ECAM
BRUSSELS ENGINEERING SCHOOL

Winter 2020
Sébastien Combéfis

PL190μ Introduction to the Go Programming Language

Coding 5: Command interpreter

This assessment evaluates the following competencies:

- *GP001 – Correctly use the syntax of Go programming* *(+1)*
- *GP101 – Write, compile and execute a single source file Go program with the command line* *(+1)*
- *GP002 – Use basic built-in data structure: array, slice and map* *(+1)*
- *GP301 – Handle rigorously the errors when calling function* *(+2)*
- *GP004 – Use functions from the Go standard library given their specification* *(+2)*

You may also be assessed on the following competencies:

- *GP401 – Understand basic Go compiler errors and warnings and fix the code accordingly* *(+1)*
- *GP003 – Manipulate structures and define methods manipulating them* *(+1)*

In this coding assessment, you have to fix and complete an existing Go program that implements a basic interpreter for commands to be defined [1]. The actual code does compile, and implements three commands: `exit` quits the program, `echo` prints the argument and `sum` computes the sum of all the numbers separated by spaces. First, you have to understand the code and fix two issues:

- the `exit` command is not working;
- the `sum` command has not been implemented correctly.

Then, you have to improve the quality of the code by better using the features of the Go programming language. You have to work on two elements:

- improve the code that parses the commands and call the function corresponding to the entered command (you may use a `map` of functions or structures and interfaces, for example);
- and provide a better way to manage errors that can be raised by the functions executing the commands, by changing the return type to `(string, error)` and correctly handling the errors in the `main` function and in the `splitLine` function.

To succeed the assessment, you have to explain to the teacher how you fixed and improved the code. Then, you have to implement a new command, in live, following the specifications given by the teacher, compile the program and run it with the command line. For example, you could be asked to:

- implement a command computing the factorial of a natural number;
- implement a command to search whether a given string is a substring of another one;
- ...

[1] The code can be found here: https://github.com/ukonline/uCourse/blob/master/PL190%C2%B5/code/cli-interpreter.go