

Виртуальное ромбовидное наследование

Программная документация

1 Тест виртуального ромбовидного наследования	1
1.1 Test virtual (diamond of death) inheritance	1
2 Иерархический список классов	3
2.1 Иерархия классов	3
3 Алфавитный указатель классов	5
3.1 Классы	5
4 Классы	7
4.1 Шаблон класса A< SizeX, SizeY >	7
4.1.1 Подробное описание	8
4.1.2 Конструктор(ы)	8
4.1.3 Методы	9
4.1.4 Документация по друзьям класса и функциям, относящимся к классу	9
4.1.5 Данные класса	10
4.2 Шаблон класса B< SizeX, SizeY >	10
4.2.1 Подробное описание	12
4.2.2 Конструктор(ы)	12
4.2.3 Методы	13
4.2.4 Документация по друзьям класса и функциям, относящимся к классу	13
4.2.5 Данные класса	14
4.3 Шаблон класса C< SizeX, SizeY >	14
4.3.1 Подробное описание	15
4.3.2 Конструктор(ы)	15
4.3.3 Методы	16
4.3.4 Документация по друзьям класса и функциям, относящимся к классу	17
4.3.5 Данные класса	17
4.4 Шаблон класса D< SizeX, SizeY >	18
4.4.1 Подробное описание	19
4.4.2 Конструктор(ы)	19
4.4.3 Методы	20
4.4.4 Документация по друзьям класса и функциям, относящимся к классу	21
4.4.5 Данные класса	21
4.5 Шаблон класса E< SizeX, SizeY >	22
4.5.1 Подробное описание	23
4.5.2 Конструктор(ы)	24
4.5.3 Методы	24
4.5.4 Документация по друзьям класса и функциям, относящимся к классу	25
4.5.5 Данные класса	25
4.6 Шаблон класса F< SizeX, SizeY >	26
4.6.1 Подробное описание	27
4.6.2 Конструктор(ы)	28
4.6.3 Методы	28

4.6.4 Документация по друзьям класса и функциям, относящимся к классу	29
4.6.5 Данные класса	29
4.7 Шаблон класса <code>G< SizeX, SizeY ></code>	30
4.7.1 Подробное описание	32
4.7.2 Конструктор(ы)	32
4.7.3 Методы	33
4.7.4 Документация по друзьям класса и функциям, относящимся к классу	33
4.7.5 Данные класса	33
Предметный указатель	35

Раздел 1

Тест виртуального ромбовидного наследования

1.1 Test virtual (diamond of death) inheritance

Тест виртуального (ромбовидного) наследования. Все классы иерархии содержат матрицу armadillo и один int в качестве полей. Был удивлен, что необходимо вызывать конструктор копирования класса [B](#) в конструкторе копирования класса [G](#), поскольку был уверен, что это надо делать только для класса [F](#), но не последующих ступеней иерархии.

Test virtual inheritance (diamond of death) with classes all contain one armadillo matrix and integer fields. I was surprised that I have to call copy constructor of [B](#) in copy constructor of [G](#) class, because I thought that it is only needed if [F](#) class, being the 'top' of diamond.

Раздел 2

Иерархический список классов

2.1 Иерархия классов

Иерархия классов.

A< SizeX, SizeY >	7
B< SizeX, SizeY >	10
C< SizeX, SizeY >	14
F< SizeX, SizeY >	26
G< SizeX, SizeY >	30
D< SizeX, SizeY >	18
E< SizeX, SizeY >	22
F< SizeX, SizeY >	26

Раздел 3

Алфавитный указатель классов

3.1 Классы

Классы с их кратким описанием.

A< SizeX, SizeY >	7
B< SizeX, SizeY >	10
C< SizeX, SizeY >	14
D< SizeX, SizeY >	18
E< SizeX, SizeY >	22
F< SizeX, SizeY >	26
G< SizeX, SizeY >	30

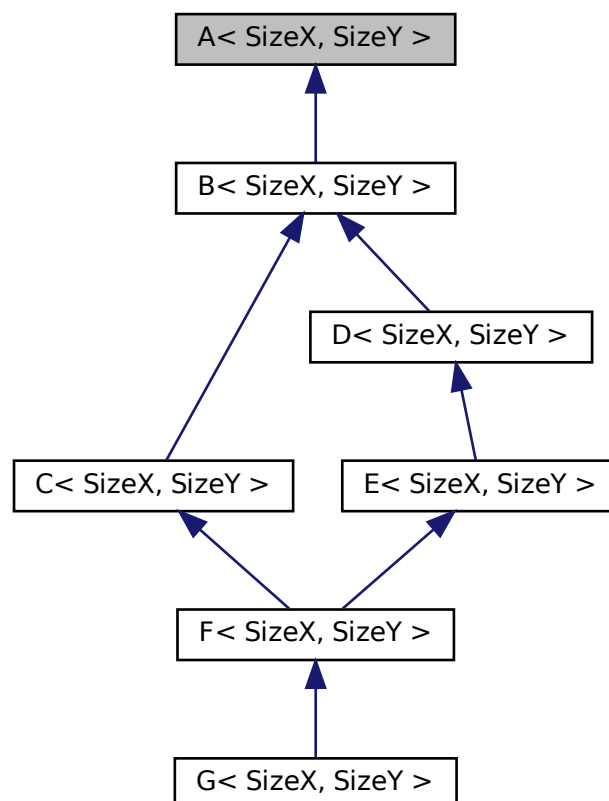
Раздел 4

Классы

4.1 Шаблон класса A< SizeX, SizeY >

```
#include <classes.h>
```

Граф наследования: A< SizeX, SizeY >:



Открытые члены

- [A](#) ()
- [A](#) (const [A](#) &other)
- [A](#) & [operator=](#) (const [A](#) &other)
- [A](#) ([A](#) &&other) noexcept
- [A](#) & [operator=](#) ([A](#) &&other) noexcept
- virtual [~A](#) ()

Открытые атрибуты

- int [IntegerA](#)
- arma::mat [MatrixA](#)

Друзья

- void [swap](#) ([A](#)< SizeX, SizeY > &lhs, [A](#)< SizeX, SizeY > &rhs) noexcept

4.1.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class A< SizeX, SizeY >
```

См. определение в файле [classes.h](#) строка [19](#)

4.1.2 Конструктор(ы)

4.1.2.1 A() [1/3]

```
template<size_t SizeX, size_t SizeY>
A< SizeX, SizeY >::A ( ) [inline]
```

См. определение в файле [classes.h](#) строка [22](#)

4.1.2.2 A() [2/3]

```
template<size_t SizeX, size_t SizeY>
A< SizeX, SizeY >::A (
    const A< SizeX, SizeY > & other ) [inline]
```

См. определение в файле [classes.h](#) строка [28](#)

4.1.2.3 A() [3/3]

```
template<size_t SizeX, size_t SizeY>
A< SizeX, SizeY >::A (
    A< SizeX, SizeY > && other ) [inline], [noexcept]
```

См. определение в файле [classes.h](#) строка 44

4.1.2.4 ~A()

```
template<size_t SizeX, size_t SizeY>
virtual A< SizeX, SizeY >::~A ( ) [inline], [virtual]
```

См. определение в файле [classes.h](#) строка 58

4.1.3 Методы

4.1.3.1 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
A & A< SizeX, SizeY >::operator= (
    A< SizeX, SizeY > && other ) [inline], [noexcept]
```

См. определение в файле [classes.h](#) строка 51

4.1.3.2 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
A & A< SizeX, SizeY >::operator= (
    const A< SizeX, SizeY > & other ) [inline]
```

См. определение в файле [classes.h](#) строка 36

4.1.4 Документация по друзьям класса и функциям, относящимся к классу

4.1.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    A< SizeX, SizeY > & lhs,
    A< SizeX, SizeY > & rhs ) [friend]
```

См. определение в файле [classes.h](#) строка 63

4.1.5 Данные класса

4.1.5.1 IntegerA

```
template<size_t SizeX, size_t SizeY>  
int A< SizeX, SizeY >::IntegerA
```

См. определение в файле [classes.h](#) строка 73

4.1.5.2 MatrixA

```
template<size_t SizeX, size_t SizeY>  
arma::mat A< SizeX, SizeY >::MatrixA
```

См. определение в файле [classes.h](#) строка 74

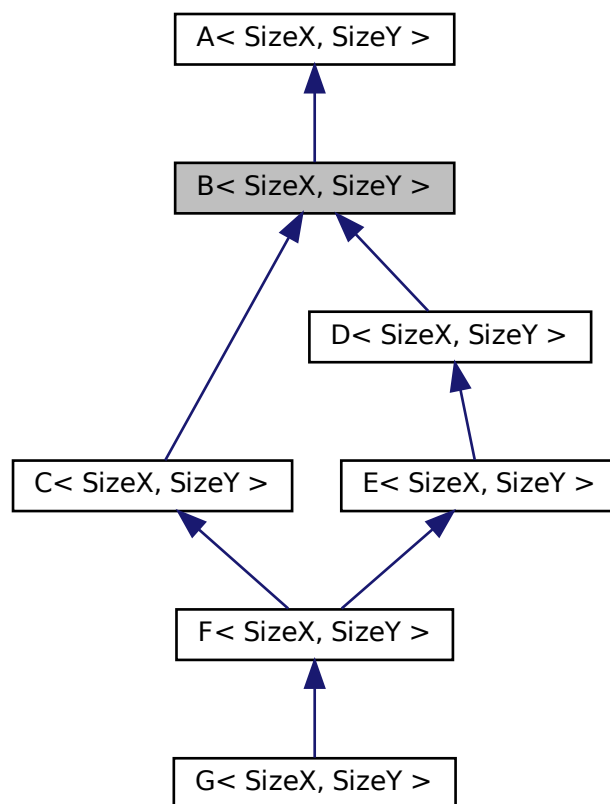
Объявления и описания членов класса находятся в файле:

- `classes.h`

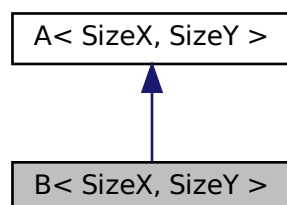
4.2 Шаблон класса B< SizeX, SizeY >

```
#include <classes.h>
```

Граф наследования: B< SizeX, SizeY >:



Граф связей класса B< SizeX, SizeY >:



Открытые члены

- [B \(\)](#)

- `B (const B &other)`
- `B & operator= (const B &other)`
- `B (B &&other) noexcept`
- `B & operator= (B &&other) noexcept`
- `virtual ~B ()`

Открытые атрибуты

- `int IntegerB`
- `arma::mat MatrixB`

Друзья

- `void swap (B< SizeX, SizeY > &lhs, B< SizeX, SizeY > &rhs)`

4.2.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class B< SizeX, SizeY >
```

См. определение в файле `classes.h` строка 78

4.2.2 Конструктор(ы)

4.2.2.1 B() [1/3]

```
template<size_t SizeX, size_t SizeY>
B< SizeX, SizeY >::B ( ) [inline]
```

См. определение в файле `classes.h` строка 83

4.2.2.2 B() [2/3]

```
template<size_t SizeX, size_t SizeY>
B< SizeX, SizeY >::B (
    const B< SizeX, SizeY > & other ) [inline]
```

См. определение в файле `classes.h` строка 89

4.2.2.3 B() [3/3]

```
template<size_t SizeX, size_t SizeY>
B< SizeX, SizeY >::B (
    B< SizeX, SizeY > && other ) [inline], [noexcept]
```

См. определение в файле [classes.h](#) строка 104

4.2.2.4 ~B()

```
template<size_t SizeX, size_t SizeY>
virtual B< SizeX, SizeY >::~~B ( ) [inline], [virtual]
```

См. определение в файле [classes.h](#) строка 117

4.2.3 Методы

4.2.3.1 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
B & B< SizeX, SizeY >::operator= (
    B< SizeX, SizeY > && other ) [inline], [noexcept]
```

См. определение в файле [classes.h](#) строка 110

4.2.3.2 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
B & B< SizeX, SizeY >::operator= (
    const B< SizeX, SizeY > & other ) [inline]
```

См. определение в файле [classes.h](#) строка 96

4.2.4 Документация по друзьям класса и функциям, относящимся к классу

4.2.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    B< SizeX, SizeY > & lhs,
    B< SizeX, SizeY > & rhs ) [friend]
```

См. определение в файле [classes.h](#) строка 122

4.2.5 Данные класса

4.2.5.1 IntegerB

```
template<size_t SizeX, size_t SizeY>  
int B< SizeX, SizeY >::IntegerB
```

См. определение в файле [classes.h](#) строка 133

4.2.5.2 MatrixB

```
template<size_t SizeX, size_t SizeY>  
arma::mat B< SizeX, SizeY >::MatrixB
```

См. определение в файле [classes.h](#) строка 134

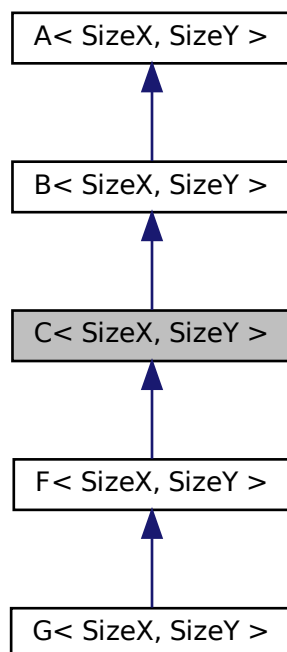
Объявления и описания членов класса находятся в файле:

- [classes.h](#)

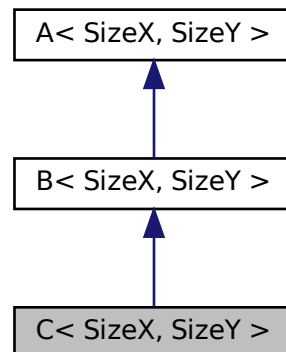
4.3 Шаблон класса C< SizeX, SizeY >

```
#include <classes.h>
```

Граф наследования:C< SizeX, SizeY >:



Граф связей класса C< SizeX, SizeY >:



Открытые члены

- `C ()`
- `C (const C &other)`
- `C & operator= (const C &other)`
- `C (C &&other) noexcept`
- `C & operator= (C &&other) noexcept`
- `virtual ~C ()`

Открытые атрибуты

- `int IntegerC`
- `arma::mat MatrixC`

Друзья

- `void swap (C< SizeX, SizeY > &lhs, C< SizeX, SizeY > &rhs) noexcept`

4.3.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class C< SizeX, SizeY >
```

См. определение в файле `classes.h` строка 138

4.3.2 Конструктор(ы)

4.3.2.1 C() [1/3]

```
template<size_t SizeX, size_t SizeY>  
C< SizeX, SizeY >::C ( ) [inline]
```

См. определение в файле [classes.h](#) строка 143

4.3.2.2 C() [2/3]

```
template<size_t SizeX, size_t SizeY>  
C< SizeX, SizeY >::C (   
    const C< SizeX, SizeY > & other ) [inline]
```

См. определение в файле [classes.h](#) строка 149

4.3.2.3 C() [3/3]

```
template<size_t SizeX, size_t SizeY>  
C< SizeX, SizeY >::C (   
    C< SizeX, SizeY > && other ) [inline], [noexcept]
```

См. определение в файле [classes.h](#) строка 164

4.3.2.4 ~C()

```
template<size_t SizeX, size_t SizeY>  
virtual C< SizeX, SizeY >::~~C ( ) [inline], [virtual]
```

См. определение в файле [classes.h](#) строка 177

4.3.3 Методы

4.3.3.1 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>  
C & C< SizeX, SizeY >::operator= (   
    C< SizeX, SizeY > && other ) [inline], [noexcept]
```

См. определение в файле [classes.h](#) строка 170

4.3.3.2 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
C & C< SizeX, SizeY >::operator= (
    const C< SizeX, SizeY > & other ) [inline]
```

См. определение в файле [classes.h](#) строка 156

4.3.4 Документация по друзьям класса и функциям, относящимся к классу

4.3.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    C< SizeX, SizeY > & lhs,
    C< SizeX, SizeY > & rhs ) [friend]
```

См. определение в файле [classes.h](#) строка 182

4.3.5 Данные класса

4.3.5.1 IntegerC

```
template<size_t SizeX, size_t SizeY>
int C< SizeX, SizeY >::IntegerC
```

См. определение в файле [classes.h](#) строка 193

4.3.5.2 MatrixC

```
template<size_t SizeX, size_t SizeY>
arma::mat C< SizeX, SizeY >::MatrixC
```

См. определение в файле [classes.h](#) строка 194

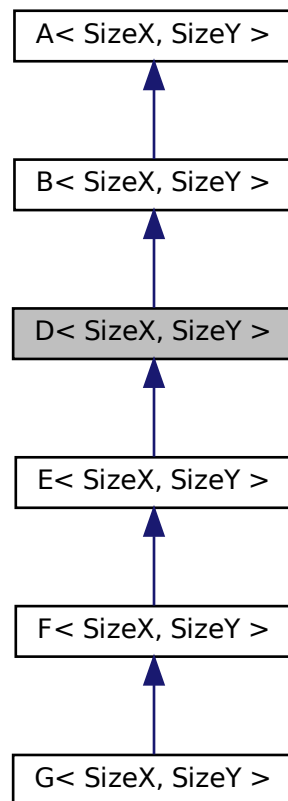
Объявления и описания членов класса находятся в файле:

- [classes.h](#)

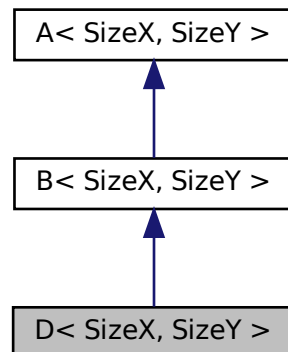
4.4 Шаблон класса D< SizeX, SizeY >

```
#include <classes.h>
```

Граф наследования: D< SizeX, SizeY >:



Граф связей класса D< SizeX, SizeY >:



Открытые члены

- [D](#) ()
- [D](#) (const [D](#) &other)
- [D](#) & [operator=](#) (const [D](#) &other)
- [D](#) ([D](#) &&other) noexcept
- [D](#) & [operator=](#) ([D](#) &&other) noexcept
- virtual [~D](#) ()

Открытые атрибуты

- int [IntegerD](#)
- arma::mat [MatrixD](#)

Друзья

- void [swap](#) ([D](#)< SizeX, SizeY > &lhs, [D](#)< SizeX, SizeY > &rhs) noexcept

4.4.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class D< SizeX, SizeY >
```

См. определение в файле [classes.h](#) строка [198](#)

4.4.2 Конструктор(ы)

4.4.2.1 D() [1/3]

```
template<size_t SizeX, size_t SizeY>  
D< SizeX, SizeY >::D ( ) [inline]
```

См. определение в файле [classes.h](#) строка 203

4.4.2.2 D() [2/3]

```
template<size_t SizeX, size_t SizeY>  
D< SizeX, SizeY >::D (   
    const D< SizeX, SizeY > & other ) [inline]
```

См. определение в файле [classes.h](#) строка 209

4.4.2.3 D() [3/3]

```
template<size_t SizeX, size_t SizeY>  
D< SizeX, SizeY >::D (   
    D< SizeX, SizeY > && other ) [inline], [noexcept]
```

См. определение в файле [classes.h](#) строка 224

4.4.2.4 ~D()

```
template<size_t SizeX, size_t SizeY>  
virtual D< SizeX, SizeY >::~~D ( ) [inline], [virtual]
```

См. определение в файле [classes.h](#) строка 237

4.4.3 Методы

4.4.3.1 operator==() [1/2]

```
template<size_t SizeX, size_t SizeY>  
D & D< SizeX, SizeY >::operator= (   
    const D< SizeX, SizeY > & other ) [inline]
```

См. определение в файле [classes.h](#) строка 216

4.4.3.2 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
D & D< SizeX, SizeY >::operator= (
    D< SizeX, SizeY > && other )    [inline], [noexcept]
```

См. определение в файле [classes.h](#) строка 230

4.4.4 Документация по друзьям класса и функциям, относящимся к классу

4.4.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    D< SizeX, SizeY > & lhs,
    D< SizeX, SizeY > & rhs )    [friend]
```

См. определение в файле [classes.h](#) строка 242

4.4.5 Данные класса

4.4.5.1 IntegerD

```
template<size_t SizeX, size_t SizeY>
int D< SizeX, SizeY >::IntegerD
```

См. определение в файле [classes.h](#) строка 253

4.4.5.2 MatrixD

```
template<size_t SizeX, size_t SizeY>
arma::mat D< SizeX, SizeY >::MatrixD
```

См. определение в файле [classes.h](#) строка 254

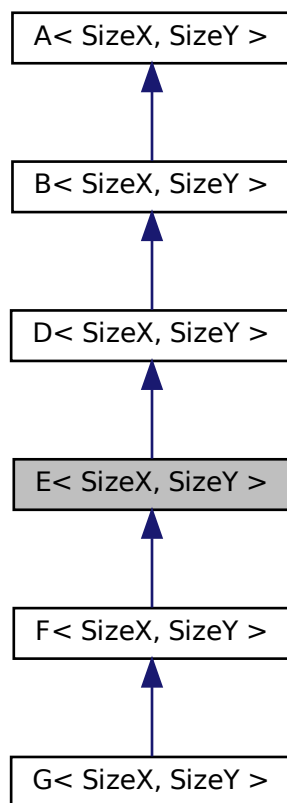
Объявления и описания членов класса находятся в файле:

- [classes.h](#)

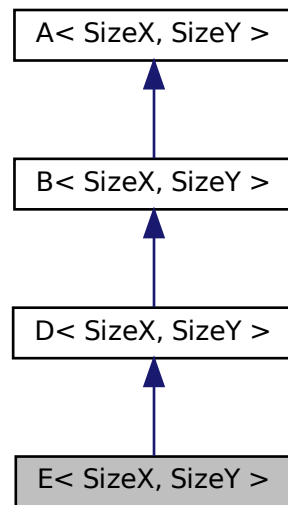
4.5 Шаблон класса E< SizeX, SizeY >

```
#include <classes.h>
```

Граф наследования: E< SizeX, SizeY >:



Граф связей класса E< SizeX, SizeY >:



Открытые члены

- [E](#) ()
- [E](#) (const [E](#) &other)
- [E](#) & [operator=](#) (const [E](#) &other)
- [E](#) ([E](#) &&other) noexcept
- [E](#) & [operator=](#) ([E](#) &&other) noexcept
- virtual [~E](#) ()

Открытые атрибуты

- int [IntegerE](#)
- arma::mat [MatrixE](#)

Друзья

- void [swap](#) ([E](#)< SizeX, SizeY > &lhs, [E](#)< SizeX, SizeY > &rhs) noexcept

4.5.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class E< SizeX, SizeY >
```

См. определение в файле [classes.h](#) строка [258](#)

4.5.2 Конструктор(ы)

4.5.2.1 E() [1/3]

```
template<size_t SizeX, size_t SizeY>  
E< SizeX, SizeY >::E ( ) [inline]
```

См. определение в файле [classes.h](#) строка 263

4.5.2.2 E() [2/3]

```
template<size_t SizeX, size_t SizeY>  
E< SizeX, SizeY >::E (  
    const E< SizeX, SizeY > & other ) [inline]
```

См. определение в файле [classes.h](#) строка 269

4.5.2.3 E() [3/3]

```
template<size_t SizeX, size_t SizeY>  
E< SizeX, SizeY >::E (  
    E< SizeX, SizeY > && other ) [inline], [noexcept]
```

См. определение в файле [classes.h](#) строка 284

4.5.2.4 ~E()

```
template<size_t SizeX, size_t SizeY>  
virtual E< SizeX, SizeY >::~~E ( ) [inline], [virtual]
```

См. определение в файле [classes.h](#) строка 297

4.5.3 Методы

4.5.3.1 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
E & E< SizeX, SizeY >::operator= (
    const E< SizeX, SizeY > & other )    [inline]
```

См. определение в файле [classes.h](#) строка 276

4.5.3.2 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
E & E< SizeX, SizeY >::operator= (
    E< SizeX, SizeY > && other )    [inline], [noexcept]
```

См. определение в файле [classes.h](#) строка 290

4.5.4 Документация по друзьям класса и функциям, относящимся к классу

4.5.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    E< SizeX, SizeY > & lhs,
    E< SizeX, SizeY > & rhs )    [friend]
```

См. определение в файле [classes.h](#) строка 302

4.5.5 Данные класса

4.5.5.1 IntegerE

```
template<size_t SizeX, size_t SizeY>
int E< SizeX, SizeY >::IntegerE
```

См. определение в файле [classes.h](#) строка 313

4.5.5.2 MatrixE

```
template<size_t SizeX, size_t SizeY>  
arma::mat E< SizeX, SizeY >::MatrixE
```

См. определение в файле [classes.h](#) строка 314

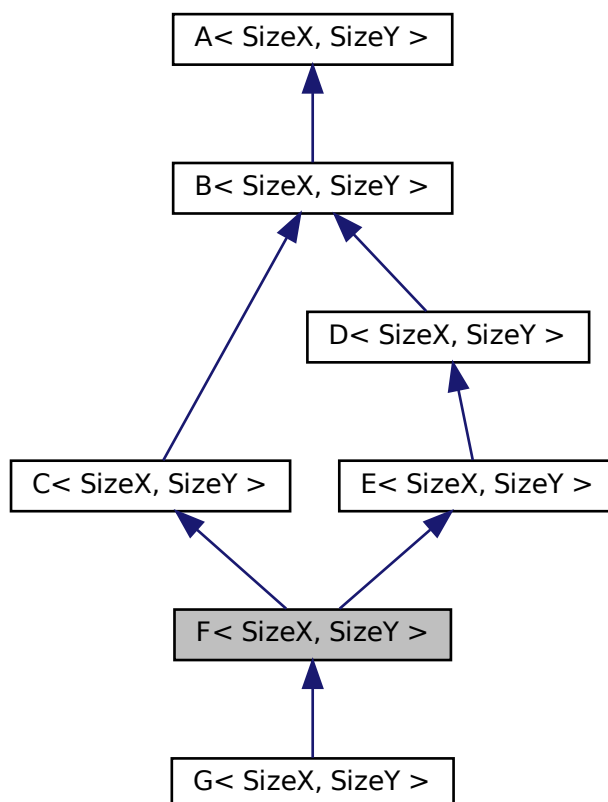
Объявления и описания членов класса находятся в файле:

- `classes.h`

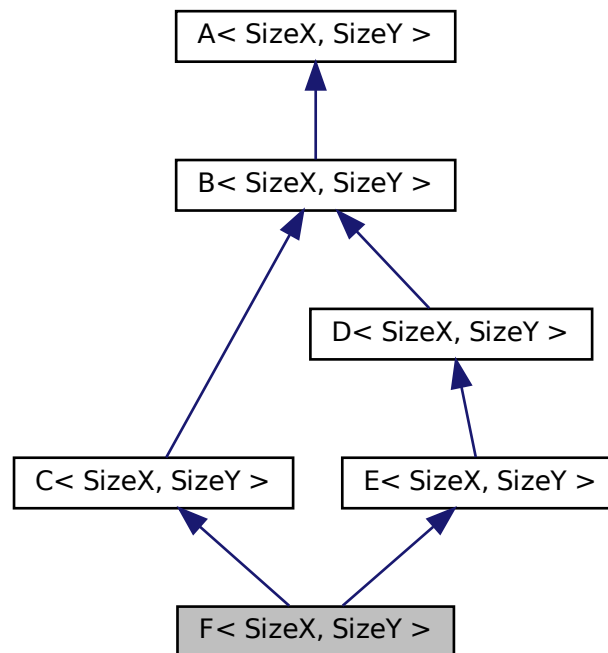
4.6 Шаблон класса F< SizeX, SizeY >

```
#include <classes.h>
```

Граф наследования:F< SizeX, SizeY >:



Граф связей класса F< SizeX, SizeY >:



Открытые члены

- `F ()`
- `F (const F &other)`
- `F & operator= (const F &other)`
- `F (F &&other) noexcept`
- `F & operator= (F &&other) noexcept`
- `virtual ~F ()`

Открытые атрибуты

- `int IntegerF`
- `arma::mat MatrixF`

Друзья

- `void swap (F< SizeX, SizeY > &lhs, F< SizeX, SizeY > &rhs) noexcept`

4.6.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
class F< SizeX, SizeY >
```

См. определение в файле `classes.h` строка 318

4.6.2 Конструктор(ы)

4.6.2.1 F() [1/3]

```
template<size_t SizeX, size_t SizeY>  
F< SizeX, SizeY >::F ( ) [inline]
```

См. определение в файле [classes.h](#) строка 324

4.6.2.2 F() [2/3]

```
template<size_t SizeX, size_t SizeY>  
F< SizeX, SizeY >::F (  
    const F< SizeX, SizeY > & other ) [inline]
```

См. определение в файле [classes.h](#) строка 330

4.6.2.3 F() [3/3]

```
template<size_t SizeX, size_t SizeY>  
F< SizeX, SizeY >::F (  
    F< SizeX, SizeY > && other ) [inline], [noexcept]
```

См. определение в файле [classes.h](#) строка 346

4.6.2.4 ~F()

```
template<size_t SizeX, size_t SizeY>  
virtual F< SizeX, SizeY >::~~F ( ) [inline], [virtual]
```

См. определение в файле [classes.h](#) строка 359

4.6.3 Методы

4.6.3.1 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
F & F< SizeX, SizeY >::operator= (
    const F< SizeX, SizeY > & other ) [inline]
```

См. определение в файле [classes.h](#) строка 338

4.6.3.2 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
F & F< SizeX, SizeY >::operator= (
    F< SizeX, SizeY > && other ) [inline], [noexcept]
```

См. определение в файле [classes.h](#) строка 352

4.6.4 Документация по друзьям класса и функциям, относящимся к классу

4.6.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    F< SizeX, SizeY > & lhs,
    F< SizeX, SizeY > & rhs ) [friend]
```

См. определение в файле [classes.h](#) строка 364

4.6.5 Данные класса

4.6.5.1 IntegerF

```
template<size_t SizeX, size_t SizeY>
int F< SizeX, SizeY >::IntegerF
```

См. определение в файле [classes.h](#) строка 377

4.6.5.2 MatrixF

```
template<size_t SizeX, size_t SizeY>  
arma::mat F< SizeX, SizeY >::MatrixF
```

См. определение в файле [classes.h](#) строка 378

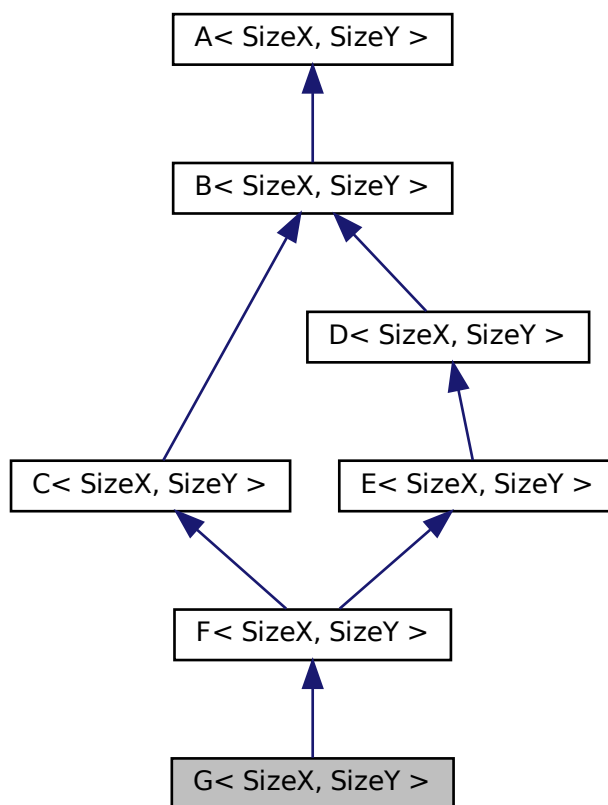
Объявления и описания членов класса находятся в файле:

- [classes.h](#)

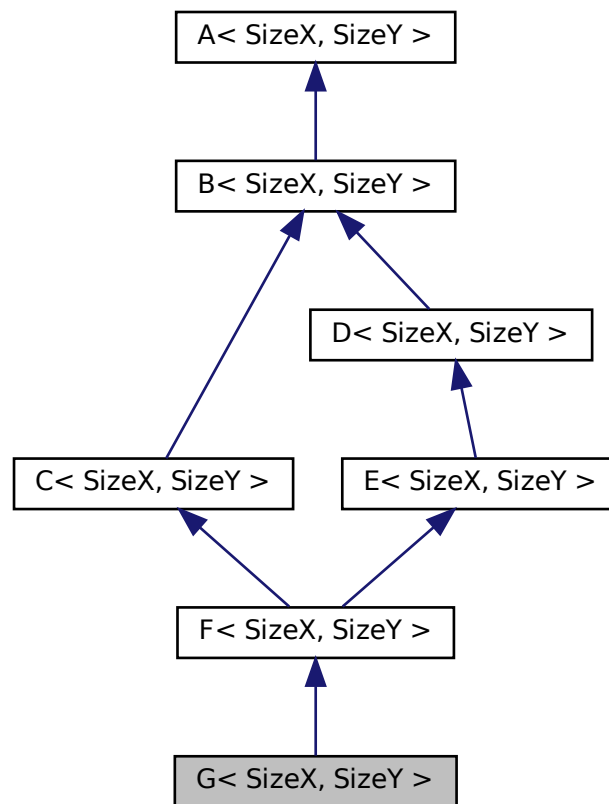
4.7 Шаблон класса G< SizeX, SizeY >

```
#include <classes.h>
```

Граф наследования:G< SizeX, SizeY >:



Граф связей класса G< SizeX, SizeY >:



Открытые члены

- `G ()`
- `G (const G &other)`
- `G & operator= (const G &other)`
- `G (G &&other) noexcept`
- `G & operator= (G &&other) noexcept`
- `virtual ~G ()`

Открытые атрибуты

- `int IntegerG`
- `arma::mat MatrixG`

Друзья

- `void swap (G< SizeX, SizeY > &lhs, G< SizeX, SizeY > &rhs)`

4.7.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>  
class G< SizeX, SizeY >
```

См. определение в файле [classes.h](#) строка 382

4.7.2 Конструктор(ы)

4.7.2.1 G() [1/3]

```
template<size_t SizeX, size_t SizeY>  
G< SizeX, SizeY >::G ( ) [inline]
```

См. определение в файле [classes.h](#) строка 387

4.7.2.2 G() [2/3]

```
template<size_t SizeX, size_t SizeY>  
G< SizeX, SizeY >::G (  
    const G< SizeX, SizeY > & other ) [inline]
```

См. определение в файле [classes.h](#) строка 393

4.7.2.3 G() [3/3]

```
template<size_t SizeX, size_t SizeY>  
G< SizeX, SizeY >::G (  
    G< SizeX, SizeY > && other ) [inline], [noexcept]
```

См. определение в файле [classes.h](#) строка 408

4.7.2.4 ~G()

```
template<size_t SizeX, size_t SizeY>  
virtual G< SizeX, SizeY >::~~G ( ) [inline], [virtual]
```

См. определение в файле [classes.h](#) строка 421

4.7.3 Методы

4.7.3.1 operator=() [1/2]

```
template<size_t SizeX, size_t SizeY>
G & G< SizeX, SizeY >::operator= (
    const G< SizeX, SizeY > & other ) [inline]
```

См. определение в файле [classes.h](#) строка 400

4.7.3.2 operator=() [2/2]

```
template<size_t SizeX, size_t SizeY>
G & G< SizeX, SizeY >::operator= (
    G< SizeX, SizeY > && other ) [inline], [noexcept]
```

См. определение в файле [classes.h](#) строка 414

4.7.4 Документация по друзьям класса и функциям, относящимся к классу

4.7.4.1 swap

```
template<size_t SizeX, size_t SizeY>
void swap (
    G< SizeX, SizeY > & lhs,
    G< SizeX, SizeY > & rhs ) [friend]
```

См. определение в файле [classes.h](#) строка 426

4.7.5 Данные класса

4.7.5.1 IntegerG

```
template<size_t SizeX, size_t SizeY>
int G< SizeX, SizeY >::IntegerG
```

См. определение в файле [classes.h](#) строка 437

4.7.5.2 MatrixG

```
template<size_t SizeX, size_t SizeY>
arma::mat G< SizeX, SizeY >::MatrixG
```

См. определение в файле [classes.h](#) строка 438

Объявления и описания членов класса находятся в файле:

- [classes.h](#)

Предметный указатель

- ~A
 - A< SizeX, SizeY >, 9
- ~B
 - B< SizeX, SizeY >, 13
- ~C
 - C< SizeX, SizeY >, 16
- ~D
 - D< SizeX, SizeY >, 20
- ~E
 - E< SizeX, SizeY >, 24
- ~F
 - F< SizeX, SizeY >, 28
- ~G
 - G< SizeX, SizeY >, 32
- A
 - A< SizeX, SizeY >, 8
- A< SizeX, SizeY >, 7
 - ~A, 9
 - A, 8
 - IntegerA, 10
 - MatrixA, 10
 - operator=, 9
 - swap, 9
- B
 - B< SizeX, SizeY >, 12
- B< SizeX, SizeY >, 10
 - ~B, 13
 - B, 12
 - IntegerB, 14
 - MatrixB, 14
 - operator=, 13
 - swap, 13
- C
 - C< SizeX, SizeY >, 15, 16
- C< SizeX, SizeY >, 14
 - ~C, 16
 - C, 15, 16
 - IntegerC, 17
 - MatrixC, 17
 - operator=, 16
 - swap, 17
- D
 - D< SizeX, SizeY >, 19, 20
- D< SizeX, SizeY >, 18
 - ~D, 20
 - D, 19, 20
- IntegerD, 21
- MatrixD, 21
- operator=, 20
- swap, 21
- E
 - E< SizeX, SizeY >, 24
- E< SizeX, SizeY >, 22
 - ~E, 24
 - E, 24
 - IntegerE, 25
 - MatrixE, 25
 - operator=, 24, 25
 - swap, 25
- F
 - F< SizeX, SizeY >, 28
- F< SizeX, SizeY >, 26
 - ~F, 28
 - F, 28
 - IntegerF, 29
 - MatrixF, 29
 - operator=, 28, 29
 - swap, 29
- G
 - G< SizeX, SizeY >, 32
- G< SizeX, SizeY >, 30
 - ~G, 32
 - G, 32
 - IntegerG, 33
 - MatrixG, 33
 - operator=, 33
 - swap, 33
- IntegerA
 - A< SizeX, SizeY >, 10
- IntegerB
 - B< SizeX, SizeY >, 14
- IntegerC
 - C< SizeX, SizeY >, 17
- IntegerD
 - D< SizeX, SizeY >, 21
- IntegerE
 - E< SizeX, SizeY >, 25
- IntegerF
 - F< SizeX, SizeY >, 29
- IntegerG
 - G< SizeX, SizeY >, 33

MatrixA

A< SizeX, SizeY >, [10](#)

MatrixB

B< SizeX, SizeY >, [14](#)

MatrixC

C< SizeX, SizeY >, [17](#)

MatrixD

D< SizeX, SizeY >, [21](#)

MatrixE

E< SizeX, SizeY >, [25](#)

MatrixF

F< SizeX, SizeY >, [29](#)

MatrixG

G< SizeX, SizeY >, [33](#)

operator=

A< SizeX, SizeY >, [9](#)B< SizeX, SizeY >, [13](#)C< SizeX, SizeY >, [16](#)D< SizeX, SizeY >, [20](#)E< SizeX, SizeY >, [24](#), [25](#)F< SizeX, SizeY >, [28](#), [29](#)G< SizeX, SizeY >, [33](#)

swap

A< SizeX, SizeY >, [9](#)B< SizeX, SizeY >, [13](#)C< SizeX, SizeY >, [17](#)D< SizeX, SizeY >, [21](#)E< SizeX, SizeY >, [25](#)F< SizeX, SizeY >, [29](#)G< SizeX, SizeY >, [33](#)