

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 структуры A< SizeX, SizeY >	7
4.1.1 Подробное описание	8
4.1.2 Конструктор(ы)	8
4.1.3 Методы	9
4.1.4 Данные класса	9
4.2 Шаблон структуры B< SizeX, SizeY >	10
4.2.1 Подробное описание	11
4.2.2 Конструктор(ы)	11
4.2.3 Данные класса	11
4.3 Шаблон структуры C< SizeX, SizeY >	12
4.3.1 Подробное описание	13
4.3.2 Конструктор(ы)	13
4.3.3 Данные класса	14
4.4 Шаблон структуры D< SizeX, SizeY >	14
4.4.1 Подробное описание	16
4.4.2 Конструктор(ы)	16
4.4.3 Данные класса	16
4.5 Шаблон структуры E< SizeX, SizeY >	17
4.5.1 Подробное описание	18
4.5.2 Конструктор(ы)	18
4.5.3 Данные класса	19
4.6 Шаблон структуры F< SizeX, SizeY >	19
4.6.1 Подробное описание	21
4.6.2 Конструктор(ы)	21
4.6.3 Данные класса	22
4.7 Шаблон структуры G< SizeX, SizeY >	22
4.7.1 Подробное описание	24
4.7.2 Конструктор(ы)	25
4.7.3 Данные класса	25
Предметный указатель	27

Раздел 1

Тест виртуального ромбовидного наследования

1.1 Test virtual (diamond of death) inheritance

Рис.1 - Структура ромбовидного наследования

Fig.1 - Diamond of death inheritance structure

<? <?

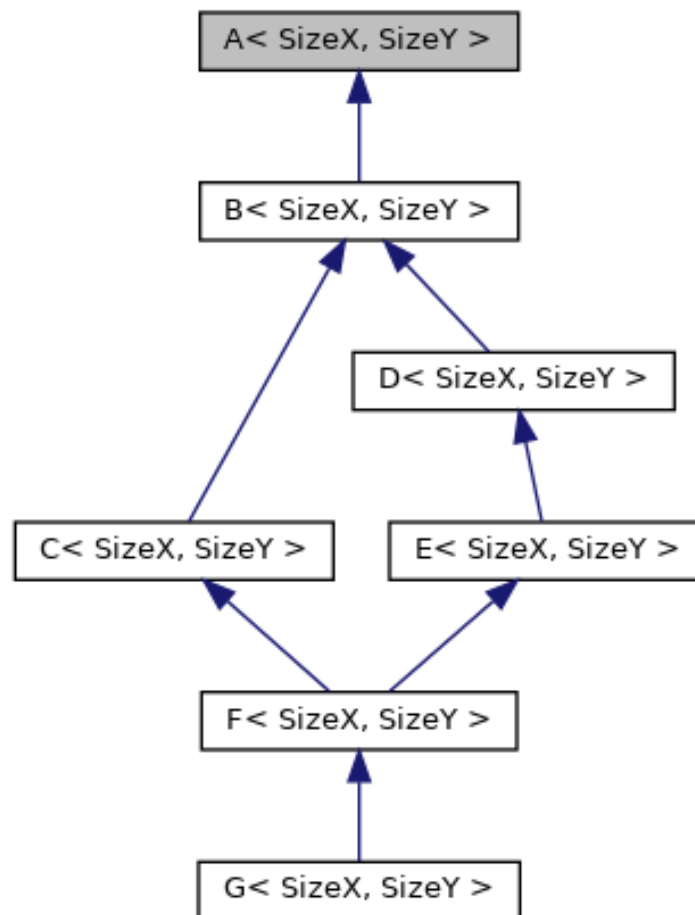


Рис. 1.1 ?>

Тест виртуального (ромбовидного) наследования. Все классы иерархии содержат матрицу `armadillo` и один `int` в качестве полей. Все работает "из коробки", если у родительского класса `A` в явном виде прописать `copy/move constructors` по умолчанию (ключевое слово `default`).

Test virtual inheritance (diamond of death) with classes all contain one `armadillo` matrix and integer fields. Everything works "out of a box" if `copy/move constructors` are provided with "default" keyword in the parent class `A`.

Раздел 2

Иерархический список классов

2.1 Иерархия классов

Иерархия классов.

| | |
|-----------------------------|----|
| A< SizeX, SizeY > | 7 |
| B< SizeX, SizeY > | 10 |
| C< SizeX, SizeY > | 12 |
| F< SizeX, SizeY > | 19 |
| G< SizeX, SizeY > | 22 |
| D< SizeX, SizeY > | 14 |
| E< SizeX, SizeY > | 17 |
| F< SizeX, SizeY > | 19 |

Раздел 3

Алфавитный указатель классов

3.1 Классы

Классы с их кратким описанием.

| | |
|-----------------------------|----|
| A< SizeX, SizeY > | 7 |
| B< SizeX, SizeY > | 10 |
| C< SizeX, SizeY > | 12 |
| D< SizeX, SizeY > | 14 |
| E< SizeX, SizeY > | 17 |
| F< SizeX, SizeY > | 19 |
| G< SizeX, SizeY > | 22 |

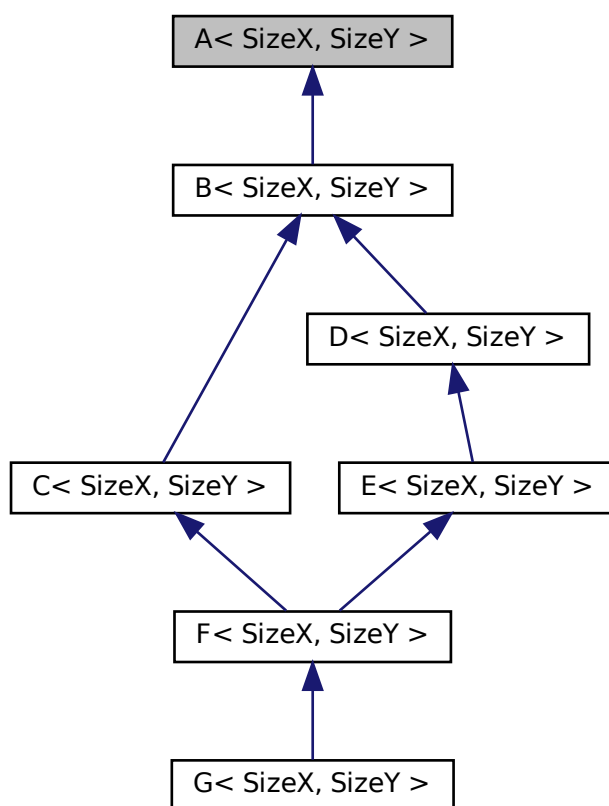
Раздел 4

Классы

4.1 Шаблон структуры A< SizeX, SizeY >

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

Граф наследования: A< SizeX, SizeY >:



Открытые члены

- `A ()`
- `A (const A &)=default`
- `A & operator= (const A &)=default`
- `A (A &&)=default`
- `A & operator= (A &&)=default`
- `virtual ~A ()=default`

Открытые атрибуты

- `int IntegerA`
- `arma::mat MatrixA`

4.1.1 Подробное описание

```
template<size_t SizeX, size_t SizeY>
struct 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 21

4.1.2.2 `A()` [2/3]

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

4.1.2.3 `A()` [3/3]

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

4.1.2.4 ~A()

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

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 > && ) [default]
```

4.1.3.2 operator=() [2/2]

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

4.1.4 Данные класса

4.1.4.1 IntegerA

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

См. определение в файле [classes.h](#) строка [29](#)

4.1.4.2 MatrixA

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

См. определение в файле [classes.h](#) строка [30](#)

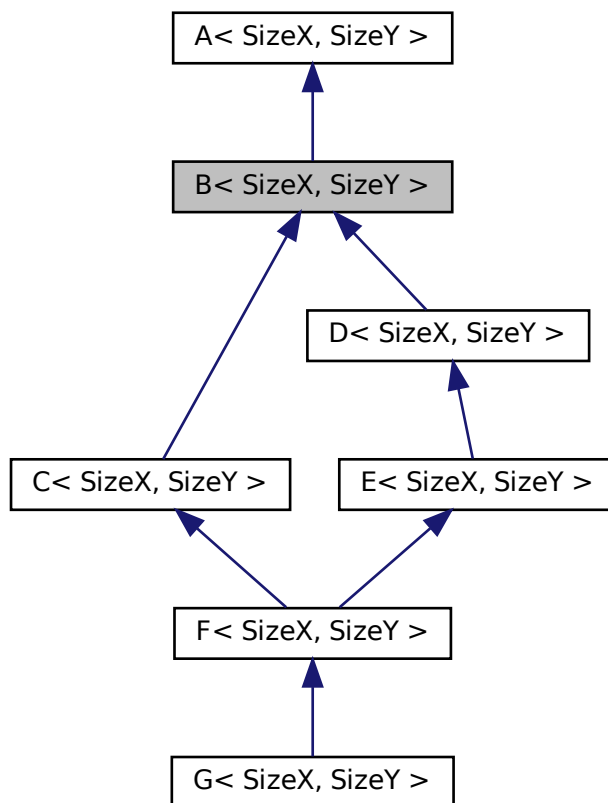
Объявления и описания членов структуры находятся в файле:

- [classes.h](#)

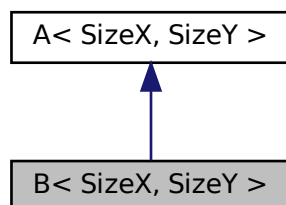
4.2 Шаблон структуры B< SizeX, SizeY >

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

Граф наследования: B< SizeX, SizeY >:



Граф связей класса B< SizeX, SizeY >:



Открытые члены

- [B](#) ()
- virtual [~B](#) ()=default

Открытые атрибуты

- int [IntegerB](#)
- arma::mat [MatrixB](#)

4.2.1 Подробное описание

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

См. определение в файле [classes.h](#) строка [34](#)

4.2.2 Конструктор(ы)

4.2.2.1 B()

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

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

4.2.2.2 ~B()

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

4.2.3 Данные класса

4.2.3.1 IntegerB

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

См. определение в файле [classes.h](#) строка [40](#)

4.2.3.2 MatrixB

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

См. определение в файле [classes.h](#) строка 41

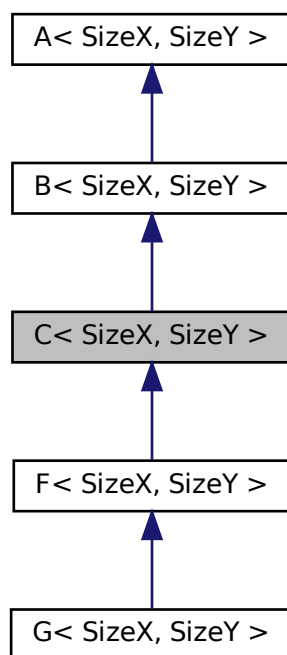
Объявления и описания членов структуры находятся в файле:

- [classes.h](#)

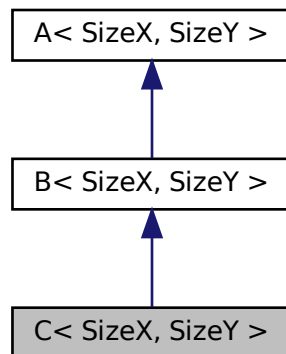
4.3 Шаблон структуры C< SizeX, SizeY >

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

Граф наследования: C< SizeX, SizeY >:



Граф связей класса C< SizeX, SizeY >:



Открытые члены

- [C\(\)](#)

Открытые атрибуты

- int [IntegerC](#)
- arma::mat [MatrixC](#)

4.3.1 Подробное описание

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

См. определение в файле [classes.h](#) строка [45](#)

4.3.2 Конструктор(ы)

4.3.2.1 C()

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

См. определение в файле [classes.h](#) строка [47](#)

4.3.3 Данные класса

4.3.3.1 IntegerC

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

См. определение в файле [classes.h](#) строка 49

4.3.3.2 MatrixC

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

См. определение в файле [classes.h](#) строка 50

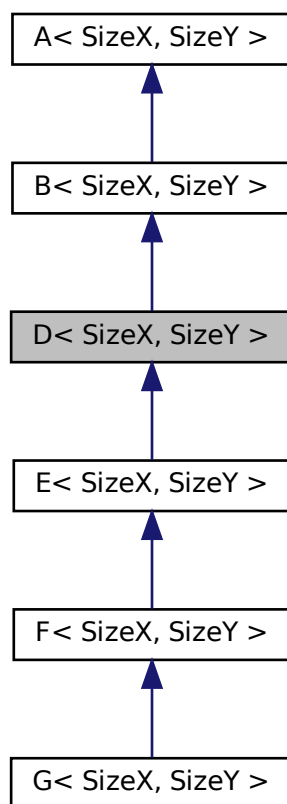
Объявления и описания членов структуры находятся в файле:

- [classes.h](#)

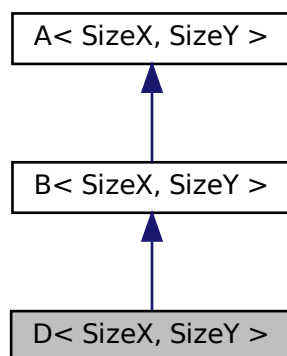
4.4 Шаблон структуры D< SizeX, SizeY >

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

Граф наследования: D< SizeX, SizeY >:



Граф связей класса D< SizeX, SizeY >:



Открытые члены

- [D](#) ()

Открытые атрибуты

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

4.4.1 Подробное описание

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

См. определение в файле [classes.h](#) строка 54

4.4.2 Конструктор(ы)

4.4.2.1 D()

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

См. определение в файле [classes.h](#) строка 56

4.4.3 Данные класса

4.4.3.1 IntegerD

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

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

4.4.3.2 MatrixD

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

См. определение в файле [classes.h](#) строка 59

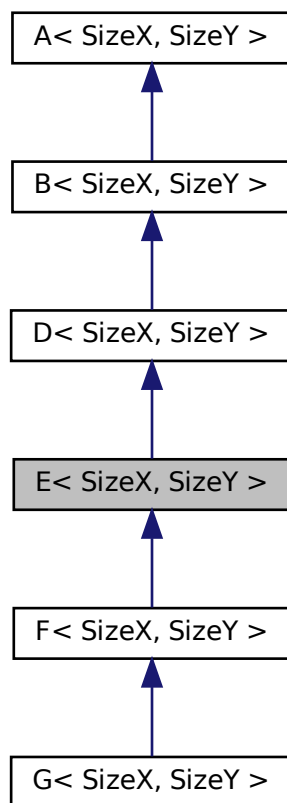
Объявления и описания членов структуры находятся в файле:

- [classes.h](#)

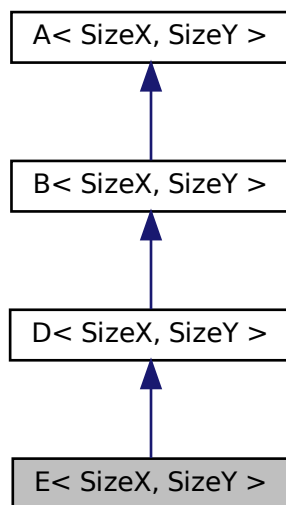
4.5 Шаблон структуры E< SizeX, SizeY >

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

Граф наследования: E< SizeX, SizeY >:



Граф связей класса `E< SizeX, SizeY >`:



Открытые члены

- [E \(\)](#)

Открытые атрибуты

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

4.5.1 Подробное описание

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

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

4.5.2 Конструктор(ы)

4.5.2.1 E()

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

См. определение в файле [classes.h](#) строка [65](#)

4.5.3 Данные класса

4.5.3.1 IntegerE

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

См. определение в файле [classes.h](#) строка 67

4.5.3.2 MatrixE

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

См. определение в файле [classes.h](#) строка 68

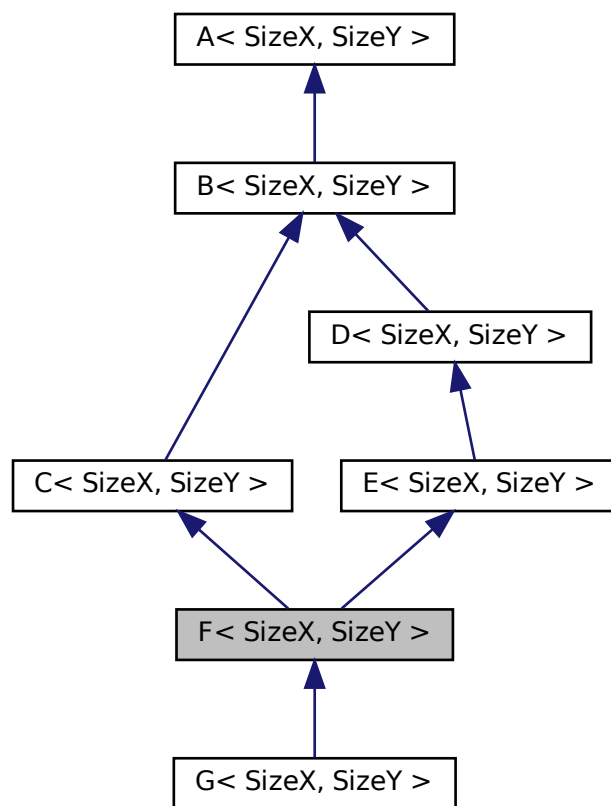
Объявления и описания членов структуры находятся в файле:

- [classes.h](#)

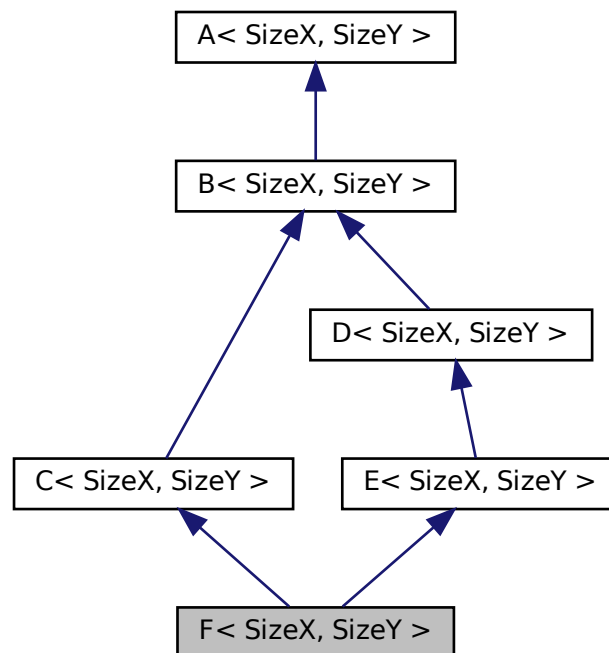
4.6 Шаблон структуры F< SizeX, SizeY >

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

Граф наследования: F < SizeX, SizeY >:



Граф связей класса F< SizeX, SizeY >:



Открытые члены

- [F \(\)](#)

Открытые атрибуты

- `int` [IntegerF](#)
- `arma::mat` [MatrixF](#)

4.6.1 Подробное описание

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

См. определение в файле [classes.h](#) строка 72

4.6.2 Конструктор(ы)

4.6.2.1 F()

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

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

4.6.3 Данные класса

4.6.3.1 IntegerF

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

См. определение в файле [classes.h](#) строка 75

4.6.3.2 MatrixF

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

См. определение в файле [classes.h](#) строка 76

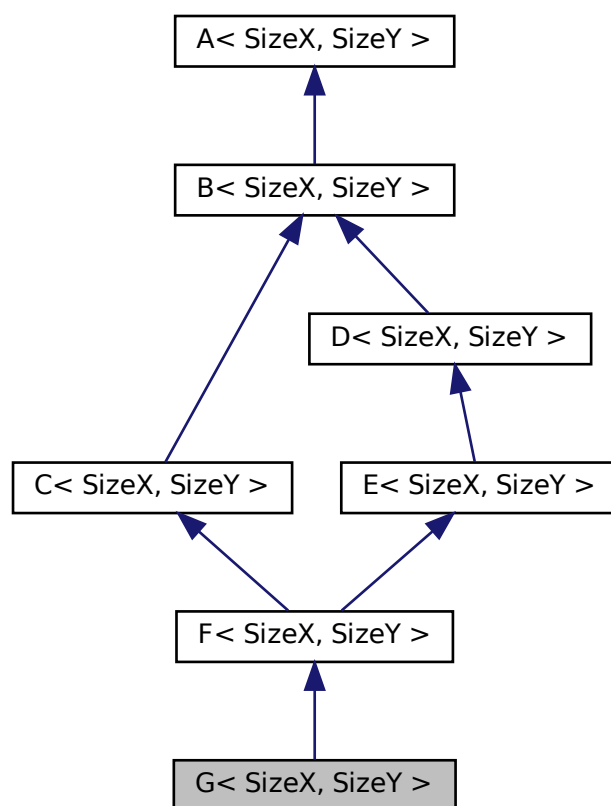
Объявления и описания членов структуры находятся в файле:

- [classes.h](#)

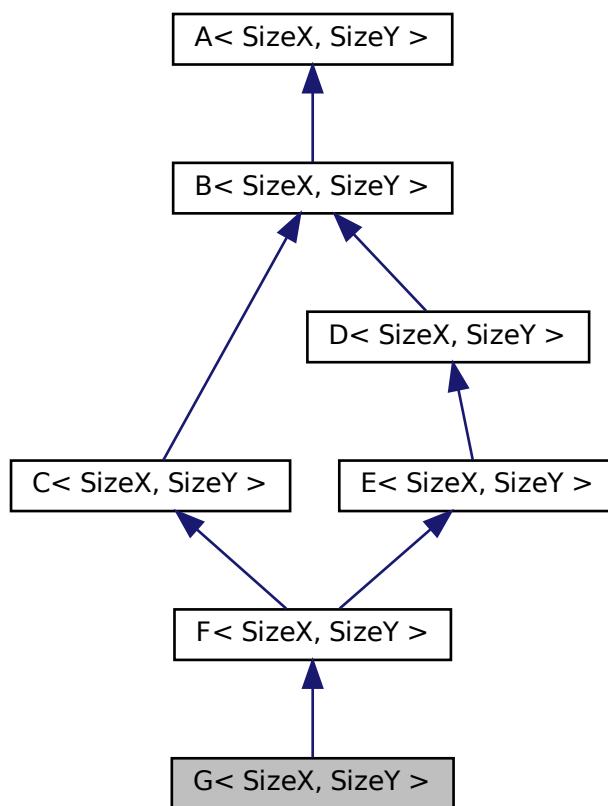
4.7 Шаблон структуры G< SizeX, SizeY >

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

Граф наследования: $G< \text{SizeX}, \text{SizeY} >$:



Граф связей класса `G< SizeX, SizeY >`:



Открытые члены

- [G\(\)](#)

Открытые атрибуты

- `int` [IntegerG](#)
- `arma::mat` [MatrixG](#)

4.7.1 Подробное описание

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

См. определение в файле [classes.h](#) строка 80

4.7.2 Конструктор(ы)

4.7.2.1 G()

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

См. определение в файле [classes.h](#) строка 82

4.7.3 Данные класса

4.7.3.1 IntegerG

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

См. определение в файле [classes.h](#) строка 84

4.7.3.2 MatrixG

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

См. определение в файле [classes.h](#) строка 85

Объявления и описания членов структуры находятся в файле:

- [classes.h](#)

Предметный указатель

~A
 A< SizeX, SizeY >, 8
~B
 B< SizeX, SizeY >, 11
A
 A< SizeX, SizeY >, 8
A< SizeX, SizeY >, 7
 ~A, 8
 A, 8
 IntegerA, 9
 MatrixA, 9
 operator=, 9
B
 B< SizeX, SizeY >, 11
B< SizeX, SizeY >, 10
 ~B, 11
 B, 11
 IntegerB, 11
 MatrixB, 11
C
 C< SizeX, SizeY >, 13
C< SizeX, SizeY >, 12
 C, 13
 IntegerC, 14
 MatrixC, 14
D
 D< SizeX, SizeY >, 16
D< SizeX, SizeY >, 14
 D, 16
 IntegerD, 16
 MatrixD, 16
E
 E< SizeX, SizeY >, 18
E< SizeX, SizeY >, 17
 E, 18
 IntegerE, 19
 MatrixE, 19
F
 F< SizeX, SizeY >, 21
F< SizeX, SizeY >, 19
 F, 21
 IntegerF, 22
 MatrixF, 22
G
 G< SizeX, SizeY >, 25
G< SizeX, SizeY >, 22
 G, 25
 IntegerG, 25
 MatrixG, 25
IntegerA
 A< SizeX, SizeY >, 9
IntegerB
 B< SizeX, SizeY >, 11
IntegerC
 C< SizeX, SizeY >, 14
IntegerD
 D< SizeX, SizeY >, 16
IntegerE
 E< SizeX, SizeY >, 19
IntegerF
 F< SizeX, SizeY >, 22
IntegerG
 G< SizeX, SizeY >, 25
MatrixA
 A< SizeX, SizeY >, 9
MatrixB
 B< SizeX, SizeY >, 11
MatrixC
 C< SizeX, SizeY >, 14
MatrixD
 D< SizeX, SizeY >, 16
MatrixE
 E< SizeX, SizeY >, 19
MatrixF
 F< SizeX, SizeY >, 22
MatrixG
 G< SizeX, SizeY >, 25
operator=
 A< SizeX, SizeY >, 9