

едение1.11Введениеsection.1.1 стирование1.22Тестированиеsection.1.2

args
2.00.01

Создано системой Doxygen 1.8.9.1

Пн 6 Июл 2015 19:26:05

Оглавление

1	Пояснительная записка проекта Args	1
1.1	Введение	1
1.2	Тестирование	2
2	Алфавитный указатель пространств имен	3
2.1	Пространства имен	3
3	Иерархический список классов	5
3.1	Иерархия классов	5
4	Алфавитный указатель классов	7
4.1	Классы	7
5	Список файлов	9
5.1	Файлы	9
6	Пространства имен	11
6.1	Пакет Args	11
6.2	Пакет test	11
6.2.1	Подробное описание	11
7	Классы	13
7.1	Класс Args.Arg	13
7.1.1	Подробное описание	16
7.1.2	Конструктор(ы)	16
7.1.2.1	Arg	16
7.1.3	Методы	16
7.1.3.1	check	16
7.1.3.2	longHelp	17
7.1.3.3	mkVHelp	17
7.1.3.4	mkVHelp	17
7.1.3.5	set	18
7.1.3.6	set	18
7.1.3.7	val	18

7.1.3.8	version	18
7.1.4	Данные класса	18
7.1.4.1	INlp	19
7.1.4.2	INm	19
7.1.4.3	required	19
7.1.4.4	sHlp	19
7.1.4.5	show	19
7.1.4.6	sNm	19
7.2	Класс Args.ArgFlg	19
7.2.1	Подробное описание	22
7.2.2	Конструктор(ы)	22
7.2.2.1	ArgFlg	22
7.2.3	Методы	22
7.2.3.1	operator bool	22
7.2.3.2	restore	22
7.2.3.3	set	22
7.2.3.4	set	22
7.2.3.5	toggle	22
7.2.3.6	val	23
7.3	Класс Args.ArgFloat	24
7.3.1	Подробное описание	26
7.3.2	Конструктор(ы)	26
7.3.2.1	ArgFloat	26
7.3.2.2	ArgFloat	26
7.3.3	Методы	26
7.3.3.1	operator double	26
7.3.3.2	operator float	26
7.3.3.3	set	26
7.3.3.4	set	26
7.3.3.5	set	26
7.3.3.6	set	26
7.3.3.7	val	27
7.3.4	Данные класса	27
7.3.4.1	v	27
7.4	Класс Args.ArgFloatMM	27
7.4.1	Подробное описание	30
7.4.2	Конструктор(ы)	30
7.4.2.1	ArgFloatMM	30
7.4.3	Методы	30
7.4.3.1	longHelp	30

7.4.3.2	set	30
7.4.3.3	set	30
7.4.3.4	set	30
7.4.3.5	set	30
7.4.3.6	setMax	31
7.4.3.7	setMin	31
7.5	Класс Args.ArgInt	31
7.5.1	Подробное описание	34
7.5.2	Конструктор(ы)	34
7.5.2.1	ArgInt	34
7.5.3	Методы	34
7.5.3.1	operator int	34
7.5.3.2	set	34
7.5.3.3	set	34
7.5.3.4	set	34
7.5.3.5	val	34
7.5.4	Данные класса	34
7.5.4.1	v	34
7.6	Класс Args.ArgIntMM	34
7.6.1	Подробное описание	37
7.6.2	Конструктор(ы)	37
7.6.2.1	ArgIntMM	37
7.6.3	Методы	37
7.6.3.1	longHelp	37
7.6.3.2	set	37
7.6.3.3	set	37
7.6.3.4	set	37
7.6.3.5	setMax	37
7.6.3.6	setMin	37
7.7	Класс Args.ArgStr	37
7.7.1	Подробное описание	40
7.7.2	Конструктор(ы)	40
7.7.2.1	ArgStr	40
7.7.3	Методы	40
7.7.3.1	operator string	40
7.7.3.2	set	40
7.7.3.3	set	40
7.7.3.4	val	40
7.8	Класс test.Program	40
7.8.1	Подробное описание	42

7.8.2	Методы	42
7.8.2.1	Main	42
7.8.2.2	usage	43
7.8.2.3	version	43
7.8.3	Данные класса	43
7.8.3.1	dbgF	43
7.8.3.2	flNm	43
7.8.3.3	hlpF	44
7.8.3.4	logLvl	44
7.8.3.5	perCent	44
7.8.3.6	tsk1	44
7.8.3.7	tsk2	44
7.8.3.8	vF	44
8	Файлы	45
8.1	Файл cs/args.cs	45
8.1.1	Подробное описание	45
8.2	Файл cs/args.txt	45
8.3	Файл cs/Assembly.txt	45
8.4	Файл cs/AssemblyInfo.cs	45
8.5	Файл AssemblyInfo.cs	46
8.6	Файл Program.cs	46
	Алфавитный указатель	47

Глава 1

Пояснительная записка проекта Args

Дата

2014-2015

1.1 Введение

Пояснительная записка к проекту args - библиотеки разбора аргументов командной строки и выдачи подсказки - страницы использования приложения (usage).

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

to test command line arguments

```
usage:
app[-?] [-v] [-l LLL] ...
options:
-?          : to see this help: True
-v          : additional info: True
-l LLL      : log level (1..8): 1
'?' means the same as 'help'
'v' means the same as 'verbose'
'l' means the same as 'log'
```

- ключами являются символы: -?, -v, -l.
- синонимами - help, verbose, log.
- описаниями - 'to see this help', 'additional info', 'log level'.
- а также присутствует один параметр ключа: LLL.

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

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 CHM 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.

Args	11
test	11

Глава 3

Иерархический список классов

3.1 Иерархия классов

Иерархия классов.

Args.Arg	13
Args.ArgFlg	19
Args.ArgFloat	24
Args.ArgFloatMM	27
Args.ArgInt	31
Args.ArgIntMM	34
Args.ArgStr	37
test.Program	40

Глава 4

Алфавитный указатель классов

4.1 Классы

Классы с их кратким описанием.

Args.Arg	Родительский класс для разбора аргументов командной строки	13
Args.ArgFlg	ключ логического типа (иначе - флаг)	19
Args.ArgFloat	24
Args.ArgFloatMM	ключ строчного типа, есть возможность ограничить максимальное и минимальное значение	27
Args.ArgInt	ключ целого типа	31
Args.ArgIntMM	ключ целого типа, есть возможность ограничить максимальное и минимальное значение	34
Args.ArgStr	ключ типа строка	37
test.Program	класс содержит глобальные переменные для ключей, тест юнит Main и функцию выдачи подсказки	40

Глава 5

Список файлов

5.1 Файлы

Полный список файлов.

AssemblyInfo.cs	46
Program.cs	46
cs/args.cs	
файл содержит классы для разбора командной строки	45
cs/AssemblyInfo.cs	45

Глава 6

Пространства имен

6.1 Пакет Args

Классы

- class [Arg](#)
Родительский класс для разбора аргументов командной строки
- class [ArgFlg](#)
ключ логического типа (иначе - флаг)
- class [ArgFloat](#)
- class [ArgFloatMM](#)
ключ строчного 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

6.2 Пакет test

Классы

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

6.2.1 Подробное описание

Пространство имен содержит тест-юнит для тестирования библиотеки args.

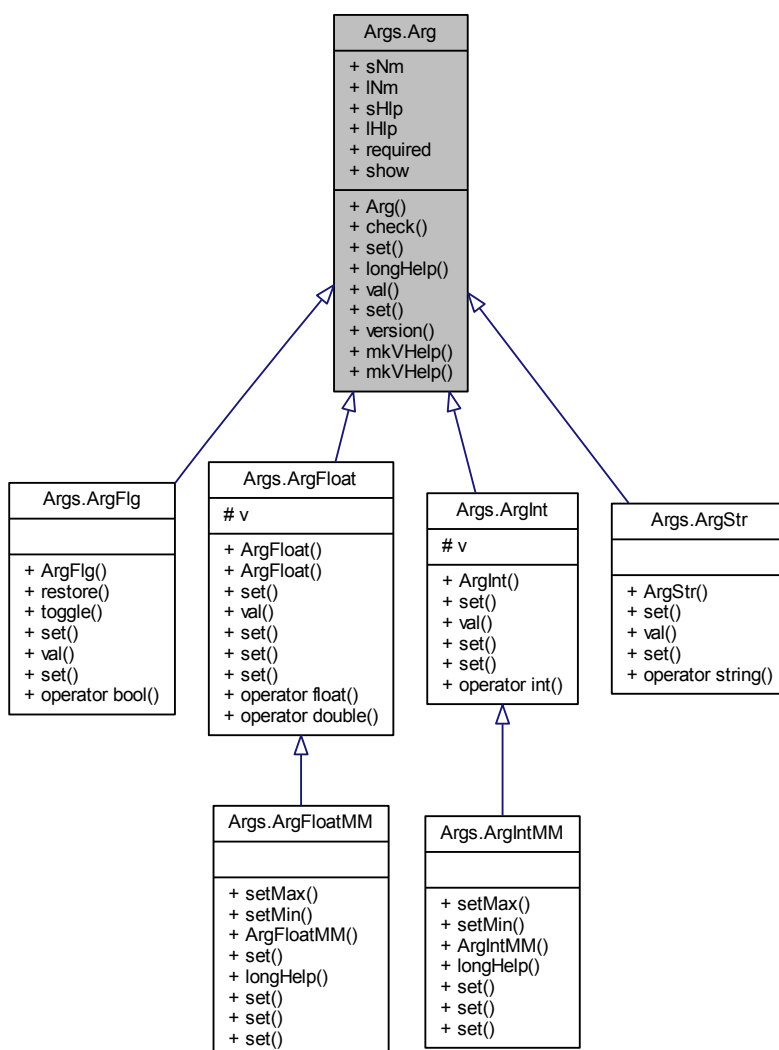
Глава 7

Классы

7.1 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
+ Arg() + check() + set() + longHelp() + val() + set() + version() + mkVHelp() + mkVHelp()

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 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

Открытые атрибуты

- string [sNm](#)
ключ

- string [lNm](#)
синоним ключа
- string [sHlp](#)
параметр 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.

7.1.1 Подробное описание

Родительский класс для разбора аргументов командной строки

7.1.2 Конструктор(ы)

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

Аргументы

<code>sNm</code>	ключ
<code>lNm</code>	синоним
<code>sHlp</code>	параметр ключа
<code>lHlp</code>	описание

7.1.3 Методы

7.1.3.1 `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

<code>i</code>	индекс текущего аргумента
<code>ps</code>	аргументы командной строки

Граф вызовов:



Граф вызова функции:



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

функция создает подсказку по ключу.

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

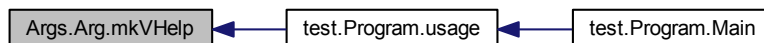
7.1.3.3 static void Args.Arg.mkVHelp (string forWhat, string add, bool verbose, params Arg[] ars) [inline], [static]

выдача подсказки

Граф вызовов:



Граф вызова функции:



7.1.3.4 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

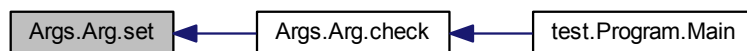
7.1.3.5 virtual void Args.Arg.set (ref int i, string[] ps) [inline], [virtual]

Аргументы

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

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

Граф вызова функции:



7.1.3.6 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.ArgStr](#) и [Args.ArgFlg](#).

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

функция переводит в строку значение ключа.

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

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

версия библиотеки

Граф вызова функции:



7.1.4 Данные класса

7.1.4.1 string Args.Arg.lHlp

описание ключа

7.1.4.2 string Args.Arg.lNm

синоним ключа

7.1.4.3 bool Args.Arg.required

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

7.1.4.4 string Args.Arg.sHlp

параметр ключа

7.1.4.5 bool Args.Arg.show

показывать ли ключ в строке вызова, по умолчанию показывать.

7.1.4.6 string Args.Arg.sNm

ключ

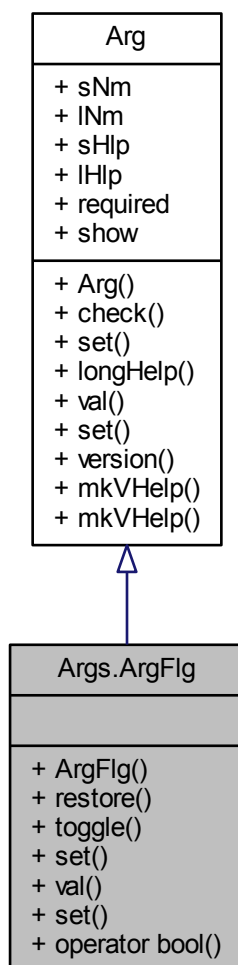
Объявления и описания членов класса находятся в файле:

- [cs/args.cs](#)

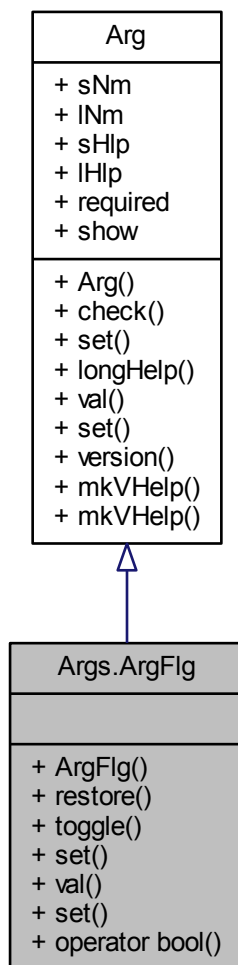
7.2 Класс Args.ArgFlg

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

7.2.1 Подробное описание

ключ логического типа (иначе - флаг)

7.2.2 Конструктор(ы)

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

Аргументы

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

7.2.3 Методы

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

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

< восстановить флаг

7.2.3.3 `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](#).

Граф вызовов:



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

восстанавливает из строки значение ключа

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

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

< переключить флаг

Граф вызова функции:



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

функция переводит в строку значение ключа.

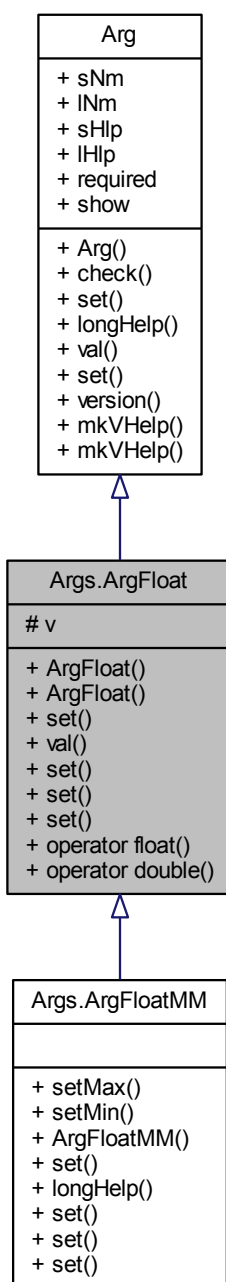
Переопределяет метод предка [Args.Arg](#).

Объявления и описания членов класса находятся в файле:

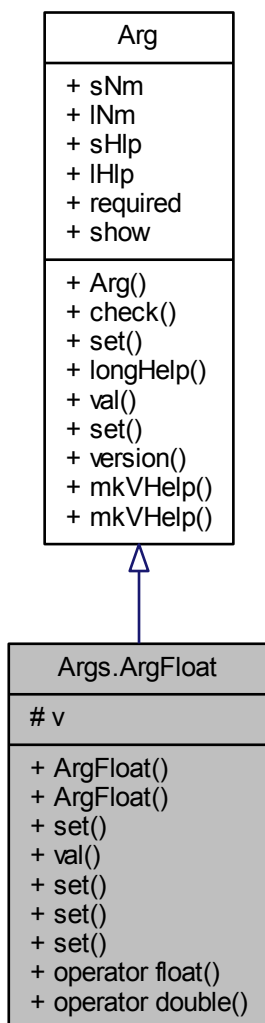
- [cs/args.cs](#)

7.3 Класс Args.ArgFloat

Граф наследования:Args.ArgFloat:



Граф связей класса Args.ArgFloat:



Открытые члены

- `ArgFloat` (float `v`, string `sNm`, string `INm`, string `IHlp`=null, string `sHlp`=null)
- `ArgFloat` (double `v`, string `sNm`, string `INm`, string `IHlp`=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

- float [v](#)

Дополнительные унаследованные члены

7.3.1 Подробное описание

ключ типа вещественное число. целая часть числа от дробной отделяется десятичной точкой, а не запятой, как в английской культуре.

7.3.2 Конструктор(ы)

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

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

7.3.3 Методы

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

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

7.3.3.3 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](#).

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

восстанавливает из строки значение ключа

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

Переопределяется в [Args.ArgFloatMM](#).

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

Переопределяется в [Args.ArgFloatMM](#).

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

Переопределяется в [Args.ArgFloatMM](#).

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

функция переводит в строку значение ключа.

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

7.3.4 Данные класса

7.3.4.1 `float Args.ArgFloat.v [protected]`

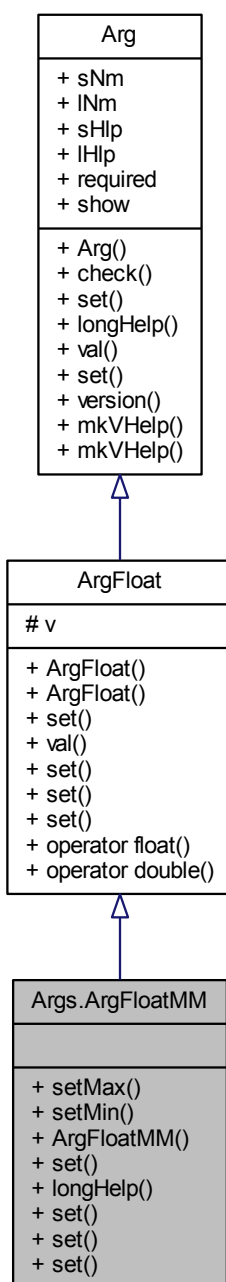
Объявления и описания членов класса находятся в файле:

- [cs/args.cs](#)

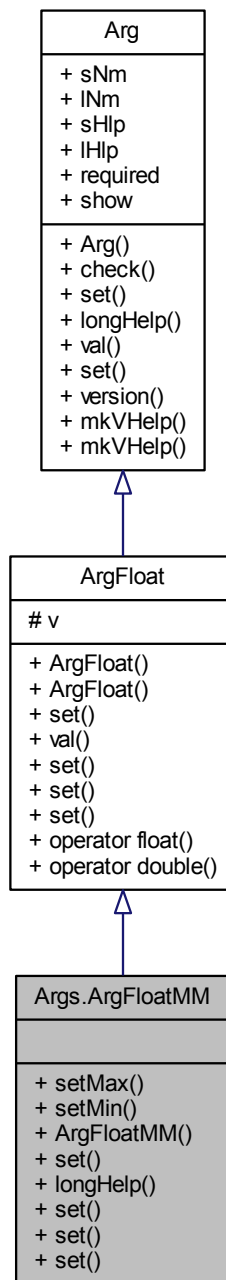
7.4 Класс Args.ArgFloatMM

ключ строчного 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

7.4.1 Подробное описание

ключ строчного типа, есть возможность ограничить максимальное и минимальное значение

7.4.2 Конструктор(ы)

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

7.4.3 Методы

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

функция создает подсказку по ключу.

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

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

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

Граф вызова функции:



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

восстанавливает из строки значение ключа

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

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

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

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

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

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

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

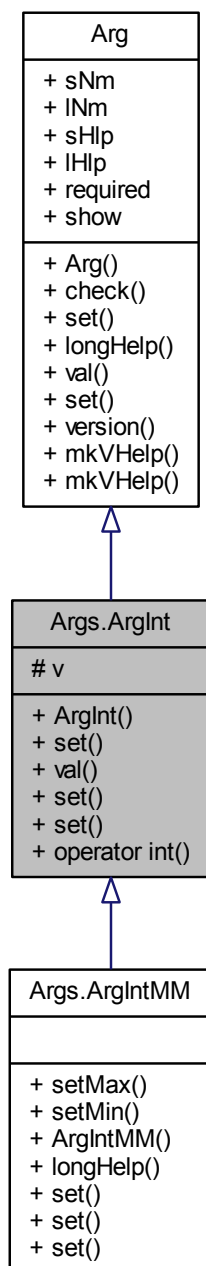
Объявления и описания членов класса находятся в файле:

- [cs/args.cs](#)

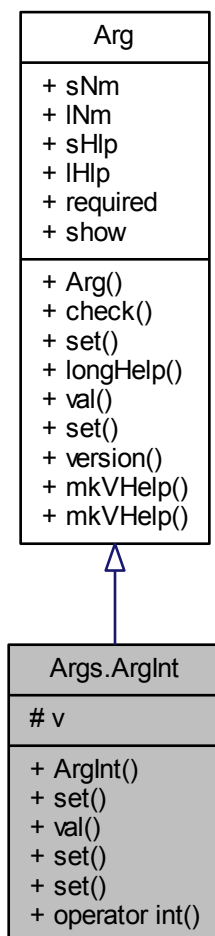
7.5 Класс Args.ArgInt

ключ целого типа

Граф наследования: Args.ArgInt:



Граф связей класса Args.ArgInt:



Открытые члены

- `ArgInt` (int `v`, string `sNm`, string `INm`, string `IHlp`=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` (int `v`)

Открытые статические члены

- static implicit `operator int` (`ArgInt` p)

Защищенные данные

- `int v`

Дополнительные унаследованные члены

7.5.1 Подробное описание

ключ целого типа

7.5.2 Конструктор(ы)

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

7.5.3 Методы

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

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

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

Переопределяется в [Args.ArgIntMM](#).

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

восстанавливает из строки значение ключа

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

Переопределяется в [Args.ArgIntMM](#).

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

Переопределяется в [Args.ArgIntMM](#).

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

функция переводит в строку значение ключа.

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

7.5.4 Данные класса

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

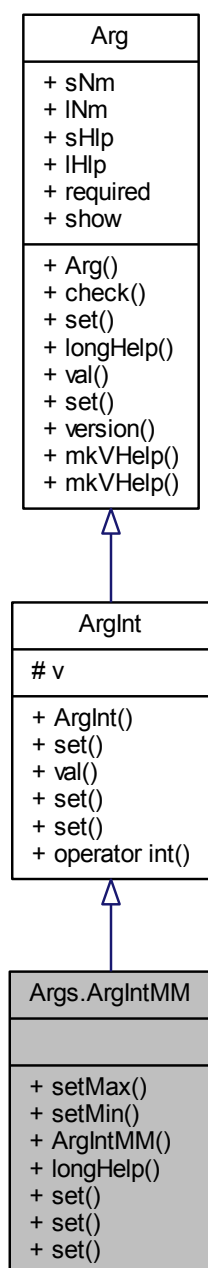
Объявления и описания членов класса находятся в файле:

- `cs/args.cs`

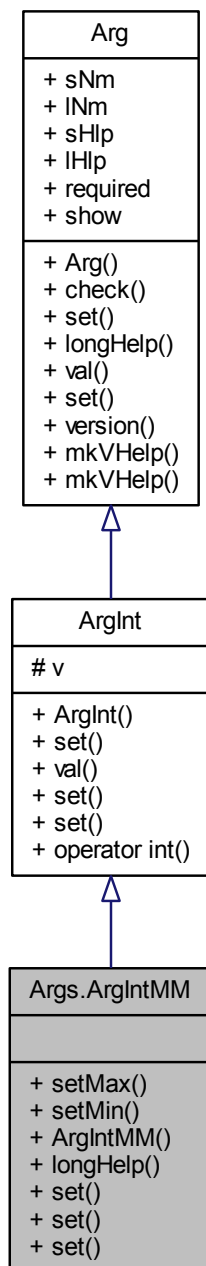
7.6 Класс Args.ArgIntMM

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

7.6.1 Подробное описание

ключ целого типа, есть возможность ограничить максимальное и минимальное значение

7.6.2 Конструктор(ы)

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

7.6.3 Методы

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

функция создает подсказку по ключу.

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

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

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

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

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

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

восстанавливает из строки значение ключа

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

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

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

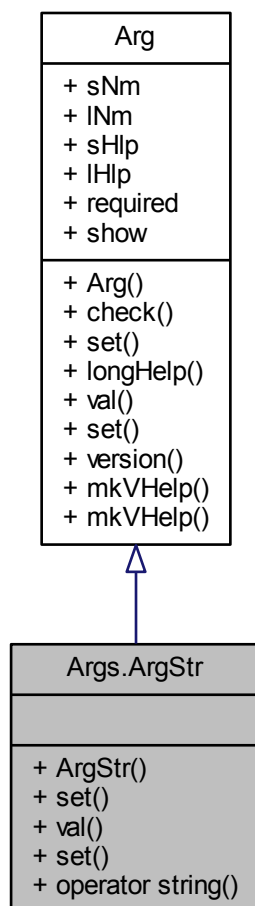
Объявления и описания членов класса находятся в файле:

- [cs/args.cs](#)

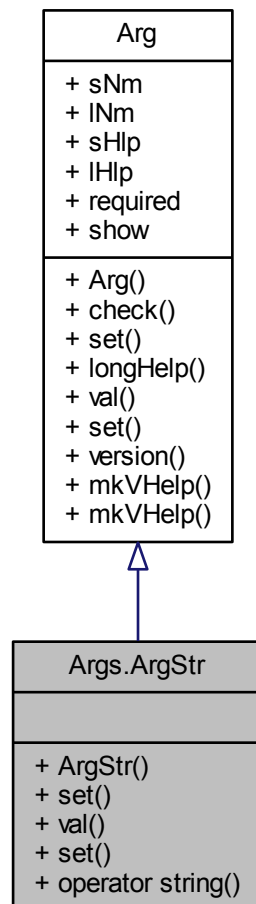
7.7 Класс Args.ArgStr

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)
- 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

7.7.1 Подробное описание

ключ типа строка

7.7.2 Конструктор(ы)

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

7.7.3 Методы

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

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

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

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

восстанавливает из строки значение ключа

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

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

функция переводит в строку значение ключа.

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

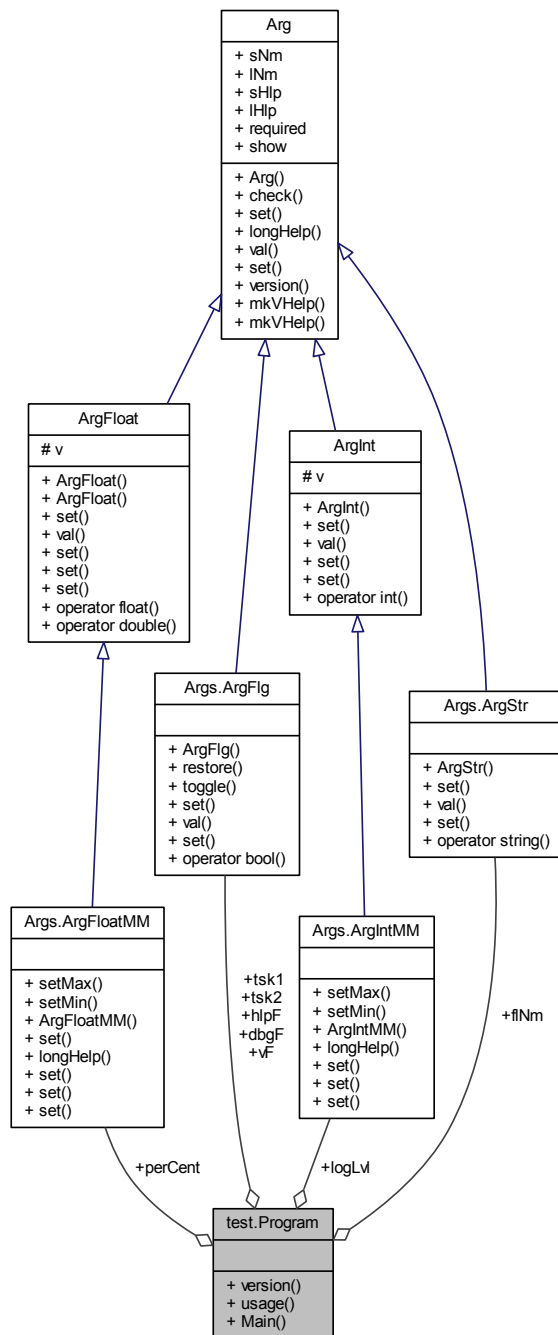
Объявления и описания членов класса находятся в файле:

- [cs/args.cs](#)

7.8 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 void [version](#) (out int major, out int minor, out int build)
- static void [usage](#) ()
- static void [Main](#) (string[] args)

Статические открытые данные

- static `ArgFlg hlpF`
флаг подсказки.
- static `ArgFlg dbgF`
флаг отладки
- static `ArgFlg vF`
флаг для дублирования журналирования в стандартный вывод
- static `ArgIntMM logLvl`
ключ целого типа
- static `ArgFloatMM perCent`
ключ плавающего типа
- static `ArgStr flNm`
ключ типа строка
- static `ArgFlg tsk1`
пример ключа для выбора работы1.
- static `ArgFlg tsk2`

7.8.1 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 и -t2 является обязательным. Они передаются в метод mkVHelp класса Arg во втором аргументе.

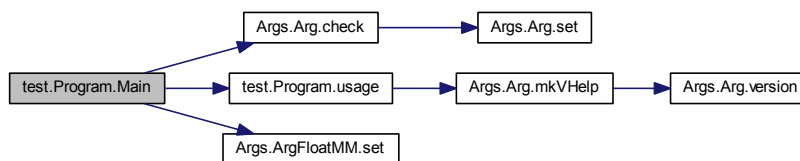
7.8.2 Методы

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

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

```
to test command line arguments (ver:2.1.0)
usage:
app [-?] [-d] [-v] [-l LLL] [-p PPP] -f FLNM -t1 | -t
options:
-?          : to see this help: True
-d          : debug mode: False
-v          : additional info: True
-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
'?' means the same as 'help'
'd' means the same as 'debug'
'v' means the same as 'verbose'
'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'
```

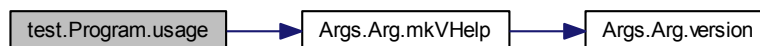
Граф вызовов:



7.8.2.2 static void test.Program.usage () [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`.

Граф вызовов:



Граф вызова функции:



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

7.8.3 Данные класса

7.8.3.1 ArgFlg test.Program.dbgF [static]

флаг отладки

7.8.3.2 ArgStr test.Program.flNm [static]

ключ типа строка

7.8.3.3 ArgFlg test.Program.hlpF [static]

[флаг](#) подсказки.

7.8.3.4 ArgIntMM test.Program.logLvl [static]

ключ целого типа

7.8.3.5 ArgFloatMM test.Program.perCent [static]

ключ плавающего типа

7.8.3.6 ArgFlg test.Program.tsk1 [static]

пример ключа для выбора работы1.

7.8.3.7 ArgFlg test.Program.tsk2 [static]

пример ключа для выбора работы2, один из ключей обязан быть в командной строке.

7.8.3.8 ArgFlg test.Program.vF [static]

флаг для дублирования журналирования в стандартный вывод

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.ArgFloat](#)
- class [Args.ArgFloatMM](#)
ключ строчного 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

- package [Args](#)

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 Program.cs

Классы

- class `test.Program`
класс содержит глобальные переменные для ключей, тест юнит Main и функцию выдачи подсказки

Пространства имен

- package `test`

Предметный указатель

Arg
 Args::Arg, 16
ArgFlg
 Args::ArgFlg, 22
ArgFloat
 Args::ArgFloat, 26
ArgFloatMM
 Args::ArgFloatMM, 30
ArgInt
 Args::ArgInt, 34
ArgIntMM
 Args::ArgIntMM, 37
ArgStr
 Args::ArgStr, 40
Args, 11
Args.Arg, 13
Args.ArgFlg, 19
Args.ArgFloat, 24
Args.ArgFloatMM, 27
Args.ArgInt, 31
Args.ArgIntMM, 34
Args.ArgStr, 37
Args::Arg
 Arg, 16
 check, 16
 IHlp, 18
 INm, 19
 longHelp, 17
 mkVHlp, 17
 required, 19
 sHlp, 19
 sNm, 19
 set, 18
 show, 19
 val, 18
 version, 18
Args::ArgFlg
 ArgFlg, 22
 operator bool, 22
 restore, 22
 set, 22
 toggle, 22
 val, 23
Args::ArgFloat
 ArgFloat, 26
 operator double, 26
 operator float, 26
 set, 26
 v, 27
 val, 26
Args::ArgFloatMM
 ArgFloatMM, 30
 longHelp, 30
 set, 30
 setMax, 30
 setMin, 31
Args::ArgInt
 ArgInt, 34
 operator int, 34
 set, 34
 v, 34
 val, 34
Args::ArgIntMM
 ArgIntMM, 37
 longHelp, 37
 set, 37
 setMax, 37
 setMin, 37
Args::ArgStr
 ArgStr, 40
 operator string, 40
 set, 40
 val, 40
AssemblyInfo.cs, 46

check
 Args::Arg, 16
cs/Assembly.txt, 45
cs/AssemblyInfo.cs, 45
cs/args.cs, 45
cs/args.txt, 45

dbgF
 test::Program, 43

flNm
 test::Program, 43

hlpF
 test::Program, 43

IHlp
 Args::Arg, 18
INm
 Args::Arg, 19
logLvl
 test::Program, 44
longHelp
 Args::Arg, 17

- Args::ArgFloatMM, [30](#)
 - Args::ArgIntMM, [37](#)
- Main
 - test::Program, [42](#)
- mkVHelp
 - Args::Arg, [17](#)
- operator bool
 - Args::ArgFlg, [22](#)
- operator double
 - Args::ArgFloat, [26](#)
- operator float
 - Args::ArgFloat, [26](#)
- operator int
 - Args::ArgInt, [34](#)
- operator string
 - Args::ArgStr, [40](#)
- perCent
 - test::Program, [44](#)
- Program.cs, [46](#)
- required
 - Args::Arg, [19](#)
- restore
 - Args::ArgFlg, [22](#)
- sHlp
 - Args::Arg, [19](#)
- sNm
 - Args::Arg, [19](#)
- set
 - Args::Arg, [18](#)
 - Args::ArgFlg, [22](#)
 - Args::ArgFloat, [26](#)
 - Args::ArgFloatMM, [30](#)
 - Args::ArgInt, [34](#)
 - Args::ArgIntMM, [37](#)
 - Args::ArgStr, [40](#)
- setMax
 - Args::ArgFloatMM, [30](#)
 - Args::ArgIntMM, [37](#)
- setMin
 - Args::ArgFloatMM, [31](#)
 - Args::ArgIntMM, [37](#)
- show
 - Args::Arg, [19](#)
- test, [11](#)
- test.Program, [40](#)
- test::Program
 - dbgF, [43](#)
 - flNm, [43](#)
 - hlpF, [43](#)
 - logLvl, [44](#)
 - Main, [42](#)
 - perCent, [44](#)
 - tsk1, [44](#)
 - tsk2, [44](#)
 - usage, [43](#)
 - vF, [44](#)
 - version, [43](#)
- toggle
 - Args::ArgFlg, [22](#)
- tsk1
 - test::Program, [44](#)
- tsk2
 - test::Program, [44](#)
- usage
 - test::Program, [43](#)
- v
 - Args::ArgFloat, [27](#)
 - Args::ArgInt, [34](#)
- vF
 - test::Program, [44](#)
- val
 - Args::Arg, [18](#)
 - Args::ArgFlg, [23](#)
 - Args::ArgFloat, [26](#)
 - Args::ArgInt, [34](#)
 - Args::ArgStr, [40](#)
- version
 - Args::Arg, [18](#)
 - test::Program, [43](#)