Namespace IbrahKit_CLI

Classes

CLI

CommandBase

Command<T, S>

Context

Class CLI

Namespace: IbrahKit_CLI

Assembly: IbrahKit-CLI.dll

```
public class CLI
```

Inheritance

object ← CLI

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Fields

FOLDER

```
public const string FOLDER = "IbrahKit"
```

Field Value

string

SPLIT_AT_SPACE

```
public const char SPLIT_AT_SPACE = ' '
```

Field Value

char

Methods

Run(string[])

```
public void Run(string[] args)
```

Parameters

args string[]

Class CommandBase

Namespace: [IbrahKit_CLI](#)

Assembly: IbrahKit-CLI.dll

```
public abstract class CommandBase
```

Inheritance

[object](#) ← CommandBase

Derived

[Command<T, S>](#)

Inherited Members

[object.Equals(object)](#) , [object.Equals(object, object)](#) , [object.GetHashCode()](#) , [object.GetType()](#) ,
[object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#)

Fields

ARG_PROCESS_SUCCES

```
public const string ARG_PROCESS_SUCCES = ""
```

Field Value

[string](#)

Methods

Execute()

Executes the command

```
public abstract string Execute()
```

Returns

[string↗](#)

A string that gets printed to the console

GetAllCommands()

Uses reflection to get the types of all commands

```
public static IEnumerable<Type> GetAllCommands()
```

Returns

[IEnumerable↗](#) <[Type↗](#)>

The types of all commands

GetData()

Returns command data

```
public abstract (string name, string desc, List<Argument> args) GetData()
```

Returns

([string↗](#) [name↗](#), [string↗](#) [desc↗](#), [List↗](#) <[Argument](#)> [args↗](#))

Returns name, description and a list of arguments that the command has

Parse()

Parses the incomming parameter

```
public abstract string Parse()
```

Returns

[string↗](#)

Returns an empty value if succcessful else it returns a non empty value

ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

[string](#)

A string that represents the current object.

Class Command<T, S>

Namespace: IbrahKit_CLI

Assembly: IbrahKit-CLI.dll

```
public abstract class Command<T, S> : CommandBase where T : Context, new() where S :
Command<T, S>
```

Type Parameters

T

S

Inheritance

object⬀ ← CommandBase ← Command<T, S>

Inherited Members

CommandBase.ARG_PROCESS_SUCCES , CommandBase.Execute() , CommandBase.GetData() ,
CommandBase.GetAllCommands() , CommandBase.ToString() , object.Equals(object)⬀ ,
object.Equals(object, object)⬀ , object.GetHashCode()⬀ , object.GetType()⬀ ,
object.MemberwiseClone()⬀ , object.ReferenceEquals(object, object)⬀

Constructors

Command(string[])

```
public Command(string[] args)
```

Parameters

args string⬀[]

Command(string[], T)

```
public Command(string[] args, T context)
```

Parameters

`args` [string](#)`[]`

`context` T

Fields

parameters

```
protected string[] parameters
```

Field Value

[string](#)`[]`

Methods

GetContext()

```
public T GetContext()
```

Returns

T

Parse()

Parses the incomming parameters

```
public override string Parse()
```

Returns

[string](#)

An empty string if successfull

Class Context

Namespace:

Assembly: IbrahKit-CLI.dll

```
public abstract class Context
```

Inheritance

[object](#) ← Context

Inherited Members

[object.Equals(object)](#) , [object.Equals(object, object)](#) , [object.GetHashCode()](#) , [object.GetType()](#) , [object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#) , [object.ToString()](#)

Namespace IbrahKit_CLI.Exceptions

Classes

[ArgumentParsingException](#)

    Thrown during the Command.Parse method

[CommandExecutionException](#)

Class ArgumentParsingException

Namespace:

Assembly: IbrahKit-CLI.dll

Thrown during the Command.Parse method

```
public class ArgumentParsingException : Exception, ISerializable
```

Inheritance

object ← Exception ← ArgumentParsingException

Implements

ISerializable

Inherited Members

Exception.GetBaseException() , Exception.GetType() , Exception.ToString() , Exception.Data ,
Exception.HelpLink , Exception.HResult , Exception.InnerException , Exception.Message ,
Exception.Source , Exception.StackTrace , Exception.TargetSite , Exception.SerializeObjectState ,
object.Equals(object) , object.Equals(object, object) , object.GetHashCode() ,
object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

ArgumentParsingException()

```
public ArgumentParsingException()
```

ArgumentParsingException(string?)

```
public ArgumentParsingException(string? message)
```

Parameters

message string

Class CommandExecutionException

Namespace: [IbrahKit_CLI](#).[Exceptions](#)

Assembly: IbrahKit-CLI.dll

```
public class CommandExecutionException : Exception, ISerializable
```

Inheritance

[object](#) ← [Exception](#) ← CommandExecutionException

Implements

[ISerializable](#)

Inherited Members

[Exception.GetBaseException()](#) , [Exception.GetType()](#) , [Exception.ToString()](#) , [Exception.Data](#) ,
[Exception.HelpLink](#) , [Exception.HResult](#) , [Exception.InnerException](#) , [Exception.Message](#) ,
[Exception.Source](#) , [Exception.StackTrace](#) , [Exception.TargetSite](#) , [Exception.SerializeObjectState](#) ,
[object.Equals(object)](#) , [object.Equals(object, object)](#) , [object.GetHashCode()](#) ,
[object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#)

Constructors

CommandExecutionException()

```
public CommandExecutionException()
```

CommandExecutionException(string?)

```
public CommandExecutionException(string? message)
```

Parameters

`message` [string](#)

Namespace IbrahKit_CLI.Params

Classes

[Argument](#)

An extension of param. Provides an additional check if there is another incomming value in the args array

[Flag](#)

[Param](#)

Class Argument

Namespace: IbrahKit_CLI.Params

Assembly: IbrahKit-CLI.dll

An extension of param. Provides an additional check if there is another incomming value in the args array

```
public class Argument : Param
```

Inheritance

object ← Param ← Argument

Inherited Members

Param.names , Param.description , Param.function , Param.GetName(int) , Param.ToString() ,
Param.TryCompare(string, out int, out int) , Param.Pass<S, T>(string[], Context, int) ,
Param.GetClosestParam(string, IEnumerable<Param>) , object.Equals(object) ,
object.Equals(object, object) , object.GetHashCode() , object.GetType() ,
object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

Argument(Func<string[], string>, string, params string[])

```
public Argument(Func<string[], string> function, string description, params string[] names)
```

Parameters

`function` Func<string[], string>

`description` string

`names` string[]

Methods

Pass<T, S>(string[], Context)

Passes the remaining arguments to a new instance of a

```
public override string Pass<T, S>(string[] args, Context cont) where T : Context, new()
where S : Command<T, S>
```

Parameters

args string[]

cont Context

Returns

string

Type Parameters

T

S

Process(string[])

Proccesses the passed input

```
public override string Process(string[] args)
```

Parameters

args string[]

The remaining unproccessed arguments

Returns

string

A string value. Empty Means success and non-empty means failiure

Class Flag

Namespace:

Assembly: IbrahKit-CLI.dll

```
public class Flag : Param
```

Inheritance

object ← Param ← Flag

Inherited Members

Param.names , Param.description , Param.function , Param.GetName(int) , Param.ToString() , Param.TryCompare(string, out int, out int) , Param.Pass<S, T>(string[], Context, int) , Param.GetClosestParam(string, IEnumerable<Param>) , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

Flag(Func<string[], string>, string, params string[])

```
public Flag(Func<string[], string> function, string description, params string[] names)
```

Parameters

`function` Func<string[], string>

`description` string

`names` string[]

Methods

Pass<T, S>(string[], Context)

Passes the remaining arguments to a new instance of a

```csharp
public override string Pass<T, S>(string[] args, Context cont) where T : Context, new()
    where S : Command<T, S>
```

Parameters

`args` [string](#)[]

`cont` [Context](#)

Returns

[string](#)

Type Parameters

`T`

`S`

Process(string[])

Proccesses the passed input

```csharp
public override string Process(string[] args)
```

Parameters

`args` [string](#)[]

The remaining unproccessed arguments

Returns

[string](#)

A string value. Empty Means success and non-empty means failiure

Class Param

Namespace: IbrahKit_CLI.Params

Assembly: IbrahKit-CLI.dll

```
public abstract class Param
```

Inheritance

object ← Param

Derived

Argument, Flag

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() ,
object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

Param(Func<string[], string>, string, params string[])

```
public Param(Func<string[], string> function, string description, params string[] names)
```

Parameters

`function` Func<string[], string>

`description` string

`names` string[]

Fields

description

```
protected string description
```

Field Value

[string](#)

function

```
protected Func<string[], string> function
```

Field Value

[Func](#) <[string](#)[], [string](#) >

names

```
protected string[] names
```

Field Value

[string](#)[]

Methods

GetClosestParam(string, IEnumerable<Param>)

Returns a param with the closest name to what was entered

```
public static string GetClosestParam(string input, IEnumerable<Param> @params)
```

Parameters

`input` [string](#)

The input to check all params against

`params` [IEnumerable](#) <[Param](#)>

The params to check the input against

Returns

[string](#)↗

  The closest Param

GetName(int)

```
public string GetName(int index)
```

Parameters

`index` [int](#)↗

Returns

[string](#)↗

Pass<T, S>(string[], Context)

Passes the remaining arguments to a new instance of a

```
public abstract string Pass<T, S>(string[] args, Context cont) where T : Context, new()
where S : Command<T, S>
```

Parameters

`args` [string](#)↗[]

`cont` [Context](#)

Returns

[string](#)↗

Type Parameters

`T`

S

Pass<S, T>(string[], Context, int)

```
protected string Pass<S, T>(string[] args, Context cont, int amountSkip) where S :
Command<T, S> where T : Context, new()
```

Parameters

args string []

cont Context

amountSkip int

Returns

string

Type Parameters

S

T

Process(string[])

Proccesses the passed input

```
public virtual string Process(string[] args)
```

Parameters

args string []

  The remaining unproccessed arguments

Returns

[string ⧉](#)

A string value. Empty Means success and non-empty means failiure

ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

[string ⧉](#)

A string that represents the current object.

TryCompare(string, out int, out int)

```
public bool TryCompare(string input, out int closestValue, out int closestIndex)
```

Parameters

input [string ⧉](#)

closestValue [int ⧉](#)

closestIndex [int ⧉](#)

Returns

[bool ⧉](#)