gender of the person
<i>height</i>	A real number that represents the height of the user in inches
<i>weight</i>	A real number that represents the weight of the user in pounds
<i>age</i>	A real number that represents how old the person is

Returns

A tuple of body fat percentage and a string with its meaning

Exceptions

<i>ValueError</i>	Throws an exception if a measurement is not valid
<i>ZeroDivisionError</i>	Throws an exception if height is 0

5.14.2.2 bodyMassIndex()

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

Calculates the body mass index of a person.

Parameters

<i>weight</i>	A real number that represents the weight of the user in pounds
<i>height</i>	A real number that represents the height of the user in inches

Returns

A tuple of bmi coefficient and a string with its meaning

Exceptions

<i>ValueError</i>	Throws an exception if height or weight is not valid
<i>ZeroDivisionError</i>	Throws an exception if height is 0

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

main calculator algorithms

Classes

- class `main_calculator.Calculator`
Calculator is a class that implements the functionality of a basic calculator.

5.15.1 Detailed Description

main calculator algorithms

Date

April 5, 2022

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

stock algorithms

Functions

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

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

Returns

The gain or loss on the stock

Exceptions

<i>ValueError</i>	Throws an exception if inputs are strings
-------------------	---

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

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

Returns

The gain or loss on the stock

Exceptions

<i>ValueError</i>	Throws an exception if inputs are strings
-------------------	---

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 30, 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 30, 2022

5.28 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()

5.28.1 Detailed Description

Provides a class to display the Stocks window.

Date

March 17, 2022

5.29 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()

5.29.1 Detailed Description

Provides a class to display the Volume window.

Date

March 18, 2022

Index

- `__init__`
 - `algebra_ui::AlgebraWindow`, 8
 - `area_ui::AreaWindow`, 9
 - `BMI_ui::BMIWindow`, 17
 - `binary_arithmetic_ui::BinArithmeticWindow`, 12
 - `binary_ui::BinaryWindow`, 13
 - `bitwise_ui::BitwiseWindow`, 14
 - `BodyFat_ui::BFWindow`, 10
 - `Conversion_ui::ConverterWindow`, 27
 - `ConversionBase_ui::ConversionBaseWindow`, 22
 - `ConversionCrypto_ui::ConversionCryptoWindow`, 23
 - `ConversionCurrency_ui::ConversionCurrencyWindow`, 24
 - `ConversionRN_ui::ConversionRNWindow`, 26
 - `floating_point_ui::FloatingPointWindow`, 28
 - `geometry_ui::GeometryWindow`, 30
 - `health_ui::HealthWindow`, 32
 - `main::MainWindow`, 34
 - `perimeter_ui::PerimeterWindow`, 38
 - `pythagore_ui::PythaWindow`, 40
 - `stock_ui::StockWindow`, 41
 - `volume_ui::VolumeWindow`, 42
- `add`
 - `gpa_ui::GPAWindow`, 31
- `addition`
 - `main::MainWindow`, 35
 - `main_calculator::Calculator`, 19
- `algebra_calculator.py`
 - `pyTheorem`, 49
 - `slopeOfLine`, 50
 - `yIntercept`, 50
- `algebra_ui::AlgebraWindow`, 7
- `algebra_ui::AlgebraWindow`
 - `__init__`, 8
- `area`
 - `area_ui::AreaWindow`, 9
- `area_ui::AreaWindow`, 8
- `area_ui::AreaWindow`
 - `__init__`, 9
 - `area`, 9
- `BMI_ui::BMIWindow`, 16
- `BMI_ui::BMIWindow`
 - `__init__`, 17
 - `bmi`, 17
- `baseconvert`
 - `ConversionBase_ui::ConversionBaseWindow`, 22
- `bf`
 - `BodyFat_ui::BFWindow`, 11
- `binAdd`
 - `binary_calculator.py`, 52
- `binArithmetic`
 - `binary_arithmetic_ui::BinArithmeticWindow`, 12
- `binDiv`
 - `binary_calculator.py`, 52
- `binMult`
 - `binary_calculator.py`, 53
- `binPow`
 - `binary_calculator.py`, 53
- `binSub`
 - `binary_calculator.py`, 54
- `binary_arithmetic_ui::BinArithmeticWindow`, 11
- `binary_arithmetic_ui::BinArithmeticWindow`
 - `__init__`, 12
 - `binArithmetic`, 12
- `binary_calculator.py`
 - `binAdd`, 52
 - `binDiv`, 52
 - `binMult`, 53
 - `binPow`, 53
 - `binSub`, 54
 - `bitwiseAND`, 54
 - `bitwiseNOT`, 55
 - `bitwiseOR`, 55
 - `bitwiseXOR`, 56
 - `lshift`, 56
 - `rshift`, 57
 - `toDecimal`, 57
 - `toFloatingPoint`, 58
- `binary_ui::BinaryWindow`, 13
- `binary_ui::BinaryWindow`
 - `__init__`, 13
- `bitwise`
 - `bitwise_ui::BitwiseWindow`, 16
- `bitwise_ui::BitwiseWindow`, 14
- `bitwise_ui::BitwiseWindow`
 - `__init__`, 14
 - `bitwise`, 16
- `bitwiseAND`
 - `binary_calculator.py`, 54
- `bitwiseNOT`
 - `binary_calculator.py`, 55
- `bitwiseOR`
 - `binary_calculator.py`, 55
- `bitwiseXOR`
 - `binary_calculator.py`, 56
- `bmi`

BMI_ui::BMIWindow, 17
 bodyFat
 health_calculator.py, 65
 BodyFat_ui.BFWindow, 10
 BodyFat_ui::BFWindow
 __init__, 10
 bf, 11
 bodyMassIndex
 health_calculator.py, 65

 calcUserGainLossCase1
 stocks_calculator.py, 67
 calcUserGainLossCase2
 stocks_calculator.py, 67
 conversion_calculator.py
 conversion_table, 61
 convertBase, 59
 convertCrypto, 59
 convertCurrency, 60
 convertRN, 60
 cryptoCVals, 61
 currencyCVals, 61
 conversion_table
 conversion_calculator.py, 61
 Conversion_ui.ConverterWindow, 27
 Conversion_ui::ConverterWindow
 __init__, 27
 ConversionBase_ui.ConversionBaseWindow, 21
 ConversionBase_ui::ConversionBaseWindow
 __init__, 22
 baseconvert, 22
 ConversionCrypto_ui.ConversionCryptoWindow, 22
 ConversionCrypto_ui::ConversionCryptoWindow
 __init__, 23
 cryptoconvert, 23
 ConversionCurrency_ui.ConversionCurrencyWindow, 24
 ConversionCurrency_ui::ConversionCurrencyWindow
 __init__, 24
 currconvert, 25
 ConversionRN_ui.ConversionRNWindow, 25
 ConversionRN_ui::ConversionRNWindow
 __init__, 26
 RNconvert, 26
 convertBase
 conversion_calculator.py, 59
 convertCrypto
 conversion_calculator.py, 59
 convertCurrency
 conversion_calculator.py, 60
 convertRN
 conversion_calculator.py, 60
 cryptoCVals
 conversion_calculator.py, 61
 cryptoconvert
 ConversionCrypto_ui::ConversionCryptoWindow, 23
 currconvert
 ConversionCurrency_ui::ConversionCurrencyWindow, 25
 currencyCVals
 conversion_calculator.py, 61

 display
 main::MainWindow, 35
 division
 main::MainWindow, 35
 main_calculator::Calculator, 19

 equals
 main::MainWindow, 35

 floating_point
 floating_point_ui::FloatingPointWindow, 29
 floating_point_ui.FloatingPointWindow, 28
 floating_point_ui::FloatingPointWindow
 __init__, 28
 floating_point, 29

 geometry_calculator.py
 getPerimeter, 62
 getVolume, 63
 geometry_ui.GeometryWindow, 29
 geometry_ui::GeometryWindow
 __init__, 30
 getMem
 main::MainWindow, 35
 main_calculator::Calculator, 19
 getPerimeter
 geometry_calculator.py, 62
 getVolume
 geometry_calculator.py, 63
 gpa
 gpa_ui::GPAWindow, 31
 gpa_calculator.py
 gpaCalculate, 64
 gpa_ui.GPAWindow, 30
 gpa_ui::GPAWindow
 add, 31
 gpa, 31
 gradeList, 31
 gpaCalculate
 gpa_calculator.py, 64
 gradeList
 gpa_ui::GPAWindow, 31

 health_calculator.py
 bodyFat, 65
 bodyMassIndex, 65
 health_ui.HealthWindow, 32
 health_ui::HealthWindow
 __init__, 32

 keyPressEvent
 main::MainWindow, 35

 left_bracket
 main::MainWindow, 36

- main_calculator::Calculator, 19
- lshift
 - binary_calculator.py, 56
- main.MainWindow, 33
- main::MainWindow
 - __init__, 34
 - addition, 35
 - display, 35
 - division, 35
 - equals, 35
 - getMem, 35
 - keyPressEvent, 35
 - left_bracket, 36
 - multiplication, 36
 - power, 36
 - reset, 36
 - right_bracket, 36
 - storeMem, 37
 - subtraction, 37
 - valueInput, 37
- main_calculator.Calculator, 18
- main_calculator::Calculator
 - addition, 19
 - division, 19
 - getMem, 19
 - left_bracket, 19
 - multiplication, 19
 - power, 19
 - reset, 20
 - right_bracket, 20
 - storeMem, 20
 - subtraction, 20
 - valueInput, 20
- multiplication
 - main::MainWindow, 36
 - main_calculator::Calculator, 19
- perimeter
 - perimeter_ui::PerimeterWindow, 38
- perimeter_ui.PerimeterWindow, 37
- perimeter_ui::PerimeterWindow
 - __init__, 38
 - perimeter, 38
- power
 - main::MainWindow, 36
 - main_calculator::Calculator, 19
- pyTheorem
 - algebra_calculator.py, 49
- pytha
 - pythagore_ui::PythaWindow, 40
- pythagore_ui.PythaWindow, 39
- pythagore_ui::PythaWindow
 - __init__, 40
 - pytha, 40
- RNconvert
 - ConversionRN_ui::ConversionRNWindow, 26
- reset
 - main::MainWindow, 36
 - main_calculator::Calculator, 20
- right_bracket
 - main::MainWindow, 36
 - main_calculator::Calculator, 20
- rshift
 - binary_calculator.py, 57
- slopeOfLine
 - algebra_calculator.py, 50
- src/main.py, 45
- src/uis/BMI_ui.py, 48
- src/uis/BodyFat_ui.py, 48
- src/uis/Calculators/algebra_calculator.py, 49
- src/uis/Calculators/binary_calculator.py, 51
- src/uis/Calculators/conversion_calculator.py, 58
- src/uis/Calculators/geometry_calculator.py, 62
- src/uis/Calculators/gpa_calculator.py, 63
- src/uis/Calculators/health_calculator.py, 64
- src/uis/Calculators/main_calculator.py, 66
- src/uis/Calculators/stocks_calculator.py, 66
- src/uis/Conversion_ui.py, 68
- src/uis/ConversionBase_ui.py, 68
- src/uis/ConversionCrypto_ui.py, 69
- src/uis/ConversionCurrency_ui.py, 69
- src/uis/ConversionRN_ui.py, 70
- src/uis/algebra_ui.py, 45
- src/uis/area_ui.py, 46
- src/uis/binary_arithmetic_ui.py, 46
- src/uis/binary_ui.py, 47
- src/uis/bitwise_ui.py, 47
- src/uis/floating_point_ui.py, 70
- src/uis/geometry_ui.py, 71
- src/uis/gpa_ui.py, 71
- src/uis/health_ui.py, 72
- src/uis/perimeter_ui.py, 72
- src/uis/pythagore_ui.py, 73
- src/uis/stock_ui.py, 73
- src/uis/volume_ui.py, 74
- stock
 - stock_ui::StockWindow, 41
- stock_ui.StockWindow, 40
- stock_ui::StockWindow
 - __init__, 41
 - stock, 41
- stocks_calculator.py
 - calcUserGainLossCase1, 67
 - calcUserGainLossCase2, 67
- storeMem
 - main::MainWindow, 37
 - main_calculator::Calculator, 20
- subtraction
 - main::MainWindow, 37
 - main_calculator::Calculator, 20
- toDecimal
 - binary_calculator.py, 57
- toFloatingPoint
 - binary_calculator.py, 58

valueInput
 main::MainWindow, [37](#)
 main_calculator::Calculator, [20](#)
volume
 volume_ui::VolumeWindow, [43](#)
volume_ui.VolumeWindow, [42](#)
volume_ui::VolumeWindow
 __init__, [42](#)
 volume, [43](#)

yIntercept
 algebra_calculator.py, [50](#)