THE EFFECT OF VISUALIZING ROLE OF VARIABLE IN OBJECT ORIENTED PROGRAMMING UNDERSTANDING

Mabroukah Amarif and Sakeenah Ahmed

Department of Computer Sciences, Sebha University, Sebha, Libya

ABSTRACT

The role of any variable is interpreted as the required task or performance of it in any part of a program. This role contributes to the easy understanding of the program and thus formulates it clearly and unambiguously. Many novice programmers face various difficulties in understanding programming, especially Object Oriented Programming. This research adopts the design of a visualization tool which includes visual model that shows the role of the reference variable (an object) within a Java program to enhance comprehension understanding for novice programmers. The model enables them to interact and thus formulate an object-oriented program in an intuitive and clear way. Based on the actual experimentation, the effectiveness of this model is improved and the importance of this research in the field of object programming is demonstrated.

KEYWORDS

Role of variable, object oriented programming, visualization, understanding

1. INTRODUCTION

The concept of the role of variable is a modern in view with respect of the lack of sufficient known about it. In the last decade, it has been used to facilitate and improve programming education [1, 2]. It is an educational implement distinct style, if used properly, to clarify the structure and meaning of the program [3]. The roles of variables contribute to the outputs of the program, indicating the achievement of the program goal or the whole system [4]. Sajaniemi et al. stated that the role of the variable certainly provide novice programmers with a new conceptual framework that enables them to rationalize and mental processing their programs in a way similar to understanding of a good code, they also stated that the role of any object-oriented variable can be defined by its properties and behavior during the program [5, 7].

Regarding of the view of experienced programmers, it was observed that their knowledge is implicitly embodied in the cognitive structures or diagrams structures for any program [2], while the new or novice programmers often find it difficult to understand a program question because they are unable to identify a variable or more, or sometimes they couldn't understand the difference between one variable and another [3, 6]. The reason lies in their inability to visualize the desired role of the variable that is determined in the program.

The variable has a dynamic property. In other words, it contains a value and then contains another value, thus successive values obtained (variable behavior). Therefore, it is called a variable that is characterized by a change in its value in any part of the program as required.

Dhinaharan Nagamalai et al. (Eds) : COSIT, AIAPP, DMA, SEC - 2019
pp. 11–20, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.90202

Looking at the object oriented programming languages; many novice programmers face many difficulties, especially in the process of correctly formulating the program [4, 7, 8]. It is realized that the most difficult part lies in the ambiguities of the variables analysis.

The data structures description of procedural programming are assigned to the programmer in terms of their inclusion, logical compilation, data types and functions, while in the object-oriented programming languages, a lot of language structures are part of the same language, which is ready and suitable for the formulation of large programs and delivered by a team of programmers. In fact, object oriented programming is an influential and effective model at present time. It's special because they have a range of tools and techniques that enable software engineers to build reusable software systems that meet user needs. The most important of these tools are class and object which used as a storage of data and operations on them through the exchange of messages between the program and objects and between objects themselves rather than functions call which found in procedural programming languages. In the other side, these tools may overwhelm the initial programmer because of their large numbers, or the difficulty of dealing with them in principle.

In this paper, we describe our developed tool called LearnOOP for teaching/learning OOP Java programming language. The tool development adopts the role of variable concept and visual forms to reflect the behaviour of variable and its roles. We also describe our experience as an experimental procedure and evaluate the developed tool using empirical evaluation approach. The following section describes the most related works to our paper while section 3 explains the developed tool description. Experimental procedure and results are described in section 4. A discussion of this paper is explained in section 5 and the conclusion is provided in section 6.

2. RELATED WORK

According to our currently search and based on what we found, a few tools have been developed for teaching/learning programming languages. Some of them adopt role of variable concept such as PlanAni [9] and ViLLE [10,11], others are not, such as OOP-ANIM [12] and Jeliot [13,14,15]. PlanAni and ViLLE tools are used for procedural programming understanding while OOP-ANIM and Jeliot are used for object oriented programming comprehension. Moreover, the animation and visualization techniques have been adopted in all of these tools but in different ways. Only PlanAni tool visualizes the role of variable from a real world component, for example; counter variable is visualized as a step foot.

ViLLE tool gives a description of role of variable in text-based form. It explains the changing values of variable at each stage of program executing.

Although the OOP-ANIM and Jeliot tools explain the object oriented programming concepts, none of them include the concept of role of variables. OOP-ANIM uses UML diagrams as a visual form of program parts, in the other side; Jeliot tool uses boxes and text form for program executing visualization.

Different studies have been carried out for evaluating these tools to know how much does role of variable concept effect in learning and understanding. From the literature, Sajaniemi et al. have conducted various studies [1, 3, 5, 7, 9]. Their results showed a positive effect of roles of variables in understanding of programming languages. They also agree that the concept of role must be enhanced by visual description that reflects the reality of its behavior.

Another research has been carried out by Shi et al. [6]. They considered the visualization and animation effects of role of variable in building a novice program. They carried out an experiment on 55 students of computer sciences. They divided the students into two groups; the first learning group learned programming in the traditional way based on the role. The second group learned with the support of PlanAni tool based on the visualizing roles of variables. Their data analysis showed a general improvement of the level of SOLO and high approval for the second group.

Al-Barakati et al. have conducted an experimental study [17]. They used a PlanAni tool to teach students an introductory programming course. Their results showed in scores on a post-test used to evaluate the effect of visual role of variable in understanding with 91 students divided into three groups. The results showed that the visual role of variable groups significantly outperformed the non visual group.

A case study have been carried out by Rajala et al. [10] on ViLLE tool focusing on the effectiveness of visualization of role of variable on learning basic programming concepts. They were randomly divided students into two groups. The control group used traditional textual material during the course, while the treatment group used the same material extensively with interactive examples using the ViLLE tool. The study found that the ViLLE tool improves student learning without prior programming experience. They also found that ViLLE tool benefit novice learners more than learners who have some previous experiences.

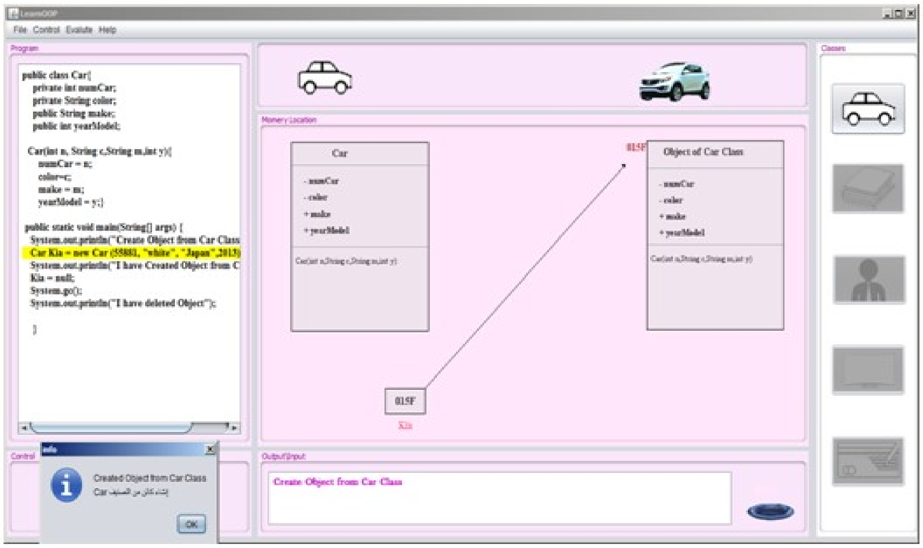
Various studies have been carried out using Jeliot tool during the development of the tool versions [15, 16]. Regarding of learning with visualization and animation within the program execution, the results show positive effect. They also concluded that the educational tool must be incorporated into classes and assignments rather than once used as an aid in teaching.

Although there are tools in object programming for increasing students' understanding, they lack of the concept of variables roles and if variable roles are included, they still need to be visualize for enhancing students' understanding. All of these motivate us to propose and develop our own tool which adopts the visualization of variables roles using real world objects for more students' perception and understanding of OOP java language.

The main goal of this paper is to justify the positive effect of visualizing the role of variable in OOP Java programming understanding through our developed tool called LearnOOP and evaluate quantitatively the effectiveness of it. For the purpose of this paper, effectiveness refers to the ability of this tool in enhancing student's perception and understanding. This goal is evaluated using an empirical evaluation approach (traditional teaching without a tool vs. using LearnOOP tool). The question is, "*Is teaching using LearnOOP tool which includes the visualization and animation of role of variables more effective than traditional teaching medium?*".

3. LEARNOOP TOOL DESCRIPTION

LearnOOP is a tool used for teaching/learning Java OOP programming language. It contains the description of object and how being created from a special class. It adopts the role of variable concept through an object description and different opeartions on it. The decriptoin of object is given by visualization and animation of his attribues and behaviour. The tool includes various examples of objects from the real world to increase the ability of undersanding OOP. For an example; an object car1 is created from a class Car. Figure 1 shows the interface of the tool and the object car1 example.



```
public class Car{
   private int numCar;
   private String color;
   public String make;
   public int yearModel;

  Car(int n, String c,String m,int y){
     numCar = n;
     color=c;
     make = m;
     yearModel = y;}

 public static void main(String[] args) {
  System.out.println("Create Object from Car Class
  Car Kia = new Car (55881, "white", "Japan",2013)
  System.out.println("I have Created Object from C
  Kia = null;
  System.gc();
  System.out.println("I have deleted Object");

   }
```

Created Object from Car Class
إنشاء كائن من الصنف Car

Create Object from Car Class

Figure 1. LearOOPTool user interface

4. LEARNOOP TOOL EVALUATION

We carried out an experiment consists of 40 students of the same lesson taught to the undergraduate Computer Science in Java Programming I (OOP) course at Sebha University of Libya during the semester I of 2017-2018 year. The students never studied OOP before. The experiment was conducted in two stages where each stage uses a different learning medium approach; the first stage uses only text-based materials (no visualization and animation of roles of variables), in the second stage, we divide the students into two groups randomly. Each group consists of 20 students. The first group called the control group which continuous in traditional teaching way while the other uses LeanOOP tool (visualizing role of variable concept). The students will be given a same test throughout the two stages. They are allowed to improve their answer after each stage. The results of the tests after each stage of the medium approach are compared.

The same topics of the lessons are given during all of the two stages. These topics are: class and object concepts, operations on object such as create, delete, copy, display, and modify object attributes. It also covers the inheritance lesson.

In this experiment, the tool SPSS [18] is used to statistically evaluate the effectiveness of LearnOOP tool using t-test and p-value.

4.1. Experimental Procedure

A total of 40 students participated in the experiment. The students are second year of Computer Science students (undergraduate students) at Sebha University of Libya. We follow the pre-test to post-test accuracy [19, 20] in order to evaluate the effectiveness of LearnOOP tool. At the first stage, all of the students were given the same lesson using traditional teaching medium followed by a pre-test. At the second stage, the students were divided into two equal groups randomly; the first group continuous with traditional teaching medium and the second group were introduced to LearnOOP tool followed by a post-test for both groups.

The experiment was controlled by delivering the same lessons to all of the students by the same teacher during one semester long. The semester was divided into two equal stages of time. In the first stage, only text-based materials were used during the lesson time with the help of electronic slides. At the end of the stage, the students were given a pre-test of three OOP questions with a time limit of one hour to answer them.

In the second stage, after the pre-test, students were divided into two equal groups; group A and group B. group A continued with traditional teaching medium. group B introduced to LearnOOP tool and to its visual interface. They were asked to experiment with some examples of real world objects such as Car, Employ, Television, Book and Account. They were also asked to experiment with the concepts of class and object, operations on objects and inheritance. At the end of the session, both groups were given a post-test of different OOP questions regarding of the same objects with a time limit of one hour to answer them.

To control the tasks performance, the students were not allowed to consult books or use any materials. Then the results of pre-test and post-tests were compared.

4.2. Experimental Results

To determine the effectiveness of LearnOOP tool, a pre-test and post-test accuracy is used. Table 1 describes the students' scores of group B for the pre-test before using LearnOOP tool and post-test after using LearnOOP tool. Notice that the maximum score for each student is 15.

Table 1. The students' scores (group B) of pre-test and post-test

No.	**Pre-test scores Without a tool**	**Post-test scores Using LearnOOP tool**
1	2	10
2	2	11
3	5	12
4	5	12
5	2	11
6	2	10
7	11	14
8	2	9
9	1	4
10	3	7
11	2	10
12	2	9
13	4	12
14	1	7
15	2	10
16	2	10
17	1	4
18	1	4
19	3	7
20	1	4
total	**54**	**177**

The means (averages) of students' scores of pre-test and post-test are calculated. Table 2 describes the means values of the group tested.

Table 2. The students' scores means of pre-test and post-test (group B)

Time	Treatment	No.	Mean
Sebha University	Without a tool	20	2.7
	Using LearnOOP tool	20	8.85

According to the chart analysis of the means values, it’s clear that the mean value using LearnOOP tool is higher than the mean value without using the tool. Figure 2 explains the idea.

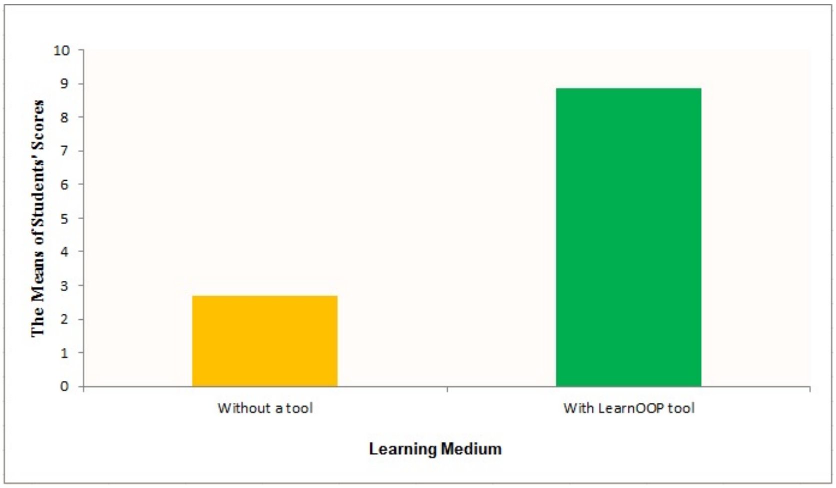


Figure 2. The means of the students’ scores (group B)

The adopted statistical analysis of this experiment is that:

- Null Hypothesis *(H_0)*: the conducted hypothesis is that there is no difference in the mean of pre-test and post-tests scores. In other words, the pre-test and post-tests scores will have equal means.
- Alternative hypothesis *(H_1)*: the alternative hypothesis is that there is at least one difference in the mean of the pre-test and post-test scores in the group tested.
- p-value: the return value of the statistical test which indicates the probability of getting a mean difference between the per-test and post-test as high as what is observed by chance. The lower the P-value, the more significant difference between the groups. The typical significance level that has been chosen in this experiment is 0.001.
- t-test: this test was run on the pre-test and post-test scores. In this experiment, the result t-test shows that there is a difference between the pre-test and post-test according to the p-value which is 0.000 and less than the significance level 0.001. Table 3 shows the result of t-test.

Table 3. The results of t-test

Treatment	No. of student	Mean	p-value	t- test
Without a tool Using LearnOOP Tool	20 20	2.7 8.85	0.000	LearnOOP tool > No tool

The test shows that there is a difference between using the LearnOOP tool and without using it based on the p-value which equal to 0.000. The p-value is less than the significance level (0.001) and that means the improvement from pre-test to post-test is statistically high significant.

Table 4 describes the students' scores of group A (without a tool) for the post. Notice that the maximum score for each student is 15.

Table 4. The students' scores (group A) of post-test

No.	Post-test scores without a tool
1	5
2	2
3	10
4	3
5	3
6	2
7	7
8	2
9	6
10	7
11	4
12	4
13	4
14	4
15	2
16	3
17	2
18	1
19	10
20	13
total	**94**

Again, The students' scores means of post-test for group A is calculated. Table 5 describes the means of both groups tested.

Table 5. The students' scores means of post-test for group A and B

Time	Treatment	No.	Mean
Sebha University	Without a tool (group A)	20	4.7
	Using LearnOOP tool (group B)	20	8.85

According to the chart analysis of the means values, it's clear that the mean value of group B, which uses LearnOOP tool, is more higher than the mean value of the group A, which does not use it. Figure 3 explains the idea.

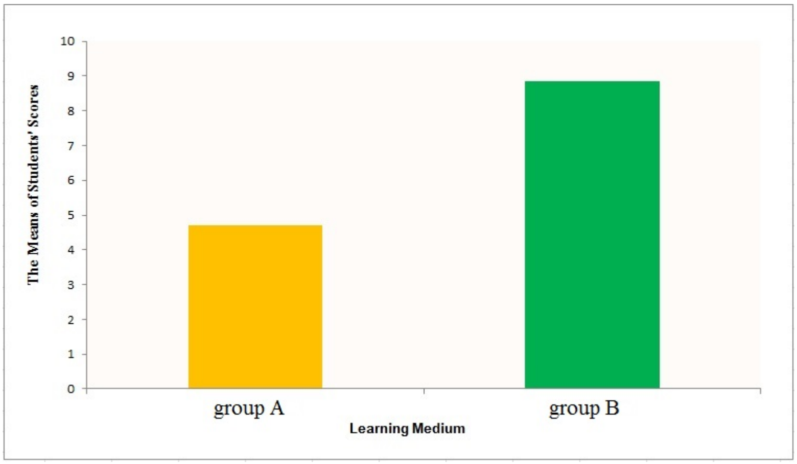


Figure 3. The means of the students' scores between group A and B

The same pervious statistical analysis is adopted for calculating the t-test; in this time, the test was run on the post-test of the two groups; A and B scores. The result t-test shows that there is a difference between the post-test of group A and post-test of group B according to the p-value which is 0.000 and less than the significance level 0.001. Table 6 shows the result of t-test.

Table 6. The results of t-test

Treatment	**No. of student**	**Mean**	**p-value**	**t- test**
Without a tool (group A) Using LearnOOPTool (group B)	20 20	4.7 8.85	0.000	LearnOOP tool > No tool

5. DISCUSSION

The results in this experiment indicate that LearnOOP Tool is more effective and efficient than traditional learning medium. According to our hypothesis testing, there is a significant difference between using LearnOOP Tool as a teaching/learning medium and text-only material. It shows that learning with visualizing role of variable significantly outperformed text-only material. The students realize many facts about object programming, the most important of which is that the object name indicates a variable that contains the object data address in the memory. They also learn from the tool how to extract the object's attributes through visual form. Meanwhile, the overall improvement of enhancing the students' ability for understanding the OOP Java using LearnOOP Tool is demonstrated and achieved. Compared with the previous studies, we almost agree with them on the need for the use of supporting education tools. We also agree on the need of adopting the role of variable concept for clarifying the programming and coding of different programming languages.

6. CONCLUSION AND FUTURE WORK

Regardless of the advancement in the area of educational techniques, it needs to be further tested with more experimental assessment and empirical evaluation, especially of using the teaching/learning visualization and animation tools during lectures and classrooms. Currently, a few researches dealt with the problem of the lack of using these kinds of tools. Furthermore, studies have shown that the role of variable concept educationally enhanced students' understanding if they were supported by visualizing forms from the real world. This paper was motivated by these observations. In particular, this paper suggested more experiments of other tools through empirical evaluation in order to improve their effectiveness and teaching/learning support in the field of programming understanding. Our future work will consider adding more OOP features to LearnOOP tool such as polymorphism. Showing the relationship between entity and objects is also an interested area of OOP programming languages. Another future work is running more experiments and empirical evaluation for LearnOOP tool and other related tools.

REFERENCES

[1] Sajaniemi, J., "Roles of variables and learning to program," in Proc. 3rd Panhellenic Conf. Didactics of Informatics, Jimoyiannis A (ed) University of Peloponnese, Korinthos, Greece, 2005.

[2] Sorva, J., Karavirta, V. & Korhonen, A., (2007) "Roles of variables in teaching," Journal of Information Technology Education, vol. 6, pp. 407-423.

[3] Byckling, P, Gerdt, P., & Sajaniemi, J., "Visualizing roles of variables to novice programmers," in Proceedings of the 14th Annual Workshop of the PPIG'02, 2002, pp. 111-127.

[4] Reenskaug, T., "Roles and Classes in Object Oriented Programming", 2007.

[5] Kuittinen, M., & Sajaniemi, J., "Teaching roles of variables in elementary programming courses," ACM SIGCSE Bulletin, vol. 36, pp. 57-61, 2004.

[6] Shi, N., Min, Z., & Zhang,P., (2017) "Effects of visualizing roles of variables with animation and IDE in novice program construction," Telematics and Informatics.

[7] Gerdt, P., & Sajaniemi, J., "Roles of Variables in Object Oriented Programming", OOPSLA'05, San Diego, California, USA, ACM 1595931937/05/0010, October 16–20, 2005.

[8] Xinogalos, S., Sartatzemi, M., & Dagdilelis, V., "Studying Students' Difficulties in an OOP Course Based on BLUEJ", Proceedings of the 9th IASTED International Conference, Computers and Advanced Technology in Education, Lima, Peru 2006.

[9] Sajaniemi, J., & Kuittinen, M., "Program Animation Based on the roles of Variables," in Proceedings of the 2003 ACM symposium on Software Visualization, 2003.

[10] Rajala, T., Laakso, M.-J., Kaila, E., & Salakoski, T., (2008) "Effectiveness of Program Visualization: A Case Study with the ViLLE Tool," Journal of Information Technology Education, vol. 7.

[11] Laakso, M.-J., Malmi, L., Korhonen, A., Rajala, T., Kaila, E., & Salakoski, T., (2008) "Using roles of variables to enhance novice's debugging work," Issues in Informing Science and Information Technology, vol. 5, pp. 281-295.

[12] Esteves, M., & Mendes, A., "OOP-Anim, a system to support learning of basic object oriented programming concepts," in Proceedings of CompSysTech'2003-International Conference on Computer Systems and Technologies. Sofia, Bulgaria, 2003.

[13] Levy, R., Ben-Ari, M., & Uronen, P. A., (2003) "The Jeliot 2000 program animation system," Computers & Education, vol. 40, pp. 1-15.

[14] Ben-Ari, M., Myller, N., Sutinen, E., & Tarhio, J., (2002) "Perspectives on program animation with Jeliot," in Software Visualization, ed: Springer, pp. 31-45.

[15]Moreno, A., Myller, N., Sutinen, E., & Ben-Ari, M., "Visualizing programs with Jeliot 3," in Proceedings of the working conference on Advanced visual interfaces, 2004, pp. 373-376.

[16] Čisar, S., M., Radosav, D., Pinter, R., Čisar, P., (2011) "Effectiveness of Program Visualization in Learning Java: a Case Study with Jeliot 3" Int. J. of Computers, Communications & Control, Vol. VI, No. 4 (December), pp. 668-680.

[17] Al-Barakati N. M., & Al-Aama, A. Y., "The effect of visualizing roles of variables on student performance in an introductory programming course," ACM SIGCSE Bulletin, vol. 41, pp. 228-232, 2009.

[18] Pallant, J., (2010) SPSS Survival Manual: A step by step guide to data analysis using SPSS. Berkshire UK: McGraw-Hill Education.

[19] Hundhausen, C. D., Douglas, S. A., & Stasko, A. T., (2002) "A meta-study of algorithm visualization effectiveness," Journal of Visual Languages and Computing, vol. 13, pp. 259-290.

[20] Yuan, X., Vega, P., Qadah, Y., Archer, R., Yu, H., & Xu, J., (2010) "Visualization Tools for Teaching Computer Security," ACM Transactions on Computing Education, vol. 9, pp. 147-155.

AUTHORS

Mabroukah Amarif: received her BSc degree in Computer Science from University of Sebha, Libya, MSc in Computer Science from Universiti Sains Malaysia, and PhD in Software Engineering from Universiti Kebangsaan Malaysia. Her interests span a wide range of topics in the area of Software Engineering, Networking, Computer Security, Visual Informatic, Computer Education and programming languages. She is currently working as Assistant Professor at the departement of computer science, Faculty of Information Technology in Sebha University of Libya.

Sakeenah Ahmed: received her BSc degree in Computer Science from University of Sebha, Libya. She is currently doing her MSc in computer Sciences at Sebha University of Libya. She interests in the area of Software Engineering, Networking, Computer Security and Computer Education. she is currently working as teacher of computer sciences in secondary school of Libya