

Assignment 1

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 | complex_adt.ComplexT Class Reference | 7 |
| 4.1.1 | Constructor & Destructor Documentation | 7 |
| 4.1.1.1 | __init__() | 7 |
| 4.2 | triangle_adt.TriangleT Class Reference | 8 |
| 4.2.1 | Constructor & Destructor Documentation | 8 |
| 4.2.1.1 | __init__() | 8 |
| 4.2.2 | Member Function Documentation | 9 |
| 4.2.2.1 | area() | 9 |
| 4.2.2.2 | equal() | 9 |
| 4.2.2.3 | get_sides() | 9 |
| 4.2.2.4 | is_valid() | 10 |
| 4.2.2.5 | perim() | 10 |
| 4.2.2.6 | tri_type() | 10 |
| 4.3 | triangle_adt.TriType Class Reference | 11 |
| 4.3.1 | Detailed Description | 11 |

| | |
|--|-----------|
| 5 File Documentation | 13 |
| 5.1 src/complex_adt.py File Reference | 13 |
| 5.1.1 Detailed Description | 14 |
| 5.1.2 Function Documentation | 14 |
| 5.1.2.1 add() | 14 |
| 5.1.2.2 conj() | 14 |
| 5.1.2.3 div() | 15 |
| 5.1.2.4 equal() | 16 |
| 5.1.2.5 get_phi() | 16 |
| 5.1.2.6 get_r() | 17 |
| 5.1.2.7 imag() | 17 |
| 5.1.2.8 mult() | 17 |
| 5.1.2.9 real() | 18 |
| 5.1.2.10 recip() | 18 |
| 5.1.2.11 sqrt() | 18 |
| 5.1.2.12 sub() | 19 |
| 5.2 src/triangle_adt.py File Reference | 19 |
| 5.2.1 Detailed Description | 19 |
| Index | 21 |

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

| | |
|----------------------------------|----|
| complex_adt.ComplexT | 7 |
| triangle_adt.TriangleT | 8 |
| Enum | |
| triangle_adt.TriType | 11 |

Chapter 2

Class Index

2.1 Class List

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

| | |
|--|----|
| complex_adt.ComplexT | 7 |
| triangle_adt.TriangleT | 8 |
| triangle_adt.TriType An enumerated data type that represents different types of triangles | 11 |

Chapter 3

File Index

3.1 File List

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

| | |
|---|--------------------|
| src/ complex_adt.py | |
| Provides methods for performing calculations on complex numbers | 13 |
| src/ triangle_adt.py | |
| Provides methods for performing calculations on complex numbers | 19 |

Chapter 4

Class Documentation

4.1 complex_adt.ComplexT Class Reference

Public Member Functions

- `def __init__`
[ComplexT](#) constructor.

Public Attributes

- `x`
- `y`

4.1.1 Constructor & Destructor Documentation

4.1.1.1 `__init__()`

```
def complex_adt.ComplexT.__init__ (
    self,
    x )
```

[ComplexT](#) constructor.

Initializes a [ComplexT](#) object with real part and an imaginary part. It also rounds the value of both real and imaginary parts upto five decimal points. Assumption is made that the both inputs are of floating point type.

Parameters

| | |
|----------------|--|
| <code>x</code> | The real part of the ComplexT object. |
| <code>y</code> | The imaginary part of the ComplexT object. |

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

- [src/complex_adt.py](#)

4.2 triangle_adt.TriangleT Class Reference

Public Member Functions

- `def __init__`
TriangleT constructor.
- `def get_sides (self)`
Gets all the lenght of sides of a triangle.
- `def equal (self, t)`
Checks if two TriangleT objects are equal or not.
- `def perim (self)`
Calculates the perimeter of a TriangleT object by adding all the lenghts of sides.
- `def area (self)`
Calculates the area of a TriangleT object by using herons formula.
- `def is_valid (self)`
Checks if TriangleT object is valid.
- `def tri_type (self)`
Checks what type of Triangle the TriangleT object is.

Public Attributes

- **s**

4.2.1 Constructor & Destructor Documentation

4.2.1.1 __init__()

```
def triangle_adt.TriangleT.__init__ (
    self,
    a )
```

[TriangleT](#) constructor.

Initializes a [TriangleT](#) object by taking three sides as input. Assumption is made that all the inputs are of integer types.

Parameters

| | |
|----------|--|
| <i>a</i> | The length of first side of triangle. |
| <i>b</i> | The length of second side of triangle. |
| <i>c</i> | The length of third side of triangle. |

4.2.2 Member Function Documentation

4.2.2.1 area()

```
def triangle_adt.TriangleT.area (
    self )
```

Calculates the area of a [TriangleT](#) object by using herons formula.

Exceptions

| | |
|-------------------------|---|
| <i>TriangleNotValid</i> | Throws TriangleNotValid if TriangleT object is not valid. |
|-------------------------|---|

Returns

Area of [TriangleT](#) object.

4.2.2.2 equal()

```
def triangle_adt.TriangleT.equal (
    self,
    t )
```

Checks if two [TriangleT](#) objects are equal or not.

param t [TriangleT](#) object evaluated for equality.

Exceptions

| | |
|-------------------------|---|
| <i>TriangleNotValid</i> | Throws TriangleNotValid if TriangleT object is not valid. |
|-------------------------|---|

Returns

Returns true if the [TriangleT](#) object is equal with the given [TriangleT](#) object; false if not

4.2.2.3 get_sides()

```
def triangle_adt.TriangleT.get_sides (
    self )
```

Gets all the lenght of sides of a triangle.

Returns

A tuple of lengths of sides in a sorted order.

4.2.2.4 is_valid()

```
def triangle_adt.TriangleT.is_valid (  
    self )
```

Checks if [TriangleT](#) object is valid.

Checks validity of [TriangleT](#) by checking if the sum of any two sides would be less or equal to the third side.

Returns

Returns true if the [TriangleT](#) object is valid; false if not.

4.2.2.5 perim()

```
def triangle_adt.TriangleT.perim (  
    self )
```

Calculates the perimeter of a [TriangleT](#) object by adding all the lengths of sides.

Exceptions

| | |
|-------------------------|---|
| <i>TriangleNotValid</i> | Throws TriangleNotValid if TriangleT object is not valid. |
|-------------------------|---|

Returns

Sum of all lengths of sides.

4.2.2.6 tri_type()

```
def triangle_adt.TriangleT.tri_type (  
    self )
```

Checks what type of Triangle the [TriangleT](#) object is.

Exceptions

| | |
|-------------------------|---|
| <i>TriangleNotValid</i> | Throws TriangleNotValid if TriangleT object is not valid. |
|-------------------------|---|

The triangle type is checked by applying different formulas for which Right, Scalene, Isosceles and Equilateral will hold true. Right Angle Triangle type objects have the highest precedence.

Returns

The type of [TriangleT](#) object.

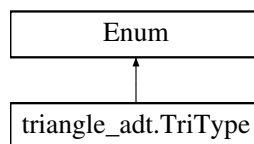
The documentation for this class was generated from the following file:

- [src/triangle_adt.py](#)

4.3 triangle_adt.TriType Class Reference

An enumerated data type that represents different types of triangles.

Inheritance diagram for triangle_adt.TriType:



Static Public Attributes

- `int equilat = 1`
- `int isosceles = 2`
- `int scalene = 3`
- `int right = 4`

4.3.1 Detailed Description

An enumerated data type that represents different types of triangles.

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

- [src/triangle_adt.py](#)

Chapter 5

File Documentation

5.1 src/complex_adt.py File Reference

Provides methods for performing calculations on complex numbers.

Classes

- class `complex_adt.ComplexT`

Functions

- def `complex_adt.real` (self)
Gets the real part of the `ComplexT` object.
- def `complex_adt.imag` (self)
imag Gets the imaginary part of the `ComplexT` object.
- def `complex_adt.get_r` (self)
get_r Calculates the modulus of the `ComplexT` object.
- def `complex_adt.get_phi` (self)
Calculates the phase(argument) between the `ComplexT` object.
- def `complex_adt.equal` (self, c)
Check if the two complex numbers are equal or not.
- def `complex_adt.conj` (self)
Does the negation of the imaginary part.
- def `complex_adt.add` (self, a)
Adds the two `ComplexT` objects.
- def `complex_adt.sub` (self, a)
Subtracts the two `ComplexT` objects.
- def `complex_adt.mult` (self, a)
Multiplies two `ComplexT` objects.
- def `complex_adt.recip` (self)
Divides one by the `ComplexT` object.
- def `complex_adt.div` (self, a)
Divides the two `ComplexT` objects.
- def `complex_adt.sqrt` (self)
Finds the square root of the `ComplexT` object.

5.1.1 Detailed Description

Provides methods for performing calculations on complex numbers.

Author

Harkanwar Singh Waraich

Date

01/14/2021

5.1.2 Function Documentation

5.1.2.1 add()

```
def complex_adt.add (
    self,
    a )
```

Adds the two ComplexT objects.

Parameters

| | |
|----------|------------------------------|
| <i>a</i> | ComplexT object to be added. |
|----------|------------------------------|

Returns

Addition of two ComplexT objects.

5.1.2.2 conj()

```
def complex_adt.conj (
    self )
```

Does the negation of the imaginary part.

Returns

The same object but with the negation of the imaginary part

5.1.2.3 div()

```
def complex_adt.div (
    self,
    a )
```

Divides the two ComplexT objects.

Exceptions

| | |
|--------------------------|--|
| <i>ZeroDivisionError</i> | Throws ZeroDivisionError if in a ComplexT object both imaginary and real parts are equal to zero it is not possible to do a division of it with ComplexT object> . |
|--------------------------|--|

This function uses complex() python function which takes real and imaginary parts of ComplexT object. After the argument is divided by class variable we can retrieve imaginary and real parts complex number which can be passed to ComplexT object.

Parameters

| | |
|----------|---|
| <i>a</i> | ComplexT object by which the the division will occur. |
|----------|---|

Returns

ComplecT object after the division.

5.1.2.4 equal()

```
def complex_adt.equal (
    self,
    c )
```

Check if the two complex numbers are equal or not.

Parameters

| | |
|----------|---|
| <i>c</i> | ComplexT object to be evaluated for equality. |
|----------|---|

Returns

Returns true if the ComplexT object is equal with the given ComplexT object; false if not

5.1.2.5 get_phi()

```
def complex_adt.get_phi (
    self )
```

Calculates the phase(argument) between the ComplexT object.

Returns

The phase(argument) in radians using phase function from cmath library

5.1.2.6 get_r()

```
def complex_adt.get_r (
    self )
```

get_r Calculates the modulus of the ComplexT object.

Returns

The modulus of the the ComplexT object.

5.1.2.7 imag()

```
def complex_adt.imag (
    self )
```

imag Gets the imaginary part of the ComplexT object.

Returns

The imaginary part of the ComplexT object.

5.1.2.8 mult()

```
def complex_adt.mult (
    self,
    a )
```

Multiplies two ComplexT objects.

This function uses complex() python function which takes real and imaginary parts of ComplexT object and the input. It then multiplies and from its result we can retrieve imaginary and real parts complex number which can be passed to ComplexT object.

Parameters

| | |
|----------|----------------------------------|
| <i>a</i> | ComplexT object to be Multiplied |
|----------|----------------------------------|

Returns

A ComplexT object after two ComplexT objects are multiplied.

5.1.2.9 real()

```
def complex_adt.real (
    self )
```

Gets the real part of the ComplexT object.

Returns

The real part of the ComplexT object.

5.1.2.10 recip()

```
def complex_adt.recip (
    self )
```

Divides one by the ComplexT object.

Exceptions

| | |
|--------------------------|---|
| <i>ZeroDivisionError</i> | Throws ZeroDivisionError if in a ComplexT object both imaginary and real parts are equal to zero it is not possible to do a reciprocal of it. |
|--------------------------|---|

This function uses complex() python function which takes real and imaginary parts of ComplexT object. After the one is divided by class variable we can retrieve imaginary and real parts complex number which can be passed to ComplexT object.

Returns

Resiprocal of the ComplexT object.

5.1.2.11 sqrt()

```
def complex_adt.sqrt (
    self )
```

Finds the square root of the ComplexT object.

This function uses complex() python function which takes real and imaginary parts of ComplexT object. After the square root of complex number we can retrieve imaginary and real parts complex number which can be passed to ComplexT object.

Returns

Square root of the ComplexT object.

5.1.2.12 sub()

```
def complex_adt.sub (
    self,
    a )
```

Subtracts the two ComplexT objects.

Parameters

| | |
|----------|-----------------------------------|
| <i>a</i> | ComplexT object to be subtracted. |
|----------|-----------------------------------|

Returns

A ComplexT object after two ComplexT objects are subtracted.

5.2 src/triangle_adt.py File Reference

Provides methods for performing calculations on complex numbers.

Classes

- class [triangle_adt.TriType](#)
An enumerated data type that represents different types of triangles.
- class [triangle_adt.TriangleT](#)

5.2.1 Detailed Description

Provides methods for performing calculations on complex numbers.

Author

Harkanwar Singh Waraich

Date

01/16/2021

Index

- `__init__`
 - `complex_adt::ComplexT`, 7
 - `triangle_adt::TriangleT`, 8
- `add`
 - `complex_adt.py`, 14
- `area`
 - `triangle_adt::TriangleT`, 9
- `complex_adt.ComplexT`, 7
- `complex_adt.py`
 - `add`, 14
 - `conj`, 14
 - `div`, 14
 - `equal`, 16
 - `get_phi`, 16
 - `get_r`, 16
 - `imag`, 17
 - `mult`, 17
 - `real`, 17
 - `recip`, 18
 - `sqrt`, 18
 - `sub`, 18
- `complex_adt::ComplexT`
 - `__init__`, 7
- `conj`
 - `complex_adt.py`, 14
- `div`
 - `complex_adt.py`, 14
- `equal`
 - `complex_adt.py`, 16
 - `triangle_adt::TriangleT`, 9
- `get_phi`
 - `complex_adt.py`, 16
- `get_r`
 - `complex_adt.py`, 16
- `get_sides`
 - `triangle_adt::TriangleT`, 9
- `imag`
 - `complex_adt.py`, 17
- `is_valid`
 - `triangle_adt::TriangleT`, 10
- `mult`
 - `complex_adt.py`, 17
- `perim`
 - `triangle_adt::TriangleT`, 10
- `real`
 - `complex_adt.py`, 17
- `recip`
 - `complex_adt.py`, 18
- `sqrt`
 - `complex_adt.py`, 18
- `src/complex_adt.py`, 13
- `src/triangle_adt.py`, 19
- `sub`
 - `complex_adt.py`, 18
- `tri_type`
 - `triangle_adt::TriangleT`, 10
- `triangle_adt.TriType`, 11
- `triangle_adt.TriangleT`, 8
- `triangle_adt::TriangleT`
 - `__init__`, 8
 - `area`, 9
 - `equal`, 9
 - `get_sides`, 9
 - `is_valid`, 10
 - `perim`, 10
 - `tri_type`, 10