

Sitnov\_IKBO-65-23

Создано системой Doxygen 1.12.0



1	Алфавитный указатель пространств имен	1
1.1	Пространства имен	1
2	Иерархический список классов	3
2.1	Иерархия классов	3
3	Алфавитный указатель классов	5
3.1	Классы	5
4	Список файлов	7
4.1	Файлы	7
5	Пространства имен	9
5.1	Пространство имен a	9
5.2	Пространство имен b	9
5.3	Пространство имен c	9
5.4	Пространство имен d	9
5.5	Пространство имен main	9
5.5.1	Функции	10
5.5.1.1	main()	10
6	Классы	11
6.1	Класс a.A	11
6.1.1	Подробное описание	12
6.1.2	Конструктор(ы)	12
6.1.2.1	__init__()	12
6.1.3	Методы	12
6.1.3.1	display()	12
6.1.3.2	get_value()	12
6.1.3.3	set_value()	13
6.1.4	Данные класса	13
6.1.4.1	_value	13
6.2	Класс b.B	13
6.2.1	Подробное описание	14
6.2.2	Конструктор(ы)	15
6.2.2.1	__init__()	15
6.2.3	Методы	15
6.2.3.1	display()	15
6.2.4	Данные класса	15
6.2.4.1	extra_value	15
6.3	Класс c.C	16
6.3.1	Подробное описание	17
6.3.2	Конструктор(ы)	17
6.3.2.1	__init__()	17
6.3.3	Методы	17

---

6.3.3.1 display()	17
6.3.4 Данные класса	18
6.3.4.1 new_field	18
6.4 Класс d.D	18
6.4.1 Подробное описание	18
6.4.2 Конструктор(ы)	18
6.4.2.1 __init__()	18
6.4.3 Методы	19
6.4.3.1 display()	19
6.4.4 Данные класса	19
6.4.4.1 c_instance	19
6.4.4.2 new_field	19
7 Файлы	21
7.1 Файл a.py	21
7.2 Файл b.py	21
7.3 Файл c.py	21
7.4 Файл d.py	22
7.5 Файл main.py	22
Предметный указатель	23

# Глава 1

## Алфавитный указатель пространств имен

### 1.1 Пространства имен

Полный список пространств имен.

<a href="#">a</a>	.....	9
<a href="#">b</a>	.....	9
<a href="#">c</a>	.....	9
<a href="#">d</a>	.....	9
<a href="#">main</a>	.....	9



## Глава 2

# Иерархический список классов

### 2.1 Иерархия классов

Иерархия классов.

a.A . . . . .	11
b.B . . . . .	13
c.C . . . . .	16
d.D . . . . .	18





## Глава 3

# Алфавитный указатель классов

### 3.1 Классы

Классы с их кратким описанием.

<a href="#">a.A</a>	.....	<a href="#">11</a>
<a href="#">b.B</a>	.....	<a href="#">13</a>
<a href="#">c.C</a>	.....	<a href="#">16</a>
<a href="#">d.D</a>	.....	<a href="#">18</a>



## Глава 4

# Список файлов

### 4.1 Файлы

Полный список файлов.

<a href="#">a.py</a>	21
<a href="#">b.py</a>	21
<a href="#">c.py</a>	21
<a href="#">d.py</a>	22
<a href="#">main.py</a>	22



## Глава 5

# Пространства имен

### 5.1 Пространство имен a

Классы

- class [A](#)

### 5.2 Пространство имен b

Классы

- class [B](#)

### 5.3 Пространство имен c

Классы

- class [C](#)

### 5.4 Пространство имен d

Классы

- class [D](#)

### 5.5 Пространство имен main

Функции

- [main](#) ()

## 5.5.1 Функции

### 5.5.1.1 `main()`

```
main.main ()
```

Основная функция для демонстрации работы с классами C и D.

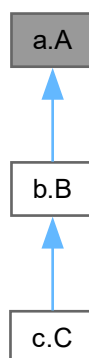
В этой функции создаются экземпляры классов C и D, а затем вызывается метод `display()` для каждого объекта, чтобы продемонстрировать работу с данными, хранящимися в этих классах.

## Глава 6

# Классы

### 6.1 Класс a.A

Граф наследования: a.A:



Открытые члены

- `__init__` (self, value)
- `get_value` (self)
- `set_value` (self, value)
- `display` (self)

Защищенные данные

- `_value` = value

### 6.1.1 Подробное описание

Базовый класс А.

Атрибуты:

value (str): Значение, которое хранит объект.

Методы:

get\_value(): Возвращает значение.

set\_value(value): Устанавливает значение.

display(): Отображает значение.

### 6.1.2 Конструктор(ы)

#### 6.1.2.1 \_\_init\_\_()

```
a.A.__init__(
    self,
    value)
```

Инициализация класса А.

Параметры:

value (str): Значение, которое будет установлено.

Переопределяется в [b.B](#) и [c.C](#).

### 6.1.3 Методы

#### 6.1.3.1 display()

```
a.A.display (
    self)
```

Вывести значение объекта.

Переопределяется в [b.B](#) и [c.C](#).

#### 6.1.3.2 get\_value()

```
a.A.get_value (
    self)
```

Получить значение.

Возвращает:

str: Значение объекта.



### 6.1.3.3 set\_value()

```
a.A.set_value (
    self,
    value)
```

Установить значение.

Параметры:  
value (str): Значение для установки.

## 6.1.4 Данные класса

### 6.1.4.1 \_value

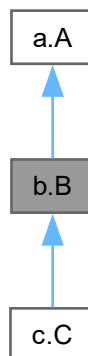
```
a.A._value = value [protected]
```

Объявления и описания членов класса находятся в файле:

- [a.py](#)

## 6.2 Класс b.B

Граф наследования:b.B:



Граф связей класса b.B:



Открытые члены

- `__init__` (self, value, `extra_value`)
- `display` (self)

Открытые члены унаследованные от [a.A](#)

- `get_value` (self)
- `set_value` (self, value)

Открытые атрибуты

- `extra_value` = `extra_value`

Дополнительные унаследованные члены

Защищенные данные унаследованные от [a.A](#)

- `_value` = value

### 6.2.1 Подробное описание

Класс В, наследующий от класса А.

Атрибуты:

`extra_value` (str): Дополнительное значение для объекта В.

Методы:

`display()`: Отображает значения из класса А и В.

### 6.2.2 Конструктор(ы)

#### 6.2.2.1 `__init__()`

```
b.B.__init__ (
    self,
    value,
    extra_value)
```

Инициализация класса B.

Параметры:  
value (str): Значение для инициализации базового класса.  
extra\_value (str): Дополнительное значение для класса B.

Переопределяет метод предка [a.A](#).

Переопределяется в [c.C](#).

### 6.2.3 Методы

#### 6.2.3.1 `display()`

```
b.B.display (
    self)
```

Вывести значения объекта B.

Переопределяет метод предка [a.A](#).

Переопределяется в [c.C](#).

### 6.2.4 Данные класса

#### 6.2.4.1 `extra_value`

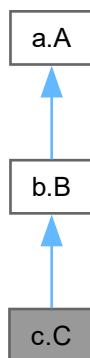
```
b.B.extra_value = extra_value
```

Объявления и описания членов класса находятся в файле:

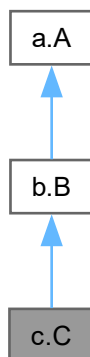
- [b.py](#)

### 6.3 Класс c.C

Граф наследования: c.C:



Граф связей класса c.C:



Открытые члены

- `__init__` (self, value, [extra\\_value](#), [new\\_field](#))
- `display` (self)

Открытые члены унаследованные от [b.B](#)

Открытые члены унаследованные от [a.A](#)

- `get_value` (self)
- `set_value` (self, value)

Открытые атрибуты

- `new_field` = `new_field`

Открытые атрибуты унаследованные от [b.B](#)

- `extra_value` = `extra_value`

Дополнительные унаследованные члены

Защищенные данные унаследованные от [a.A](#)

- `_value` = `value`

### 6.3.1 Подробное описание

Класс C, наследующий от класса B.

Атрибуты:

`new_field` (str): Дополнительное поле для объекта C.

Методы:

`display()`: Отображает значения из класса A, B и C.

### 6.3.2 Конструктор(ы)

#### 6.3.2.1 `__init__()`

```
c.C.__init__(
    self,
    value,
    extra_value,
    new_field)
```

Инициализация класса C.

Параметры:

`value` (str): Значение для инициализации базового класса.

`extra_value` (str): Значение для инициализации класса B.

`new_field` (str): Новое поле для класса C.

Переопределяет метод предка [b.B](#).

### 6.3.3 Методы

#### 6.3.3.1 `display()`

```
c.C.display (
    self)
```

Вывести значения объекта C.

Переопределяет метод предка [b.B](#).

### 6.3.4 Данные класса

#### 6.3.4.1 new\_field

```
c.C.new_field = new_field
```

Объявления и описания членов класса находятся в файле:

- [c.py](#)

## 6.4 Класс d.D

Открытые члены

- [\\_\\_init\\_\\_](#) (self, value, extra\_value, [new\\_field](#))
- [display](#) (self)

Открытые атрибуты

- [c\\_instance](#) = [C](#)(value, extra\_value, [new\\_field](#))
- [new\\_field](#) = new\_field

### 6.4.1 Подробное описание

Класс D, который работает с классом C.

Атрибуты:

[c\\_instance](#) (C): Экземпляр класса C.

[new\\_field](#) (str): Дополнительное поле для объекта D.

Методы:

[display\(\)](#): Отображает значения из класса C и D.

### 6.4.2 Конструктор(ы)

#### 6.4.2.1 \_\_init\_\_()

```
d.D.__init__ (
    self,
    value,
    extra_value,
    new_field)
```

Инициализация класса D.

Параметры:

[value](#) (str): Значение для инициализации класса C.

[extra\\_value](#) (str): Дополнительное значение для инициализации класса C.

[new\\_field](#) (str): Дополнительное поле для класса D.

### 6.4.3 Методы

#### 6.4.3.1 display()

```
d.D.display (  
    self)
```

Выводит значения из класса C и D.

Класс C отображает значения из классов A и B, а также из своего поля.  
Класс D отображает дополнительные значения, присущие только ему.

### 6.4.4 Данные класса

#### 6.4.4.1 c\_instance

```
d.D.c_instance = C(value, extra_value, new_field)
```

#### 6.4.4.2 new\_field

```
d.D.new_field = new_field
```

Объявления и описания членов класса находятся в файле:

- [d.py](#)





## Глава 7

# Файлы

### 7.1 Файл a.py

Классы

- class [a.A](#)

Пространства имен

- namespace [a](#)

### 7.2 Файл b.py

Классы

- class [b.B](#)

Пространства имен

- namespace [b](#)

### 7.3 Файл c.py

Классы

- class [c.C](#)

Пространства имен

- namespace [c](#)

## 7.4 Файл d.py

### Классы

- class [d.D](#)

### Пространства имен

- namespace [d](#)

## 7.5 Файл main.py

### Пространства имен

- namespace [main](#)

### Функции

- [main.main](#) ()

# Предметный указатель

- `--init__`
    - `a.A`, [12](#)
    - `b.B`, [15](#)
    - `c.C`, [17](#)
    - `d.D`, [18](#)
  - `_value`
    - `a.A`, [13](#)
- `a`, [9](#)
- `a.A`, [11](#)
  - `--init__`, [12](#)
  - `_value`, [13](#)
  - `display`, [12](#)
  - `get_value`, [12](#)
  - `set_value`, [12](#)
- `a.py`, [21](#)
- `b`, [9](#)
- `b.B`, [13](#)
  - `--init__`, [15](#)
  - `display`, [15](#)
  - `extra_value`, [15](#)
- `b.py`, [21](#)
- `c`, [9](#)
- `c.C`, [16](#)
  - `--init__`, [17](#)
  - `display`, [17](#)
  - `new_field`, [18](#)
- `c.py`, [21](#)
- `c_instance`
  - `d.D`, [19](#)
- `d`, [9](#)
- `d.D`, [18](#)
  - `--init__`, [18](#)
  - `c_instance`, [19](#)
  - `display`, [19](#)
  - `new_field`, [19](#)
- `d.py`, [22](#)
- `display`
  - `a.A`, [12](#)
  - `b.B`, [15](#)
  - `c.C`, [17](#)
  - `d.D`, [19](#)
- `extra_value`
  - `b.B`, [15](#)
- `get_value`
- `a.A`, [12](#)
- `main`, [9](#)
  - `main`, [10](#)
- `main.py`, [22](#)
- `new_field`
  - `c.C`, [18](#)
  - `d.D`, [19](#)
- `set_value`
  - `a.A`, [12](#)