2.3 Programming Paradigm

Return to DIDO CLI Background

Programming paradigms are a way to classify programming languages based on their features and characteristics. For example, is the language Imperative or Declarative in nature.

- *An **Imperative Language** uses a sequence of statements to determine how to reach a certain goal. These statements are said to change the state of the program as each one is executed in turn.* [1)]
- ***Declarative languages**, also called **Non-Procedural** or very high level, are programming languages in which (ideally) a program specifies what is to be done rather than how to do it. In such languages there is less difference between the specification of a program and its implementation than in the procedural languages described so far.*[2)]

An import aspect of any programming language is to determine the programming paradigm(s) features needed to properly and meet the requirements of the language. The DIDO-CLI is no exception. In order to discuss the programming paradigm for DIDO-CLI, a brief overview of the kinds of (i.e., classification) of some of the most common paradigms is presented.

Some paradigms are place a lot of emphasis on the implications for the execution model of the language, such as allowing side effects, or whether the sequence of operations is defined by the execution model. Other paradigms are concerned mainly with the way that code is organized, such as grouping a code into units along with the state that is modified by the code. Yet others are concerned mainly with the style of syntax and grammar.

There are many kinds of Programming Paradigms[3)]. Only those that are considered relevant to the DIDO-CLI are reviewed here.

- Procedural Programming
- Object Oriented Programming
- Functional Programming

The DIDO-CLI is expected to be a hybrid of these paradigms.

Paradigm Details

- 2.1.3.1 Procedural Programming
- 2.1.3.2 Object Oriented Programming
- 2.1.3.3 Functional Programming
- 2.1.3.4 Hybrid of Functional and Procedural Languages

1)

user2102654, Stackoverflow, What is difference between functional and imperative programming languages?, 24 July 2013, Accessed 12 April 2021, https://stackoverflow.com/questions/17826380/what-is-difference-between-functional-and-imperative-pro

gramming-languages
2)

Encyclopedia Britannica, Accessed: 12 April 2021, https://www.britannica.com/technology/computer-programming-language/Visual-Basic#ref849838
3)

GeeksForGeeks, Difference between Procedural and Non-Procedural language, Accessed: 15 April 2021, https://www.geeksforgeeks.org/difference-between-procedural-and-non-procedural-language/