**Programming Languages Report**

Nikol Kirilova Stoyanova, Student ID: 17069455

**Introduction**

The aim of this report is to compare the following programming languages Smalltalk, Python, JavaScript, Clojure and Haskell, and their key concepts. As a result, this will conclude in an evaluation and reflection of the studied languages as well as a brief discussion of working with different languages.

**Comparison of programming languages**

* **Smalltalk**

Smalltalk is a pure Object-oriented programming language or in other words, objects, classes, control structures, etc. are accepted as objects. The instance variables are always protected meaning that they can be accessed only by the subclasses of the class in which they have been declared. The variables are type-less which allows the language to be dynamically typed. Moreover, Smalltalk is reflective – it can inspect its own structure. Smalltalk is a class-based system. It has meta structure – all class methods belong to it and all meta classes are subclasses of Class. The class objects act as templates and support single inheritance. Smalltalk objects are instantiated on the heap and use garbage collection and implicit de-referencing. They can hold a state – reference to variables and receive or send messages from/to other objects.

* **Python**  
  Python is another Object-oriented programming language. Python is a multiparadigm programming language - generic, reflective and structured. It is also interactive and interpreted – allowing testing programming features while programming **[1]**. Some of its features support functional programming as well - the language supports functions as map, filter and reduce, and types as lists, sets and dictionaries **[2]**. Python has an automatic garbage collection. All basic types and containers are objects. Python also has control structures. The language is dynamically typed as the variables are not declared – they are type-less. In Python if a variable is only referenced in a function it is a global one. However, if the variable is assigned a value within a function then it is local unless declared global. One of the most important features of Python is that the code emphasis is on readability. As Python is Off-side rule language this is achieved by indentation of the code.
* **JavaScript**  
  JavaScript is a well-recognised dynamically typed object-based scripting language. It is a multiparadigm language as it is imperative, interpreted, event-driven and reflective. When classified, JavaScript not only supports features of Object-oriented programming but also ones of Functional programming. It is also partially concurrent as it consists of promises for handling asynchronous tasks. The language is weakly typed which means that it might produce implicit type conversions and imitates polymorphism. Variables’ types are not declared. JavaScript also has garbage-collection. It also implements weak encapsulation with getters and setters.
* **Clojure**  
  Clojure is an almost pure functional programming language – it is not Object-oriented. The language is declarative as it attempts to minimise the side-effects. It focuses on parallelism and concurrency - it has multi-threads and the computations are executed not in sequence but during overlapping time periods. A prominent feature of Clojure is that the programming language has immutable data structures. Most importantly, Clojure is a List-based language as it is based on the list type data structure. It is an Interactive mode language - it is reflective as it allows you to enter commands and inspect the results. Moreover, Clojure can be dynamically compiled **[3]** and is garbage-collected.
* **Haskell**  
  Haskell is a pure functional programming language - all its data types are immutable. The language similarly to Clojure is also declarative – it allows no side-effects and is concurrent. Haskell does not have any constructs such as while and for. It is statically typed and garbage-collected **[4]**. Haskell is also generic, literate and off-side rule language. Indentation is important as code would not be executed otherwise. The programming language is classified as Lazy. This means that Haskell delays the evaluation of an expression until its value is needed and avoids repeated evaluations **[5]**. Haskell also allows re-definition of identifiers **[6]**. Other features that Haskell has are pattern matching, scoped type variables and monads.

In conclusion, all five languages are multiparadigm ones. All of them are garbage-collected and are interactive mode languages – expressions can be entered one at a time and the result can be evaluated immediately. Smalltalk, Python and Haskell are compiled languages which means that they are typically processed by compilers. All four languages excluding Haskell are reflective. While JavaScript and Clojure are impure functional programming languages, Haskell is pure, whereas Smalltalk and Python are object-oriented. However, Python can also implement functional programming. JavaScript and Smalltalk do not multiple inheritance in contrast with Python. Smalltalk and Clojure are more orthogonal than Haskell, which makes their readability, writability and reliability better. Clojure is considered highly expressive. The same can be said about Haskell. However, Python is less expressive, whereas JavaScript is considered highly inexpressive. Python and JavaScript are easy to read languages as they are off-side rule-based meaning that indentation is important. JavaScript and Clojure on the other hand use brackets. All five languages are categorised as High-Level programming languages which means that it is easier to understand what a command in the code does. While object-oriented languages use mutable data, functional ones use immutable data.

According to Stack overflow Developer Survey in the year 2019, JavaScript and Python are the most popular programming languages. On the other hand, Clojure is second to last **[7]**.

**Evaluation and Reflection**

I found very easy to use JavaScript as I have coded in JavaScript before. Because it is object-based language and has object-oriented characteristics and is weakly-typed it was easier to understand and implement solutions using it. I found Smalltalk easy to understand even though understanding the Pharo environment was a bit hard in the beginning. Python was the language I really enjoyed using and found as easy as JavaScript to use. What I found useful in Smalltalk was that everything is an object that can be used only in the class it is declared and its subclasses. That made programming enjoyable. Moreover, Python doesn’t have too much punctuality which made typing and coding more practical and faster.

On the other hand, Clojure and Haskell proved to be difficult for me as they are functional, and I have never programmed with functional languages before. Moreover, I found particularly infuriating the many brackets in Clojure which made coding in the language harder than in all the other four. Haskell was hard to use because of the pattern matching. However, I liked Haskell as similarly to Python it had indentation rule which made reading the code and checking it for mistakes easier.

I found that I am a very good programmer using object-oriented languages. However, I need to improve my functional programming as it took more time to develop the solutions for the assignment tasks while using Clojure and Haskell.

Overall, both functional and object-oriented languages helped me understand why they are different and what their usage is.

**Discussion**

In conclusion, object-oriented and functional languages have their differences. While functional languages can be used for manipulating data through developing functions, in object-oriented programming data is stored into objects. However, both languages try to provide codes that are bug-free, easily understandable and well managed. I think working with all these programming languages will be highly helpful in my career as I will be able to identify which language to use to produce the best programming solution. I believe experience in Python and JavaScript is essential for most jobs and programming using them can be beneficial for my career development. However, I wouldn’t disregard Clojure and Haskell as they might prove useful. I would like to learn Ruby because it has easy syntaxis and implements procedural programming similarly as JavaScript as well as object-oriented and functional programming.

**References:**

1. The Python Software Foundation (2020), “General Python FAQ”, [Online] Available at: <https://docs.python.org/3/faq/general.html#what-is-python>
2. Hettinger, Raymond (2002). "PEP 289 – Generator Expressions". Python Enhancement Proposals. Python Software Foundation, [Online] Available at: <https://www.python.org/dev/peps/pep-0289/>
3. Rich Hickey (2019), “Dynamic Development”. Clojure, [Online] Available at: <https://clojure.org/about/dynamic>
4. Dr Cunningham, R. (2019), “Haskell”, Manchester Metropolitan University
5. David Anthony Watt; William Findlay (2004). Programming language design concepts. John Wiley and Sons. pp. 367–368.
6. Dr Cunningham, R. (2019), “Haskell II”, Manchester Metropolitan University
7. “Most popular Technologies” (source: Stack Overflow, 2019: online), <https://insights.stackoverflow.com/survey/2019#most-popular-technologies>