

едение1.11Введениesection.1.1 стирование1.23Тестированиесеction.1.2



args  
2.10.00

Создано системой Doxygen 1.8.13



# Оглавление

1	Пояснительная записка проекта Args	1
1.1	Введение . . . . .	1
1.2	Тестирование . . . . .	3
2	Алфавитный указатель пространств имен	5
2.1	Пространства имен . . . . .	5
3	Иерархический список классов	7
3.1	Иерархия классов . . . . .	7
4	Алфавитный указатель классов	9
4.1	Классы . . . . .	9
5	Список файлов	11
5.1	Файлы . . . . .	11
6	Пространства имен	13
6.1	Пространство имен Args . . . . .	13
6.1.1	Функции . . . . .	13
6.1.1.1	handler() . . . . .	13
6.2	Пространство имен test . . . . .	14
6.2.1	Подробное описание . . . . .	14
6.3	Пространство имен wnd . . . . .	14

---

7	Классы	15
7.1	Класс <code>wnd._Button</code>	15
7.1.1	Конструктор(ы)	16
7.1.1.1	<code>_Button()</code> [1/2]	16
7.1.1.2	<code>_Button()</code> [2/2]	16
7.1.2	Методы	17
7.1.2.1	<code>handler()</code>	17
7.2	Класс <code>wnd._Label</code>	17
7.2.1	Конструктор(ы)	18
7.2.1.1	<code>_Label()</code>	18
7.3	Класс <code>wnd._LabelInfo</code>	19
7.3.1	Конструктор(ы)	20
7.3.1.1	<code>_LabelInfo()</code>	20
7.4	Класс <code>wnd._TextBox</code>	20
7.4.1	Конструктор(ы)	22
7.4.1.1	<code>_TextBox()</code>	22
7.4.2	Методы	22
7.4.2.1	<code>ordNo()</code>	22
7.4.2.2	<code>restore()</code>	23
7.4.2.3	<code>set()</code>	23
7.4.3	Данные класса	23
7.4.3.1	<code>defVal</code>	24
7.4.3.2	<code>retArg</code>	24
7.4.3.3	<code>txtChanged</code>	24
7.5	Класс <code>wnd._ToolBarButton</code>	24
7.5.1	Конструктор(ы)	25
7.5.1.1	<code>_ToolBarButton()</code>	25
7.6	Класс <code>Args.Arg</code>	26
7.6.1	Подробное описание	28
7.6.2	Конструктор(ы)	28

---

---

7.6.2.1	Arg()	28
7.6.3	Методы	29
7.6.3.1	check()	29
7.6.3.2	checkInput()	30
7.6.3.3	longHelp()	30
7.6.3.4	mkArgs() [1/2]	31
7.6.3.5	mkArgs() [2/2]	31
7.6.3.6	mkVHelp() [1/2]	32
7.6.3.7	mkVHelp() [2/2]	32
7.6.3.8	set() [1/2]	33
7.6.3.9	set() [2/2]	33
7.6.3.10	val()	34
7.6.3.11	version()	34
7.6.4	Данные класса	34
7.6.4.1	edit	34
7.6.4.2	isPassword	35
7.6.4.3	lHlp	35
7.6.4.4	lNm	35
7.6.4.5	required	35
7.6.4.6	sHlp	35
7.6.4.7	show	35
7.6.4.8	sNm	36
7.6.4.9	txtChanged	36
7.6.4.10	vals	36
7.7	Класс Args.ArgChar	36
7.7.1	Подробное описание	39
7.7.2	Конструктор(ы)	39
7.7.2.1	ArgChar()	39
7.7.3	Методы	39
7.7.3.1	operator char()	39

---

7.7.3.2	set() [1/2]	39
7.7.3.3	set() [2/2]	40
7.7.3.4	val()	40
7.8	Класс Args.ArgFlg	40
7.8.1	Подробное описание	43
7.8.2	Конструктор(ы)	43
7.8.2.1	ArgFlg()	43
7.8.3	Методы	43
7.8.3.1	operator bool()	43
7.8.3.2	restore()	43
7.8.3.3	set() [1/2]	44
7.8.3.4	set() [2/2]	44
7.8.3.5	toggle()	44
7.8.3.6	val()	44
7.9	Класс Args.ArgFloat	45
7.9.1	Конструктор(ы)	47
7.9.1.1	ArgFloat() [1/2]	47
7.9.1.2	ArgFloat() [2/2]	47
7.9.2	Методы	47
7.9.2.1	operator double()	47
7.9.2.2	operator float()	48
7.9.2.3	set() [1/4]	48
7.9.2.4	set() [2/4]	48
7.9.2.5	set() [3/4]	48
7.9.2.6	set() [4/4]	48
7.9.2.7	val()	49
7.9.3	Данные класса	49
7.9.3.1	v	49
7.10	Класс Args.ArgFloatMM	49
7.10.1	Подробное описание	52



---

7.10.2	Конструктор(ы)	52
7.10.2.1	ArgFloatMM()	52
7.10.3	Методы	52
7.10.3.1	longHelp()	52
7.10.3.2	set() [1/4]	53
7.10.3.3	set() [2/4]	53
7.10.3.4	set() [3/4]	53
7.10.3.5	set() [4/4]	53
7.10.3.6	setMax()	54
7.10.3.7	setMin()	54
7.11	Класс Args.ArgInt	54
7.11.1	Подробное описание	57
7.11.2	Конструктор(ы)	57
7.11.2.1	ArgInt()	57
7.11.3	Методы	57
7.11.3.1	checkInput()	57
7.11.3.2	operator int()	57
7.11.3.3	set() [1/3]	58
7.11.3.4	set() [2/3]	58
7.11.3.5	set() [3/3]	58
7.11.3.6	val()	58
7.11.4	Данные класса	58
7.11.4.1	v	59
7.12	Класс Args.ArgIntMM	59
7.12.1	Подробное описание	62
7.12.2	Конструктор(ы)	62
7.12.2.1	ArgIntMM()	62
7.12.3	Методы	62
7.12.3.1	longHelp()	62
7.12.3.2	set() [1/3]	62

---

---

7.12.3.3	set() [2/3]	63
7.12.3.4	set() [3/3]	63
7.12.3.5	setMax()	63
7.12.3.6	setMin()	63
7.13	Класс Args.ArgStr	64
7.13.1	Подробное описание	66
7.13.2	Конструктор(ы)	66
7.13.2.1	ArgStr()	66
7.13.3	Методы	66
7.13.3.1	operator string()	66
7.13.3.2	set() [1/2]	66
7.13.3.3	set() [2/2]	66
7.13.3.4	val()	67
7.14	Класс wnd.BaseDialog	68
7.14.1	Конструктор(ы)	70
7.14.1.1	BaseDialog() [1/3]	70
7.14.1.2	BaseDialog() [2/3]	71
7.14.1.3	BaseDialog() [3/3]	71
7.14.2	Методы	72
7.14.2.1	addBox()	72
7.14.2.2	addComboBox()	73
7.14.2.3	addFld2()	73
7.14.2.4	addLabel()	74
7.14.2.5	DoStuff()	75
7.14.2.6	EnableTextBoxes()	75
7.14.2.7	loadH()	75
7.14.2.8	setWidth()	76
7.14.3	Данные класса	76
7.14.3.1	bSz2	76
7.14.3.2	cb1	76

---

7.14.3.3	dbg	76
7.14.3.4	justShow	76
7.14.3.5	l	77
7.14.3.6	ll	77
7.14.3.7	mainTable	77
7.14.3.8	OK_but	77
7.14.3.9	ps	77
7.14.3.10	t1	77
7.14.3.11	txtNo	77
7.14.3.12	w	78
7.15	Класс wnd.check	78
7.15.1	Подробное описание	78
7.15.2	Методы	78
7.15.2.1	isIn()	78
7.16	Структура wnd.CNST	79
7.16.1	Данные класса	80
7.16.1.1	_ADD	80
7.16.1.2	_DEL	80
7.16.1.3	_EDIT	80
7.16.1.4	_ESC	80
7.16.1.5	_EXIT	80
7.16.1.6	_EXP	81
7.16.1.7	_FLT	81
7.16.1.8	_FND	81
7.16.1.9	_NRD	81
7.16.1.10	_OK	81
7.16.1.11	_ORD	81
7.16.1.12	_REF	81
7.16.1.13	CANCEL	81
7.16.1.14	ERR	82

---

---

7.16.1.15 NO	82
7.16.1.16 NOTITEM	82
7.16.1.17 NRD	82
7.16.1.18 NSF	82
7.16.1.19 NULL	82
7.16.1.20 OK	82
7.16.1.21 stop0	82
7.16.1.22 stop1	83
7.16.1.23 stop2	83
7.16.1.24 UNK_TXT	83
7.16.1.25 WHEEL_DELTA	83
7.16.1.26 WPAR	83
7.16.1.27 YES	83
7.17 Класс wnd.DialogModel	84
7.17.1 Конструктор(ы)	85
7.17.1.1 DialogModel()	85
7.17.2 Методы	86
7.17.2.1 countBaseSize()	86
7.17.3 Данные класса	86
7.17.3.1 bSz	86
7.18 Класс wnd.inscr	87
7.18.1 Данные класса	87
7.18.1.1 CURRENT	87
7.18.1.2 HLP	87
7.18.1.3 OK	88
7.18.1.4 PAR1	88
7.18.1.5 PAR2	88
7.18.1.6 START	88
7.18.1.7 TRACK	88
7.18.1.8 VERB	88

---

7.18.1.9	WND	88
7.19	Класс <code>wnd.mk</code>	89
7.19.1	Методы	89
7.19.1.1	<code>Info()</code>	89
7.20	Класс <code>wnd.Names</code>	89
7.20.1	Методы	90
7.20.1.1	<code>Add()</code> [1/3]	90
7.20.1.2	<code>Add()</code> [2/3]	90
7.20.1.3	<code>Add()</code> [3/3]	91
7.20.1.4	<code>Text()</code>	91
7.20.2	Данные класса	91
7.20.2.1	<code>en</code>	91
7.20.2.2	<code>lang</code>	92
7.20.2.3	<code>ru</code>	92
7.20.2.4	<code>ua</code>	92
7.21	Класс <code>wnd.OkCancelDlg</code>	92
7.21.1	Подробное описание	95
7.21.2	Конструктор(ы)	95
7.21.2.1	<code>OkCancelDlg()</code> [1/3]	95
7.21.2.2	<code>OkCancelDlg()</code> [2/3]	95
7.21.2.3	<code>OkCancelDlg()</code> [3/3]	95
7.21.3	Методы	95
7.21.3.1	<code>DoStuff()</code>	96
7.21.4	Данные класса	96
7.21.4.1	<code>ESC_but</code>	96
7.22	Класс <code>wnd.OkDlg</code>	96
7.22.1	Подробное описание	99
7.22.2	Конструктор(ы)	99
7.22.2.1	<code>OkDlg()</code> [1/3]	99
7.22.2.2	<code>OkDlg()</code> [2/3]	99

---

7.22.2.3	OkDlg() [3/3]	99
7.22.3	Методы	99
7.22.3.1	DoStuff()	100
7.22.3.2	initBtn()	100
7.22.4	Данные класса	100
7.22.4.1	current	100
7.23	Класс test.Program	101
7.23.1	Подробное описание	102
7.23.2	Конструктор(ы)	102
7.23.2.1	Program()	102
7.23.3	Методы	103
7.23.3.1	check()	103
7.23.3.2	Main()	103
7.23.3.3	version()	105
7.23.4	Данные класса	105
7.23.4.1	dbgF	106
7.23.4.2	hlpF	106
7.23.4.3	tsk2	106
7.23.4.4	tsk3	106
7.23.4.5	vF	106
7.24	Структура wnd.SZ	107
7.24.1	Данные класса	107
7.24.1.1	FONT	107
7.24.1.2	TD_BUF	108
7.24.1.3	X_BUTTON	108
7.24.1.4	X_FLD	108
7.24.1.5	X_LABEL1	108
7.24.1.6	X_LABEL2	108
7.24.1.7	X_SPC	108
7.24.1.8	X_TEXT	108
7.24.1.9	Y	108
7.24.1.10	Y_FLD	109
7.24.1.11	Y_SPC	109
7.25	Класс wnd.warning	109
7.25.1	Методы	109
7.25.1.1	NRD()	109
7.25.1.2	NTD()	110
7.26	Класс test.WCNST	110
7.26.1	Данные класса	110
7.26.1.1	colors	110

---

---

8	Файлы	111
8.1	Файл cs/args.cs . . . . .	111
8.1.1	Подробное описание . . . . .	112
8.2	Файл cs/args.txt . . . . .	112
8.3	Файл cs/Assembly.txt . . . . .	112
8.4	Файл cs/AssemblyInfo.cs . . . . .	112
8.5	Файл AssemblyInfo.cs . . . . .	112
8.6	Файл cs/ut.cs . . . . .	112
8.7	Файл cs/wBtnLbl.cs . . . . .	112
8.8	Файл cs/wDlgBase.cs . . . . .	113
8.8.1	Макросы . . . . .	113
8.8.1.1	PANEL . . . . .	113
8.9	Файл cs/wOkCancel.cs . . . . .	113
8.10	Файл Program.cs . . . . .	113
	Алфавитный указатель	115





# Глава 1

## Пояснительная записка проекта Args

Дата

2014-2017

### 1.1 Введение

Пояснительная записка к проекту args - библиотеки ввода и разбора параметров приложения и выдачи подсказки - страницы использования приложения (usage). Параметры приложения могут вводиться как аргументы командной строки либо как поля автоматически генерируемого диалогового окна.

Разработчики А.Г.Пискунов, Д.И.Гись, О.С.Мисуна.

В пояснительной записке используются следующие термины:

- аргумент командной строки - элемент массива args из static void Main(string[] args);
- ключ, это аргумент командной строки, начинающийся на символ '-' или '/'. Библиотека игнорирует регистр букв;
- синоним ключа, далее - синоним. Библиотека игнорирует регистр букв;
- описание ключа;
- параметр ключа.

например, на следующей странице использования приложения для тестирования библиотеки

```
to test command line arguments (ver:2.10.0)
usage:
a [-?] [-d] [-v] [-ln NNN] [-l LLL] [-p PPP] -f FLNM -t1 | -t2 | -t3
options:
-?           : to see this help: True
-d           : debug mode: False
-v           : additional info: True
-ln NNN     : log level names:{Spam Debug Warning Stats Error FatalError Info Ignore}: Error
-l LLL      : log level (1..8): 1
-p PPP      : percent for something (..100): 0.05
-f FLNM     : data file: somefile.dat
-t1         : to do some work: False
-t2         : to do another work: False
```

```

-t3          : to show dialog window: False
'?' means the same as 'help'
'd' means the same as 'debug'
'v' means the same as 'verbose'
'ln' means the same as 'logName'
'l' means the same as 'log'
'p' means the same as 'percent'
'f' means the same as 'file'
't1' means the same as 'workOne'
't2' means the same as 'workNext'
't3' means the same as 'window'

```

- ключами являются символы: `-?`, `-v`, `-l`.
- синонимами: `help`, `verbose`, `log`.
- описаниями: `'to see this help'`, `'additional info'`, `'log level'`.
- параметрами ключа: `LLL`, `PPP`, `FLNM`.

Родительским классом для разбора аргументов является [Arg](#).

Страница подсказки выдается

- в случае консольного приложения - в стандартный вывод ошибок,
- в случае приложения Win32 - в диалоговое окно. Свойство приложения, которое позволяет ему выдавать подсказку толи в стандартный вывод ошибок, толи в диалоговое окно, появилась в библиотеке благодаря помощи участника форума [sql.ru](#)

При генерации диалогового окна для различных полей ввода можно задавать

- признак того, что поле не надо редактировать.
- признак того, что поля используются для ввода пароля и не надо отображать текст.
- метод - делегат для проверки корректности введенного в поле значения. В случае некорректного ввода, блокируются возможность вводить что либо в другие поля и нажимать кнопку Ок.
- набор корректных альтернатив для ввода в поле.

Файл был создан утилитами [Doxygen](#) и [Microsoft 's HTML Help Workshop](#). С утилитой [Doxygen](#) можно познакомиться документе [DOXYGEN И GRAPHVIZ : ДОКУМЕНТИРОВАНИЕ ПРОЕКТОВ НА C#](#).

При создании текущей Пояснительной записки использовались следующие параметры файла конфигурации утилиты [Doxygen](#):

- `DOXYFILE_ENCODING = CP1251;`
- `GENERATE_HTMLHELP = YES`. Параметр для создания CHM файла - файла подсказки из упакованного HTML. В каталог, где строится документация выводится файл `index.hhp`, который потом используется утилитой `hhc.exe` из [Microsoft 's HTML Help Workshop](#). для построения файла подсказки;
- `CHM_FILE = report.chm`. Задать имя CHM файла;
- `CHM_INDEX_ENCODING = CP1251;`

При кодировке UTF-8 не удалось получить корректного отображения СНМ файла.

Кроме того, были использованы следующие команды форматирования текста и синонимы для них (в оригинале markdown):

- элемент списка (markdown) начинается с символа минус '-' или плюс '+';
- новый параграф (markdown) - начинается пустой строкой;
- символ обратная кавычка "" начинает и заканчивает небольшой фрагмент кода (текст, который не надо форматировать);
- три минуса подряд с дополнительными символами новой строки (markdown) - начинают и заканчивают неформатированный текст, вокруг которого рисуется линии;
- текст [Doxygen](<http://doxygen.org>) вставляет в документ внешние гиперссылки;
- текст [Arg](@ref [Args.Arg](#)) вставляет в документ внутренние гиперссылки;
- '\mainpage' - изменение начальной страницы документа.

## 1.2 Тестирование

Для тестирования библиотеки используется специальный [Тестюнит](#).



## Глава 2

# Алфавитный указатель пространств имен

### 2.1 Пространства имен

Полный список пространств имен.

<a href="#">Args</a>	.....	13
<a href="#">test</a>	.....	14
<a href="#">wnd</a>	.....	14



## Глава 3

# Иерархический список классов

### 3.1 Иерархия классов

Иерархия классов.

Args.Arg . . . . .	26
Args.ArgChar . . . . .	36
Args.ArgFlg . . . . .	40
Args.ArgFloat . . . . .	45
Args.ArgFloatMM . . . . .	49
Args.ArgInt . . . . .	54
Args.ArgIntMM . . . . .	59
Args.ArgStr . . . . .	64
Button	
wnd._Button . . . . .	15
wnd.check . . . . .	78
wnd.CNST . . . . .	79
Form	
wnd.DialogModel . . . . .	84
wnd.BaseDialog . . . . .	68
wnd.OkDlg . . . . .	96
wnd.OkCancelDlg . . . . .	92
wnd.inscr . . . . .	87
Label	
wnd._Label . . . . .	17
wnd._LabelInfo . . . . .	19
wnd.mk . . . . .	89
wnd.Names . . . . .	89
test.Program . . . . .	101
wnd.SZ . . . . .	107
TextBox	
wnd._TextBox . . . . .	20
ToolBarButton	
wnd._ToolBarButton . . . . .	24
wnd.warning . . . . .	109
test.WCNST . . . . .	110





## Глава 4

# Алфавитный указатель классов

### 4.1 Классы

Классы с их кратким описанием.

<code>wnd._Button</code>	15
<code>wnd._Label</code>	17
<code>wnd._LabelInfo</code>	19
<code>wnd._TextBox</code>	20
<code>wnd._ToolBarButton</code>	24
<code>Args.Arg</code>	
Родительский класс для разбора аргументов командной строки	26
<code>Args.ArgChar</code>	
ключ типа символ	36
<code>Args.ArgFlg</code>	
ключ логического типа (иначе - флаг)	40
<code>Args.ArgFloat</code>	45
<code>Args.ArgFloatMM</code>	
ключ вещественного типа, есть возможность ограничить максимальное и минимальное значение	49
<code>Args.ArgInt</code>	
ключ целого типа	54
<code>Args.ArgIntMM</code>	
ключ целого типа, есть возможность ограничить максимальное и минимальное значение	59
<code>Args.ArgStr</code>	
ключ типа строка	64
<code>wnd.BaseDialog</code>	68
<code>wnd.check</code>	
класс для разного рода проверок	78
<code>wnd.CNST</code>	79
<code>wnd.DialogModel</code>	84
<code>wnd.inscr</code>	87
<code>wnd.mk</code>	89
<code>wnd.Names</code>	89
<code>wnd.OkCancelDlg</code>	
класс для ввода значений можно отменить ввод некоторых полей	92
<code>wnd.OkDlg</code>	
класс для демонстрации значений (вводить нельзя)	96

<a href="#">test.Program</a>	класс содержит глобальные переменные для ключей, тест юнит Main и функцию выдачи подсказки . . . . .	101
<a href="#">wnd.SZ</a>	. . . . .	107
<a href="#">wnd.warning</a>	. . . . .	109
<a href="#">test.WCNST</a>	. . . . .	110

## Глава 5

# Список файлов

### 5.1 Файлы

Полный список файлов.

<a href="#">AssemblyInfo.cs</a> . . . . .	112
<a href="#">Program.cs</a> . . . . .	113
<a href="#">cs/args.cs</a>	
файл содержит классы для разбора командной строки . . . . .	111
<a href="#">cs/AssemblyInfo.cs</a> . . . . .	112
<a href="#">cs/ut.cs</a> . . . . .	112
<a href="#">cs/wBtnLbl.cs</a> . . . . .	112
<a href="#">cs/wDlgBase.cs</a> . . . . .	113
<a href="#">cs/wOkCancel.cs</a> . . . . .	113



## Глава 6

# Пространства имен

### 6.1 Пространство имен Args

#### Классы

- class [Arg](#)  
Родительский класс для разбора аргументов командной строки
- class [ArgChar](#)  
ключ типа символ
- class [ArgFlg](#)  
ключ логического типа (иначе - флаг)
- class [ArgFloat](#)
- class [ArgFloatMM](#)  
ключ вещественного типа, есть возможность ограничить максимальное и минимальное значение
- class [ArgInt](#)  
ключ целого типа
- class [ArgIntMM](#)  
ключ целого типа, есть возможность ограничить максимальное и минимальное значение
- class [ArgStr](#)  
ключ типа строка

#### Функции

- delegate bool [handler](#) (object src, string val)  
указатель для диалоговых окон

#### 6.1.1 Функции

##### 6.1.1.1 handler()

```
delegate bool Args.handler (  
    object src,  
    string val )
```

указатель для диалоговых окон

## 6.2 Пространство имен test

### Классы

- class [Program](#)  
класс содержит глобальные переменные для ключей, тест юнит Main и функцию выдачи подсказки
- class [WCNST](#)

### 6.2.1 Подробное описание

Пространство имен содержит тест-юнит для тестирования библиотеки args.

## 6.3 Пространство имен wnd

### Классы

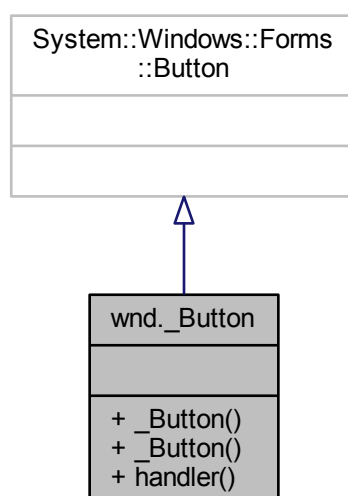
- class [\\_Button](#)
- class [\\_Label](#)
- class [\\_LabelInfo](#)
- class [\\_TextBox](#)
- class [\\_ToolBarButton](#)
- class [BaseDialog](#)
- class [check](#)  
класс для разного рода проверок
- struct [CNST](#)
- class [DialogModel](#)
- class [inscr](#)
- class [mk](#)
- class [Names](#)
- class [OkCancelDlg](#)  
класс для ввода значений можно отменить ввод некоторых полей.
- class [OkDlg](#)  
класс для демонстрации значений (вводить нельзя)
- struct [SZ](#)
- class [warning](#)

## Глава 7

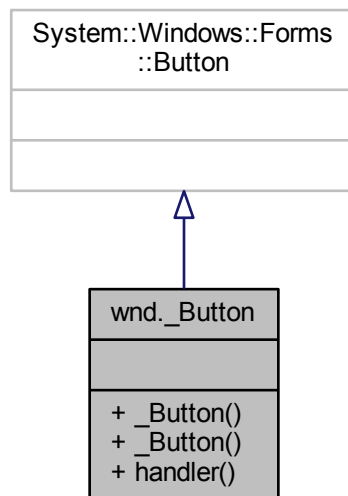
# Классы

### 7.1 Класс wnd.\_Button

Граф наследования:wnd.\_Button:



Граф связей класса `wnd._Button`:



Открытые члены

- `_Button()`
- `_Button(string nm)`
- `void handler(EventHandler click)`

### 7.1.1 Конструктор(ы)

#### 7.1.1.1 `_Button()` [1/2]

`wnd._Button._Button()` [inline]

#### 7.1.1.2 `_Button()` [2/2]

`wnd._Button._Button(string nm)` [inline]

Граф вызовов:





### 7.1.2 Методы

#### 7.1.2.1 handler()

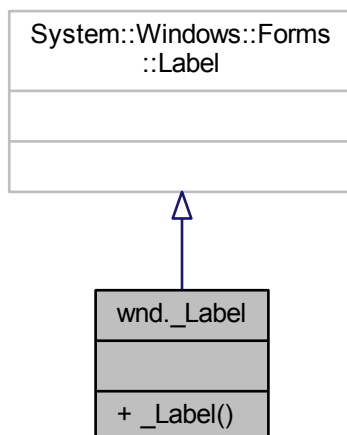
```
void wnd._Button.handler (
    EventHandler click ) [inline]
```

Объявления и описания членов класса находятся в файле:

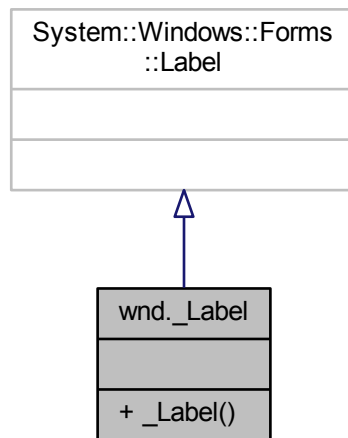
- [cs/wBtnLbl.cs](#)

## 7.2 Класс wnd.\_Label

Граф наследования:wnd.\_Label:



Граф связей класса `wnd._Label`:



Открытые члены

- [\\_Label](#) (string nm, int p=0)

## 7.2.1 Конструктор(ы)

### 7.2.1.1 \_Label()

```
wnd._Label._Label (
    string nm,
    int p = 0 ) [inline]
```

Граф вызовов:

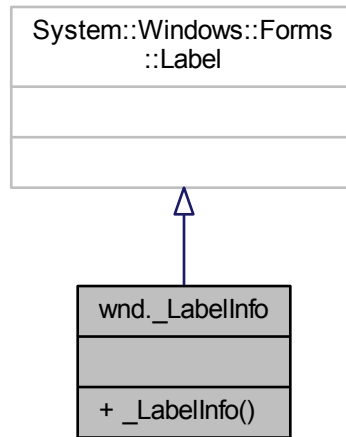


Объявления и описания членов класса находятся в файле:

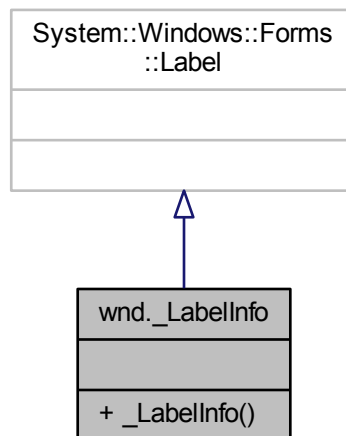
- [cs/wBtnLbl.cs](#)

## 7.3 Класс wnd.\_LabelInfo

Граф наследования:wnd.\_LabelInfo:



Граф связей класса wnd.\_LabelInfo:



Открытые члены

- [\\_LabelInfo \(\)](#)

### 7.3.1 Конструктор(ы)

#### 7.3.1.1 \_LabelInfo()

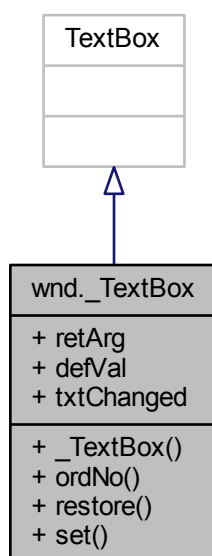
```
wnd._LabelInfo._LabelInfo ( ) [inline]
```

Объявления и описания членов класса находятся в файле:

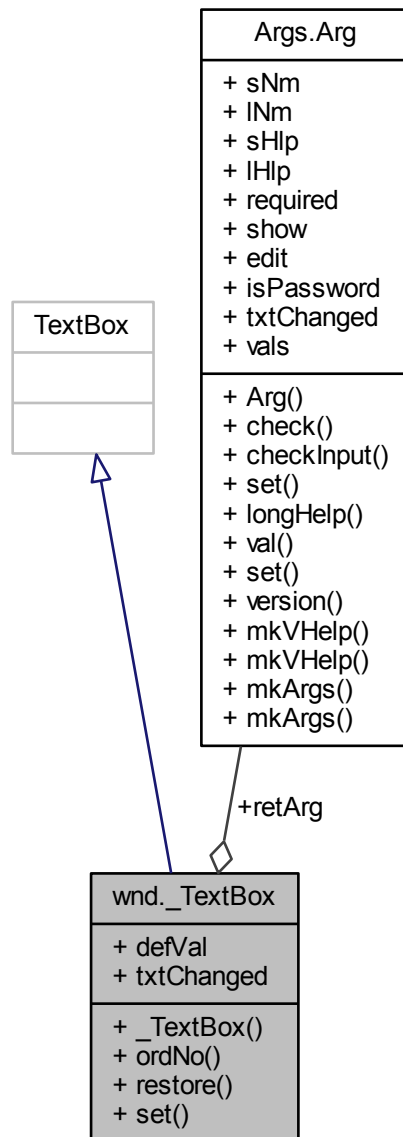
- [cs/wBtnLbl.cs](#)

### 7.4 Класс wnd.\_TextBox

Граф наследования:wnd.\_TextBox:



Граф связей класса wnd.\_TextBox:



### Открытые члены

- `_TextBox (Arg a)`
- `int ordNo ()`  
возвращает порядковый номер поля ввода
- `virtual void restore ()`  
восстановить значения по умолчанию
- `virtual void set ()`  
вернуть введенный текст и восстановить значение по умолчанию.

## Открытые атрибуты

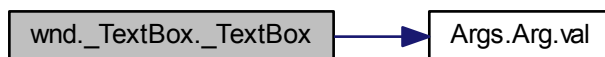
- [Arg retArg](#)  
переменная для возврата введенного значения
- `string defVal`  
умолчательное значение
- `handler txtChanged`

### 7.4.1 Конструктор(ы)

#### 7.4.1.1 `_TextBox()`

```
wnd._TextBox._TextBox (  
    Arg a ) [inline]
```

взяли значение по умолчанию. Граф вызовов:



### 7.4.2 Методы

#### 7.4.2.1 `ordNo()`

```
int wnd._TextBox.ordNo ( ) [inline]
```

возвращает порядковый номер поля ввода

## 7.4.2.2 restore()

```
virtual void wnd._TextBox.restore ( ) [inline], [virtual]
```

восстановить значения по умолчанию

Граф вызовов:



Граф вызова функции:

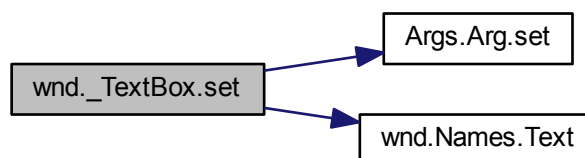


## 7.4.2.3 set()

```
virtual void wnd._TextBox.set ( ) [inline], [virtual]
```

вернуть введенный текст и восстановить значение по умолчанию.

Граф вызовов:



## 7.4.3 Данные класса

### 7.4.3.1 defVal

string wnd.\_TextBox.defVal

умолчательное значение

### 7.4.3.2 retArg

[Arg](#) wnd.\_TextBox.retArg

переменная для возврата введенного значения

### 7.4.3.3 txtChanged

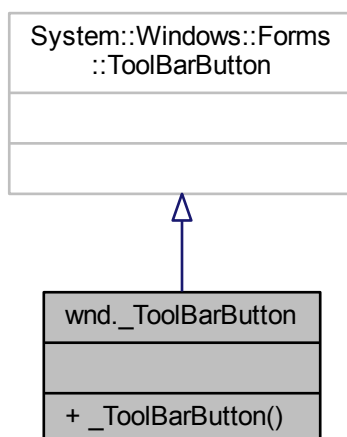
handler wnd.\_TextBox.txtChanged

Объявления и описания членов класса находятся в файле:

- [cs/wBtnLbl.cs](#)

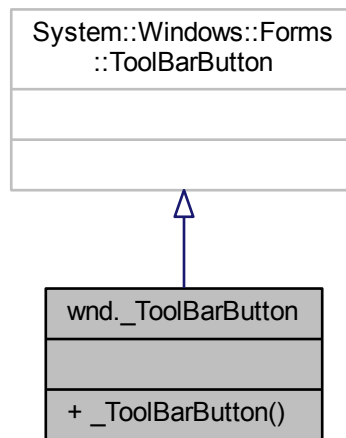
## 7.5 Класс wnd.\_ToolBarButton

Граф наследования:wnd.\_ToolBarButton:





Граф связей класса wnd.\_ToolBarButton:



Открытые члены

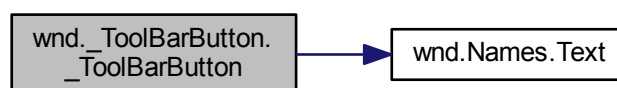
- `_ToolBarButton` (string nm)

### 7.5.1 Конструктор(ы)

#### 7.5.1.1 `_ToolBarButton()`

```
wnd._ToolBarButton._ToolBarButton (  
    string nm ) [inline]
```

Граф вызовов:



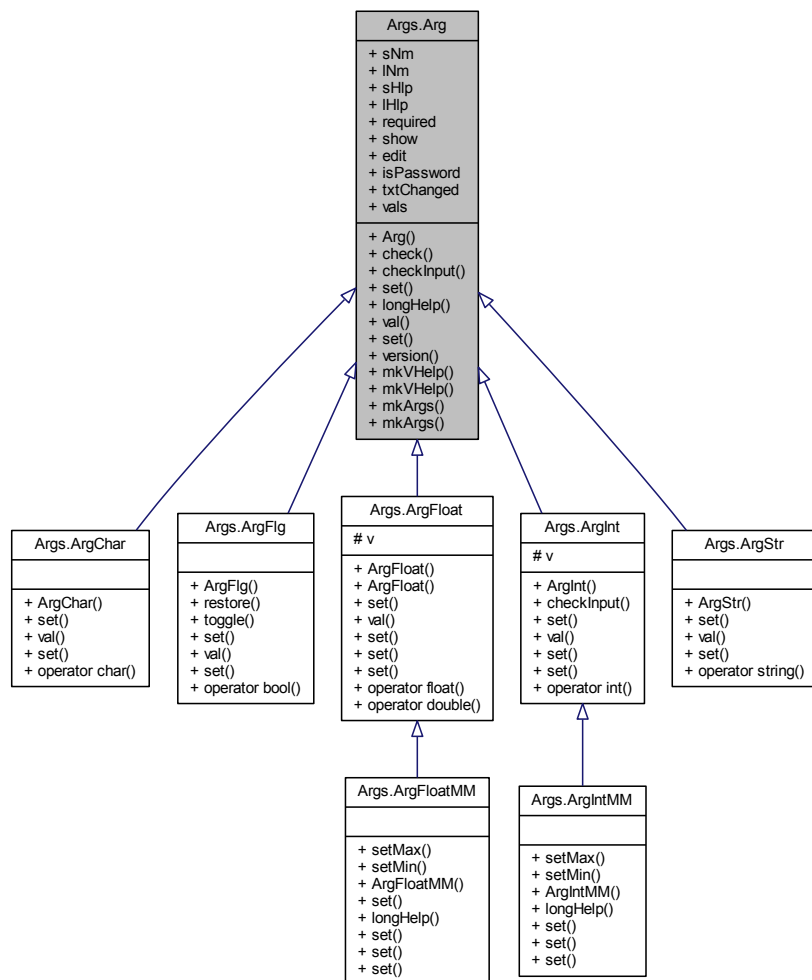
Объявления и описания членов класса находятся в файле:

- [cs/wBtnLbl.cs](#)

## 7.6 Класс Args.Arg

Родительский класс для разбора аргументов командной строки

Граф наследования: Args.Arg:



Граф связей класса Args.Arg:

Args.Arg
+ sNm + lNm + sHlp + lHlp + required + show + edit + isPassword + txtChanged + vals
+ Arg() + check() + checkInput() + set() + longHelp() + val() + set() + version() + mkVHelp() + mkVHelp() + mkArgs() + mkArgs()

### Открытые члены

- [Arg](#) (string sNm, string lNm, string sHlp, string lHlp)
- bool [check](#) (ref int i, string [] ps)
  - проверить аргументы командной строки на наличие текущего ключа
- virtual bool [checkInput](#) (object sender, string str)
- virtual void [set](#) (ref int i, string[] ps)
- virtual string [longHelp](#) ()
  - функция создает подсказку по ключу.
- virtual string [val](#) ()
  - функция переводит в строку значение ключа.
- virtual void [set](#) (string v)
  - восстанавливает из строки значение ключа

### Открытые статические члены

- static void [version](#) (out int major, out int minor, out int build)
  - версия библиотеки
- static void [mkVHelp](#) (string forWhat, string add, bool verbose, params [Arg](#) [] ars)
  - выдача подсказки

- static void `mkVHelp` (string forWhat, string ver, string after, bool verbose, params `Arg[]` ars)  
собственно выдача подсказки с версией приложения
- static `Arg []` `mkArgs` (params `Arg[]` ars)  
генерация текста для подсказки и для вывода в журнал запуска
- static string `mkArgs` (`Arg[]` ars, int ip)

## Открытые атрибуты

- string `sNm`  
ключ
- string `lNm`  
синоним ключа
- string `sHlp`  
паоаметр ключа
- string `lHlp`  
описание ключа
- bool `required`  
обязательный ключ - не делать квадратных скобок, по умолчанию - делать
- bool `show`  
показывать ли ключ в строке вызова, по умолчанию показывать.
- bool `edit`  
флаг редактирования в диалоговом окне
- bool `isPassword`  
в поле вводят пароль (если true текст шифруется '\*')
- `handler txtChanged`  
обработчик проверки значения в поле диалогового окна
- string [] `vals`

### 7.6.1 Подробное описание

Родительский класс для разбора аргументов командной строки

### 7.6.2 Конструктор(ы)

#### 7.6.2.1 Arg()

```
Args.Arg.Arg (
    string sNm,
    string lNm,
    string sHlp,
    string lHlp ) [inline]
```

< можно редактировать

< шифровать при вводе

## Аргументы

sNm	ключ
INm	синоним
sHp	параметр ключа
INp	описание

Граф вызовов:



### 7.6.3 Методы

#### 7.6.3.1 check()

```
bool Args.Arg.check (  
    ref int i,  
    string [] ps ) [inline]
```

проверить аргументы командной строки на наличие текущего ключа

функция задает текущей переменной [Arg](#) правильное значение из списка аргументов командной строки и сдвигает текущий аргумент. Используется позднее связывание, так как для разных типов ключей выполняются различные преобразования, включая проверку на диапазон допустимых значений.

## Аргументы

i	индекс текущего аргумента
ps	аргументы командной строки

Граф вызова функции:



### 7.6.3.2 checkInput()

```
virtual bool Args.Arg.checkInput (
    object sender,
    string str ) [inline], [virtual]
```

Переопределяется в [Args.ArgInt](#).

Граф вызова функции:



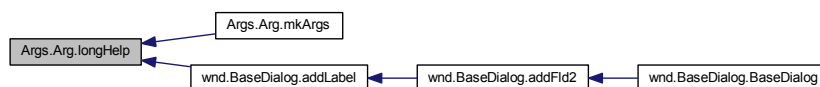
### 7.6.3.3 longHelp()

```
virtual string Args.Arg.longHelp ( ) [inline], [virtual]
```

функция создает подсказку по ключу.

Переопределяется в [Args.ArgIntMM](#) и [Args.ArgFloatMM](#).

Граф вызова функции:

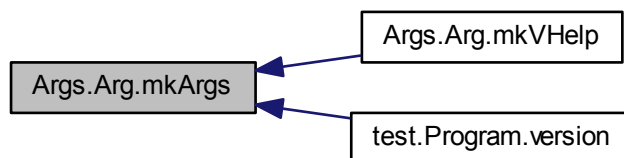


## 7.6.3.4 mkArgs() [1/2]

```
static Arg [] Args.Arg.mkArgs (  
    params Arg [] ars ) [inline], [static]
```

генерация текста для подсказки и для вывода в журнал запуска

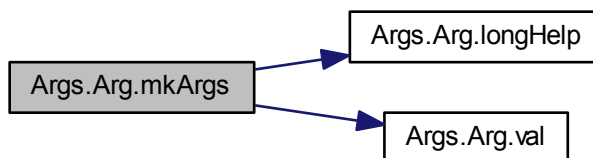
Граф вызова функции:



## 7.6.3.5 mkArgs() [2/2]

```
static string Args.Arg.mkArgs (  
    Arg [] ars,  
    int ip ) [inline], [static]
```

Граф вызовов:



## 7.6.3.6 mkVHelp() [1/2]

```
static void Args.Arg.mkVHelp (
    string forWhat,
    string add,
    bool verbose,
    params Arg [] ars ) [inline], [static]
```

выдача подсказки

Граф вызовов:



Граф вызова функции:



## 7.6.3.7 mkVHelp() [2/2]

```
static void Args.Arg.mkVHelp (
    string forWhat,
    string ver,
    string after,
    bool verbose,
    params Arg [] ars ) [inline], [static]
```

собственно выдача подсказки с версией приложения

Аргументы

forWhat	краткое описание приложения
ver	версия приложения
after	дополнительная часть строки usage тут можно нестандартным образом показать ключи
verbose	добавлять ли список синонимов в подсказку



Граф вызовов:



#### 7.6.3.8 set() [1/2]

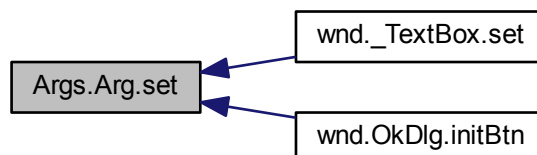
```
virtual void Args.Arg.set (  
    ref int i,  
    string [] ps ) [inline], [virtual]
```

Аргументы

ps	задать значение ключа из текущего аргумента
----	---

Переопределяется в [Args.ArgIntMM](#), [Args.ArgInt](#), [Args.ArgFloatMM](#), [Args.ArgFloat](#), [Args.ArgChar](#), [Args.ArgStr](#) и [Args.ArgFlg](#).

Граф вызова функции:



#### 7.6.3.9 set() [2/2]

```
virtual void Args.Arg.set (  
    string v ) [inline], [virtual]
```

восстанавливает из строки значение ключа

Переопределяется в [Args.ArgIntMM](#), [Args.ArgInt](#), [Args.ArgFloatMM](#), [Args.ArgFloat](#), [Args.ArgChar](#), [Args.ArgStr](#) и [Args.ArgFlg](#).

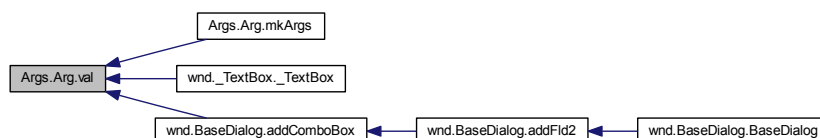
## 7.6.3.10 val()

```
virtual string Args.Arg.val ( ) [inline], [virtual]
```

функция переводит в строку значение ключа.

Переопределяется в [Args.ArgInt](#), [Args.ArgFloat](#), [Args.ArgChar](#), [Args.ArgStr](#) и [Args.ArgFlg](#).

Граф вызова функции:

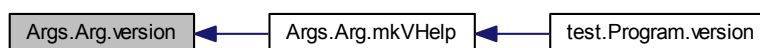


## 7.6.3.11 version()

```
static void Args.Arg.version (
    out int major,
    out int minor,
    out int build ) [inline], [static]
```

версия библиотеки

Граф вызова функции:



## 7.6.4 Данные класса

## 7.6.4.1 edit

```
bool Args.Arg.edit
```

флаг редактирования в диалоговом окне

## 7.6.4.2 isPassword

bool Args.Arg.isPassword

в поле вводят пароль (если true текст шифруется '\*')

## 7.6.4.3 lHlp

string Args.Arg.lHlp

описание ключа

## 7.6.4.4 lNm

string Args.Arg.lNm

синоним ключа

## 7.6.4.5 required

bool Args.Arg.required

обязательный ключ - не делать квадратных скобок, по умолчанию - делать

## 7.6.4.6 sHlp

string Args.Arg.sHlp

паоаметр ключа

## 7.6.4.7 show

bool Args.Arg.show

показывать ли ключ в строке вызова, по умолчанию показывать.

#### 7.6.4.8 sNm

string Args.Arg.sNm

ключ

#### 7.6.4.9 txtChanged

[handler](#) Args.Arg.txtChanged

обработчик проверки значения в поле диалогового окна

#### 7.6.4.10 vals

string [] Args.Arg.vals

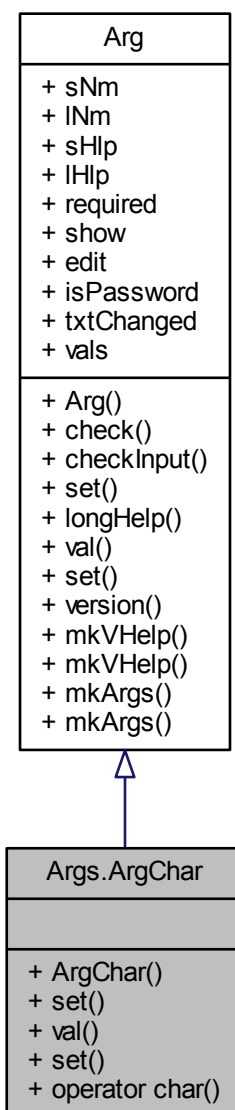
Объявления и описания членов класса находятся в файле:

- [cs/args.cs](#)

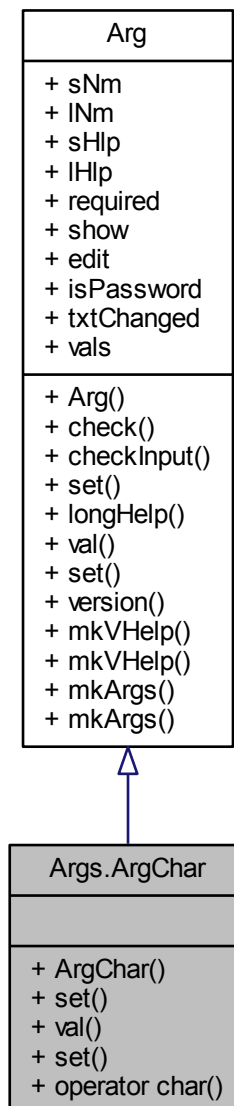
## 7.7 Класс Args.ArgChar

ключ типа СИМВОЛ

Граф наследования:Args.ArgChar:



Граф связей класса Args.ArgChar:



### Открытые члены

- **ArgChar** (char v, string sNm, string lNm, string lHlp=null, string sHlp=null, string[] vals=null)
- override void **set** (ref int i, string[] ps)
- override string **val** ()
  - функция переводит в строку значение ключа.
- override void **set** (string v)
  - восстанавливает из строки значение ключа

### Открытые статические члены

- static implicit **operator char** (ArgChar p)

## Дополнительные унаследованные члены

## 7.7.1 Подробное описание

ключ типа символ

## 7.7.2 Конструктор(ы)

## 7.7.2.1 ArgChar()

```
Args.ArgChar.ArgChar (
    char v,
    string sNm,
    string lNm,
    string lHlp = null,
    string sHlp = null,
    string [] vals = null ) [inline]
```

## Аргументы

v	умолчательное значение
sNm	идентификатор параметра
lNm	длинное имя
lHlp	длинный хелп
sHlp	короткий идентификатор для командной строки

## 7.7.3 Методы

## 7.7.3.1 operator char()

```
static implicit Args.ArgChar.operator char (
    ArgChar p ) [inline], [static]
```

## 7.7.3.2 set() [1/2]

```
override void Args.ArgChar.set (
    ref int i,
    string [] ps ) [inline], [virtual]
```

Переопределяет метод предка [Args.Arg](#).

### 7.7.3.3 set() [2/2]

```
override void Args.ArgChar.set (  
    string v ) [inline], [virtual]
```

восстанавливает из строки значение ключа

Переопределяет метод предка [Args.Arg](#).

### 7.7.3.4 val()

```
override string Args.ArgChar.val ( ) [inline], [virtual]
```

функция переводит в строку значение ключа.

Переопределяет метод предка [Args.Arg](#).

Объявления и описания членов класса находятся в файле:

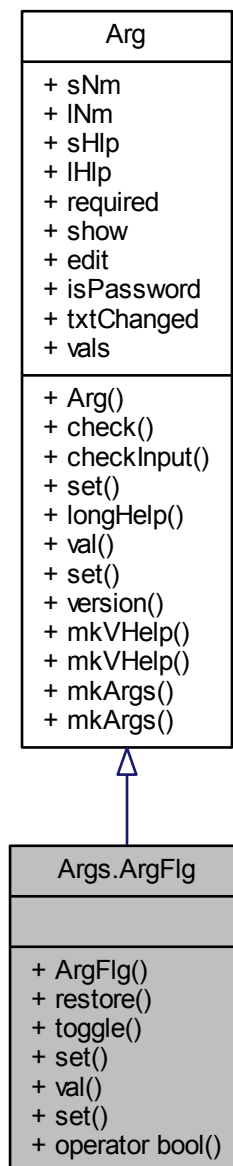
- [cs/args.cs](#)

## 7.8 Класс Args.ArgFlg

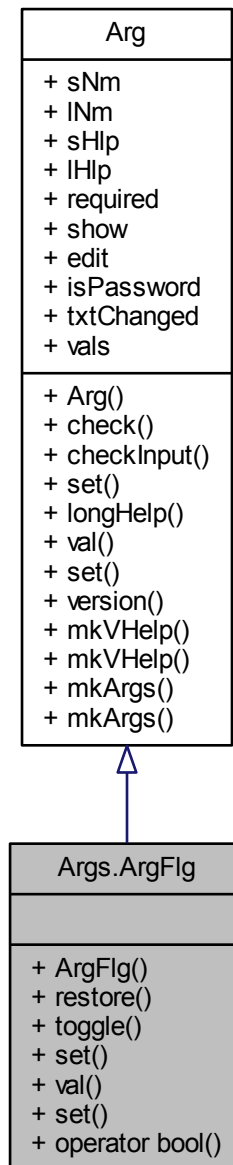
ключ логического типа (иначе - флаг)



Граф наследования:Args.ArgFlg:



Граф связей класса Args.ArgFlg:



### Открытые члены

- **ArgFlg** (bool v, string sNm, string INm, string IHlp=null, string sHlp=null)
- void **restore** ()
- void **toggle** ()
- override void **set** (ref int i, string[] ps)
  - переключение флага, не сдвигает индекс i текущего аргумента.
- override string **val** ()
  - функция переводит в строку значение ключа.
- override void **set** (string v)
  - восстанавливает из строки значение ключа

Открытые статические члены

- static implicit operator bool (ArgFlg p)

Дополнительные унаследованные члены

### 7.8.1 Подробное описание

ключ логического типа (иначе - флаг)

### 7.8.2 Конструктор(ы)

#### 7.8.2.1 ArgFlg()

```
Args.ArgFlg.ArgFlg (
    bool v,
    string sNm,
    string lNm,
    string lHlp = null,
    string sHlp = null ) [inline]
```

Аргументы

v	начальное значение ключа
---	--------------------------

### 7.8.3 Методы

#### 7.8.3.1 operator bool()

```
static implicit Args.ArgFlg.operator bool (
    ArgFlg p ) [inline], [static]
```

#### 7.8.3.2 restore()

```
void Args.ArgFlg.restore ( ) [inline]
```

< восстановить флаг

### 7.8.3.3 set() [1/2]

```
override void Args.ArgFlg.set (  
    ref int i,  
    string [] ps ) [inline], [virtual]
```

переключение флага, не сдвигает индекс *i* текущего аргумента.

Переопределяет метод предка [Args.Arg](#).

### 7.8.3.4 set() [2/2]

```
override void Args.ArgFlg.set (  
    string v ) [inline], [virtual]
```

восстанавливает из строки значение ключа

Переопределяет метод предка [Args.Arg](#).

### 7.8.3.5 toggle()

```
void Args.ArgFlg.toggle ( ) [inline]
```

< переключить флаг

### 7.8.3.6 val()

```
override string Args.ArgFlg.val ( ) [inline], [virtual]
```

функция переводит в строку значение ключа.

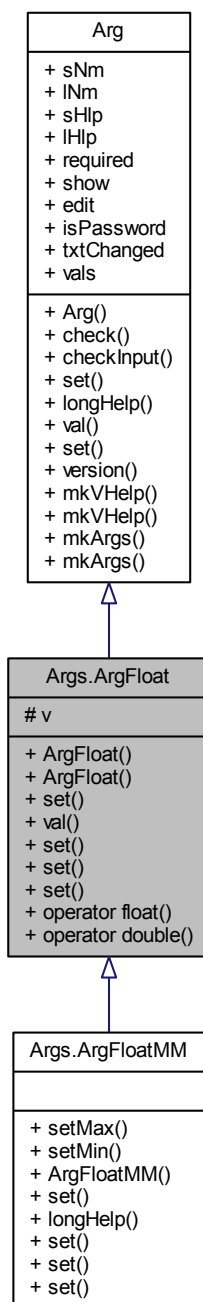
Переопределяет метод предка [Args.Arg](#).

Объявления и описания членов класса находятся в файле:

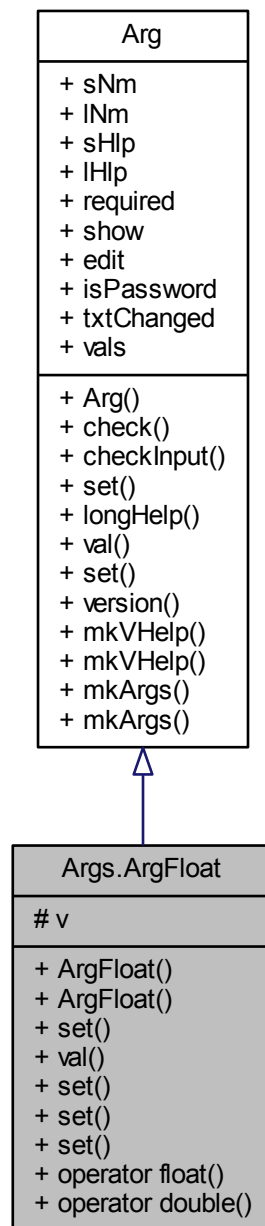
- [cs/args.cs](#)

## 7.9 Класс Args.ArgFloat

Граф наследования:Args.ArgFloat:



Граф связей класса Args.ArgFloat:



### Открытые члены

- ArgFloat (float v, string sNm, string lNm, string lHlp=null, string sHlp=null)
- ArgFloat (double v, string sNm, string lNm, string lHlp=null, string sHlp=null)
- override void set (ref int i, string [] ps)
- override string val ()
  - функция переводит в строку значение ключа.
- override void set (string v)

восстанавливает из строки значение ключа

- virtual void `set` (double `v`)
- virtual void `set` (float `v`)

#### Открытые статические члены

- static implicit `operator float` (`ArgFloat p`)
- static implicit `operator double` (`ArgFloat p`)

#### Защищенные данные

- double `v`

#### Дополнительные унаследованные члены

### 7.9.1 Конструктор(ы)

#### 7.9.1.1 `ArgFloat()` [1/2]

```
Args.ArgFloat.ArgFloat (  
    float v,  
    string sNm,  
    string lNm,  
    string lHlp = null,  
    string sHlp = null ) [inline]
```

#### 7.9.1.2 `ArgFloat()` [2/2]

```
Args.ArgFloat.ArgFloat (  
    double v,  
    string sNm,  
    string lNm,  
    string lHlp = null,  
    string sHlp = null ) [inline]
```

### 7.9.2 Методы

#### 7.9.2.1 `operator double()`

```
static implicit Args.ArgFloat.operator double (  
    ArgFloat p ) [inline], [static]
```

### 7.9.2.2 operator float()

```
static implicit Args.ArgFloat.operator float (  
    ArgFloat p ) [inline], [static]
```

### 7.9.2.3 set() [1/4]

```
override void Args.ArgFloat.set (  
    ref int i,  
    string [] ps ) [inline], [virtual]
```

разбор аргументов командной строки2

Переопределяет метод предка [Args.Arg](#).

Переопределяется в [Args.ArgFloatMM](#).

### 7.9.2.4 set() [2/4]

```
override void Args.ArgFloat.set (  
    string v ) [inline], [virtual]
```

восстанавливает из строки значение ключа

Переопределяет метод предка [Args.Arg](#).

Переопределяется в [Args.ArgFloatMM](#).

### 7.9.2.5 set() [3/4]

```
virtual void Args.ArgFloat.set (  
    double v ) [inline], [virtual]
```

Переопределяется в [Args.ArgFloatMM](#).

### 7.9.2.6 set() [4/4]

```
virtual void Args.ArgFloat.set (  
    float v ) [inline], [virtual]
```

Переопределяется в [Args.ArgFloatMM](#).



## 7.9.2.7 val()

```
override string Args.ArgFloat.val ( ) [inline], [virtual]
```

функция переводит в строку значение ключа.

Переопределяет метод предка [Args.Arg](#).

## 7.9.3 Данные класса

## 7.9.3.1 v

```
double Args.ArgFloat.v [protected]
```

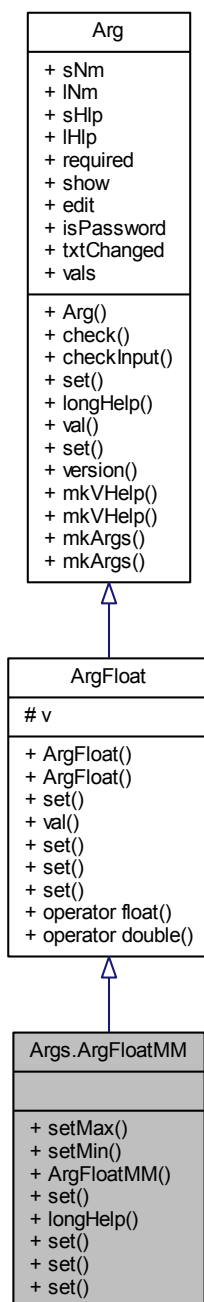
Объявления и описания членов класса находятся в файле:

- [cs/args.cs](#)

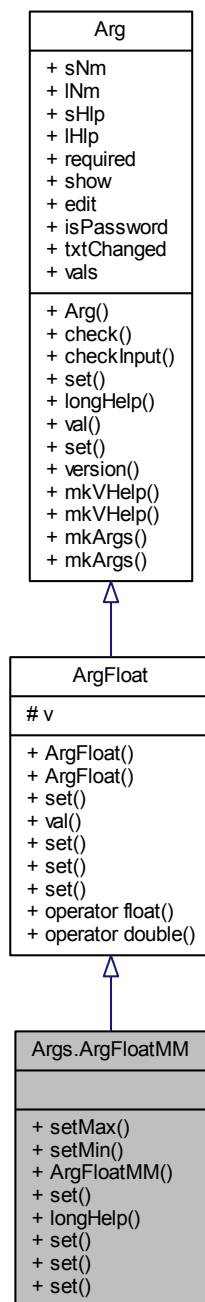
## 7.10 Класс Args.ArgFloatMM

ключ вещественного типа, есть возможность ограничить максимальное и минимальное значение

Граф наследования: Args.ArgFloatMM:



Граф связей класса Args.ArgFloatMM:



### Открытые члены

- void `setMax` (double max)
- void `setMin` (double min)
- `ArgFloatMM` (double v, string sNm, string lNm, string lHlp=null, string sHlp=null)
- override void `set` (ref int i, string[] ps)
- override string `longHelp` ()

- функция создает подсказку по ключу.
- override void `set` (string `v`)  
восстанавливает из строки значение ключа
  - override void `set` (double `v`)
  - override void `set` (float `v`)

## Дополнительные унаследованные члены

### 7.10.1 Подробное описание

ключ вещественного типа, есть возможность ограничить максимальное и минимальное значение

### 7.10.2 Конструктор(ы)

#### 7.10.2.1 ArgFloatMM()

```
Args.ArgFloatMM.ArgFloatMM (  
    double v,  
    string sNm,  
    string lNm,  
    string lHlp = null,  
    string sHlp = null ) [inline]
```

### 7.10.3 Методы

#### 7.10.3.1 longHelp()

```
override string Args.ArgFloatMM.longHelp ( ) [inline], [virtual]
```

функция создает подсказку по ключу.

Переопределяет метод предка [Args.Arg](#).

## 7.10.3.2 set() [1/4]

```
override void Args.ArgFloatMM.set (  
    ref int i,  
    string [] ps ) [inline], [virtual]
```

Переопределяет метод предка [Args.ArgFloat](#).

Граф вызова функции:



## 7.10.3.3 set() [2/4]

```
override void Args.ArgFloatMM.set (  
    string v ) [inline], [virtual]
```

восстанавливает из строки значение ключа

Переопределяет метод предка [Args.ArgFloat](#).

## 7.10.3.4 set() [3/4]

```
override void Args.ArgFloatMM.set (  
    double v ) [inline], [virtual]
```

Переопределяет метод предка [Args.ArgFloat](#).

## 7.10.3.5 set() [4/4]

```
override void Args.ArgFloatMM.set (  
    float v ) [inline], [virtual]
```

Переопределяет метод предка [Args.ArgFloat](#).

### 7.10.3.6 setMax()

```
void Args.ArgFloatMM.setMax (  
    double max ) [inline]
```

Граф вызова функции:



### 7.10.3.7 setMin()

```
void Args.ArgFloatMM.setMin (  
    double min ) [inline]
```

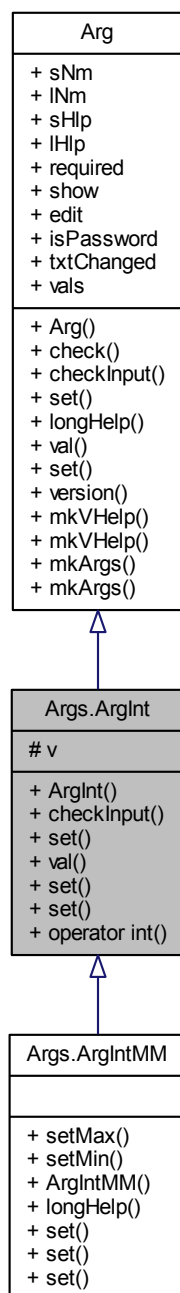
Объявления и описания членов класса находятся в файле:

- [cs/args.cs](#)

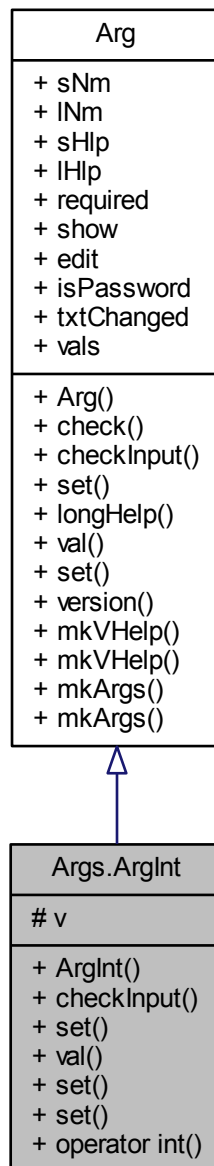
## 7.11 Класс Args.ArgInt

ключ целого типа

Граф наследования:Args.ArgInt:



Граф связей класса Args.ArgInt:



### Открытые члены

- `ArgInt` (int `v`, string `sNm`, string `lNm`, string `lHlp`=null, string `sHlp`=null)
- override bool `checkInput` (object sender, string str)
- override void `set` (ref int i, string[] ps)
- override string `val` ()
  - функция переводит в строку значение ключа.
- override void `set` (string `v`)
  - восстанавливает из строки значение ключа
- virtual void `set` (int `v`)



## Открытые статические члены

- static implicit operator int (ArgInt p)

## Защищенные данные

- int v

## Дополнительные унаследованные члены

## 7.11.1 Подробное описание

ключ целого типа

## 7.11.2 Конструктор(ы)

## 7.11.2.1 ArgInt()

```
Args.ArgInt.ArgInt (  
    int v,  
    string sNm,  
    string lNm,  
    string lHlp = null,  
    string sHlp = null ) [inline]
```

## 7.11.3 Методы

## 7.11.3.1 checkInput()

```
override bool Args.ArgInt.checkInput (  
    object sender,  
    string str ) [inline], [virtual]
```

Переопределяет метод предка [Args.Arg](#).

## 7.11.3.2 operator int()

```
static implicit Args.ArgInt.operator int (  
    ArgInt p ) [inline], [static]
```

7.11.3.3 `set()` [1/3]

```
override void Args.ArgInt.set (  
    ref int i,  
    string [] ps ) [inline], [virtual]
```

Переопределяет метод предка [Args.Arg](#).

Переопределяется в [Args.ArgIntMM](#).

7.11.3.4 `set()` [2/3]

```
override void Args.ArgInt.set (  
    string v ) [inline], [virtual]
```

восстанавливает из строки значение ключа

Переопределяет метод предка [Args.Arg](#).

Переопределяется в [Args.ArgIntMM](#).

7.11.3.5 `set()` [3/3]

```
virtual void Args.ArgInt.set (  
    int v ) [inline], [virtual]
```

Переопределяется в [Args.ArgIntMM](#).

7.11.3.6 `val()`

```
override string Args.ArgInt.val ( ) [inline], [virtual]
```

функция переводит в строку значение ключа.

Переопределяет метод предка [Args.Arg](#).

## 7.11.4 Данные класса

7.11.4.1 v

```
int Args.ArgInt.v [protected]
```

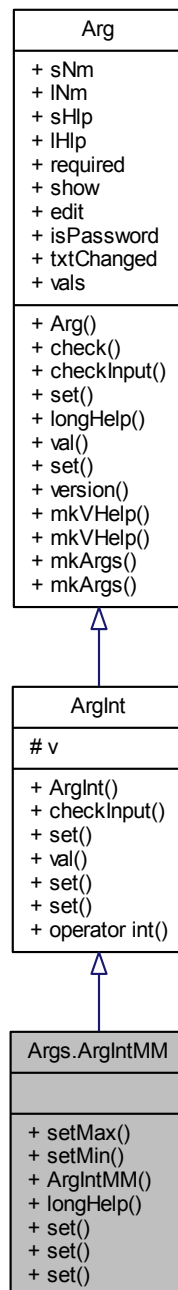
Объявления и описания членов класса находятся в файле:

- [cs/args.cs](#)

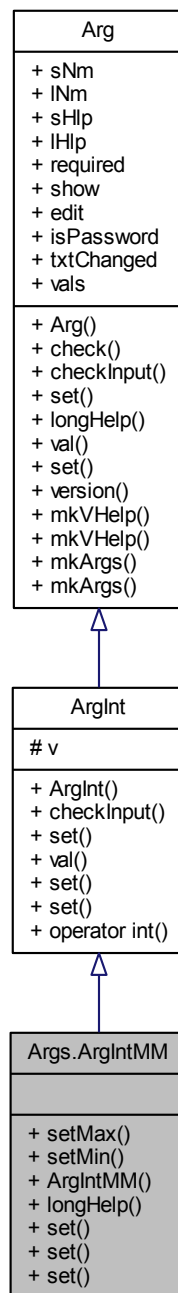
## 7.12 Класс Args.ArgIntMM

ключ целого типа, есть возможность ограничить максимальное и минимальное значение

Граф наследования: Args.ArgIntMM:



Граф связей класса Args.ArgIntMM:



### Открытые члены

- void `setMax` (int max)
- void `setMin` (int min)
- `ArgIntMM` (int v, string sNm, string lNm, string lHlp=null, string sHlp=null)
- override string `longHelp` ()  
функция создает подсказку по ключу.

- override void `set` (ref int i, string[] ps)
- override void `set` (int v)
- override void `set` (string v)

восстанавливает из строки значение ключа

## Дополнительные унаследованные члены

### 7.12.1 Подробное описание

ключ целого типа, есть возможность ограничить максимальное и минимальное значение

### 7.12.2 Конструктор(ы)

#### 7.12.2.1 ArgIntMM()

```
Args.ArgIntMM.ArgIntMM (
    int v,
    string sNm,
    string lNm,
    string lHlp = null,
    string sHlp = null ) [inline]
```

### 7.12.3 Методы

#### 7.12.3.1 longHelp()

```
override string Args.ArgIntMM.longHelp ( ) [inline], [virtual]
```

функция создает подсказку по ключу.

Переопределяет метод предка [Args.Arg](#).

#### 7.12.3.2 set() [1/3]

```
override void Args.ArgIntMM.set (
    ref int i,
    string [] ps ) [inline], [virtual]
```

Переопределяет метод предка [Args.ArgInt](#).

## 7.12.3.3 set() [2/3]

```
override void Args.ArgIntMM.set (  
    int v ) [inline], [virtual]
```

Переопределяет метод предка [Args.ArgInt](#).

## 7.12.3.4 set() [3/3]

```
override void Args.ArgIntMM.set (  
    string v ) [inline], [virtual]
```

восстанавливает из строки значение ключа

Переопределяет метод предка [Args.ArgInt](#).

## 7.12.3.5 setMax()

```
void Args.ArgIntMM.setMax (  
    int max ) [inline]
```

Граф вызова функции:



## 7.12.3.6 setMin()

```
void Args.ArgIntMM.setMin (  
    int min ) [inline]
```

Граф вызова функции:



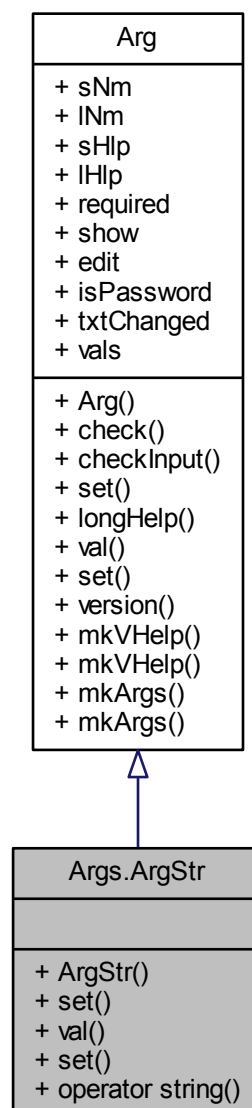
Объявления и описания членов класса находятся в файле:

- [cs/args.cs](#)

## 7.13 Класс Args.ArgStr

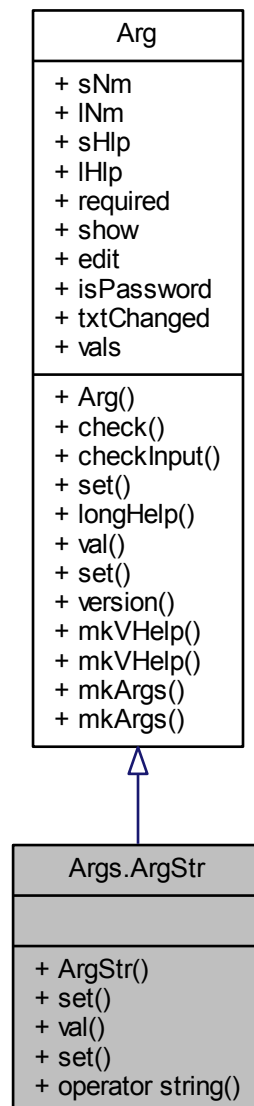
ключ типа строка

Граф наследования: Args.ArgStr:





Граф связей класса Args.ArgStr:



#### Открытые члены

- `ArgStr` (string v, string sNm, string lNm, string lHlp=null, string sHlp=null, string[] vals=null)
- override void `set` (ref int i, string[] ps)
- override string `val` ()
  - функция переводит в строку значение ключа.
- override void `set` (string v)
  - восстанавливает из строки значение ключа

#### Открытые статические члены

- static implicit `operator string` (`ArgStr` p)

## Дополнительные унаследованные члены

### 7.13.1 Подробное описание

ключ типа строка

### 7.13.2 Конструктор(ы)

#### 7.13.2.1 ArgStr()

```
Args.ArgStr.ArgStr (  
    string v,  
    string sNm,  
    string lNm,  
    string lHlp = null,  
    string sHlp = null,  
    string [] vals = null ) [inline]
```

### 7.13.3 Методы

#### 7.13.3.1 operator string()

```
static implicit Args.ArgStr.operator string (  
    ArgStr p ) [inline], [static]
```

#### 7.13.3.2 set() [1/2]

```
override void Args.ArgStr.set (  
    ref int i,  
    string [] ps ) [inline], [virtual]
```

Переопределяет метод предка [Args.Arg](#).

#### 7.13.3.3 set() [2/2]

```
override void Args.ArgStr.set (  
    string v ) [inline], [virtual]
```

восстанавливает из строки значение ключа

Переопределяет метод предка [Args.Arg](#).

## 7.13.3.4 val()

```
override string Args.ArgStr.val ( ) [inline], [virtual]
```

функция переводит в строку значение ключа.

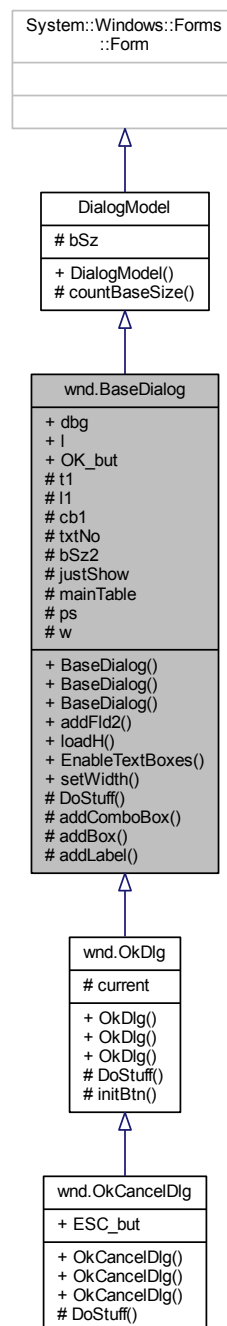
Переопределяет метод предка [Args.Arg](#).

Объявления и описания членов класса находятся в файле:

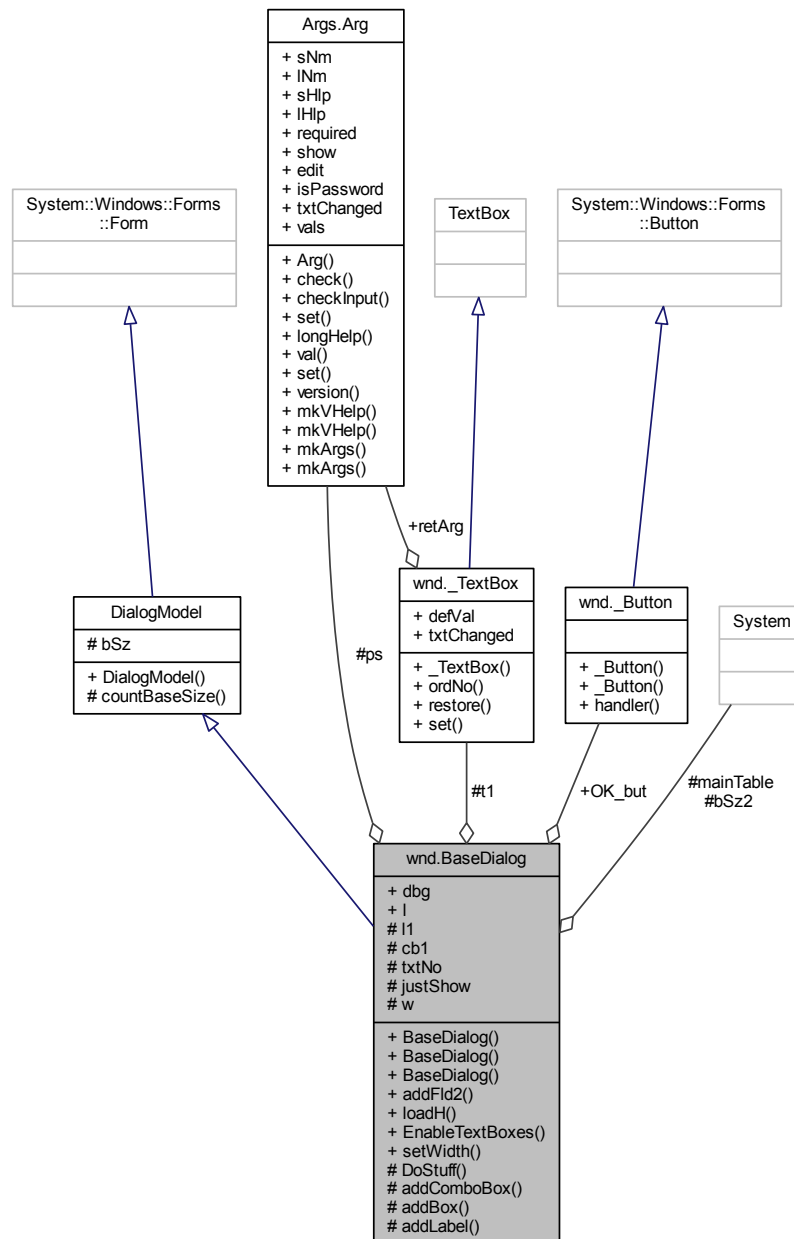
- [cs/args.cs](#)

## 7.14 Класс wnd.BaseDialog

Граф наследования:wnd.BaseDialog:



Граф связей класса wnd.BaseDialog:



### Открытые члены

- `BaseDialog` (string name, Loger ll=null)
- `BaseDialog` (string name, Loger ll, params `Arg[] ps`)  
этот конструктор для создания нескольких полей ввода
- `BaseDialog` (string name, `Arg p`, Loger ll=null)
- `void addFld2` (int n, `Arg par`)  
метод добавляет в окно новое поле ввода
- `void loadH` (object sender, System.EventArgs e)
- `void EnableTextBoxes` (bool b, string name)
- `void setWidth` ()

## Открытые атрибуты

- bool `dbg`
- `Logger l`
- `_Button OK_but`

## Защищенные члены

- abstract void `DoStuff ()`
- void `addComboBox (int n, Arg par, Panel p, string[] values)`
- void `addBox (int n, Arg par, Panel p)`
- void `addLabel (int n, Arg par, Control p)`

## Защищенные данные

- `_TextBox t1`  
поле ввода
- `Label l1`  
метка для поля ввода
- `ComboBox cb1`
- `int txtNo`  
здесь записано кол-во полей ввода
- `System.Drawing.Size bSz2`
- `bool justShow = false`  
отмена редактирования
- `System.Windows.Forms.TableLayoutPanel mainTable`
- `Arg [] ps`
- `int w`

## 7.14.1 Конструктор(ы)

## 7.14.1.1 BaseDialog() [1/3]

```
wnd.BaseDialog.BaseDialog (
    string name,
    Logger ll = null ) [inline]
```

ширина для поля ввода

## Аргументы

name	для наследования функции addFld в окно для ввода многих строчек имя окна ввода
ll	журнал для отладки окна

Граф вызовов:



#### 7.14.1.2 BaseDialog() [2/3]

```

wnd.BaseDialog.BaseDialog (
    string name,
    Loger ll,
    params Arg [] ps ) [inline]
  
```

Этот конструктор для создания нескольких полей ввода

Аргументы

name	заголовок окна
ps	список переменных для создания полей ввода

Граф вызовов:



#### 7.14.1.3 BaseDialog() [3/3]

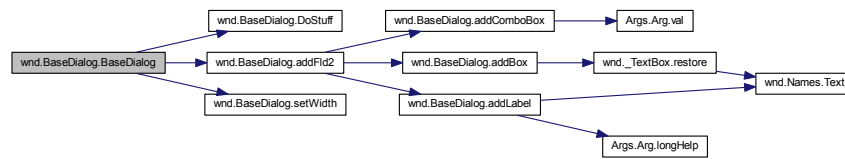
```

wnd.BaseDialog.BaseDialog (
    string name,
    Arg p,
    Loger ll = null ) [inline]
  
```

Аргументы

name	заголовок окна
p	переменная для ввода
ll	журнал для отладки

Граф вызовов:



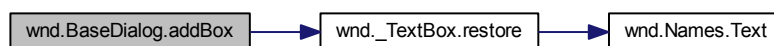
## 7.14.2 Методы

### 7.14.2.1 addBox()

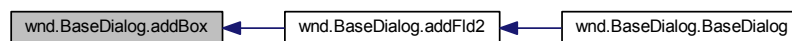
```

void wnd.BaseDialog.addBox (
    int n,
    Arg par,
    Panel p ) [inline], [protected]
  
```

Граф вызовов:



Граф вызова функции:

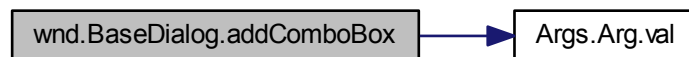




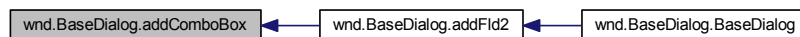
## 7.14.2.2 addComboBox()

```
void wnd.BaseDialog.addComboBox (
    int n,
    Arg par,
    Panel p,
    string [] values ) [inline], [protected]
```

Граф вызовов:



Граф вызова функции:

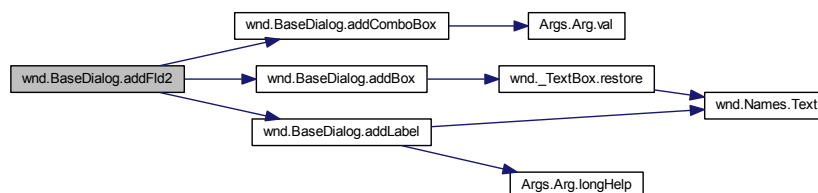


## 7.14.2.3 addFld2()

```
void wnd.BaseDialog.addFld2 (
    int n,
    Arg par ) [inline]
```

метод добавляет в окно новое поле ввода

панель содержит метку и поле ввода Граф вызовов:



Граф вызова функции:

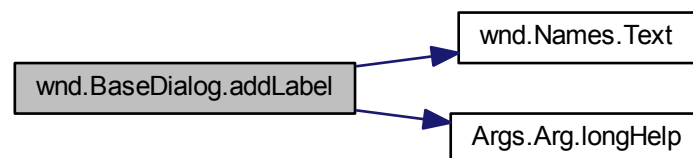


#### 7.14.2.4 addLabel()

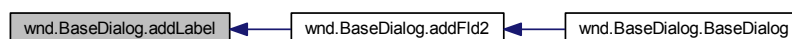
```

void wnd.BaseDialog.addLabel (
    int n,
    Arg par,
    Control p ) [inline], [protected]
  
```

создать новую строку Граф вызовов:



Граф вызова функции:



## 7.14.2.5 DoStuff()

```
abstract void wnd.BaseDialog.DoStuff ( ) [protected], [pure virtual]
```

Замещается в [wnd.OkCancelDlg](#) и [wnd.OkDlg](#).

Граф вызова функции:



## 7.14.2.6 EnableTextBoxes()

```
void wnd.BaseDialog.EnableTextBoxes (
    bool b,
    string name ) [inline]
```

Граф вызова функции:



## 7.14.2.7 loadH()

```
void wnd.BaseDialog.loadH (
    object sender,
    System.EventArgs e ) [inline]
```

Граф вызова функции:



7.14.2.8 `setWidth()`

```
void wnd.BaseDialog.setWidth ( ) [inline]
```

Граф вызова функции:



## 7.14.3 Данные класса

7.14.3.1 `bSz2`

```
System.Drawing.Size wnd.BaseDialog.bSz2 [protected]
```

это размер поля ввода, в два раза шире стандартной кнопки.

7.14.3.2 `cb1`

```
ComboBox wnd.BaseDialog.cb1 [protected]
```

7.14.3.3 `dbg`

```
bool wnd.BaseDialog.dbg
```

7.14.3.4 `justShow`

```
bool wnd.BaseDialog.justShow = false [protected]
```

отмена редактирования

## 7.14.3.5 l

Logger wnd.BaseDialog.l

## 7.14.3.6 l1

Label wnd.BaseDialog.l1 [protected]

метка для поля ввода

## 7.14.3.7 mainTable

System.Windows.Forms.TableLayoutPanel wnd.BaseDialog.mainTable [protected]

## 7.14.3.8 OK\_but

[\\_Button](#) wnd.BaseDialog.OK\_but

## 7.14.3.9 ps

[Arg \[\]](#) wnd.BaseDialog.ps [protected]

## 7.14.3.10 t1

[\\_TextBox](#) wnd.BaseDialog.t1 [protected]

поле ввода

## 7.14.3.11 txtNo

int wnd.BaseDialog.txtNo [protected]

здесь записано кол-во полей ввода

### 7.14.3.12 w

int wnd.BaseDialog.w [protected]

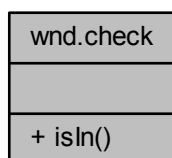
Объявления и описания членов класса находятся в файле:

- [cs/wDlgBase.cs](#)

## 7.15 Класс wnd.check

класс для разного рода проверок

Граф связей класса wnd.check:



### Открытые статические члены

- static bool [isIn](#) (int x, int y, Size clientArea)  
проверка принадлежит ли точка прямоугольнику

### 7.15.1 Подробное описание

класс для разного рода проверок

### 7.15.2 Методы

#### 7.15.2.1 isIn()

```
static bool wnd.check.isIn (  
    int x,  
    int y,  
    Size clientArea ) [inline], [static]
```

проверка принадлежит ли точка прямоугольнику

Аргументы

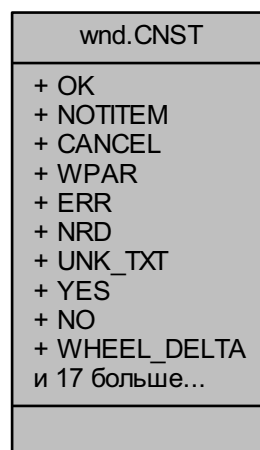
x	икс точки
y	игрек точки
clientArea	размер контрола

Объявления и описания членов класса находятся в файле:

- [cs/ut.cs](#)

## 7.16 Структура wnd.CNST

Граф связей класса wnd.CNST:



Открытые атрибуты

- const int `OK` = 0
- const int `NOTITEM` = 1
- const int `CANCEL` = 10
- const int `WPAR` = 3
- const int `ERR` = 5
- const int `NRD` = 6
- const string `UNK_TXT` = ""
- const string `YES` = "yes"
- const string `NO` = "no"
- const int `WHEEL_DELTA` = 120
- const string `_NRD` = "is not ready!"
- const string `_OK` = "OK"
- const string `_ESC` = "Cancel"

- `const string _EXIT = "Close"`
- `const string _ADD = "Add"`
- `const string _DEL = "Delete"`
- `const string _EDIT = "Edit"`
- `const string _REF = "Refresh"`
- `const string _FLT = "Filter"`
- `const string _FND = "Find"`
- `const string _ORD = "Order"`
- `const string _EXP = "Export"`
- `const string NULL = "<none>"`
- `const string NSF = "<missing at server>"`
- `const string stop0 = "*stop*"`
- `const string stop1 = "_STOP_"`
- `const string stop2 = "Continue"`

### 7.16.1 Данные класса

#### 7.16.1.1 \_ADD

```
const string wnd.CNST._ADD = "Add"
```

#### 7.16.1.2 \_DEL

```
const string wnd.CNST._DEL = "Delete"
```

#### 7.16.1.3 \_EDIT

```
const string wnd.CNST._EDIT = "Edit"
```

#### 7.16.1.4 \_ESC

```
const string wnd.CNST._ESC = "Cancel"
```

#### 7.16.1.5 \_EXIT

```
const string wnd.CNST._EXIT = "Close"
```



## 7.16.1.6 \_EXP

```
const string wnd.CNST._EXP = "Export"
```

## 7.16.1.7 \_FLT

```
const string wnd.CNST._FLT = "Filter"
```

## 7.16.1.8 \_FND

```
const string wnd.CNST._FND = "Find"
```

## 7.16.1.9 \_NRD

```
const string wnd.CNST._NRD = "is not ready!"
```

## 7.16.1.10 \_OK

```
const string wnd.CNST._OK = "OK"
```

## 7.16.1.11 \_ORD

```
const string wnd.CNST._ORD = "Order"
```

## 7.16.1.12 \_REF

```
const string wnd.CNST._REF = "Refresh"
```

## 7.16.1.13 CANCEL

```
const int wnd.CNST.CANCEL = 10
```

## 7.16.1.14 ERR

```
const int wnd.CNST.ERR = 5
```

## 7.16.1.15 NO

```
const string wnd.CNST.NO = "no"
```

## 7.16.1.16 NOTITEM

```
const int wnd.CNST.NOTITEM = 1
```

## 7.16.1.17 NRD

```
const int wnd.CNST.NRD = 6
```

## 7.16.1.18 NSF

```
const string wnd.CNST.NSF = "'<missing at server>'"
```

## 7.16.1.19 NULL

```
const string wnd.CNST.NULL = "<none>"
```

## 7.16.1.20 OK

```
const int wnd.CNST.OK = 0
```

## 7.16.1.21 stop0

```
const string wnd.CNST.stop0 = "*stop*"
```

## 7.16.1.22 stop1

```
const string wnd.CNST.stop1 = "_STOP_"
```

## 7.16.1.23 stop2

```
const string wnd.CNST.stop2 = "Continue"
```

## 7.16.1.24 UNK\_TXT

```
const string wnd.CNST.UNK_TXT = ""
```

## 7.16.1.25 WHEEL\_DELTA

```
const int wnd.CNST.WHEEL_DELTA = 120
```

## 7.16.1.26 WPAR

```
const int wnd.CNST.WPAR = 3
```

## 7.16.1.27 YES

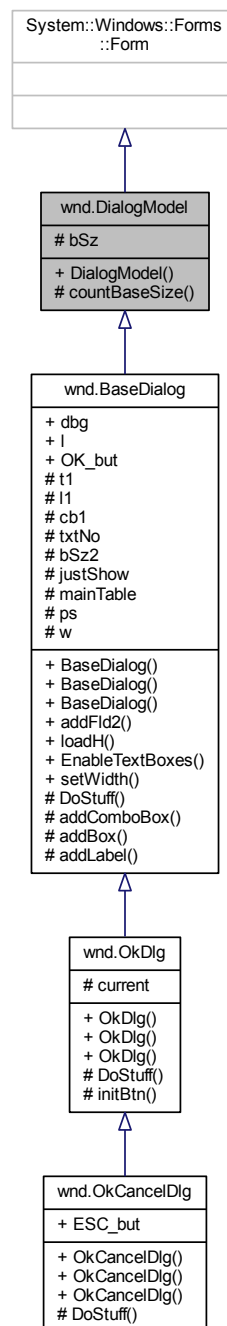
```
const string wnd.CNST.YES = "yes"
```

Объявления и описания членов структуры находятся в файле:

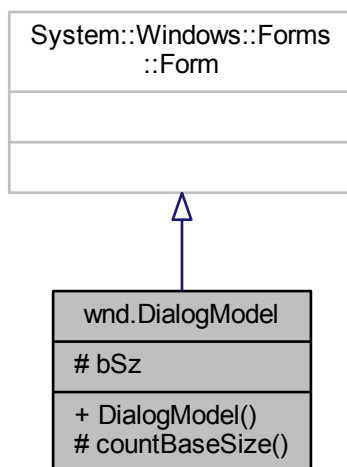
- [cs/ut.cs](#)

## 7.17 Класс wnd.DialogModel

Граф наследования:wnd.DialogModel:



Граф связей класса wnd.DialogModel:



Открытые члены

- `DialogModel` (string q="question for OkCancel window", int www=1)

Защищенные члены

- Size `countBaseSize` ()

Защищенные данные

- Size `bSz`

### 7.17.1 Конструктор(ы)

#### 7.17.1.1 DialogModel()

```
wnd.DialogModel.DialogModel (
    string q = "question for OkCancel window",
    int www = 1 ) [inline]
```

Аргументы

q	заголовок окна
www	минимальное количество дополнительных панелей между кнопками ок, канел

Граф вызовов:



## 7.17.2 Методы

### 7.17.2.1 countBaseSize()

Size wnd.DialogModel.countBaseSize ( ) [inline], [protected]

Граф вызова функции:



## 7.17.3 Данные класса

### 7.17.3.1 bSz

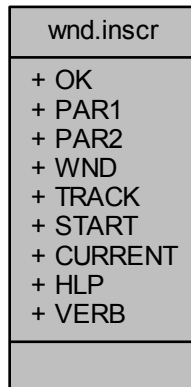
Size wnd.DialogModel.bSz [protected]

Объявления и описания членов класса находятся в файле:

- [cs/wDlgBase.cs](#)

## 7.18 Класс wnd.inscr

Граф связей класса wnd.inscr:



### Открытые атрибуты

- `const string OK = "OK"`
- `const string PAR1 = "Arguments"`
- `const string PAR2 = "Parameters"`
- `const string WND = "Window for"`
- `const string TRACK = "track"`
- `const string START = "start"`
- `const string CURRENT = "current"`
- `const string HLP = "help"`
- `const string VERB = "verbose"`

#### 7.18.1 Данные класса

##### 7.18.1.1 CURRENT

```
const string wnd.inscr.CURRENT = "current"
```

##### 7.18.1.2 HLP

```
const string wnd.inscr.HLP = "help"
```

## 7.18.1.3 OK

```
const string wnd.inscr.OK = "OK"
```

## 7.18.1.4 PAR1

```
const string wnd.inscr.PAR1 = "Arguments"
```

## 7.18.1.5 PAR2

```
const string wnd.inscr.PAR2 = "Parameters"
```

## 7.18.1.6 START

```
const string wnd.inscr.START = "start"
```

## 7.18.1.7 TRACK

```
const string wnd.inscr.TRACK = "track"
```

## 7.18.1.8 VERB

```
const string wnd.inscr.VERB = "verbose"
```

## 7.18.1.9 WND

```
const string wnd.inscr.WND = "Window for"
```

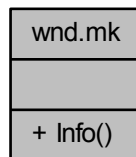
Объявления и описания членов класса находятся в файле:

- [cs/wBtnLbl.cs](#)



## 7.19 Класс wnd.mk

Граф связей класса wnd.mk:



Открытые статические члены

- static string [Info](#) (string dll)

### 7.19.1 Методы

#### 7.19.1.1 Info()

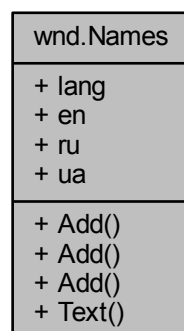
```
static string wnd.mk.Info (  
    string dll ) [inline], [static]
```

Объявления и описания членов класса находятся в файле:

- [cs/ut.cs](#)

## 7.20 Класс wnd.Names

Граф связей класса wnd.Names:



## Открытые статические члены

- static void [Add](#) (string key)
- static void [Add](#) (string key, string ruT)
- static void [Add](#) (string key, string ruT, string uaT)
- static string [Text](#) (string key)

## Статические открытые данные

- static int [lang](#)
- static StringDictionary [en](#)
- static StringDictionary [ru](#)
- static StringDictionary [ua](#)

### 7.20.1 Методы

#### 7.20.1.1 Add() [1/3]

```
static void wnd.Names.Add (  
    string key ) [inline], [static]
```

Граф вызова функции:



#### 7.20.1.2 Add() [2/3]

```
static void wnd.Names.Add (  
    string key,  
    string ruT ) [inline], [static]
```

## 7.20.1.3 Add() [3/3]

```
static void wnd.Names.Add (  
    string key,  
    string ruT,  
    string uaT ) [inline], [static]
```

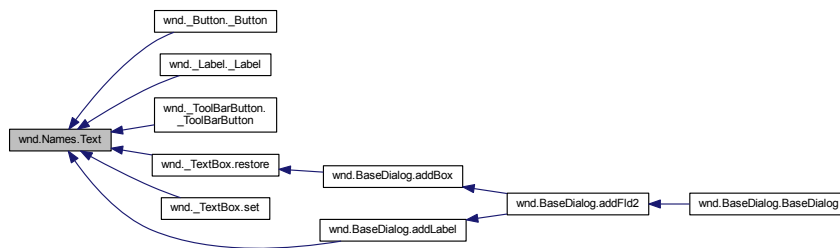
Граф вызовов:



## 7.20.1.4 Text()

```
static string wnd.Names.Text (  
    string key ) [inline], [static]
```

Граф вызова функции:



## 7.20.2 Данные класса

## 7.20.2.1 en

```
StringDictionary wnd.Names.en [static]
```

### 7.20.2.2 lang

```
int wnd.Names.lang [static]
```

### 7.20.2.3 ru

```
StringDictionary wnd.Names.ru [static]
```

### 7.20.2.4 ua

```
StringDictionary wnd.Names.ua [static]
```

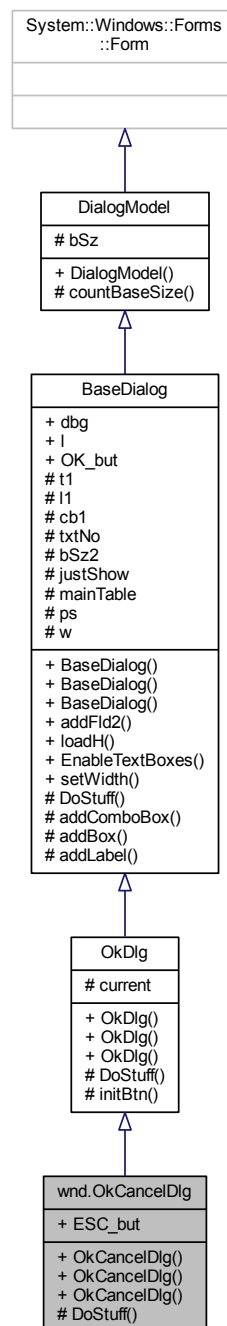
Объявления и описания членов класса находятся в файле:

- [cs/wBtnLbl.cs](#)

## 7.21 Класс wnd.OkCancelDlg

класс для ввода значений можно отменить ввод некоторых полей.

Граф наследования: wnd.OkCancelDlg:





## Открытые атрибуты

- [\\_Button ESC\\_but](#)

## Защищенные члены

- override void [DoStuff](#) ()

## Дополнительные унаследованные члены

### 7.21.1 Подробное описание

класс для ввода значений можно отменить ввод некоторых полей.

### 7.21.2 Конструктор(ы)

#### 7.21.2.1 OkCancelDlg() [1/3]

```
wnd.OkCancelDlg.OkCancelDlg (  
    string name,  
    Logger ll = null ) [inline]
```

#### 7.21.2.2 OkCancelDlg() [2/3]

```
wnd.OkCancelDlg.OkCancelDlg (  
    string name,  
    Logger ll,  
    params Arg [] ps ) [inline]
```

#### 7.21.2.3 OkCancelDlg() [3/3]

```
wnd.OkCancelDlg.OkCancelDlg (  
    string name,  
    Arg p,  
    Logger ll = null ) [inline]
```

### 7.21.3 Методы

### 7.21.3.1 DoStuff()

override void wnd.OkCancelDlg.DoStuff ( ) [inline], [protected], [virtual]

Переопределяет метод предка [wnd.OkDlg](#).

### 7.21.4 Данные класса

#### 7.21.4.1 ESC\_but

[\\_Button](#) wnd.OkCancelDlg.ESC\_but

Объявления и описания членов класса находятся в файле:

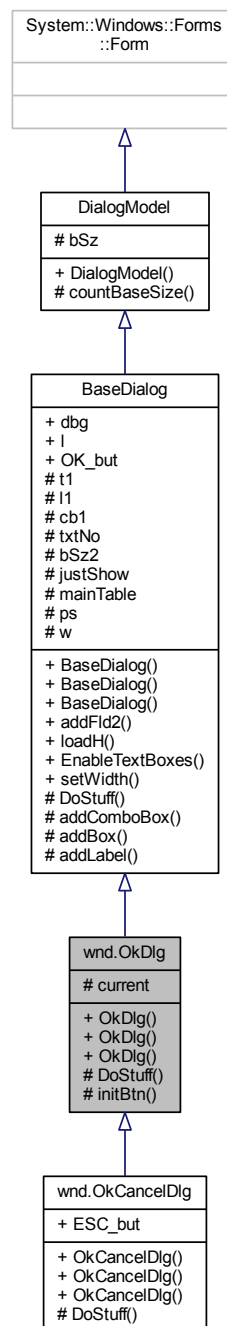
- [cs/wOkCancel.cs](#)

## 7.22 Класс wnd.OkDlg

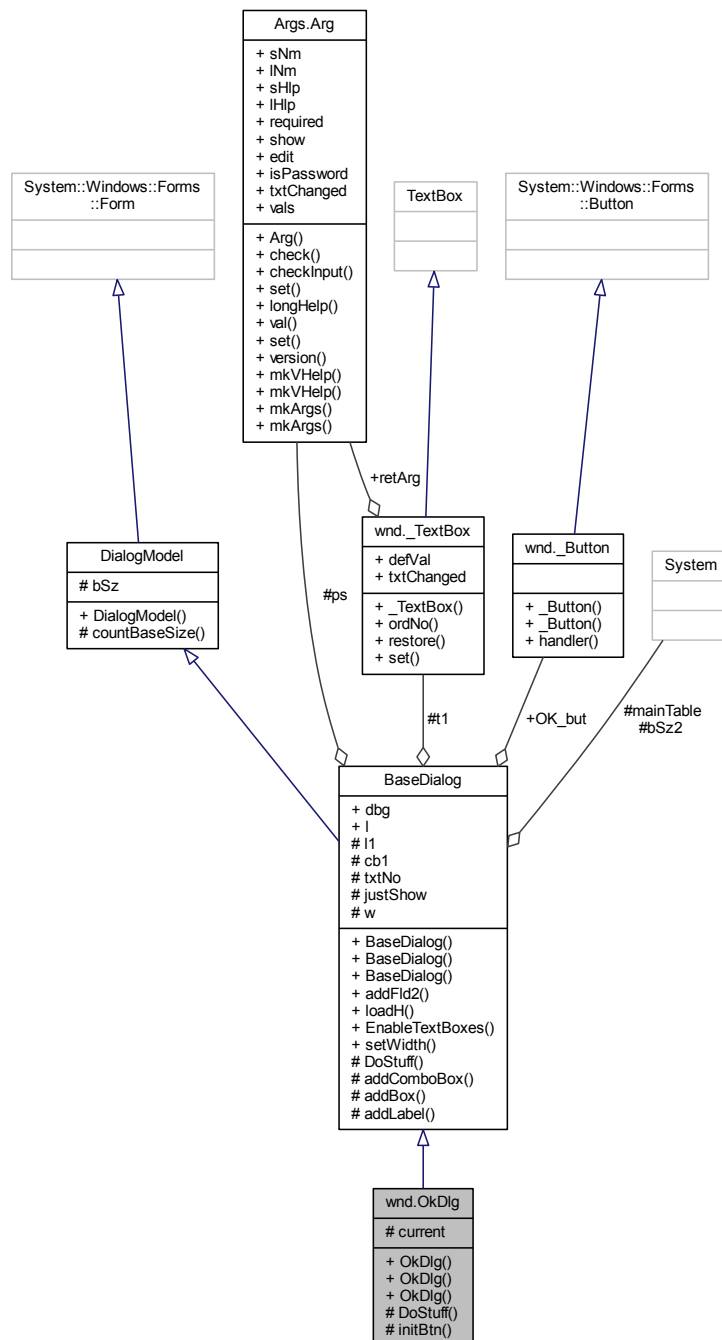
класс для демонстрации значений (вводить нельзя)



Граф наследования:wnd.OkDlg:



Граф связей класса wnd.OkDlg:



Открытые члены

- `OkDlg` (string name, Logger ll=null)
- `OkDlg` (string name, Logger ll, params `Arg[] ps`)
- `OkDlg` (string name, `Arg p`, Logger ll=null)

### Защищенные члены

- override void [DoStuff](#) ()
- void [initBtn](#) ([\\_Button](#) b, string name, string text, DialogResult result)

### Защищенные данные

- int [current](#)

### Дополнительные унаследованные члены

#### 7.22.1 Подробное описание

класс для демонстрации значений (вводить нельзя)

#### 7.22.2 Конструктор(ы)

##### 7.22.2.1 OkDlg() [1/3]

```
wnd.OkDlg.OkDlg (  
    string name,  
    Logger ll = null ) [inline]
```

##### 7.22.2.2 OkDlg() [2/3]

```
wnd.OkDlg.OkDlg (  
    string name,  
    Logger ll,  
    params Arg [] ps ) [inline]
```

##### 7.22.2.3 OkDlg() [3/3]

```
wnd.OkDlg.OkDlg (  
    string name,  
    Arg p,  
    Logger ll = null ) [inline]
```

#### 7.22.3 Методы

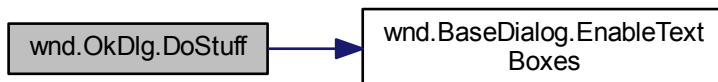
### 7.22.3.1 DoStuff()

override void wnd.OkDlg.DoStuff ( ) [inline], [protected], [virtual]

Замещает [wnd.BaseDialog](#).

Переопределяется в [wnd.OkCancelDlg](#).

Граф вызовов:



### 7.22.3.2 initBtn()

```
void wnd.OkDlg.initBtn (
    _Button b,
    string name,
    string text,
    DialogResult result ) [inline], [protected]
```

Граф вызовов:



## 7.22.4 Данные класса

### 7.22.4.1 current

```
int wnd.OkDlg.current [protected]
```

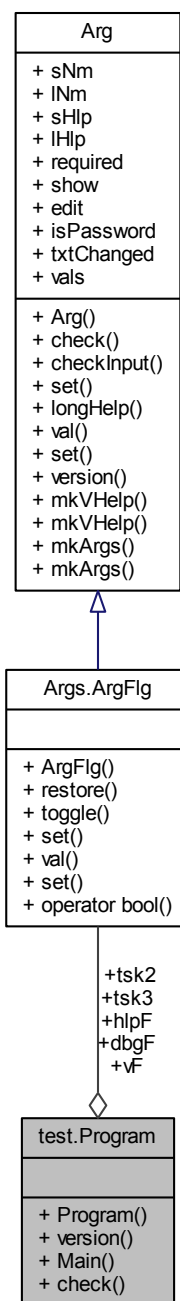
Объявления и описания членов класса находятся в файле:

- [cs/wOkCancel.cs](#)

## 7.23 Класс test.Program

класс содержит глобальные переменные для ключей, тест юнит Main и функцию выдачи подсказки

Граф связей класса test.Program:



Открытые статические члены

- static [Program](#) ()

- static void `version` (out int major, out int minor, out int build)
- static void `Main` (string[] args)
- static bool `check` (object sender, string email)  
чтобы продемонстрировать проверку ввода

## Статические открытые данные

- static `ArgFlg hlpF`  
флаг подсказки.
- static `ArgFlg dbgF`  
флаг отладки
- static `ArgFlg vF`  
флаг для дублирования журналирования в стандартный вывод
- static `ArgFlg tsk2`  
пример ключа для выбора работы2,.
- static `ArgFlg tsk3`

### 7.23.1 Подробное описание

класс содержит глобальные переменные для ключей, тест юнит Main и функцию выдачи подсказки

Приложение демонстрирует использование библиотеки разбора командной строки. Использует систему журналирования LOGGER. Есть ключи всех типов - логический (так же называется флагом), вещественный, целый и строчный. Занимается тем, что записывает в журнал системы сообщения всех уровней важности, что задается параметром ключа -l. Если ключу -p присвоить значение 0.5, то он будет установлен в 0.1. Один из ключей -t1, -t2 и t3 является обязательным. Они передаются в метод `mkVHelp` класса `Arg` во втором аргументе и не обрамляются квадратными скобками.

### 7.23.2 Конструктор(ы)

#### 7.23.2.1 Program()

static test.Program.Program ( ) [inline], [static]

Пример создания объектов, задаются значение по умолчанию, короткое имя, синоним, подсказка и обозначены параметра аргумента.

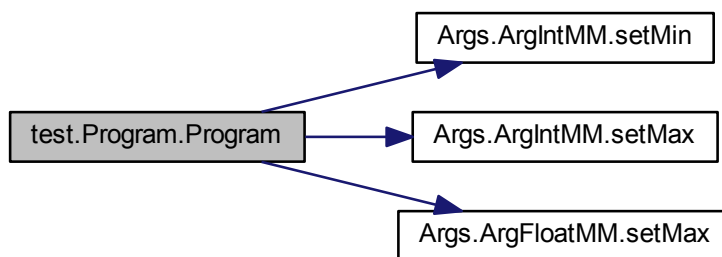
```
tsk3 = new ArgFlg(false, "t3", "window", "to show dialog window");
tsk3.show = false; // отменить автоматическое добавление аргумента в ком.строку
подсказки,
// аргумент добавляется к подсказке во втором параметре
// Arg.mkVHelp

logLvl = new ArgIntMM(1, "l", "log", "log level", "LLL");
logLvl.setMin(1); // задаем минимальные и максимальные значения аргумента
logLvl.setMax(8); //

//logLvl.show = false;
fNm = new ArgStr("somefile.dat", "f", "file", "data file", "FLNM");
fNm.required = true; // нет квадратных скобок
// fNm.show = false;

perCent = new ArgFloatMM(0.05, "p", "percent", "percent for something", "PPP");
perCent.setMax(100.0);
```

Граф вызовов:



### 7.23.3 Методы

#### 7.23.3.1 check()

```
static bool test.Program.check (  
    object sender,  
    string email ) [inline], [static]
```

чтобы продемонстрировать проверку ввода

Граф вызова функции:



#### 7.23.3.2 Main()

```
static void test.Program.Main (  
    string [] args ) [inline], [static]
```

Показать строку подсказки и прекратить работу приложения. В командной строке оператор обязан задать один из ключей `-t1` или `-t2`, поэтому у переменных `tsk1` и `tsk2` задано свойство не показывать, а они появляются во втором параметре функции `mkVHelp`.

Пример создания метода `usage`

```
static public void usage(){
    Arg.mkVHelp("to test command line arguments" // подсказка по утилите
        , "-t1 | -t2 | -t3" // автоматическая вставка этих аргументов в ком. строку
        , vF
        , Arg.mkArgs(
            , hlpF
            , dbgF
            , vF
            , logNm
            , logLvl
            , perCent
            , flNm
            , tsk1
            , tsk2
            , tsk3
        )
    );
    Environment.Exit(1);
}
```

Работа тест юнита при введенных ключах '-v' и '-?' заключается в вызове данного метода `test.usage()`, который в стандартный вывод ошибок выводит следующий текст

```
to test command line arguments (ver:2.10.0)
usage:
a [-?] [-d] [-v] [-ln NNN] [-l LLL] [-p PPP] -f FLNM -t1 | -t2 | -t3
options:
-?          : to see this help: True
-d          : debug mode: False
-v          : additional info: True
-ln NNN     : log level names:{Spam Debug Warning Stats Error FatalError Info Ignore}; Error
-l LLL      : log level (1..8): 1
-p PPP      : percent for something (..100): 0.05
-f FLNM     : data file: somefile.dat
-t1         : to do some work: False
-t2         : to do another work: False
-t3         : to show dialog window: False
'?' means the same as 'help'
'd' means the same as 'debug'
'v' means the same as 'verbose'
'ln' means the same as 'logName'
'l' means the same as 'log'
'p' means the same as 'percent'
'f' means the same as 'file'
't1' means the same as 'workOne'
't2' means the same as 'workNext'
't3' means the same as 'window'
```

Пример создания окна диалогового для ввода параметров

```
if (tsk3) {
    ArgStr user = new ArgStr("NuFoo", "pwd", "user", "input your login here");
    user.edit = false; // запрет редактирования

    ArgStr pwd = new ArgStr("default", "pwd", "password", "input your password here");
    pwd.isPassword = true; // забить звездочками вводимое значение

    ArgStr email = new ArgStr("email@example.com", "email", "email", "email of user");
    email.txtChanged = check; // вызвать метод check для проверки правильности ввода
    ArgStr test_Str_Arr = new ArgStr("Black", "c", "color", "color", "chose some color"
        , WCNST.colors // показать список возможных альтернатив
    );
}

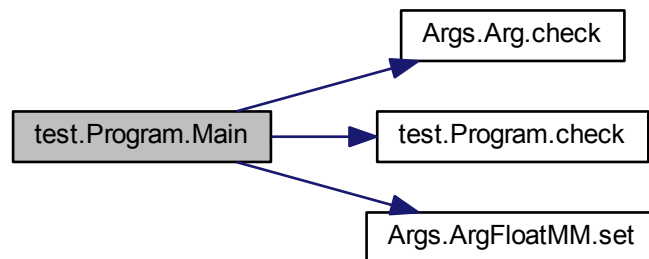
OkCancelDlg it = new OkCancelDlg( "window to input some params", Logger
    , user
    , pwd
    , email
    , perCent
    , test_Str_Arr );

DialogResult rc = it.ShowDialog();
```



```
if (rc == DialogResult.OK) {  
    Console.WriteLine ("your input : {0}/{1}/{2}/{3}/{4}"  
        , (string)user  
        , (string)pwd  
        , (string)email  
        , (string)test_Str_Arr  
        , (double) perCent  
    );  
}  
else  
    Console.WriteLine ("nothing to show");  
}
```

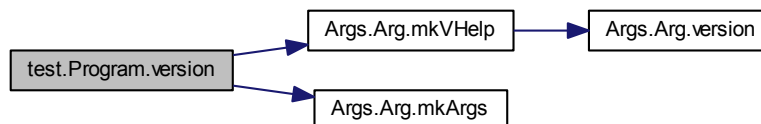
Граф ВЫЗОВОВ:



### 7.23.3.3 version()

```
static void test.Program.version (  
    out int major,  
    out int minor,  
    out int build ) [inline], [static]
```

Граф ВЫЗОВОВ:



### 7.23.4 Данные класса

## 7.23.4.1 dbgF

`ArgFlg test.Program.dbgF [static]`

флаг отладки

## 7.23.4.2 hlpF

`ArgFlg test.Program.hlpF [static]`

флаг подсказки.

## 7.23.4.3 tsk2

`ArgFlg test.Program.tsk2 [static]`

пример ключа для выбора работы2,.

Пример объявления целых, строчных плавающих и логических переменных для ввода аргументов командной строки

```
static public ArgIntMM logLvl ;
static public ArgStr logNm ;
static public ArgFloatMM perCent ;
static public ArgStr flNm ;
static public ArgFlg tsk1 ; //< пример ключа для выбора работы1.
```

## 7.23.4.4 tsk3

`ArgFlg test.Program.tsk3 [static]`

пример ключа для покаать окно ввода, параметров один из ключей обязан быть в командной строке.

## 7.23.4.5 vF

`ArgFlg test.Program.vF [static]`

флаг для дублирования журналирования в стандартный вывод

Объявления и описания членов класса находятся в файле:

- [Program.cs](#)

## 7.24 Структура wnd.SZ

Граф связей класса wnd.SZ:



### Открытые атрибуты

- `const int TD_BUF = 500`
- `const int Y = 24`
- `const int X_BUTTON = 72`
- `const int X_LABEL1 = 64`
- `const int X_LABEL2 = 124`
- `const int X_TEXT = 124`
- `const int X_FLD = 1`
- `const int Y_FLD = 2`
- `const int X_SPC = 8`
- `const int Y_SPC = 4`
- `const int FONT = 9`

### 7.24.1 Данные класса

#### 7.24.1.1 FONT

```
const int wnd.SZ.FONT = 9
```

## 7.24.1.2 TD\_BUF

```
const int wnd.SZ.TD_BUF = 500
```

## 7.24.1.3 X\_BUTTON

```
const int wnd.SZ.X_BUTTON = 72
```

## 7.24.1.4 X\_FLD

```
const int wnd.SZ.X_FLD = 1
```

## 7.24.1.5 X\_LABEL1

```
const int wnd.SZ.X_LABEL1 = 64
```

## 7.24.1.6 X\_LABEL2

```
const int wnd.SZ.X_LABEL2 = 124
```

## 7.24.1.7 X\_SPC

```
const int wnd.SZ.X_SPC = 8
```

## 7.24.1.8 X\_TEXT

```
const int wnd.SZ.X_TEXT = 124
```

## 7.24.1.9 Y

```
const int wnd.SZ.Y = 24
```

## 7.24.1.10 Y\_FLD

```
const int wnd.SZ.Y_FLD = 2
```

## 7.24.1.11 Y\_SPC

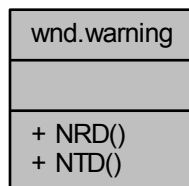
```
const int wnd.SZ.Y_SPC = 4
```

Объявления и описания членов структуры находятся в файле:

- [cs/ut.cs](#)

## 7.25 Класс wnd.warning

Граф связей класса wnd.warning:



## Открытые статические члены

- static void [NRD](#) (object sender, EventArgs e)
- static void [NTD](#) (object sender, EventArgs e)

## 7.25.1 Методы

## 7.25.1.1 NRD()

```
static void wnd.warning.NRD (  
    object sender,  
    EventArgs e ) [inline], [static]
```

### 7.25.1.2 NTD()

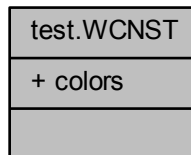
```
static void wnd.warning.NTD (  
    object sender,  
    EventArgs e ) [inline], [static]
```

Объявления и описания членов класса находятся в файле:

- [cs/ut.cs](#)

## 7.26 Класс test.WCNST

Граф связей класса test.WCNST:



Статические открытые данные

- `static string [] colors`

### 7.26.1 Данные класса

#### 7.26.1.1 colors

```
string [] test.WCNST.colors [static]
```

Объявления и описания членов класса находятся в файле:

- [Program.cs](#)

## Глава 8

# Файлы

### 8.1 Файл cs/args.cs

файл содержит классы для разбора командной строки

#### Классы

- class [Args.Arg](#)  
Родительский класс для разбора аргументов командной строки
- class [Args.ArgFlg](#)  
ключ логического типа (иначе - флаг)
- class [Args.ArgStr](#)  
ключ типа строка
- class [Args.ArgChar](#)  
ключ типа символ
- class [Args.ArgFloat](#)
- class [Args.ArgFloatMM](#)  
ключ вещественного типа, есть возможность ограничить максимальное и минимальное значение
- class [Args.ArgInt](#)  
ключ целого типа
- class [Args.ArgIntMM](#)  
ключ целого типа, есть возможность ограничить максимальное и минимальное значение

#### Пространства имен

- namespace [Args](#)

#### Функции

- delegate bool [Args.handler](#) (object src, string val)  
указатель для диалоговых окон

### 8.1.1 Подробное описание

файл содержит классы для разбора командной строки

## 8.2 Файл cs/args.txt

## 8.3 Файл cs/Assembly.txt

## 8.4 Файл cs/AssemblyInfo.cs

## 8.5 Файл AssemblyInfo.cs

## 8.6 Файл cs/ut.cs

### Классы

- class [wnd.check](#)  
класс для разного рода проверок
- class [wnd.mk](#)
- class [wnd.warning](#)
- struct [wnd.CNST](#)
- struct [wnd.SZ](#)

### Пространства имен

- namespace [wnd](#)

## 8.7 Файл cs/wBtnLbl.cs

### Классы

- class [wnd.Names](#)
- class [wnd.\\_Button](#)
- class [wnd.\\_Label](#)
- class [wnd.\\_LabelInfo](#)
- class [wnd.\\_ToolBarButton](#)
- class [wnd.inscr](#)
- class [wnd.\\_TextBox](#)

### Пространства имен

- namespace [wnd](#)



## 8.8 Файл cs/wDlgBase.cs

### Классы

- class [wnd.BaseDialog](#)
- class [wnd.DialogModel](#)

### Пространства имен

- namespace [wnd](#)

### Макросы

- `#define` [PANEL](#)

### 8.8.1 Макросы

#### 8.8.1.1 PANEL

```
#define PANEL
```

## 8.9 Файл cs/wOkCancel.cs

### Классы

- class [wnd.OkDlg](#)  
класс для демонстрации значений (вводить нельзя)
- class [wnd.OkCancelDlg](#)  
класс для ввода значений можно отменить ввод некоторых полей.

### Пространства имен

- namespace [wnd](#)

## 8.10 Файл Program.cs

### Классы

- class [test.Program](#)  
класс содержит глобальные переменные для ключей, тест юнит Main и функцию выдачи подсказки
- class [test.WCNST](#)

### Пространства имен

- namespace [test](#)



# Предметный указатель

- \_ADD
  - wnd::CNST, 80
- \_Button
  - wnd::\_Button, 16
- \_DEL
  - wnd::CNST, 80
- \_EDIT
  - wnd::CNST, 80
- \_ESC
  - wnd::CNST, 80
- \_EXP
  - wnd::CNST, 80
- \_EXIT
  - wnd::CNST, 80
- \_FLT
  - wnd::CNST, 81
- \_FND
  - wnd::CNST, 81
- \_Label
  - wnd::\_Label, 18
- \_LabelInfo
  - wnd::\_LabelInfo, 20
- \_NRD
  - wnd::CNST, 81
- \_OK
  - wnd::CNST, 81
- \_ORD
  - wnd::CNST, 81
- \_REF
  - wnd::CNST, 81
- \_TextBox
  - wnd::\_TextBox, 22
- \_ToolBarButton
  - wnd::\_ToolBarButton, 25

Add

- wnd::Names, 90

addBox

- wnd::BaseDialog, 72

addComboBox

- wnd::BaseDialog, 72

addFld2

- wnd::BaseDialog, 73

addLabel

- wnd::BaseDialog, 74

Arg

- Args::Arg, 28

ArgChar

- Args::ArgChar, 39

ArgFlg

- Args::ArgFlg, 43

ArgFloat

- Args::ArgFloat, 47

ArgFloatMM

- Args::ArgFloatMM, 52

ArgInt

- Args::ArgInt, 57

ArgIntMM

- Args::ArgIntMM, 62

ArgStr

- Args::ArgStr, 66

Args, 13

- handler, 13

Args.Arg, 26

Args.ArgChar, 36

Args.ArgFlg, 40

Args.ArgFloat, 45

Args.ArgFloatMM, 49

Args.ArgInt, 54

Args.ArgIntMM, 59

Args.ArgStr, 64

Args::Arg

- Arg, 28

- check, 29

- checkInput, 30

- edit, 34

- isPassword, 34

- IHlp, 35

- INm, 35

- longHelp, 30

- mkArgs, 30, 31

- mkVHlp, 31, 32

- required, 35

- sHlp, 35

- sNm, 35

- set, 33

- show, 35

- txtChanged, 36

- val, 33

- vals, 36

- version, 34

Args::ArgChar

- ArgChar, 39

- operator char, 39

- set, 39

- val, 40

Args::ArgFlg

- ArgFlg, 43

- operator bool, 43
- restore, 43
- set, 43, 44
- toggle, 44
- val, 44
- Args::ArgFloat
  - ArgFloat, 47
  - operator double, 47
  - operator float, 47
  - set, 48
  - v, 49
  - val, 48
- Args::ArgFloatMM
  - ArgFloatMM, 52
  - longHelp, 52
  - set, 52, 53
  - setMax, 53
  - setMin, 54
- Args::ArgInt
  - ArgInt, 57
  - checkInput, 57
  - operator int, 57
  - set, 57, 58
  - v, 58
  - val, 58
- Args::ArgIntMM
  - ArgIntMM, 62
  - longHelp, 62
  - set, 62, 63
  - setMax, 63
  - setMin, 63
- Args::ArgStr
  - ArgStr, 66
  - operator string, 66
  - set, 66
  - val, 66
- AssemblyInfo.cs, 112
- bSz
  - wnd::DialogModel, 86
- bSz2
  - wnd::BaseDialog, 76
- BaseDialog
  - wnd::BaseDialog, 70, 71
- CANCEL
  - wnd::CNST, 81
- CURRENT
  - wnd::inscr, 87
- cb1
  - wnd::BaseDialog, 76
- check
  - Args::Arg, 29
  - test::Program, 103
- checkInput
  - Args::Arg, 30
  - Args::ArgInt, 57
- colors
  - test::WCNST, 110
- countBaseSize
  - wnd::DialogModel, 86
- cs/Assembly.txt, 112
- cs/AssemblyInfo.cs, 112
- cs/args.cs, 111
- cs/args.txt, 112
- cs/ut.cs, 112
- cs/wBtnLbl.cs, 112
- cs/wDlgBase.cs, 113
- cs/wOkCancel.cs, 113
- current
  - wnd::OkDlg, 100
- dbg
  - wnd::BaseDialog, 76
- dbgF
  - test::Program, 105
- defVal
  - wnd::\_TextBox, 23
- DialogModel
  - wnd::DialogModel, 85
- DoStuff
  - wnd::BaseDialog, 74
  - wnd::OkCancelDlg, 95
  - wnd::OkDlg, 99
- ERR
  - wnd::CNST, 81
- ESC\_but
  - wnd::OkCancelDlg, 96
- edit
  - Args::Arg, 34
- en
  - wnd::Names, 91
- EnableTextBoxes
  - wnd::BaseDialog, 75
- FONT
  - wnd::SZ, 107
- HLP
  - wnd::inscr, 87
- handler
  - Args, 13
  - wnd::\_Button, 17
- hlpF
  - test::Program, 106
- Info
  - wnd::mk, 89
- initBtn
  - wnd::OkDlg, 100
- isIn
  - wnd::check, 78
- isPassword
  - Args::Arg, 34
- justShow
  - wnd::BaseDialog, 76

- l
  - wnd::BaseDialog, 76
- ll
  - wnd::BaseDialog, 77
- lHlp
  - Args::Arg, 35
- INm
  - Args::Arg, 35
- lang
  - wnd::Names, 91
- loadH
  - wnd::BaseDialog, 75
- longHelp
  - Args::Arg, 30
  - Args::ArgFloatMM, 52
  - Args::ArgIntMM, 62
- Main
  - test::Program, 103
- mainTable
  - wnd::BaseDialog, 77
- mkArgs
  - Args::Arg, 30, 31
- mkVHelp
  - Args::Arg, 31, 32
- NOTITEM
  - wnd::CNST, 82
- NRD
  - wnd::CNST, 82
  - wnd::warning, 109
- NSF
  - wnd::CNST, 82
- NTD
  - wnd::warning, 109
- NULL
  - wnd::CNST, 82
- NO
  - wnd::CNST, 82
- OK\_but
  - wnd::BaseDialog, 77
- OK
  - wnd::CNST, 82
  - wnd::inscr, 87
- OkCancelDlg
  - wnd::OkCancelDlg, 95
- OkDlg
  - wnd::OkDlg, 99
- operator bool
  - Args::ArgFlg, 43
- operator char
  - Args::ArgChar, 39
- operator double
  - Args::ArgFloat, 47
- operator float
  - Args::ArgFloat, 47
- operator int
  - Args::ArgInt, 57
- operator string
  - Args::ArgStr, 66
- ordNo
  - wnd::\_TextBox, 22
- PANEL
  - wDlgBase.cs, 113
- PAR1
  - wnd::inscr, 88
- PAR2
  - wnd::inscr, 88
- Program
  - test::Program, 102
- Program.cs, 113
- ps
  - wnd::BaseDialog, 77
- required
  - Args::Arg, 35
- restore
  - Args::ArgFlg, 43
  - wnd::\_TextBox, 22
- retArg
  - wnd::\_TextBox, 24
- ru
  - wnd::Names, 92
- sHlp
  - Args::Arg, 35
- sNm
  - Args::Arg, 35
- START
  - wnd::inscr, 88
- set
  - Args::Arg, 33
  - Args::ArgChar, 39
  - Args::ArgFlg, 43, 44
  - Args::ArgFloat, 48
  - Args::ArgFloatMM, 52, 53
  - Args::ArgInt, 57, 58
  - Args::ArgIntMM, 62, 63
  - Args::ArgStr, 66
  - wnd::\_TextBox, 23
- setMax
  - Args::ArgFloatMM, 53
  - Args::ArgIntMM, 63
- setMin
  - Args::ArgFloatMM, 54
  - Args::ArgIntMM, 63
- setWidth
  - wnd::BaseDialog, 75
- show
  - Args::Arg, 35
- stop0
  - wnd::CNST, 82
- stop1
  - wnd::CNST, 82
- stop2
  - wnd::CNST, 83

- t1
  - wnd::BaseDialog, 77
- TD\_BUF
  - wnd::SZ, 107
- TRACK
  - wnd::inscr, 88
- test, 14
- test.Program, 101
- test.WCNST, 110
- test::Program
  - check, 103
  - dbgF, 105
  - hlpF, 106
  - Main, 103
  - Program, 102
  - tsk2, 106
  - tsk3, 106
  - version, 105
  - vF, 106
- test::WCNST
  - colors, 110
- Text
  - wnd::Names, 91
- toggle
  - Args::ArgFlg, 44
- tsk2
  - test::Program, 106
- tsk3
  - test::Program, 106
- txtChanged
  - Args::Arg, 36
  - wnd::\_TextBox, 24
- txtNo
  - wnd::BaseDialog, 77
- UNK\_TXT
  - wnd::CNST, 83
- ua
  - wnd::Names, 92
- v
  - Args::ArgFloat, 49
  - Args::ArgInt, 58
- VERB
  - wnd::inscr, 88
- val
  - Args::Arg, 33
  - Args::ArgChar, 40
  - Args::ArgFlg, 44
  - Args::ArgFloat, 48
  - Args::ArgInt, 58
  - Args::ArgStr, 66
- vals
  - Args::Arg, 36
- version
  - Args::Arg, 34
  - test::Program, 105
- vF
  - test::Program, 106
- w
  - wnd::BaseDialog, 77
- wDlgBase.cs
  - PANEL, 113
- WHEEL\_DELTA
  - wnd::CNST, 83
- WND
  - wnd::inscr, 88
- WPAR
  - wnd::CNST, 83
- wnd, 14
- wnd.\_Button, 15
- wnd.\_Label, 17
- wnd.\_LabelInfo, 19
- wnd.\_TextBox, 20
- wnd.\_ToolBarButton, 24
- wnd.BaseDialog, 68
- wnd.CNST, 79
- wnd.check, 78
- wnd.DialogModel, 84
- wnd.inscr, 87
- wnd.mk, 89
- wnd.Names, 89
- wnd.OkCancelDlg, 92
- wnd.OkDlg, 96
- wnd.SZ, 107
- wnd.warning, 109
- wnd::\_Button
  - \_Button, 16
  - handler, 17
- wnd::\_Label
  - \_Label, 18
- wnd::\_LabelInfo
  - \_LabelInfo, 20
- wnd::\_TextBox
  - \_TextBox, 22
  - defVal, 23
  - ordNo, 22
  - restore, 22
  - retArg, 24
  - set, 23
  - txtChanged, 24
- wnd::\_ToolBarButton
  - \_ToolBarButton, 25
- wnd::BaseDialog
  - addBox, 72
  - addComboBox, 72
  - addFld2, 73
  - addLabel, 74
  - bSz2, 76
  - BaseDialog, 70, 71
  - cb1, 76
  - dbg, 76
  - DoStuff, 74
  - EnableTextBoxes, 75
  - justShow, 76
  - l, 76
  - l1, 77

- loadH, 75
- mainTable, 77
- OK\_but, 77
- ps, 77
- setWidth, 75
- t1, 77
- txtNo, 77
- w, 77
- wnd::CNST
  - \_ADD, 80
  - \_DEL, 80
  - \_EDIT, 80
  - \_ESC, 80
  - \_EXP, 80
  - \_EXIT, 80
  - \_FLT, 81
  - \_FND, 81
  - \_NRD, 81
  - \_OK, 81
  - \_ORD, 81
  - \_REF, 81
  - CANCEL, 81
  - ERR, 81
  - NOTITEM, 82
  - NRD, 82
  - NSF, 82
  - NULL, 82
  - NO, 82
  - OK, 82
  - stop0, 82
  - stop1, 82
  - stop2, 83
  - UNK\_TXT, 83
  - WHEEL\_DELTA, 83
  - WPAR, 83
  - YES, 83
- wnd::DialogModel
  - bSz, 86
  - countBaseSize, 86
  - DialogModel, 85
- wnd::Names
  - Add, 90
  - en, 91
  - lang, 91
  - ru, 92
  - Text, 91
  - ua, 92
- wnd::OkCancelDlg
  - DoStuff, 95
  - ESC\_but, 96
  - OkCancelDlg, 95
- wnd::OkDlg
  - current, 100
  - DoStuff, 99
  - initBtn, 100
  - OkDlg, 99
- wnd::SZ
  - FONT, 107
  - TD\_BUF, 107
  - X\_BUTTON, 108
  - X\_FLD, 108
  - X\_LABEL1, 108
  - X\_LABEL2, 108
  - X\_SPC, 108
  - X\_TEXT, 108
  - Y, 108
  - Y\_FLD, 108
  - Y\_SPC, 109
- wnd::check
  - isIn, 78
- wnd::inscr
  - CURRENT, 87
  - HLP, 87
  - OK, 87
  - PAR1, 88
  - PAR2, 88
  - START, 88
  - TRACK, 88
  - VERB, 88
  - WND, 88
- wnd::mk
  - Info, 89
- wnd::warning
  - NRD, 109
  - NTD, 109
- X\_BUTTON
  - wnd::SZ, 108
- X\_FLD
  - wnd::SZ, 108
- X\_LABEL1
  - wnd::SZ, 108
- X\_LABEL2
  - wnd::SZ, 108
- X\_SPC
  - wnd::SZ, 108
- X\_TEXT
  - wnd::SZ, 108
- Y
  - wnd::SZ, 108
- Y\_FLD
  - wnd::SZ, 108
- Y\_SPC
  - wnd::SZ, 109
- YES
  - wnd::CNST, 83