

COMMAND LINE PARSER

C#

Michal Kolcun & Michael Ševčík

USE CASES AND FEATURES

- Common parsing of a command line argument input.
 - `--file=1.txt --access-type=read`
- Support of both named options and typed plain arguments.
 - E.g. “-o” and “--option” – short and long names.
 - Allows mixing of plain arguments and options.
- Extendable with user-defined types via predefined interfaces.

API

Key Concepts:

1. Interfaces representing Options
 - IOption
 - IParametrizedOption
 - IMultipleParameterOption
2. Interface representing Plain Argument(s)
3. Parser class

TIME COMMAND

-f FORMAT, --format=FORMAT

Specify output format, possibly overriding the format specified in the environment variable TIME.

```
enum Format : byte
{
    format1,
    format2,
    format3
}
```

```
Action<Format?> formatAction = format => Console.WriteLine(format);
var FormatOption = IParametrizedOption.CreateParameterOption<Format?>
    (formatAction, false, true, new char[] { 'f' }, new string[] { "format" });
FormatOption.SetHelpString("HS...");
```

--help Print a usage message on standard output and exit successfully.

```
Action markHelp = () => Console.WriteLine("was Help");
var helpOption = IOption.CreateNoParameterOption(markAppend, false, null, new string[] { "help" });
helpOption.SetHelpString("(Used together with -o.) Do not overwrite but append.");
```

TIME COMMAND

```
// Create Parser.  
Parser parser = new();  
parser.SetPlainArgumentHelpString("Terminate option list.");  
  
// Fill parser with already created options.  
parser.Add(FormatOption);  
parser.Add(portabilityOption);  
parser.Add(outputFileOption);  
parser.Add(appendOption);  
parser.Add(verboseOutputOption);  
parser.Add(helpOption);  
  
// Parse command-line input.  
parser.ParseCommandLine(args);
```

HIGHLIGHTS




- Factory methods for creating option instances

```
public static IOption CreateNoParameterOption(  
    Action action,  
    bool isMandatory,  
    char[]? shortSynonyms = null,  
    string[]? longSynonyms = null  
)
```

- Easily extensible for your own types of options and plain arguments!

```
0 references  
public interface IOption{  
    0 references  
    public bool IsMandatory { get; }  
    0 references  
    public char[]? ShortSynonyms { get; }  
    0 references  
    public string[]? LongSynonyms { get; }  
    0 references  
    public string HelpString { get; set; }  
    0 references  
    public void TakeAction();  
}
```

REVIEW POINTS

- Minor naming conventions errors. 
- Minor errors in documentation, due to ongoing development and missing implementation. 
- Suggestion to use exceptions for error handling.
 - Considered but not used. 
- Suggestion to start the Readme with “quick-start” section. 