

Многомодульный проект

Создано системой Doxygen 1.8.9.1

Вс 14 Фев 2016 13:09:52

Оглавление

1	Иерархический список классов	1
1.1	Иерархия классов	1
2	Алфавитный указатель классов	3
2.1	Классы	3
3	Классы	5
3.1	Класс Array	5
3.2	Класс ArrayApp	5
3.3	Класс ArrayAppDemo	5
3.4	Класс CircleGame	6
3.4.1	Подробное описание	6
3.4.2	Конструктор(ы)	6
3.4.2.1	CircleGame	6
3.4.3	Методы	6
3.4.3.1	determineWinner	6
3.4.3.2	setKicking	7
3.4.3.3	setPlayers	8
3.5	Класс Meter	8
3.5.1	Подробное описание	8
3.5.2	Методы	8
3.5.2.1	fromInches	8
3.5.2.2	getCm	9
3.5.2.3	getInches	9
3.5.2.4	getM	9
3.5.2.5	getMm	9
3.6	Структура Meters	9
3.7	Класс Number	10
3.7.1	Подробное описание	10
3.7.2	Конструктор(ы)	10
3.7.2.1	Number	10
3.7.2.2	Number	10

3.7.3	Методы	10
3.7.3.1	getNum	10
3.7.3.2	isPalindrome	11
3.8	Класс OutOfRange	11
3.9	Класс Phrases	11
3.9.1	Подробное описание	11
3.9.2	Конструктор(ы)	11
3.9.2.1	Phrases	11
3.9.3	Методы	12
3.9.3.1	getStr	12
3.9.3.2	upper_case_phrases	12
3.10	Класс Route	12
3.10.1	Подробное описание	12
3.10.2	Конструктор(ы)	12
3.10.2.1	Route	12
3.10.2.2	Route	13
3.10.2.3	Route	13
3.10.3	Методы	13
3.10.3.1	distance	13
3.10.3.2	timeOfHalfWay	13
Алфавитный указатель		15

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

Array	5
ArrayApp	5
ArrayAppDemo	5
CircleGame	6
exception	
OutOfRangeException	11
Meter	8
Meters	9
Number	10
Phrases	11
Route	12

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Array	5
ArrayApp	5
ArrayAppDemo	5
CircleGame	
Круговая игра	6
Meter	
Представление числа в дюймовой и метрической системе	8
Meters	9
Number	
Число-палиндром	10
OutOfRange	11
Phrases	
Представление фраз в тексте	11
Route	
Расчет времени, затраченного на первую половину пути	12

Глава 3

Классы

3.1 Класс Array

Открытые члены

- Array (int size=10)
- Array (Array &)
- int getSize () const
- void setItem (const int, const int)
- Array & operator= (Array &)
- int operator[] (int) const

Объявления и описания членов классов находятся в файлах:

- cpplib/array.h
- cpplib/array.cpp

3.2 Класс ArrayApp

Открытые члены

- ArrayApp (int size=10)
- void setSize (int)

Друзья

- istream & operator>> (istream &, ArrayApp &)
- ostream & operator<< (ostream &, ArrayApp &)

Объявления и описания членов классов находятся в файлах:

- cpplib/arrayapp.h
- cpplib/arrayapp.cpp

3.3 Класс ArrayAppDemo

Открытые члены

- void demo () const

Объявления и описания членов классов находятся в файлах:

- `srrapp/arrayappdemo.h`
- `srrapp/arrayappdemo.cpp`

3.4 Класс CircleGame

Круговая игра

```
#include <circlegame.h>
```

Открытые члены

- `CircleGame` (`const int num_of_players`, `const int num_for_kick`)
Конструктор
- `void setPlayers` (`const int num_of_players`)
Установить значение поля `players`, равное количеству игроков
- `void setKicking` (`const int num_for_kick`)
Установить значение поля `kicking`, равное номеру человека, которого выводят
- `int determineWinner` () `const`
Определить победителя

3.4.1 Подробное описание

Круговая игра

По кругу располагаются `num_of_players` человек. Ведущий считает по кругу, начиная с первого, и выводит `num_for_kick`-го человека. Круг смыкается, счет возобновляется со следующего. Так продолжается, пока в круге не останется только один человек. Найти номер этого человека.

3.4.2 Конструктор(ы)

3.4.2.1 `CircleGame::CircleGame (const int num_of_players, const int num_for_kick)`

Конструктор

Аргументы

<code>num_of_players</code>	количество игроков, участвующих в игре
<code>num_for_kick</code>	номер человека, которого выводят из игры

3.4.3 Методы

3.4.3.1 `int CircleGame::determineWinner () const`

Определить победителя

Данная функция высчитывает номер игрока-победителя. Ожидается что значения `players` и `kicking` известны.

Возвращает

Номер победившего игрока

3.4.3.2 void CircleGame::setKicking (const int num_for_kick)

Установить значение поля kicking, равное номеру человека, которого выводят

Аргументы

num_for_kick	новый номер человека, которого выводят из игры
--------------	--

3.4.3.3 void CircleGame::setPlayers (const int num_of_players)

Установить значение поля players, равное количеству игроков

Аргументы

num_of_players ←	новое количество игроков, участвующих в игре
------------------	--

Объявления и описания членов классов находятся в файлах:

- cpplib/circlegame.h
- cpplib/circlegame.cpp

3.5 Класс Meter

Представление числа в дюймовой и метрической системе

```
#include <meter.h>
```

Открытые члены

- [Meter](#) ()
Конструктор
- void [fromInches](#) (const int inches)
Перевести из дюймов в метрическую систему
- int [getInches](#) () const
Получить число дюймов
- int [getM](#) () const
Получить количество метров
- int [getCm](#) () const
Получить количество сантиметров
- double [getMm](#) () const
Получить количество миллиметров

3.5.1 Подробное описание

Представление числа в дюймовой и метрической системе

Перевести длину отрезка из дюймов в метры, сантиметры и миллиметры.

3.5.2 Методы

3.5.2.1 void Meter::fromInches (const int inches)

Перевести из дюймов в метрическую систему

Аргументы

inches	число дюймов
--------	--------------

3.5.2.2 `int Meter::getCm () const` [inline]

Получить количество сантиметров

Возвращает

количество сантиметров

3.5.2.3 `int Meter::getInches () const` [inline]

Получить число дюймов

Возвращает

число дюймов

3.5.2.4 `int Meter::getM () const` [inline]

Получить количество метров

Возвращает

количество метров

3.5.2.5 `double Meter::getMm () const` [inline]

Получить количество миллиметров

Возвращает

количество миллиметров

Объявления и описания членов классов находятся в файлах:

- `cpplib/meter.h`
- `cpplib/meter.cpp`

3.6 Структура Meters

Открытые атрибуты

- `int m`
- `int cm`
- `double mm`

Объявления и описания членов структуры находятся в файле:

- `library/inch_to_cm.h`

3.7 Класс Number

Число-палиндром

```
#include <number.h>
```

Открытые члены

- `Number` (const int int_num)
Конструктор, получающий число
- `Number` (const string str_num)
Конструктор, получающий строку
- bool `isPalindrome` () const
Метод, определяющий, является ли число палиндромом
- string `getNum` () const
Метод, запрашивающий число в строковом формате

3.7.1 Подробное описание

Число-палиндром

Проверить, является ли заданное число NUMBER палиндромом (симметричным, первая цифра равна последней и так далее).

3.7.2 Конструктор(ы)

3.7.2.1 Number::Number (const int int_num)

Конструктор, получающий число

Аргументы

int_num	число в целом формате, которое нужно проверить на палиндром
---------	---

3.7.2.2 Number::Number (const string str_num)

Конструктор, получающий строку

Аргументы

str_num	число в строковом формате, которое нужно проверить на палиндром
---------	---

3.7.3 Методы

3.7.3.1 string Number::getNum () const [inline]

Метод, запрашивающий число в строковом формате

Возвращает

число в строковом формате

3.7.3.2 bool Number::isPalindrome () const

Метод, определяющий, является ли число палиндромом

Ожидается, что число задано

Возвращает

true - если число является палиндромом, false - если число таковым не является

Объявления и описания членов классов находятся в файлах:

- cpplib/number.h
- cpplib/number.cpp

3.8 Класс OutOfRange

Граф наследования: OutOfRange:

3.9 Класс Phrases

Представление фраз в тексте

```
#include <phrases.h>
```

Открытые члены

- [Phrases](#) (const string text)
Конструктор
- void [upper_case_phrases](#) ()
Метод, исправляющий в тексте фразы
- string [getStr](#) () const
Метод, запрашивающий текст

3.9.1 Подробное описание

Представление фраз в тексте

Проверить, что все фразы в тексте начинаются с прописной буквы и при необходимости откорректировать текст.

3.9.2 Конструктор(ы)

3.9.2.1 Phrases::Phrases (const string text)

Конструктор

Аргументы

text	текст, в котором необходимо проверить, начинаются ли фразы с прописной буквы
------	--

3.9.3 Методы

3.9.3.1 `string Phrases::getStr () const [inline]`

Метод, запрашивающий текст

Возвращает

текст в формате `string`

3.9.3.2 `void Phrases::upper_case_phrases ()`

Метод, исправляющий в тексте фразы

Ожидается, что текст задан

Объявления и описания членов классов находятся в файлах:

- `cpplib/phrases.h`
- `cpplib/phrases.cpp`

3.10 Класс Route

Расчет времени, затраченного на первую половину пути

```
#include <route.h>
```

Открытые члены

- [Route](#) (`const double velocity[], const double time[]`)
Конструктор, получающий массивы
- [Route](#) (`const vector< double > &velocity, const vector< double > &time`)
Конструктор, получающий `vector`.
- [Route](#) (`const double v1, const double t1, const double v2, const double t2, const double v3, const double t3`)
Конструктор, последовательно получающий скорости и время движения
- `double distance () const`
Рассчитать расстояние движения
- `double timeOfHalfWay () const`
Рассчитать время, затраченное на прохождение половины пути

3.10.1 Подробное описание

Расчет времени, затраченного на первую половину пути

Определить, за какое время путник одолел первую половину пути, двигаясь T1 часов со скоростью V1, T2 часов со скоростью V2, T3 часов со скоростью V3.

3.10.2 Конструктор(ы)

3.10.2.1 `Route::Route (const double velocity[], const double time[])`

Конструктор, получающий массивы

Аргументы

velocity[]	массив, содержащий скорости на последовательных участках
time[]	массив, содержащий время движения на последовательных участках

3.10.2.2 Route::Route (const vector< double > & velocity, const vector< double > & time)

Конструктор, получающий vector.

Аргументы

velocity	вектор, содержащий скорости на последовательных участках
time	вектор, содержащий время движения на последовательных участках

3.10.2.3 Route::Route (const double v1, const double t1, const double v2, const double t2, const double v3, const double t3)

Конструктор, последовательно получающий скорости и время движения

Аргументы

v1	скорость на первом участке
t1	время движения на первом участке
v2	скорость на втором участке
t2	время движения на втором участке
v3	скорость на третьем участке
t3	время движения на третьем участке

3.10.3 Методы

3.10.3.1 double Route::distance () const

Рассчитать расстояние движения

Возвращает

расстояние движения

3.10.3.2 double Route::timeOfHalfWay () const

Рассчитать время, затраченное на прохождение половины пути

Возвращает

время, затраченное на половину пути

Объявления и описания членов классов находятся в файлах:

- cpplib/route.h
- cpplib/route.cpp

Предметный указатель

- Array, [5](#)
- ArrayApp, [5](#)
- ArrayAppDemo, [5](#)
- CircleGame, [6](#)
 - CircleGame, [6](#)
 - determineWinner, [6](#)
 - setKicking, [6](#)
 - setPlayers, [8](#)
- determineWinner
 - CircleGame, [6](#)
- distance
 - Route, [13](#)
- fromInches
 - Meter, [8](#)
- getCm
 - Meter, [9](#)
- getInches
 - Meter, [9](#)
- getM
 - Meter, [9](#)
- getMm
 - Meter, [9](#)
- getNum
 - Number, [10](#)
- getStr
 - Phrases, [12](#)
- isPalindrome
 - Number, [10](#)
- Meter, [8](#)
 - fromInches, [8](#)
 - getCm, [9](#)
 - getInches, [9](#)
 - getM, [9](#)
 - getMm, [9](#)
- Meters, [9](#)
- Number, [10](#)
 - getNum, [10](#)
 - isPalindrome, [10](#)
 - Number, [10](#)
- OutOfRange, [11](#)
- Phrases, [11](#)
 - getStr, [12](#)
 - Phrases, [11](#)
 - upper_case_phrases, [12](#)
- Route, [12](#)
 - distance, [13](#)
 - Route, [12](#), [13](#)
 - timeOfHalfWay, [13](#)
- setKicking
 - CircleGame, [6](#)
- setPlayers
 - CircleGame, [8](#)
- timeOfHalfWay
 - Route, [13](#)
- upper_case_phrases
 - Phrases, [12](#)