

Testes Trunfo

JOSE TARCISIO PRATA JUNIOR

PIETRO URCINE DE ABREU

Versão 1.0

Quarta, 21 de Dezembro de 2022

Sumário

Table of contents

Índice Hierárquico

Hierarquia de Classes

Esta lista de hierarquias está parcialmente ordenada (ordem alfabética):

CPPUNIT_NS::TestFixture	
newtestclass.....	5
CPPUNIT_NS::TestListener	
ProgressListener.....	6

Índice dos Componentes

Lista de Classes

Aqui estão as classes, estruturas, uniões e interfaces e suas respectivas descrições:

newtestclass	5
ProgressListener	6

Índice dos Arquivos

Lista de Arquivos

Esta é a lista de todos os arquivos e suas respectivas descrições:

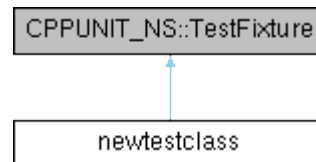
newtestclass.cpp	7
newtestclass.h	8
newtestrunner.cpp	10

Classes

Referência da Classe newtestclass

```
#include <newtestclass.h>
```

Diagrama de hierarquia para newtestclass:



Membros Públicos

- **newtestclass ()**
- **virtual ~newtestclass ()**
- **void setUp ()**
- **void tearDown ()**

Construtores e Destrutores

newtestclass::newtestclass ()

```
15         {  
16     }
```

newtestclass::~~newtestclass () [virtual]

```
18         {  
19     }
```

Funções membros

void newtestclass::setUp ()

```
21         {  
22     }
```

void newtestclass::tearDown ()

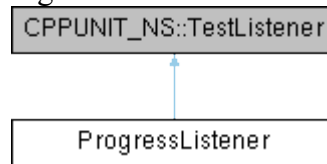
```
24         {  
25     }
```

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- **newtestclass.h**
- **newtestclass.cpp**

Referência da Classe ProgressListener

Diagrama de hierarquia para ProgressListener:



Membros Públicos

- `ProgressListener ()`
- `~ProgressListener ()`
- `void startTest (CPPUNIT_NS::Test *test)`
- `void addFailure (const CPPUNIT_NS::TestFailure &failure)`
- `void endTest (CPPUNIT_NS::Test *test)`

Construtores e Destrutores

`ProgressListener::ProgressListener () [inline]`

```
22 : m_lastTestFailed(false) {  
23 }
```

`ProgressListener::~~ProgressListener () [inline]`

```
25 {  
26 }
```

Funções membros

`void ProgressListener::addFailure (const CPPUNIT_NS::TestFailure & failure) [inline]`

```
36 {  
37     CPPUNIT_NS::stdCout() << " : " << (failure.isError() ? "error" :  
38     "assertion");  
39     m_lastTestFailed = true;  
39 }
```

`void ProgressListener::endTest (CPPUNIT_NS::Test * test) [inline]`

```
41 {  
42     if (!m_lastTestFailed)  
43         CPPUNIT_NS::stdCout() << " : OK";  
44     CPPUNIT_NS::stdCout() << "\n";  
45 }
```

`void ProgressListener::startTest (CPPUNIT_NS::Test * test) [inline]`

```
28 {  
29     CPPUNIT_NS::stdCout() << test->getName();  
30     CPPUNIT_NS::stdCout() << "\n";  
31     CPPUNIT_NS::stdCout().flush();  
32     m_lastTestFailed = false;  
33 }  
34 }
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- `newtestrunner.cpp`

Arquivos

Referência do Arquivo newtestclass.cpp

```
#include "newtestclass.h"  
#include "../Carta.h"  
#include "../Carro.h"  
#include "../TipoCarta.h"  
#include "../Baralho.h"  
#include <limits>  
#include <math.h>
```

Funções

- CPPUNIT_TEST_SUITE_REGISTRATION (newtestclass)
-

Funções

CPPUNIT_TEST_SUITE_REGISTRATION (newtestclass)

Referência do Arquivo newtestclass.h

```
#include <cppunit/extensions/HelperMacros.h>
```

Componentes

- class `newtestclass`

newtestclass.h

Vá para a documentação desse arquivo.1

```
5 #ifndef NEWTESTCLASS_H
6 #define NEWTESTCLASS_H
7
8 #include <cppunit/extensions/HelperMacros.h>
9
10 class newtestclass : public CPPUNIT_NS::TestFixture {
11     CPPUNIT_TEST_SUITE(newtestclass);
12     CPPUNIT_TEST(testCarta);
13     CPPUNIT_TEST(testCarro);
14     CPPUNIT_TEST(testBaralho);
15     CPPUNIT_TEST_SUITE_END();
16 public:
17     newtestclass();
18     virtual ~newtestclass();
19     void setUp();
20     void tearDown();
21 private:
22     void testCarta();
23     void testCarro();
24     void testBaralho();
25 };
26
27 #endif /* NEWTESTCLASS_H */
```

Referência do Arquivo newtestrunner.cpp

```
#include <cppunit/BriefTestProgressListener.h>
#include <cppunit/CompilerOutputter.h>
#include <cppunit/extensions/TestFactoryRegistry.h>
#include <cppunit/TestResult.h>
#include <cppunit/TestResultCollector.h>
#include <cppunit/TestRunner.h>
#include <cppunit/Test.h>
#include <cppunit/TestFailure.h>
#include <cppunit/portability/Stream.h>
```

Componentes

- class **ProgressListener**

Funções

- int **main ()**

Funções

int main ()

```
58     {
59         // Create the event manager and test controller
60         CPPUNIT_NS::TestResult controller;
61
62         // Add a listener that collects test result
63         CPPUNIT_NS::TestResultCollector result;
64         controller.addListener(&result);
65
66         // Add a listener that print dots as test run.
67         ProgressListener progress;
68         controller.addListener(&progress);
69
70         // Add the top suite to the test runner
71         CPPUNIT_NS::TestRunner runner;
72         runner.addTest(CPPUNIT_NS::TestFactoryRegistry::getRegistry().makeTest());
73         runner.run(controller);
74
75         // Print test in a compiler compatible format.
76         CPPUNIT_NS::CompilerOutputter outputter(&result, CPPUNIT_NS::stdCOut());
77         outputter.write();
78
79         return result.wasSuccessful() ? 0 : 1;
80     }
```

Sumário

INDEX