

/home/bblaskova/NetBeansProjects/IVS-New-Meta

Generated by Doxygen 1.8.13

Contents

1	IVS-New-Meta	2
2	Hierarchical Index	2
2.1	Class Hierarchy	2
3	Class Index	2
3.1	Class List	2
4	File Index	3
4.1	File List	3
5	Class Documentation	3
5.1	Calculator Class Reference	3
5.1.1	Member Function Documentation	4
5.2	CreditsController Class Reference	4
5.2.1	Member Function Documentation	5
5.3	GUI_Controller Class Reference	5
5.3.1	Member Function Documentation	6
5.4	Help_controller Class Reference	14
5.5	math Class Reference	14
5.5.1	Member Function Documentation	15
5.6	mathTest Class Reference	22
5.6.1	Member Function Documentation	22
6	File Documentation	24
6.1	src/Calculator.java File Reference	24
6.1.1	Detailed Description	25
6.2	src/GUI_Controller.java File Reference	25
6.2.1	Detailed Description	25
6.3	src/math.java File Reference	25
6.3.1	Detailed Description	25
6.4	test/mathTest.java File Reference	25
6.4.1	Detailed Description	25

Index	27
-----------------------	----

1 IVS-New-Meta

Prostredi

Ubuntu 32bit Ubuntu 64bit Windows 32bit Windows 64bit

Autori

Nazev tymu

- xblask04 Barbora Blašková
- xcrkon00 Jakub Crkoň
- xkosti05 Gabriel Košťalik

Licence

GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

2 Hierarchical Index

2.1 Class Hierarchy

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

math	14
mathTest	22
Application	
Calculator	3
Initializable	
CreditsController	4
GUI_Controller	5
Help_controller	14

3 Class Index

3.1 Class List

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

Calculator	3
CreditsController	4
GUI_Controller	5
Help_controller	14
math	14
mathTest	22

4 File Index

4.1 File List

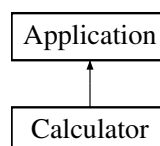
Here is a list of all documented files with brief descriptions:

src/Calculator.java Main class - Calculator	24
src/GUI_Controller.java Contains button actions	25
src/math.java Class with mathematic operation used in calculator	25
test/mathTest.java Tests for math.java library	25

5 Class Documentation

5.1 Calculator Class Reference

Inheritance diagram for Calculator:



Public Member Functions

- void **start** (Stage stage) throws Exception

Static Public Member Functions

- static void [main](#) (String[] args)
Main function of the calculator.

5.1.1 Member Function Documentation

5.1.1.1 main()

```
static void Calculator.main (
    String [] args ) [inline], [static]
```

Main function of the calculator.

Parameters

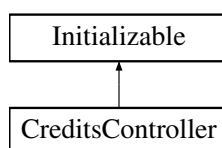
<i>args</i>	the command line arguments
-------------	----------------------------

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

- [src/Calculator.java](#)

5.2 CreditsController Class Reference

Inheritance diagram for CreditsController:



Public Member Functions

- void [initialize](#) (URL url, ResourceBundle rb)

Private Member Functions

- void **GNU_link** (ActionEvent event)
- void **creditsAction** (ActionEvent event)
- void **closeAction** (ActionEvent event)

Private Attributes

- Button **closeAbout**
- ToggleButton **creditsButton**
- Pane **bg_pane**
- TextArea **bg_pane_text**
- TextField **dsg**
- TextField **crt**
- TextField **dcm**
- Hyperlink **link**

5.2.1 Member Function Documentation

5.2.1.1 initialize()

```
void CreditsController.initialize (
    URL url,
    ResourceBundle rb ) [inline]
```

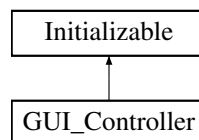
Initializes the controller class.

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

- src/CreditsController.java

5.3 GUI_Controller Class Reference

Inheritance diagram for GUI_Controller:



Public Member Functions

- void **initialize** (URL url, ResourceBundle rb)

Private Member Functions

- void **zeroAction** (ActionEvent event)
Zero button pressed.
- void **oneAction** (ActionEvent event)
Button '1' pressed.
- void **twoAction** (ActionEvent event)
Button '2' pressed.
- void **threeAction** (ActionEvent event)
Button '3' pressed.
- void **fourAction** (ActionEvent event)
Button '4' pressed.
- void **fiveAction** (ActionEvent event)
Button '5' pressed.
- void **sixAction** (ActionEvent event)
Button '6' pressed.
- void **sevenAction** (ActionEvent event)
Button '7' pressed.
- void **eighthAction** (ActionEvent event)

- Button '8' pressed.*
- void [nineAction](#) (ActionEvent event)
- Button '9' pressed.*
- void [DELAction](#) (ActionEvent event)
- Button 'DEL' pressed.*
- void [CAAction](#) (ActionEvent event)
- Button 'CA' pressed.*
- void [equalAction](#) (ActionEvent event)
- Button '=' pressed.*
- void [dotAction](#) (ActionEvent event)
- Button '.' pressed.*
- void [plusAction](#) (ActionEvent event)
- Button '+' pressed.*
- void [minusAction](#) (ActionEvent event)
- Button '-' pressed.*
- void [moduloAction](#) (ActionEvent event)
- Button " pressed.*
- void [sqrtAction](#) (ActionEvent event)
- Button 'sqrt' pressed.*
- void [powAction](#) (ActionEvent event)
- Button '^' pressed.*
- void [multiAction](#) (ActionEvent event)
- Button '*' pressed.*
- void [divAction](#) (ActionEvent event)
- Button '/' pressed.*
- void [factAction](#) (ActionEvent event)
- Button '!' pressed.*
- void [helpAction](#) (ActionEvent event)
- Button '?' pressed.*
- void [mid_result](#) ()
- Function handling more than one operation in row.*
- long [is_int](#) (double x)
- Function will check whether the given number is integer or not.*
- void [reset](#) ()
- noooo idea*

Private Attributes

- TextField **display**
- TextField **OP_display**
- double **operand_one**
- double **operand_two**
- boolean **dot_flag**
- boolean **reset_D**
- int **operation**

5.3.1 Member Function Documentation

5.3.1.1 CAction()

```
void GUI_Controller.CAction (
    ActionEvent event ) [inline], [private]
```

Button 'CA' pressed.

Parameters

<i>event</i>	<i>todo</i>
--------------	-------------

5.3.1.2 DELAction()

```
void GUI_Controller.DELAction (  
    ActionEvent event ) [inline], [private]
```

Button 'DEL' pressed.

Parameters

<i>event</i>	<i>todo</i>
--------------	-------------

5.3.1.3 divAction()

```
void GUI_Controller.divAction (  
    ActionEvent event ) [inline], [private]
```

Button '/' pressed.

Parameters

<i>event</i>	<i>todo</i>
--------------	-------------

5.3.1.4 dotAction()

```
void GUI_Controller.dotAction (  
    ActionEvent event ) [inline], [private]
```

Button '.' pressed.

Parameters

<i>event</i>	<i>todo</i>
--------------	-------------

5.3.1.5 eigthAction()

```
void GUI_Controller.eigthAction (  
    ActionEvent event ) [inline], [private]
```

Button '8' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.6 equalAction()

```
void GUI_Controller.equalAction (
    ActionEvent event ) [inline], [private]
```

Button '=' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.7 factAction()

```
void GUI_Controller.factAction (
    ActionEvent event ) [inline], [private]
```

Button '!' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.8 fiveAction()

```
void GUI_Controller.fiveAction (
    ActionEvent event ) [inline], [private]
```

Button '5' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.9 fourAction()

```
void GUI_Controller.fourAction (
    ActionEvent event ) [inline], [private]
```

Button '4' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.10 helpAction()

```
void GUI_Controller.helpAction (  
   (ActionEvent event ) [inline], [private]
```

Button '?' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.11 is_int()

```
long GUI_Controller.is_int (  
    double x ) [inline], [private]
```

Function will check whether the given number is integer or not.

Parameters

<i>x</i>	Number to be checked.
----------	-----------------------

Returns

Integer representation of number if number was x.0, or -1 if not.

5.3.1.12 minusAction()

```
void GUI_Controller.minusAction (  
   (ActionEvent event ) [inline], [private]
```

Button '-' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.13 moduloAction()

```
void GUI_Controller.moduloAction (
    ActionEvent event ) [inline], [private]
```

Button " pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.14 multiAction()

```
void GUI_Controller.multiAction (
    ActionEvent event ) [inline], [private]
```

Button '*' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.15 nineAction()

```
void GUI_Controller.nineAction (
    ActionEvent event ) [inline], [private]
```

Button '9' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.16 oneAction()

```
void GUI_Controller.oneAction (
    ActionEvent event ) [inline], [private]
```

Button '1' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.17 plusAction()

```
void GUI_Controller.plusAction (  
    ActionEvent event ) [inline], [private]
```

Button '+' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.18 powAction()

```
void GUI_Controller.powAction (  
    ActionEvent event ) [inline], [private]
```

Button '^' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.19 sevenAction()

```
void GUI_Controller.sevenAction (  
    ActionEvent event ) [inline], [private]
```

Button '7' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.20 sixAction()

```
void GUI_Controller.sixAction (  
    ActionEvent event ) [inline], [private]
```

Button '6' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.21 sqrtAction()

```
void GUI_Controller.sqrtAction (
    ActionEvent event ) [inline], [private]
```

Button 'sqrt' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.22 threeAction()

```
void GUI_Controller.threeAction (
    ActionEvent event ) [inline], [private]
```

Button '3' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.23 twoAction()

```
void GUI_Controller.twoAction (
    ActionEvent event ) [inline], [private]
```

Button '2' pressed.

Parameters

<i>event</i>	todo
--------------	------

5.3.1.24 zeroAction()

```
void GUI_Controller.zeroAction (
    ActionEvent event ) [inline], [private]
```

Zero button pressed.

Parameters

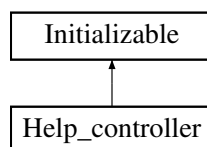
<i>event</i>	todo
--------------	------

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

- [src/GUI_Controller.java](#)

5.4 Help_controller Class Reference

Inheritance diagram for Help_controller:



Public Member Functions

- void **initialize** (URL url, ResourceBundle rb)

Private Member Functions

- void **closeAction** (ActionEvent event)
- void **creditsAction** (ActionEvent event)

Private Attributes

- Button **closeWindow**

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

- [src/Help_controller.java](#)

5.5 math Class Reference

Static Public Member Functions

- static double [add](#) (double x, double y)
Function will add two double numbers.
- static long [add](#) (long x, long y)
Function will add two integer numbers.
- static double [add](#) (double x, long y)
Function will add two numbers - one integer and one double.
- static double [add](#) (long x, double y)
Function will add two numbers - one integer and one double.
- static double [sub](#) (double x, double y)
Substraction of two numbers.
- static double [multiply](#) (double x, double y)

Function will multiply two double numbers.

- static long **multiply** (long x, long y)

Function will multiply two integer numbers.

- static double **multiply** (long x, double y)

Function will multiply two numbers - one double and one integer.

- static double **multiply** (double x, long y)

Function will multiply two numbers - one double and one integer.

- static double **divide** (double x, double y)

Division of two numbers.

- static long **factorial** (long num)

Function will calculate a factorial of a given number.

- static double **pow** (double x, int y)
- static int **pow** (int x, int y)
- static long **mod** (long x, long y)

Modulo function.

- static double **root** (double x, double root)

Sqrt function.

5.5.1 Member Function Documentation

5.5.1.1 **add()** [1/4]

```
static double math.add (
    double x,
    double y ) [inline], [static]
```

Function will add two double numbers.

Parameters

x	first number for operation
y	second number for operation

Returns

result of adding two parameters

5.5.1.2 **add()** [2/4]

```
static long math.add (
    long x,
    long y ) [inline], [static]
```

Function will add two integer numbers.

Parameters

<i>x</i>	first number for operation
<i>y</i>	second number for operation

Returns

result of adding two parameters

5.5.1.3 add() [3/4]

```
static double math.add (  
    double x,  
    long y ) [inline], [static]
```

Function will add two numbers - one integer and one double.

Parameters

<i>x</i>	first number for operation
<i>y</i>	second number for operation

Returns

result of adding two parameters

5.5.1.4 add() [4/4]

```
static double math.add (  
    long x,  
    double y ) [inline], [static]
```

Function will add two numbers - one integer and one double.

Parameters

<i>x</i>	first number for operation
<i>y</i>	second number for operation

Returns

result of adding two parameters

5.5.1.5 divide()

```
static double math.divide (  
    double x,  
    double y ) [inline], [static]
```

Division of two numbers.

Parameters

<i>x</i>	divident
<i>y</i>	divisor

Returns

result of dividing two numbers

5.5.1.6 factorial()

```
static long math.factorial (  
    long num ) [inline], [static]
```

Function will calculate a factorial of a given number.

Parameters

<i>num</i>	number from which we want factorial to be calculated from
------------	---

Returns

value of the factorial

5.5.1.7 mod()

```
static long math.mod (  
    long x,  
    long y ) [inline], [static]
```

Modulo function.

Parameters

<i>x</i>	divident
<i>y</i>	divisor

Returns

the remainder from division

5.5.1.8 multiply() [1/4]

```
static double math.multiply (  
    double x,  
    double y ) [inline], [static]
```

Function will multiply two double numbers.

Parameters

<i>x</i>	first number for operation
<i>y</i>	second number for operation

Returns

result of multiplying two numbers

5.5.1.9 multiply() [2/4]

```
static long math.multiply (  
    long x,  
    long y ) [inline], [static]
```

Function will multiply two integer numbers.

Parameters

<i>x</i>	first number for operation
<i>y</i>	second number for operation

Returns

result of multiplying two numbers

5.5.1.10 multiply() [3/4]

```
static double math.multiply (  
    long x,  
    double y ) [inline], [static]
```

Function will multiply two numbers - one double and one integer.

Parameters

<i>x</i>	first number for operation
<i>y</i>	second number for operation

Returns

result of multiplying two numbers

5.5.1.11 multiply() [4/4]

```
static double math.multiply (  
    double x,  
    long y ) [inline], [static]
```

Function will multiply two numbers - one double and one integer.

Parameters

<i>x</i>	first number for operation
<i>y</i>	second number for operation

Returns

result of multiplying two numbers

5.5.1.12 pow() [1/2]

```
static double math.pow (  
    double x,  
    int y ) [inline], [static]
```

Parameters

<i>x</i>	floating point base value
<i>y</i>	power value

Returns

result of raising x to the power y

5.5.1.13 pow() [2/2]

```
static int math.pow (  
    int x,  
    int y )  [inline], [static]
```

Parameters

<i>x</i>	floating point base value
<i>y</i>	power value

Returns

result of raising *x* to the power *y*

5.5.1.14 root()

```
static double math.root (  
    double x,  
    double root ) [inline], [static]
```

Sqrt function.

Parameters

<i>x</i>	number to be rooted
<i>root</i>	root of the number

Returns

result of operation

5.5.1.15 sub()

```
static double math.sub (  
    double x,  
    double y ) [inline], [static]
```

Substraction of two numbers.

Parameters

<i>x</i>	first number for operation
<i>y</i>	second number for operation

Returns

result of subtracting two numbers

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

- [src/math.java](#)

5.6 mathTest Class Reference

Public Member Functions

- void **setUp** () throws Exception
- void **tearDown** () throws Exception
- void [testAdd](#) ()
- void [testSub](#) ()
- void [testMultiply](#) ()
- void [testDivide](#) ()
- void [testFactorial](#) ()
- void [testPow](#) ()
- void [testMod](#) ()
- void [testAdd_double_double](#) ()
- void [testAdd_long_long](#) ()
- void [testAdd_double_long](#) ()
- void [testAdd_long_double](#) ()
- void [testMultiply_double_double](#) ()
- void [testMultiply_long_long](#) ()
- void [testMultiply_long_double](#) ()
- void [testMultiply_double_long](#) ()
- void [testPow_double_int](#) ()
- void [testPow_int_int](#) ()
- void [testRoot](#) ()

Static Public Member Functions

- static void **setUpClass** () throws Exception
- static void **tearDownClass** () throws Exception

5.6.1 Member Function Documentation

5.6.1.1 testAdd()

```
void mathTest.testAdd ( ) [inline]
```

Test of add method, of class math.

5.6.1.2 testAdd_double_double()

```
void mathTest.testAdd_double_double ( ) [inline]
```

Test of add method, of class math. Double, Double

5.6.1.3 testAdd_double_long()

```
void mathTest.testAdd_double_long ( ) [inline]
```

Test of add method, of class math. Double, Long

5.6.1.4 testAdd_long_double()

```
void mathTest.testAdd_long_double ( ) [inline]
```

Test of add method, of class math. Long, Double

5.6.1.5 testAdd_long_long()

```
void mathTest.testAdd_long_long ( ) [inline]
```

Test of add method, of class math. Long, Long

5.6.1.6 testDivide()

```
void mathTest.testDivide ( ) [inline]
```

Test of divide method, of class math.

5.6.1.7 testFactorial()

```
void mathTest.testFactorial ( ) [inline]
```

Test of factorial method, of class math.

5.6.1.8 testMod()

```
void mathTest.testMod ( ) [inline]
```

Test of mod method, of class math.

5.6.1.9 testMultiply()

```
void mathTest.testMultiply ( ) [inline]
```

Test of multiply method, of class math.

5.6.1.10 testMultiply_double_double()

```
void mathTest.testMultiply_double_double ( ) [inline]
```

Test of multiply method, of class math. Double, Double

5.6.1.11 testMultiply_double_long()

```
void mathTest.testMultiply_double_long ( ) [inline]
```

Test of multiply method, of class math. Double, Long

5.6.1.12 testMultiply_long_double()

```
void mathTest.testMultiply_long_double ( ) [inline]
```

Test of multiply method, of class math. Long, Double

5.6.1.13 testMultiply_long_long()

```
void mathTest.testMultiply_long_long ( ) [inline]
```

Test of multiply method, of class math. Long, Long

5.6.1.14 testPow()

```
void mathTest.testPow ( ) [inline]
```

Test of pow method, of class math.

5.6.1.15 testPow_double_int()

```
void mathTest.testPow_double_int ( ) [inline]
```

Test of pow method, of class math. Double, Int

5.6.1.16 testPow_int_int()

```
void mathTest.testPow_int_int ( ) [inline]
```

Test of pow method, of class math. Int, Int

5.6.1.17 testRoot()

```
void mathTest.testRoot ( ) [inline]
```

Test of root method, of class math.

5.6.1.18 testSub()

```
void mathTest.testSub ( ) [inline]
```

Test of sub method, of class math.

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

- test/[mathTest.java](#)

6 File Documentation

6.1 src/Calculator.java File Reference

Main class - [Calculator](#).

Classes

- class [Calculator](#)

6.1.1 Detailed Description

Main class - [Calculator](#).

Author

xcrkon00

6.2 src/GUI_Controller.java File Reference

Contains button actions.

Classes

- class [GUI_Controller](#)

6.2.1 Detailed Description

Contains button actions.

FXML Controller class

Author

xcrkon00

6.3 src/math.java File Reference

Class with mathematic operation used in calculator.

Classes

- class [math](#)

6.3.1 Detailed Description

Class with mathematic operation used in calculator.

Author

xblask04

6.4 test/mathTest.java File Reference

Tests for [math.java](#) library.

Classes

- class [mathTest](#)

6.4.1 Detailed Description

Tests for [math.java](#) library.

Author

xkosti05

Index

- add
 - math, [15](#), [16](#)
- CAction
 - GUI_Controller, [6](#)
- Calculator, [3](#)
 - main, [4](#)
- CreditsController, [4](#)
 - initialize, [5](#)
- DELAAction
 - GUI_Controller, [8](#)
- divAction
 - GUI_Controller, [8](#)
- divide
 - math, [16](#)
- dotAction
 - GUI_Controller, [8](#)
- eighthAction
 - GUI_Controller, [8](#)
- equalAction
 - GUI_Controller, [9](#)
- factAction
 - GUI_Controller, [9](#)
- factorial
 - math, [17](#)
- fiveAction
 - GUI_Controller, [9](#)
- fourAction
 - GUI_Controller, [9](#)
- GUI_Controller, [5](#)
 - CAction, [6](#)
 - DELAAction, [8](#)
 - divAction, [8](#)
 - dotAction, [8](#)
 - eighthAction, [8](#)
 - equalAction, [9](#)
 - factAction, [9](#)
 - fiveAction, [9](#)
 - fourAction, [9](#)
 - helpAction, [10](#)
 - is_int, [10](#)
 - minusAction, [10](#)
 - moduloAction, [10](#)
 - multiAction, [11](#)
 - nineAction, [11](#)
 - oneAction, [11](#)
 - plusAction, [12](#)
 - powAction, [12](#)
 - sevenAction, [12](#)
 - sixAction, [12](#)
 - sqrtAction, [13](#)
 - threeAction, [13](#)
 - twoAction, [13](#)
 - zeroAction, [13](#)
- Help_controller, [14](#)
- helpAction
 - GUI_Controller, [10](#)
- initialize
 - CreditsController, [5](#)
- is_int
 - GUI_Controller, [10](#)
- main
 - Calculator, [4](#)
- math, [14](#)
 - add, [15](#), [16](#)
 - divide, [16](#)
 - factorial, [17](#)
 - mod, [17](#)
 - multiply, [18](#), [19](#)
 - pow, [19](#)
 - root, [21](#)
 - sub, [21](#)
- mathTest, [22](#)
 - testAdd, [22](#)
 - testAdd_double_double, [22](#)
 - testAdd_double_long, [22](#)
 - testAdd_long_double, [22](#)
 - testAdd_long_long, [23](#)
 - testDivide, [23](#)
 - testFactorial, [23](#)
 - testMod, [23](#)
 - testMultiply, [23](#)
 - testMultiply_double_double, [23](#)
 - testMultiply_double_long, [23](#)
 - testMultiply_long_double, [23](#)
 - testMultiply_long_long, [23](#)
 - testPow, [24](#)
 - testPow_double_int, [24](#)
 - testPow_int_int, [24](#)
 - testRoot, [24](#)
 - testSub, [24](#)
- minusAction
 - GUI_Controller, [10](#)
- mod
 - math, [17](#)
- moduloAction
 - GUI_Controller, [10](#)
- multiAction
 - GUI_Controller, [11](#)
- multiply
 - math, [18](#), [19](#)
- nineAction
 - GUI_Controller, [11](#)
- oneAction

- GUI_Controller, 11
- plusAction
 - GUI_Controller, 12
- pow
 - math, 19
- powAction
 - GUI_Controller, 12
- root
 - math, 21
- sevenAction
 - GUI_Controller, 12
- sixAction
 - GUI_Controller, 12
- sqrtAction
 - GUI_Controller, 13
- src/Calculator.java, 24
- src/GUI_Controller.java, 25
- src/math.java, 25
- sub
 - math, 21
- test/mathTest.java, 25
- testAdd
 - mathTest, 22
- testAdd_double_double
 - mathTest, 22
- testAdd_double_long
 - mathTest, 22
- testAdd_long_double
 - mathTest, 22
- testAdd_long_long
 - mathTest, 23
- testDivide
 - mathTest, 23
- testFactorial
 - mathTest, 23
- testMod
 - mathTest, 23
- testMultiply
 - mathTest, 23
- testMultiply_double_double
 - mathTest, 23
- testMultiply_double_long
 - mathTest, 23
- testMultiply_long_double
 - mathTest, 23
- testMultiply_long_long
 - mathTest, 23
- testPow
 - mathTest, 24
- testPow_double_int
 - mathTest, 24
- testPow_int_int
 - mathTest, 24
- testRoot
 - mathTest, 24
- testSub
 - mathTest, 24
- threeAction
 - GUI_Controller, 13
- twoAction
 - GUI_Controller, 13
- zeroAction
 - GUI_Controller, 13