SquareEquation

Generated by Doxygen 1.9.2

Chapter 1

SquareEquation

This program solves quadratic equations

To use the program, you need to click on the Code button on github and copy the link

Then insert the following command into the console: git clone link Here link - the link that was copied earlier

Then you can open the project in code blocks and use the program

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

File Documentation

3.1 C:/Codes/SquareEquation/main.cpp File Reference

```
#include <cstdio>
#include <cassert>
#include <math.h>
#include "solveEquation.h"
#include "tests.h"
```

Macros

- #define TESTS

Functions

- void printRoots (const double x1, const double x2, const int roots_number)
- void launchProgram ()
- int main ()

3.1.1 Macro Definition Documentation

3.1.1.1 TESTS

```
#define TESTS
```

3.1.2 Function Documentation

3.1.2.1 launchProgram()

```
void launchProgram ( )
```

This function reads data and prints answer

Returns

The roots of the equation

3.1.2.2 main()

```
int main ( )
```

3.1.2.3 printRoots()

```
void printRoots (
            const double x1,
            const double x2,
            const int roots_number )
```

This function prints the roots of the equation (or their number)

Parameters

x_1	- the root of the equation
x_2	- the root of the equation
roots_number	- number of the roots

3.2 C:/Codes/SquareEquation/README.md File Reference

3.3 C:/Codes/SquareEquation/solveEquation.cpp File Reference

```
#include <cassert>
#include <math.h>
#include <stdio.h>
#include "solveEquation.h"
```

Functions

- int solveSquareEquation (double a, double b, double c, double ∗x1, double ∗x2)
- int solveLinearEquation (double a, double b, double ∗x)
- int isZero (double a)

Variables

- const double PRECISION = 1e-6

3.3.1 Function Documentation

3.3.1.1 isZero()

```
int isZero (
            double a )
```

This function checks whether the number is zero

Parameters

in	*a*	- the number that is compared with zero

Returns

if a is zero the program returns 1, else 0

3.3.1.2 solveLinearEquation()

```
int solveLinearEquation (
            double a,
            double b,
            double * x )
```

This function solves linear equation

Parameters

in	*a*	- coefficient before x
in	*b*	- the free term of the equation
out	*x*	- root of the equation. The program changes the value at the address

Returns

The number of the roots

3.3.1.3 solveSquareEquation()

```
int solveSquareEquation (
            double a,
            double b,
            double c,
            double * x1,
            double * x2 )
```

This program solves quadratic equation
The function returns the number of roots of the quadratic equation
And also finds the roots of the quadratic equation.

Parameters

in	*a*	- coefficient before x^2
in	*b*	- coefficient before x
in	*c*	- the free term of the equation
out	*x_1*	- root of the equation. The program changes the value at the address
out	*x_2*	- root of the equation. The program changes the value at the address

Returns

The number of the roots

3.3.2 Variable Documentation

3.3.2.1 PRECISION

```
const double PRECISION = 1e-6
```

3.4 C:/Codes/SquareEquation/solveEquation.h File Reference

Enumerations

- enum roots_number { infRoots = -1 , noRoots , oneRoot , twoRoots }

Functions

- int solveSquareEquation (double a, double b, double c, double *x1, double *x2)
- int solveLinearEquation (double a, double b, double *x)
- int isZero (double a)

3.4.1 Enumeration Type Documentation

3.4.1.1 roots_number

```
enum roots_number
```

Enumerator

infRoots	
noRoots	
oneRoot	
twoRoots	

3.4.2 Function Documentation

3.4.2.1 isZero()

```
int isZero (
            double a )
```

This function checks whether the number is zero

Parameters

in	*a*	- the number that is compared with zero

Returns

if a is zero the program returns 1, else 0

3.4.2.2 solveLinearEquation()

```
int solveLinearEquation (
             double a,
             double b,
             double * x )
```

This function solves linear equation

Parameters

in	*a*	- coefficient before x
in	*b*	- the free term of the equation
out	*x*	- root of the equation. The program changes the value at the address

Returns

The number of the roots

3.4.2.3 solveSquareEquation()

```
int solveSquareEquation (
            double a,
            double b,
            double c,
            double * x1,
            double * x2 )
```

This program solves quadratic equation
The function returns the number of roots of the quadratic equation
And also finds the roots of the quadratic equation.

Parameters

in	*a*	- coefficient before x^2
in	*b*	- coefficient before x
in	*c*	- the free term of the equation
out	*x_1*	- root of the equation. The program changes the value at the address
out	*x_2*	- root of the equation. The program changes the value at the address

Returns

The number of the roots

3.5 solveEquation.h

Go to the documentation of this file.

```
1 #ifndef SOLVEEQUATION_H_INCLUDED
2 #define SOLVEEQUATION_H_INCLUDED
3
4 enum roots_number {
5     infRoots = -1,
6     noRoots,
7     oneRoot,
8     twoRoots
9 };
10
23 int solveSquareEquation(double a, double b, double c, double *x1, double *x2);
24
32 int solveLinearEquation(double a, double b, double *x);
33
39 int isZero(double a);
40
41 #endif // SOLVEEQUATION_H_INCLUDED
```

3.6 C:/Codes/SquareEquation/tests.cpp File Reference

```
#include <cassert>
#include <cmath>
#include "tests.h"
#include "solveEquation.h"
#include <cstdio>
```

Functions

- void startAllTestsForSquareEquation ()
- void testSquareEquationAllCoeffsZero ()
- void testSquareEquationAllCoeffsZeroExceptC ()
- void testSquareEquationAllCoeffsZeroExceptB ()
- void testSquareEquationAisZeroTheLastestTwoCoeffsIsNotZero ()
- void testSquareEquationAllCoeffsZeroExceptA ()
- void testSquareEquationsTheSquareRootOfNegativeNumber ()
- void testSquareEquationsTheSquareRootOfPositiveNumber ()
- void testSquareEquationDiscriminantIsNegative ()
- void testSquareEquationDiscriminantIsZero ()
- void testSquareEquationDiscriminantIsPositive ()

3.6.1 Function Documentation

3.6.1.1 startAllTestsForSquareEquation()

```
void startAllTestsForSquareEquation ( )
```

This function launches other tests
These tests check whether the program gives the correct answers in particular cases If the program gives the correct answer nothing will happen Otherwise, there will be an error message

Returns

if the program passed all the tests, nothing will happen. Otherwise, an error message will be displayed

3.6.1.2 testSquareEquationAisZeroTheLastestTwoCoeffsIsNotZero()

```
void testSquareEquationAisZeroTheLastestTwoCoeffsIsNotZero ( )
```

3.6.1.3 testSquareEquationAllCoeffsZero()

```
void testSquareEquationAllCoeffsZero ( )
```

3.6.1.4 testSquareEquationAllCoeffsZeroExceptA()

```
void testSquareEquationAllCoeffsZeroExceptA ( )
```

3.6.1.5 testSquareEquationAllCoeffsZeroExceptB()

```
void testSquareEquationAllCoeffsZeroExceptB ( )
```

3.6.1.6 testSquareEquationAllCoeffsZeroExceptC()

```
void testSquareEquationAllCoeffsZeroExceptC ( )
```

3.6.1.7 testSquareEquationDiscriminantIsNegative()

```
void testSquareEquationDiscriminantIsNegative ( )
```

3.6.1.8 testSquareEquationDiscriminantIsPositive()

```
void testSquareEquationDiscriminantIsPositive ( )
```

3.6.1.9 testSquareEquationDiscriminantIsZero()

```
void testSquareEquationDiscriminantIsZero ( )
```

3.6.1.10 testSquareEquationsTheSquareRootOfNegativeNumber()

```
void testSquareEquationsTheSquareRootOfNegativeNumber ( )
```

3.6.1.11 testSquareEquationsTheSquareRootOfPositiveNumber()

```
void testSquareEquationsTheSquareRootOfPositiveNumber ( )
```

3.7 C:/Codes/SquareEquation/tests.h File Reference

Functions

- void startAllTestsForSquareEquation ()
- void testSquareEquationAllCoeffsZero ()
- void testSquareEquationAllCoeffsZeroExceptC ()
- void testSquareEquationAllCoeffsZeroExceptB ()
- void testSquareEquationAisZeroTheLastestTwoCoeffsIsNotZero ()
- void testSquareEquationAllCoeffsZeroExceptA ()
- void testSquareEquationsTheSquareRootOfNegativeNumber ()
- void testSquareEquationsTheSquareRootOfPositiveNumber ()
- void testSquareEquationDiscriminantIsNegative ()
- void testSquareEquationDiscriminantIsZero ()
- void testSquareEquationDiscriminantIsPositive ()

Variables

- double PRECISION

3.7.1 Function Documentation

3.7.1.1 startAllTestsForSquareEquation()

```
void startAllTestsForSquareEquation ( )
```

This function launches other tests
These tests check whether the program gives the correct answers in particular cases If the program gives the correct answer nothing will happen Otherwise, there will be an error message

Returns

if the program passed all the tests, nothing will happen. Otherwise, an error message will be displayed

3.7.1.2 testSquareEquationAisZeroTheLastestTwoCoeffsIsNotZero()

```
void testSquareEquationAisZeroTheLastestTwoCoeffsIsNotZero ( )
```

3.7.1.3 testSquareEquationAllCoeffsZero()

```
void testSquareEquationAllCoeffsZero ( )
```

3.7.1.4 testSquareEquationAllCoeffsZeroExceptA()

```
void testSquareEquationAllCoeffsZeroExceptA ( )
```

3.7.1.5 testSquareEquationAllCoeffsZeroExceptB()

```
void testSquareEquationAllCoeffsZeroExceptB ( )
```

3.7.1.6 testSquareEquationAllCoeffsZeroExceptC()

```
void testSquareEquationAllCoeffsZeroExceptC ( )
```

3.7.1.7 testSquareEquationDiscriminantIsNegative()

```
void testSquareEquationDiscriminantIsNegative ( )
```

3.7.1.8 testSquareEquationDiscriminantIsPositive()

```
void testSquareEquationDiscriminantIsPositive ( )
```

3.7.1.9 testSquareEquationDiscriminantIsZero()

```
void testSquareEquationDiscriminantIsZero ( )
```

3.7.1.10 testSquareEquationsTheSquareRootOfNegativeNumber()

```
void testSquareEquationsTheSquareRootOfNegativeNumber ( )
```

3.7.1.11 testSquareEquationsTheSquareRootOfPositiveNumber()

```
void testSquareEquationsTheSquareRootOfPositiveNumber ( )
```

3.7.2 Variable Documentation

3.7.2.1 PRECISION

```
double PRECISION  [extern]
```

3.8 tests.h

Go to the documentation of this file.

```
1 #ifndef TESTS_H_INCLUDED
2 #define TESTS_H_INCLUDED
3
4 extern double PRECISION;
5
15 void startAllTestsForSquareEquation();
16
17 void testSquareEquationAllCoeffsZero();
18 void testSquareEquationAllCoeffsZeroExceptC();
19 void testSquareEquationAllCoeffsZeroExceptB();
20 void testSquareEquationAisZeroTheLastestTwoCoeffsIsNotZero();
21 void testSquareEquationAllCoeffsZeroExceptA();
22 void testSquareEquationsTheSquareRootOfNegativeNumber();
23 void testSquareEquationsTheSquareRootOfPositiveNumber();
24 void testSquareEquationDiscriminantIsNegative();
25 void testSquareEquationDiscriminantIsZero();
26 void testSquareEquationDiscriminantIsPositive();
27
28 #endif // TESTS_H_INCLUDED
```

Index