Investigating the Usability of Java by Mining the GitHub Repositories

Rabia Jamro
Department of Software Engineering
Mehran University of Engineering & Technology
Jamshoro, Pakistan
rabiaqadir42@gmail.com

Sania Bhatti
Department of Software Engineering
Mehran University of Engineering & Technology
Jamshoro, Pakistan
sania.bhatti@faculty.muet.edu.pk

Salahuddin Saddar
Department of Software Engineering
Mehran University of Engineering & Technology
Jamshoro, Pakistan
salahuddin.saddar@faculty.muet.edu.pk

Faisal Memon
Fauji Fertilizer Bin Qasim Limited,
Karachi, Pakistan
faisal.memon@ffbl.com

***Abstract*— Programming languages have a major role in the software industry and so many programming languages have been introduced till now. There is a need to determine the usability, goodness, and quality of programming languages, which make them differ from one another and find out their purpose which suits them. This paper will help novice developers with efficient coding. The paper presents the usability of a Java language on the GitHub repository. For this, we analyzed 55831 latest java projects and from this dataset, we find out the usage of different java features, and finally, the analysis was performed on features of java. From analysis results, we find out the usability facts of java language.**

Keywords*— *Java, data mining, usability, programming language. GitHub

I. Introduction

The First High-level programming language was developed in 1950 [1] named FORTRAN. After that new languages were introduced and till now, thousands of languages have been introduced. Here the question arises that what makes a programming language useful and different from one another?

In software development, some factors are affecting to make a good programming language like the length of code, speed, performance, error handling, and utilization of memory [2]. There is no single programming language that fits with all these all factors. Some are good at error handling and some good at performance and speed. Scripting languages (PHP, JavaScript) contain less code than object-oriented programming languages (Java) [2]. Developers always want to make good programs to achieve some useful factors. There is an approach in which we can measure the goodness of language by taking real users, providing them certain tasks and they are free to choose any programming language. This approach can give some better results. But it is time-consuming and not sufficient in every aspect [3].

This paper defines a way through which usability and goodness of Java programming language can be measured by mining the GitHub [4], a popular coding platform. There are some reasons to choose this repository. One reason is that we can retrieve popular projects from GitHub due to star activity [5]. Another reason is that the GitHub repository can update the main repository if any developer applies changes on a specific project [6]. This approach solves many difficulties (time factor and limited scope) faced in the real users' studies and it also enhances the scope of the work. This paper focuses on the most important features of java language that help the developers for understanding its basics. For conditional statements, if-else and switch are used. This study differentiates them with the help of their usability over the open-source platform.

The paper determines the usability of most used features that is important for understanding the basics and syntax of java. We mined 55831 medium sized database consisting of recent projects of java from open-source GitHub. For accomplishing the mining task on GitHub we used the Boa infrastructure [7].

The work presented in this paper is an addition to the state of art research focusing on the usability of Java. This paper will increase the interest of novice developers in determining the usability of new features which are added in new versions of programming languages and they can also determine the usability of new features over GitHub.

II. LITERATURE REVIEW

Some studies have been conducted concerning the mining of repositories but most of them are not related to defining the usability of any programming language [8,9]. These studies performed mining tasks on the GitHub repository to determine the similar application code available on it. The purpose of those studies was to save project costs and also help the developers. The study used Boa infrastructure analysis formed on data science projects stored over GitHub [10] in order to find that how many data science projects are using the python language and which are the popular libraries used by developers to find the errors and enhancing code in python language. Using Eclipse IDE parser usability of java was determined in the study [3]. Study [3] is a preliminary step towards the analysis of usability of Java and our work presents more reliable results in comparison to [3] because a large number of Github Java projects are analyzed. . GitHub repository was mined to achieve some objectives that include, classifying the Readme files of projects into points for better understanding [11], how the collaboration has been done between the developers on GitHub [12], and which software engineering practices are focused during the application development is defined in [13]. Another study focused to determine the usability of the single go-to statement in C language [14]. The study used the

ISBN-978-969-23372-1-2

GitHub platform to mining the available C language projects to extract the “go-to” statement.

III. METHODOLOGY & IMPLEMENTATION

At first, the most popular programming language “Java” was selected and used for determining its usability through features. GitHub, a social coding platform that is not only used for development but it can also be used for data mining tasks to get useful mined data from such enormous data available on GitHub. To get the data from GitHub we used Bao. Boa is a platform that interacts with the Hadoop program to get useful data from the social coding repository like GitHub. Fig. 1 defines the query submission and query processing steps done on Boa. By using the Boa web interface, we write queries, send input to the dataset, and submit to it, this is known as the query submission step. The query is compiled into a Hadoop program, then deployed on the cluster. Cluster gets the data according to query needs from the cache because GitHub stores its repositories on a cache. When data available on the cache match with the query, the cluster returns the result to the Bao through the web known as the processing of the queries. After both steps, we got the results in textual form.

The paper mined some popular features and 55831 recent projects of java from the GitHub repository. To avoid duplication random projects are selected from the large data set. Once we gathered the usability of each feature, we have enough knowledge about different features used by the experienced developers, and this information we are using to give suggestions to the new developers. A detailed discussion of each feature of java is mentioned in the Results and discussion section of the paper.

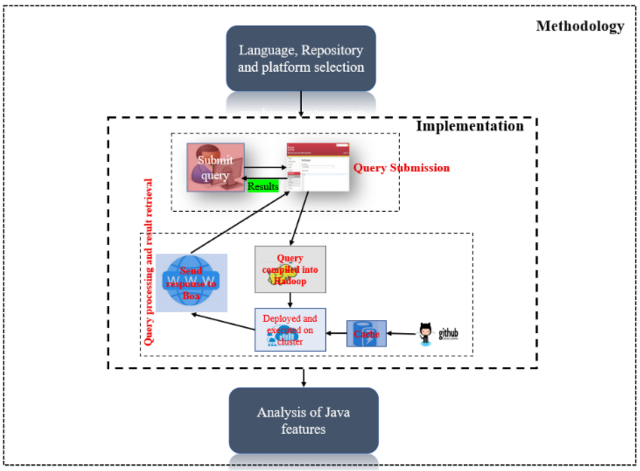
Methodology
Language, Repository and platform selection
Implementation
Submit query
Results
Query Submission
Query processing and result retrieval
Send response to Boa
Query compiled into Hadoop
Deployed and executed on cluster
Cache
github
Analysis of Java features

Fig. 1 Methodology & Implementation

IV. RESULTS AND DISCUSSION

We mined around 55831 latest java projects; from these projects we find out the most used features of java over the GitHub social coding platform. Fig. 2 illustrates the different features of the java programming language which are analyzed in this study.

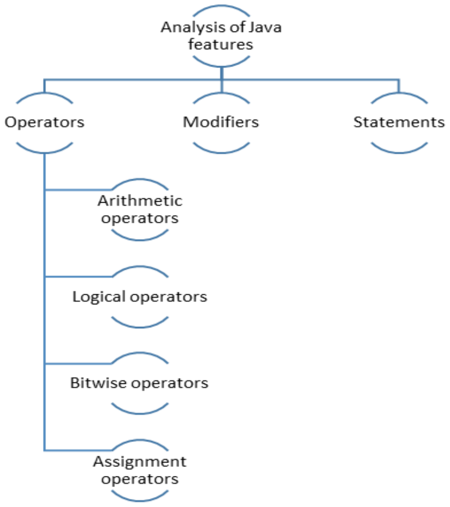

Fig. 2 Features of Java analyzed

A. *Java Operators*

The operator is a sign used between the variables and values to perform the operations. Java language is rich in operators which are categorized in certain groups like arithmetic, logical, assignments, bitwise, and so on. This paper defines the usage of mostly used groups of operators. It also provides a comparison between groups that which group is highly used by the developers during coding.

1) Arithmetic Operators:

For mathematical calculation, arithmetic operators are used, like for addition, increment and division. The results in table are presented in declining order.

TABLE I. ARITHMETIC OPERATORS

Arithmetic Operators	Count	Average Count/Project
Addition (+)	5,071,866	90.84
Subtraction (-)	3,017,891	54.05
Increment (++)	1,795,751	32.16
Multiplication (*)	1,160,147	20.77
Division (/)	619,679	11.09
Decrement (--)	184,710	3.30
Modulus (%)	100,452	1.79

In TABLE I., we analyzed that the addition operator has the highest use because it can also be used for concatenation along with addition and modulus has the least usage. The main reason to construct this table is that the novice user can become familiar with the usage of these operators during mathematical calculations.

2) Logical Operators

Logical operators are used for determining the true and false (Boolean) values. Java language support AND, OR, and NOT logical operators. A list of possible logical operators is shown in table 2, there is little difference among the usage of logical And and Not operator in average count per project, however least commonly used logical operator is (||) depending upon the requirement of the project.

TABLE II. LOGICAL OPERATORS

Logical Operators	Count	Average Count/Project
Logical_And (&&)	1,635,241	29.28
Logical_Not (!)	1,566,119	28.05
Logical_Or (||)	937,295	16.78

3) Bitwise Operators

There is a type of operator that exists, which operates on each bit is called bitwise operators. Highly used programming language java supports a wide range of bitwise operators, which can be applied to almost all data types like byte short, char, long, and int. From count and average count/project value of table 3, the & bitwise operator has the highest priority and ~ has the lowest priority depending upon the logics used in the particular projects.

TABLE III. BITWISE OPERATORS

Bitwise Operators	Count	Average Count/Project
Bit_And (&)	462,637	8.28
Bit_Lshift (<<)	194,429	3.48
Bit_Or (|)	173,120	3.10
Bit_Rshift (>>)	104,396	1.86
Bit_Unsignedshift (>>>)	67,395	1.20
Bit_Xor (^)	38,521	0.68
Bit_Not (~)	31,376	0.56

4) Assignment Operators:

Assignment operators allocate or assign values to the variables, the "=" sign is used in assignment operators. In assignment operators, the variable is used before assignment operator and value after =, in case of variable and value. e.g. x+=10.

Java supports many assignment operators. From table 4, we have noticed that if bitwise and arithmetic are used alone or with assignment sign, operator priority varies. Like in bitwise operators the & has the highest value but as = operator used with them, the bitwise OR operator has priority more than AND.

TABLE IV. ASSIGNMENT OPERATORS

Assignment Operators	Count	Average Count/Project
Assign_Add (+=)	613,188	10.98
Assign_Sub (-=)	191,519	3.43
Assign_Bitor (|=)	81,782	1.46
Assign_Mult (*=)	44,013	0.78
Assign_Bitand (&=)	20,020	0.35
Assign_Div (/=)	15,668	0.28
Assign_Bitxor (^=)	14,987	0.26
Assign_Lshift (<<=)	8,040	0.14
Assign_Rshift (>>=)	4,256	0.076
Assign_Unsignedrshift (>>>=)	3,554	0.06
Assign_Mod (%=)	2,951	0.05

After comparing all the above-mentioned operators' groups in java, we can say that the arithmetic group and logical group of operators are highly used over the open-source GitHub platform because to make conditions we mostly use these two groups. One must be familiar with all these groups of operators for the understanding of language basics. Figure 3 presents the comparison of four groups of operators and average count/project.

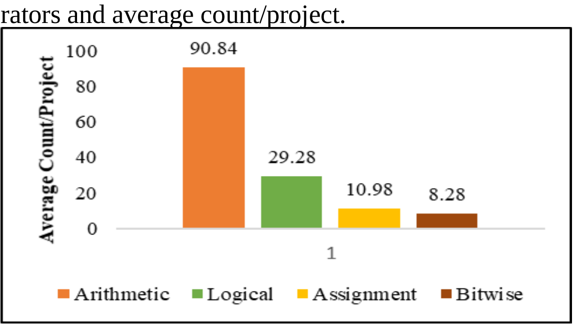

Fig. 3 Highest value of each group

B. Modifiers:

Modifiers are used to add a description with variables, classes, and methods. These are the keywords used before the method, variable, and class name defining its accessibility and non-accessibility. The paper focuses on the modifiers which are non-assessable because these modifiers are common in classes, variables, and methods. Like abstract modifier can be used with methods and classes.

In table 5, the count value of each modifier changes depending upon its usage, the final modifier has highest priority over all other modifiers, the reason is that this modifier can be used with variables, classes, and methods. Static is used with variables and methods. That's why it has high value than abstract as abstract is only used with methods and classes, and a class or method can contain many variables. Synchronized is only used in threads and has the least count value. Figure 4 shows the modifiers usage in terms of average count/project.

TABLE V. MODIFIERS

Modifiers	Count	Average Count/Project
Final	6,838,517	122.48
Static	5,444,375	97.51
Abstract	328,140	5.87
Synchronized	166,392	2.98

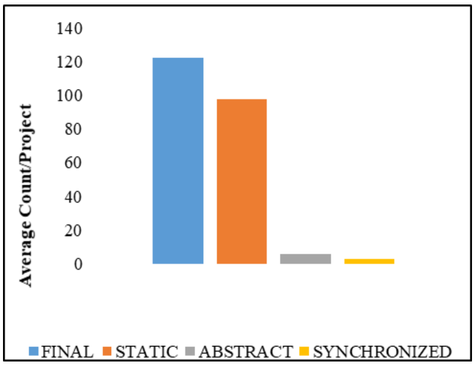

Fig. 4 Modifiers Average count/project

C. Java Language Statements:

To learn any programming language, one must become familiar with the basics of it. Statements are said to be basics of languages. Statements are divided into certain categories like looping statements, conditional statements, block statements, try-catch statements, and so on.

TABLE VI. STATEMENTS

Statements	Counts	Average Count/Project
Return	11,396,557	204.12
If	11,358,454	203.44
For	2,090,839	37.44
Case	1,861,692	33.34
Catch	1,593,975	28.54
Try	1,537,115	27.53
Throw	1,489,737	26.68
Break	1,057,785	18.94
While	538,607	9.64
Switch	312,345	5.59
Continue	210,922	3.77
Synchronized	157,603	2.82
Assert	102,259	1.83
Do While	55,453	0.99
Labeled	46,098	0.82
Empty	32,083	0.57

Table 6 shows the most used statements from higher priority to lower priority by using count and average count per project values. From table 6, it is obvious that the most widely used loop is For, and do-while is rarely used. In control statements, IF has priority over the switch statement. As we know the switch is faster than IF, even that developers use if statements due to its simplicity and variable values as switch contain fixed values. In brief, we can say that among all statements return statement is most popular, which is used in java methods and the empty statement is least popular.

V. COMAPRISON WITH THE DISTINCT STUDY OF JAVA USABILITY VIA MINING

Table 7 presents comparison of our study with [3]. In study [3], authors observed the usability issues of Java source code by mining the features of 1746 publically available GitHub projects. However, in this work we have measured the usability of Java by mining 55831 GitHub projects.

TABLE VII. COMPARISON

	Lemay et al.[3]	Our study
Year of study	2018	2020
Number of projects	1746	55831
Method	Eclipse IDE parser	Boa Infrastructure
Arithmetic operators	Limited	All
Logical operators	Limited	All
Assignment operators	No	All
Bitwise operators	Limited	All
Modifiers	No	All
Null checks	All	Future work
Standard Method call	Yes	Future work

VI. CONCLUSION

Previous studies have determined the language quality, usability and goodness in different perspective. This paper has presented the usability of Java language by mining process. This is the first time that analysis is performed on commonly used features of java after mined results, which will be better for understanding the difference between similar features. This paper will be helpful for the new developers to make quality code, and also for researches to extend the work in this area hopefully.

VII. FUTURE WORK

This work can be extended by comparing the Java language usage in different types of projects for the advanced level for experienced developers. Another future work direction could be to find out the usability of other programming languages like python, JavaScript etc. We can also perform the comparative analysis among different programming languages like object-oriented programming languages (java, C#), scripting languages (JavaScript, php) through features.

REFERENCES

[1] Al-Qahtani, Sultan S., et al. "Comparing selected criteria of programming languages java, php, c++, perl, haskell, aspectj, ruby, cobol, bash scripts and scheme revision 1.0-a team cplgroup comp6411-s10 term report." arXiv preprint arXiv:1008.3434 (2010).

[2] S. Nanz and C. A. Furia, “A Comparative Study of Programming Languages in Rosetta Code,” 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, 2015.

[3] Lemay, Mark J. "Understanding Java usability by mining GitHub repositories." 9th Workshop on Evaluation and Usability of Programming Languages and Tools (PLATEAU 2018). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[4] C. Wanstrath, "GitHub: Where the world builds software", GitHub, 2020. [Online]. Available: https://github.com/. [Accessed: 03- Jun- 2020].H. Borges and M. T. Valente, “What’s in a GitHub Star? Understanding Repository Starring Practices in a Social Coding Platform,” Journal of Systems and Software, vol. 146, pp. 112–129, 2018.

[5] H. Borges, A. Hora, and M. T. Valente, “Understanding the Factors That Impact the Popularity of GitHub Repositories,” 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME), 2016.

[6] H. Rajan, "About the Boa Language and Infrastructure - Boa - Iowa State University", Boa.cs.iastate.edu, 2020. [Online]. Available: http://boa.cs.iastate.edu/. [Accessed: 18- Aug- 2020].

[7] K. W. Nafi, B. Roy, C. K. Roy, and K. A. Schneider, “A universal cross language software similarity detector for open source software categorization,” Journal of Systems and Software, vol. 162, p. 110491, 2020.

[8] Zhang, Yun, et al. "Detecting similar repositories on GitHub." 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE, 2017.

[9] S. Biswas, M. J. Islam, Y. Huang, and H. Rajan, “Boa Meets Python: A Boa Dataset of Data Science Software in Python Language,” 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR), 2019.

[10] Prana, Gede Artha Azriadi, et al. "Categorizing the content of GitHub README files." Empirical Software Engineering 24.3 (2019): 1296-1327.

[11] Cosentino, Valerio, Javier L. Cánovas Izquierdo, and Jordi Cabot. "A systematic mapping study of software development with GitHub." IEEE Access 5 (2017): 7173-7192.

[12] Munaiah, Nuthan, et al. "Curating GitHub for engineered software projects." Empirical Software Engineering 22.6 (2017): 3219-3253.

[13] Meiyappan Nagappan, Romain Robbes, Yasutaka Kamei, Éric Tanter, Shane McIntosh, Audris Mockus, and Ahmed E Hassan. An empirical study of goto in C code from GitHub repositories. In Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, pages 404–414. ACM, 2015.