DPL

(P1)

Yeray Méndez Romero

[yeray.mendez@udc.es](mailto:yeray.mendez@udc.es)

Daniel Rivera López

[d.rivera1@udc.es](mailto:d.rivera1@udc.es)

Selected languages:

-OCaml

-Python

-Go

-Kotlin

-Groovy

-Lua

OCaml:

a)

-Compiler: version 4.02.3

-System: Ubuntu 16.04 LTS

b)

OCaml is an industrial strength programming language supporting functional, imperative and object-oriented styles and have automatic memory management (garbage collector). The OCaml source code can be compiled for a virtual machine or compiled to machine code for many different architectures, which is our case, in this last compilation option it runs with an efficiency similar to C/C++. OCaml also have a static type analysis with type inference, with a strong type system, with first class functional values, parametrized polymorphism, pattern calling, exception handling and other advanced features.

c)

\*Advantages:

- User custom types which make us able to stay more faithful to the original implementation and avoid to make use of OO.

-The memory management is done automatically by the OCaml which release us from explicit management of the memory.

-Explicit pointer management which make us able to stay more faithful to the original implementation.

-Allow imperative programming which make us able to stay more faithful to the original implementation

\*Disadvantages:

d)

Because all OCaml function need to return a value we were forced to return unit in attempt to maintain the essence of the original implementation

-The error function of the original implementation no longer exists in the OCaml implementation due to the automatic memory management of OCaml

e)

- Iterative methods force us to lose the functional potential of OCaml.

Python:

a)

-Interpreter: Python 3.4.4 – Python 3.5.2

-System: Windows 10 – Ubuntu 16.04

b)

Python is a simple and easy to learn high level programming language which supports multiple programming paradigms including Object-Oriented, imperative, functional programming and procedural styles. Python also features a dynamic and strong typing and automatic memory management. Python is an interpreted programming language which makes python programs much more portable than a compiled language. Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords)

c)

\*Advantages:

-Python allows imperative programming which make us able to do the iterative methods.

-The memory management is done automatically by the Python memory manager which release us from explicit management of the memory.

\*Disadvantages:

-Python does not allow the creation of custom data types without importing a library and because of that the custom type is made by objects.

d)

-As we previously said we were forced to use a class and objects from it to mimic the custom type of Pascal, otherwise we would have needed to import a library.

-The error function of the original implementation no longer exists in the Python implementation due to the automatic memory management of Python.

e)

-If we could import libraries we could have lower memory usage during execution against the actual object oriented version.

Go:

a)

-Interpreter: go1.9 windows/amd64 – go1.6.2 linux/amd64

-System: Windows 10 – Ubuntu 16.04

b)

Go It is a compiled, imperative, Object-Oriented, statically typed language, with garbage collection, limited structural typing, memory safety features and style concurrent programming features added. The compiler and other language tools originally developed by Google.

c)

\*Advantages:

- User custom types which make us able to stay more faithful to the original implementation and avoid to make use of OO.

-The memory management is done automatically by the Go memory manager which release us from explicit management of the memory.

-Because Go is imperative, the structure of the functions is similar to the original.

-The explicit pointer management make us able to no return parameters as it was in the original implementation.

\*Disadvantages:

d)

-The error function of the original implementation no longer exists in the Go implementation due to the automatic memory management of Go.

-The library “fmt” is imported into the program to make us able to print values on the screen.

e)

Kotlin:

a)

-Interpreter: Kotlin version 1.1.50 (JRE 1.8.0\_92-b14)

-System: Windows 10 – Ubuntu 16.04

b)

Kotlin is a statically typed, functional, Object-Oriented programming language that targets the JVM (Java Virtual Machine), Android, JavaScript and Native. The memory management on Kotlin is automatically done by the JVM if it is compiled to a .jar file and will enable different solutions to memory management when compiled to native in the future, because the native compiler is in a pre-release state. Kotlin also features nullable types to prevent null pointer exceptions.

\*Advantages:

-The memory management is done automatically by the JVM garbage collector which release us from explicit management of the memory.

-Kotlin is imperative as Pascal and because of that the implementation is similar.

\*Disadvantages:

-Kotlin force the programmer to use objects in order to create custom data types.

d)

-The error function of the original implementation no longer exists in the Kotlin implementation due to the automatic memory management of the JVM.

e)