JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 7, ISSUE 08, 2020

PERFORMANCE ANALYSIS OF OBJECT ORIENTATION OF HASKELL AND JAVA LANGUAGES

Kongu Vel. S, A. Kumaravel

[1]Research Scholar, Department of Management Science and Humanities, Bharath Institute of Higher Education and Research, India. E-mail: kongunkl@gmail.com

[2]Professor, Department of Information Technology, Bharath Institute of Higher Education and Research, India. E-mail: drkumaravel@gmail.com

ABSTRACT

Object Oriented Language is the familiar concepts in the programming environment.The core of Object-Oriented Programming is the object. OOPS language is widely used for creating larger projects.A class is a model, a blue print and describes the common features.**]**A class contains Functions, variables called fields, and behaviors.Objects are called instances of the class.**[**Haskell programming can support Functional concepts as well as object-oriented concepts. It supports type class bounded and to get different parameter-based polymorphism as same like Java and Dot net-orientedlanguages. Haskell supports for the OOPS concept of encapsulation,mutable state, inheritance,overriding.We are taking two languages of Haskell and Java for parameters for testing.We implement Object oriented concepts to both languages and process the programmed for getting result. Based on that result,we conclude which language are working faster than others.

Keywords: Haskell, Java, Object Oriented Programming, Time complexity.

1. INTRODUCTION

Programming languages are restraint entire behavior of computer hardware devices.Several languages are gone out of date and some languages are used on developing environment.These kinds of programming languages become very familiar, because of easy to learning,efficiency and power of expression.In this task we are taking two kinds of programming languages.We taking measurement of two parameters in memory consumption and running time requirement are tested.

Object Orientation fundamentals are widely used concepts in programming environment.

Most of the programming languages are used to concept of Object Orientation for developing code.[5] Objects are blue print of classes. This real-world object holds two characteristics.ie behavior and state. This characteristic is access entire programmed. It is holding the entire class of behavior and state with in itself.Software objects are sculptural is additionally having state and behavior. A computer code objects maintains its state in one or additional variables and implements its characteristics in strategies.

Encapsulation canrestrict the information from the outside world and binding together related data and methods. So, itguarantees that the individual [2] [4] object can be completely autonomous from external interference. Inheritance is used to share the characteristics of the object and behaviors. Inheritance is used to avoid repetition and access the details of the common futures in a common class. Repetition makes the code for getting errors. If you are using inheritance concepts, eliminates so many lines of code for the same purpose in the programme. So, execution time of code faster than we make the whole programme.

Common class are called Parent Class. From the parent class,we inherited properties and behaviors for the other class. Interface also known as Protocol,is an alternative to inheritance for two unrelated [3] classes to communicate with each other. Same operation of objects is using different classes. Function Overloading describes same name with different arguments to pass to a Function and each function getting different output of that particular of this

ISSN- 2394-5125 VOL 7, ISSUE 08, 2020

programme. Polymorphism in a class usually Child class is redefined for the entire parenting class for the wherever inherited.

Haskell supports Functional language as well as OOPS Concepts. It supports type class bounded and to get different parameter-based polymorphism as same like Java and Dot Net-oriented languages. Whether Haskell supports for the OOPS concept of encapsulation, mutable state inheritance,overriding.

In an intellectual sense,Object type system,inheritance,subtyping,virtual methods are adaptable for the entire advanced Haskell system.In a[8] practical point of view,one may wonder practically whether we can successfully transfer entire concepts of the Object to Haskell environment.Because of Haskell execution engine is supported to highly supported techniques. From language style perspective, Haskell encompasses a sturdy record in prototyping and encryption abstraction mechanisms, whatever new idea comes to implements and write and modify a compiler.In an educational point of view,both developers are used concepts of functional and objects-oriented concepts in the same language and used Haskell type systems.

Classes are category as a function that generates objects and state are maintained throughout entire life cycle of programme. Methods are able to access programme state and [6] behavior through by the object.One of the foremost powerful options in Haskell are higher order functions and first-class monadic values. These optionsmake it powerful language for Haskell.In this feature are implemented to other language are very difficult.That is the very challenge to other languages.

In Haskell point of view,[7] object-oriented concepts are difficult to implement in the programming environment.Object familiarized languages are create international use of implicit intimidation between a subtype and supertypes. Although Haskell lack of thought of subtyping.Haskell additionally supported Overloading ideas. All the over laden behavior and state should share a typical kind pattern. In comparisonof different object bound languages permit one methodology name to be overladen at unrelated sorts.

2. MATERIALS AND METHODS

2.1 Inheritance

In object-oriented programming environments, Inheritance is allowed access state and behavior of one class is access to another class.There are commonly used fields are describing both classes in the Inheritance.Each subclass can retrieve the information from the parent class. Sub class variables can access directly from the base class.

2.2 Encapsulation

In object-oriented programming, Encapsulation of concepts aresaying it is holding all the state and behavior of entire class within a single entity.It binds the entire code and data to manipulate.It is secure and accessibility the code from outside. Encapsulation is not allowingthe data cannot be accessed from outside of any other the class.

2.3 Function Overloading

In object-oriented programming, Function overloading is using a same function name to more than number of functions. During compile time,it cannot through the error.because of every function having different parameters.Those type of mechanism is accessing during runtime based on parameter types.

2.4 HASKELL

This type of functional programming language that supports parallel execution.Its also increase significantly programmer efficiency such as clearer, shorter and code manageable and more reliable of a smaller number of errors.The smaller gap between the programmer and language its wide spectrum to implement variety of applications. Its more efficiency to maintain and mor modifiable.

2.5 JAVA

Java language is more powerful object-oriented programming language. To implement more efficiency application and supported functional programming. It has built in sustenance for Functional programming of mathematical notation. Later on, the version of Java 8 is supports for Functional Programming. Java 8 provides with a new Functional Interface, Lambda Expressions, Lazy evaluation and Stream API which is quite good.

3. EXPERIMENTAL SETUP AND IMPLEMENTAION

For the experimental results are used for the Java and Haskell programming language code implementation compare and measure comprehension evaluation. The time complexity takes measurement of programming execution time. When compare and measure the two-functionprogramming like Java and Haskellto check the time complexity As Figure 1 shows the time taken by the Java program to compute even numbers using the Object Orientated concepts approximately getting result between 0.0019 to 0.0043 Seconds and the Haskell language approximately getting

ISSN- 2394-5125 VOL 7, ISSUE 08, 2020

result between 0.01 to 0.03 seconds .The time variation between both languages are bigger difference. The Code difference between Java and Haskell functional programming language as given as:

The inheritance of Two Languages of Java and Haskell

```
public class inher {

 public static void main (String args []) {

long startTime = System.nanoTime();

 masters s = new masters();

 s. student_Details ();

 s. masters_Details ();

long stopTime = System.nanoTime();

long elapsedTime =stopTime - startTime;

double seconds = (double)elapsedTime / 1_000_000_000.0;

System.out.println(BigDecimal.valueOf(seconds).toPlainString()); }}
```

```
data Counter = Counter

{

    tick :: Counter, display :: Int

  }

nkCounter :: (Int -> Counter) -> (Int -> Counter)

nkCounter self n = Counter {

    tick    = self (n + 1)

  , display = n

  }

twice :: (Int -> Counter) -> (Int -> Counter)

twice self n = (nklCounter self n) {

    display = n * 2
```

ISSN- 2394-5125 VOL 7, ISSUE 08, 2020

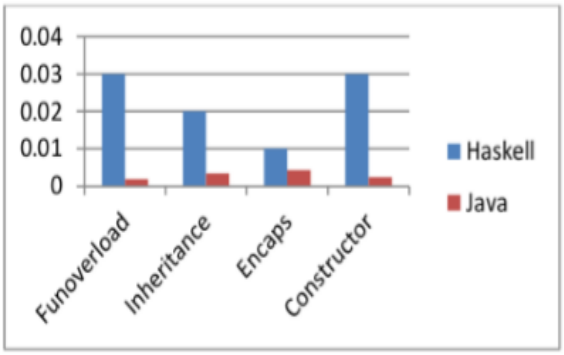

Figure 1: Performance of Java & Haskell Programming

The Table 1 shows the execution of function overloading, inheritance, encapsulation, and function constructor of java and Haskell programming language. The efficiency time complexity between these two functional languages was executed and proved the language.

Function	Haskell (In Sec)	Java (In Sec)
Inheritance	0.03	0.0019
Function Overloading	0.02	0.0034
Encapsulation	0.01	0.0043
Constructor	0.03	0.0024

4. CONCLUSION

The aim of this research paper the implementation of this two functional programming namely Java and Haskell were evaluated and to establish the computational time is compared and proved the Java programming language is lesser than the Haskell Language. However, the Object Orientation is efficient, when compared and proved that the fundamental construct program can providemore efficient code when translating Object Oriented code to some list of concepts. To analyze the execution time of Java and Haskell functional language was more efficient and less time to execute to code. It's additionally ended that a set of the loop facility are often used as if it were a listing comprehension facility. In future, we compared more functional language and get better time complexity.

REFERENCES

1. Mark Shields,Simon Peyton Jones,Object-Oriented Style Overloading for Haskell , 2005.
2. Gregory Neverov,Paul Roe,A Multi-stage, Object-Oriented Programming Language,2004.
3. Abadi, M.,Cardelli, L.: A Theory of Objects. Springer, Heidelberg ,1996.
4. L.Augustsson, “Implementing Haskell overloading”,1993.
5. Oguntunde,BosedeOyenike,Comparative analysis of some programming languages,2012
6. Manuel M.T.Chakravarty,Gabriele Keller,Functional Array Fusion
7. RF Pointon, S Priebe, HW Loidl, R Loogen ,Functional vs object-oriented distributed languages, 2001.
8. R Harrison, LG Samaraweera, MR Dobie ,Comparing programming paradigms: an evaluation of functional and object-oriented programs,1996.
9. Kumaravel“A.,Comparison of two multi-classification approaches for detecting network attacks”, World Applied Sciences Journal,V-27,I-11,PP-1461-1465,Y-2013.
10. Tariq J., Kumaravel“A.,Construction of cellular automata over hexagonal and triangular tessellations for path planning of multi-robots”,2016 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2016,V-,I-,PP--,Y-2017.
11. Sudha M., Kumaravel“A.,Analysis and measurement of wave guides using poissonmethod”,Indonesian Journal of Electrical Engineering and Computer Science,V-8,I-2,PP-546-548,Y-2017