



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijircce.com](http://www.ijircce.com)

Vol. 5, Issue 3, March 2017

## Readability Improvement in JAVA language by Using Segmentation Techniques

Pooja Kundu\*, Shivam Wadhwa\*\*, Vivek Kamboj\*\*, Renuka Dhillon\*\*,

Assistant Prof. ,Dept. of CSE, Panipat Institute of Engineering & Technology, Panipat Haryana, India\*

B.E. Student, Dept. of CSE, Panipat Institute of Engineering & Technology, Panipat Haryana, India\*\*

**ABSTRACT:** With the goal of increasing program readability for easier understanding, coding guidelines often include formatting standards such as indenting loop and conditional branch body statements. Similarly, good programming practice suggests that programmers use blank lines to visibly delineate between code segments that represent different algorithmic steps or high level actions. Unfortunately, programmers do not always follow these guidelines. While editors and IDEs can easily indent code based on syntax, they do not currently support automatic blank line insertion, which presents more significant challenges involving the semantics. This paper presents a heuristic solution to the automatic blank line insertion problem, by leveraging both program structure and naming information to identify “meaningful blocks”, consecutive statements that logically implement a high level action. It is a technique to do segmentation in the Java code. Such Segmentation in the code will result in an efficient way of handling the program, finding the bugs and readability of the program. The technique relies on identifying those blocks which are meaningful and have a comprehend structure.

**KEYWORDS:** Formatting, Segmentation block, JAVA, Readability.

### I. INTRODUCTION

Software maintenance is an important task as the efficiency of a project lies on its maintenance. A complex program is hard to debug and difficult to understand. Moreover, lots of time is wasted if you read a poorly maintained program. So, in order to efficiently understand and maintain the program, its readability should be improved.

There are various text translators in market which help in converting the text so that it becomes readable. They work by inserting blank lines in between the text and maintain the flow of the program. Sometimes the semantic meaning of the program is lost after the transformation. [13][14]

Java is one of the most widely used language but there is no such technique to create a readable program. Since, the program created in Java language are complex we need to insert blank lines[10] by doing so certain keywords and blocks are easily understandable such transformations are clearly used to change the syntactic look inspite of changing the syntactic meaning[11].

Programmers can themselves follow certain formats while writing a program but it is observed such formats are not followed widely[12]. Even manual inspection cannot sort out the problem of poor readability so there is an need to be automatic blank line insertion. We have proposed a segmentation technique to do the same. We identify different blocks and then remove the overlapping blocks, if we get a short block then we merge it with other blocks.

In our research paper, we have improved the readability of the java code with the help of segmentation technique. Meaningful segments in the program are categorized and large segments are divided into smaller ones for better modularization .The rest of the paper is divided as follows. Section 2 presents the related work in the field of improving the segment of code. Proposed work is discussed in Section 3 along with the flow chart. This paper is concluded in Section 4.

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirccce.com](http://www.ijirccce.com)

Vol. 5, Issue 3, March 2017

## II. RELATED WORK

R.Brooks [3] has comprehended a program with the help of four sources. Behavior of the programming language is observed along with the documentation of language. In his Research he has analysis a structure of knowledge by organizing the information into various domains with the help of his reconstruction process he has provided a reconstruction process to comprehend the program.

In [4] beakon is used in order to create coding pattern so that program structure is understood well enough. They have provided experimental analysis results as well to compare the results of advanced and intermediate groups and algorithm is sub-divided into various areas.

Micro patterns of java code are studied in [5] which also made abstraction clear. The author has proposed a way to studied these pattern because they are require and are easily recognized. He has presented 27 such patterns. The result provided in this research paper are sustained in the software evaluation . Also the code quality is improved.

Extract Method [6] decomposes large and complex methods. Refactoring opportunities are easily identified with the proposed word .Dependencies of the rules is also preserved. Design flaws vary easily identified . Various coding standard are discussed in order to maintain uniform structure through out the code in [7]-[9]. Improvement in code readability to improve the quality [10] in a language can be provided by grouping the code similar to a usability and a test group.

## III. PROPOSED WORK

Our proposed work identifies the different segments in the JAVA language. Different segments are categorized according to their functionality. The flowchart of our work is shown in figure 1. We have divided our algorithm into 3 parts. First part identifies the meaningful blocks. Segments of codes are refined in the second part. Short blocks are merged together.

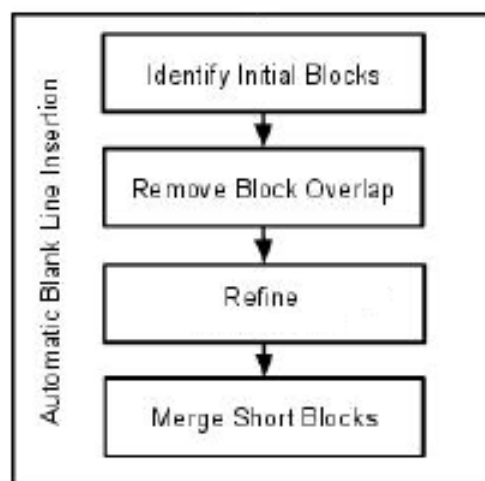


Fig. 1 Segmentation technique

### A. Identification of large blocks

In this part we have divided the program into the meaningful blocks. We have considered a block to be meaningful, if certain lines are consecutive and have a close meaning together. We have identified the blocks which are syntactically same. Also each return statement is considered at the end of the block. Blank lines are inserted before and



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijircce.com](http://www.ijircce.com)

Vol. 5, Issue 3, March 2017

after the comment. Syntactic Categorization is done on the basis of following categories: Init , Method Call, Object Method call, Throw , Return, Prefix Expression, Postfix Expression, Infix Expression and Assignment operator.

## B. Refine blocks

There should be no overlapping of the statements because that also makes a program hard to read and understand some Recursive procedures are overlapped. Also there are certain cases where some blocks are logically related. So, we consider the Naming Conventions to separate out overlapping blocks, we find out the similarity level. We assign three different levels namely, Top, Middle, Bottom. Top level similarity means strong statement pair. Bottom level similarity means weakest statement pair. This similarity pair categorization is discussed in table No. 1.

Table 1: Similarity Index

Structure	Bottom	Middle	Top
Init	Type+name+right assignment	Type+name/type+assignment	Type+right assignment
Return	Return to parent method	-	-
Method call	Name of method	-	-
Expression	Infix, postfix, prefix	-	-
Assignment operator	-	Name/right assignment	Right assignment

We construct a similarity function to calculate the similarity level of a structure in Java language. For Top level similarity 3 is returned, For Middle 2 is returned , For Bottom 1 is returned else 0 is returned . This is shown in figure No. 2

```
if(top)
    return 3;
else if(middle)
    return 2;
else if(bottom)
    return 1;
else
    return 0;
```

Figure 2: Similarity Function



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirccce.com](http://www.ijirccce.com)

Vol. 5, Issue 3, March 2017

## Blocks Merging

After the second phase we refine our blocks by deciding whether to segment the block or not to segment the block. We make the decision of segmentation by considering the similarity level and block length. Those blocks which have same similarity level are clustered together. At each point while scanning, difference between similarity level is calculated. If the difference is 0 then we insert a blank line. After inserting Blank lines some blocks of very small size might created. So, we target blocks with a single line and merge them with their nearest block.

## IV. CONCLUSION

Code readability need to be improved. A code with an efficient readable format makes a program easy to read, maintain and debug. In this research paper we have proposed a technique to improve the readability of the program. We have discussed a segmentation technique. Our technique divides the code into meaningful blocks and creates white space among such blocks. The proposed algorithm not only creates white spaces but also merges small blocks. Certain levels of similarity are given to each block.

## REFERENCES

- [1] Buse, Raymond PL, and Westley R. Weimer. "Learning a metric for code readability." IEEE Transactions on Software Engineering 36.4 (2010): 546-558.
- [2] Jackson, Stoney, PremkumarDevanbu, and Kwan-Liu Ma. "Stable, flexible, peephole pretty-printing." Science of Computer Programming 72.1 (2008): 40-51.
- [3] Brooks, Ruven. "Towards a theory of the comprehension of computer programs." International journal of man-machine studies 18.6 (1983): 543-554.
- [4] Crosby, Martha E., Jean Scholtz, and Susan Wiedenbeck. "The roles beacons play in comprehension for novice and expert programmers." 14th Workshop of the Psychology of Programming Interest Group. 2002.
- [5] Gil, Joseph Yossi, and ItayMaman. "Micro patterns in Java code." ACM SIGPLAN Notices 40.10 (2005): 97-116.
- [6] Tsantalis, Nikolaos, and Alexander Chatzigeorgiou. "Identification of extract method refactoring opportunities for the decomposition of methods." Journal of Systems and Software 84.10 (2011): 1757-1782.
- [7] W. Humphrey, Introduction to the personal software process(sm), 1st ed. Addison-Wesley Professional, 1996.
- [8] B. Zograf. (2011, Jan.) Java programming style guidelines.[Online].Available: <http://geosoft.no/development/javastyle.html>
- [9] D. Lea. (2000, Feb.) Draft java coding standard. [Online].<http://gee.cs.oswego.edu/dl/htmljava/CodingStd.html>
- [10] Haneef, Nuzhat J. "Software documentation and readability: a proposed process improvement." ACM SIGSOFT Software Engineering Notes 23.3 (1998): 75-77.
- [11] Posnett, Daryl, Abram Hindle, and PremkumarDevanbu. "A simpler model of software readability."Proceedings of the 8th working conference on mining software repositories.ACM, 2011.
- [12] Wang, Xiaoran, Lori Pollock, and K. Vijay-Shanker. "Automatic segmentation of method code into meaningful blocks to improve readability." Reverse Engineering (WCRE), 2011 18th Working Conference on. IEEE, 2011.
- [13] Schach, Stephen R. Object-oriented and classical software engineering. Vol. 6. New York: McGraw-Hill, 2002.
- [14] Wang, Xiaoran, Lori Pollock, and K. Vijay-Shanker. "Automatic segmentation of method code into meaningful blocks to improve readability." Reverse Engineering (WCRE), 2011 18th Working Conference on. IEEE, 2011.
- [15] Buse, Raymond PL, and Westley R. Weimer. "A metric for software readability."Proceedings of the 2008 international symposium on Software testing and analysis.ACM, 2008.