🌊🦈

CShargs

argument parser

Evgenia Golubeva
Jan Kytka

Overview: declarative approach with attributes

```
void Main(string[] args) {
    var arguments = new TimeArguments();
    arguments.Parse(args);

    // check version option
    if (arguments.Version) {
        Console.WriteLine("Version option present.");
    }

    if (arguments.Help) {
        // generate structured help, write it to console
        arguments.GenerateHelp(Console.Out);
    }

    // get parsed plain arguments
    var plainArgs = arguments.PlainArgs;
}
```

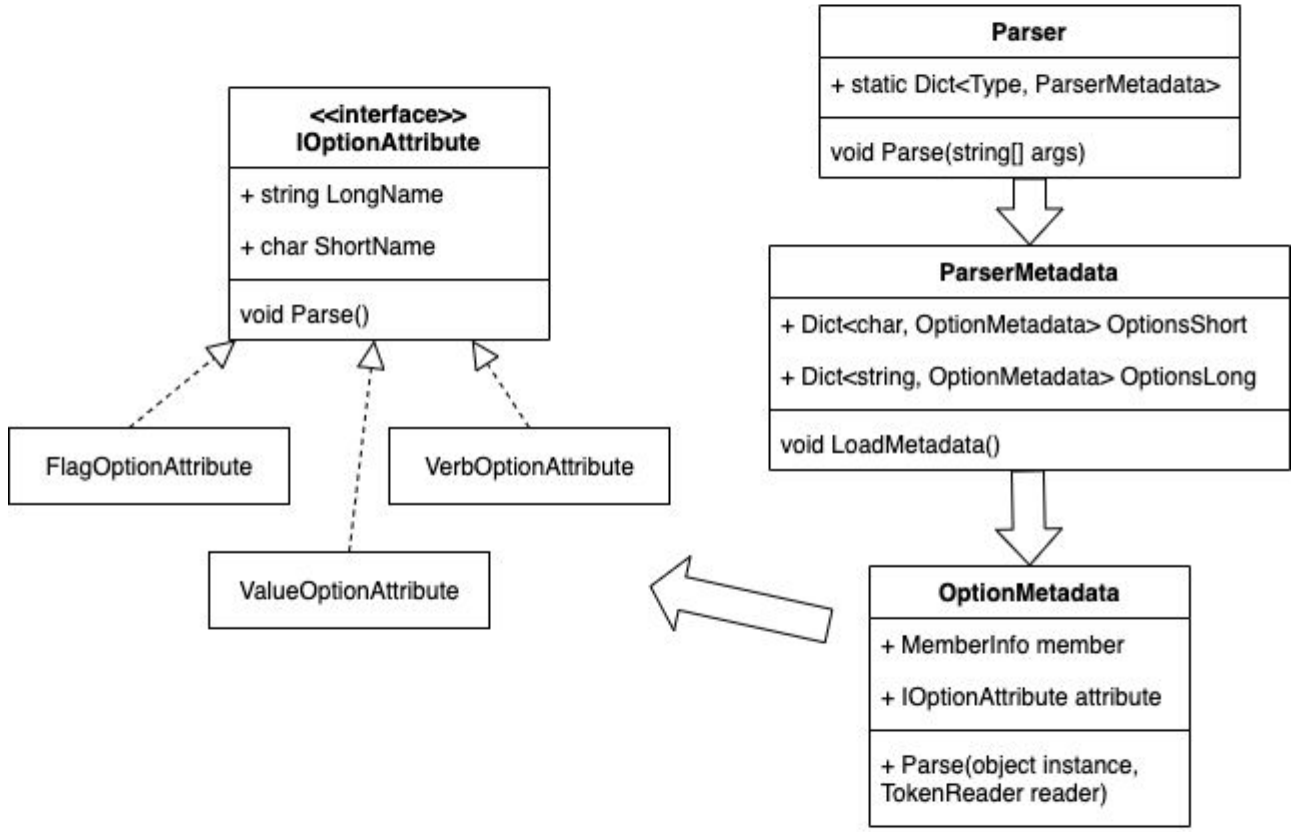
```
class TimeArgumentsParser : CShargs.Parser {

    // time  -V, --version
    [FlagOption("version", shortName: 'V', help: "Print version information.")]
    public bool Version { get; set; }

    // time --output=FILE
    [ValueOption("output", shortName:'o', required: false, help: "Do not send the results to stderr, but overwrite the
specified file.")]
    public string OutputFile { get; set; }

    // time --output=FILE -a
    [FlagOption("append", shortName:'a', useWith: nameof(OutputFile), help: "Do not overwrite but append.")]
    public bool Append { get; set; }

}
```

Overview

2 phases:

- Metadata creation using reflection
    - Create option metadata from property annotations in parser object
    - Do static checks on parser definition - unique option names etc.
- Parsing
    - Use metadata to parse cli arguments - execute user-defined parsers and methods
    - Do dynamic checks: argument dependencies, required arguments, etc.

Top down data flow



<<interface>>
IOptionAttribute
+ string LongName
+ char ShortName
void Parse()
FlagOptionAttribute
VerbOptionAttribute
ValueOptionAttribute
Parser
+ static Dict<Type, ParserMetadata>
void Parse(string[] args)
ParserMetadata
+ Dict<char, OptionMetadata> OptionsShort
+ Dict<string, OptionMetadata> OptionsLong
void LoadMetadata()
OptionMetadata
+ MemberInfo member
+ IOptionAttribute attribute
+ Parse(object instance, TokenReader reader)

Checking rules

- How to check all rules defined in parser class
    - Required options
    - Exclusive groups of options
    - Option dependencies

- Ideal abstraction vs performance

Tests

- Tests we got:
    - API tests
    - Static initialization
- Integration - different frameworks
    - xUnit
    - nUnit
- Need more tests for implementation side