ISSN 2518-1467 (Online),
ISSN 1991-3494 (Print)

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ҰЛТТЫҚ ҒЫЛЫМ АКАДЕМИЯСЫНЫҢ
Абай атындағы Қазақ ұлттық педагогикалық университетінің

Х А Б А Р Ш Ы С Ы

ВЕСТНИК

НАЦИОНАЛЬНОЙ АКАДЕМИИ НАУК РЕСПУБЛИКИ КАЗАХСТАН
Казахский национальный педагогический университет имени Абая

THE BULLETIN

THE NATIONAL ACADEMY OF SCIENCES OF THE REPUBLIC OF KAZAKHSTAN
Abai Kazakh National Pedagogical University

PUBLISHED SINCE 1944

2 (396)

MARCH – APRIL 2022

ALMATY, NAS RK

Қазақстан Республикасы Ұлттық ғылым академиясы «ҚР ҰҒА Хабаршысы» ғылыми журналының Web of Science-тің жаңаланған нұсқасы Emerging Sources Citation Index-те индекстелуге қабылданғанын хабарлайды. Бұл индекстелу барысында Clarivate Analytics компаниясы журналды одан әрі the Science Citation Index Expanded, the Social Sciences Citation Index және the Arts & Humanities Citation Index-ке қабылдау мәселесін қарастыруда. Web of Science зерттеушілер, авторлар, баспашылар мен мекемелерге контент тереңдігі мен сапасын ұсынады. ҚР ҰҒА Хабаршысының Emerging Sources Citation Index-ке енуі біздің қоғамдастық үшін ең өзекті және беделді мультидисциплинарлы контентке адалдығымызды білдіреді.

НАН РК сообщает, что научный журнал «Вестник НАН РК» был принят для индексирования в Emerging Sources CitationIndex, обновленной версии Web of Science. Содержание в этом индексировании находится в стадии рассмотрения компанией Clarivate Analytics для дальнейшего принятия журнала в the Science Citation Index Expanded, the Social Sciences Citation Index и the Arts & Humanities Citation Index. Web of Science предлагает качество и глубину контента для исследователей, авторов, издателей и учреждений. Включение Вестника НАН РК в Emerging Sources Citation Index демонстрирует нашу приверженность к наиболее актуальному и влиятельному мультидисциплинарному контенту для нашего сообщества.

NAS RK is pleased to announce that Bulletin of NAS RK scientific journal has been accepted for indexing in the Emerging Sources Citation Index, a new edition of Web of Science. Content in this index is under consideration by Clarivate Analytics to be accepted in the Science Citation Index Expanded, the Social Sciences Citation Index, and the Arts & Humanities Citation Index. The quality and depth of content Web of Science offers to researchers, authors, publishers, and institutions sets it apart from other research databases. The inclusion of Bulletin of NAS RK in the Emerging Sources Citation Index demonstrates our dedication to providing the most relevant and influential multidiscipline content to our community.

«Қазақстан Республикасы Ұлттық ғылым академиясының Хабаршысы».
ISSN 2518-1467 (Online),
ISSN 1991-3494 (Print).
Меншіктенуші: «Қазақстан Республикасының Ұлттық ғылым академиясы» РҚБ (Алматы қ.).
Қазақстан Республикасының Ақпарат және коммуникациялар министрлігінің Ақпарат комитетінде 12.02.2018 ж. берілген
№ 16895-Ж мерзімдік басылым тіркеуіне қойылу туралы куәлік.
Тақырыптық бағыты: *әлеуметтік ғылымдар саласындағы зерттеулерге арналған.*
Мерзімділігі: жылына 6 рет.
Тиражы: 300 дана.
Редакцияның мекен-жайы: 050010, Алматы қ., Шевченко көш., 28, 219 бөл., тел.: 272-13-19
http://www.bulletin-science.kz/index.php/en/

Типографияның мекен-жайы: «Аруна» ЖК, Алматы қ., Мұратбаев көш., 75.

«Вестник Национальной академии наук Республики Казахстан».
ISSN 2518-1467 (Online),
ISSN 1991-3494 (Print).
Собственник: РОО «Национальная академия наук Республики Казахстан» (г. Алматы). Свидетельство о постановке на учет периодического печатного издания в Комитете информации Министерства информации и коммуникаций и Республики Казахстан **№ 16895-Ж,** выданное 12.02.2018 г.
Тематическая направленность: *посвящен исследованиям в области социальных наук.*
Периодичность: 6 раз в год.
Тираж: 300 экземпляров.
Адрес редакции: 050010, г. Алматы, ул. Шевченко, 28, ком. 219, тел. 272-13-19
http://www.bulletin-science.kz/index.php/en/

Адрес типографии: ИП «Аруна», г. Алматы, ул. Муратбаева, 75.

Bulletin of the National Academy of Sciences of the Republic of Kazakhstan.
ISSN 2518-1467 (Online),
ISSN 1991-3494 (Print).
Owner: RPA «National Academy of Sciences of the Republic of Kazakhstan» (Almaty). The certificate of registration of
a periodical printed publication in the Committee of information of the Ministry of Information and Communications
of the Republic of Kazakhstan **No. 16895-Ж,** issued on 12.02.2018.
Thematic focus: *it is dedicated to research in the field of social sciences.*
Periodicity: 6 times a year.
Circulation: 300 copies.
Editorial address: 28, Shevchenko str., of. 220, Almaty, 050010, tel. 272-13-19
http://www.bulletin-science.kz/index.php/en/

Address of printing house: ST «Aruna», 75, Muratbayev str, Almaty.

BULLETIN OF NATIONAL ACADEMY OF SCIENCES
OF THE REPUBLIC OF KAZAKHSTAN
ISSN 1991-3494
Volume 2, Number 396 (2022), 44-64
https://doi.org/10.32014/2022.2518-1467.271

Bayanbek Amantay

National School of Physics & Math Almaty, Almaty, Kazakhstan.

TEACHING MAIN CONCEPTS OF OBJECT-ORIENTED PROGRAMMING USING GREENFOOT INTEGRATED DEVELOPMENT ENVIRONMENT

Abstract. In this paper, teaching object-oriented programming at school will be considered using the Greenfoot visual programming environment. Greenfoot is a Java Interactive Development Environment (IDE) designed primarily for educational purposes at the high school and undergraduate level. This simplifies the development of 2D graphics applications such as simulations and interactive games. Greenfoot is intended to be used in educational environments, although anyone can use it to learn the language at their own pace. Students got to know the program in computer science course and so far they learn a lot from Greenfoot. Also, this program does not actually teach puples Java, but instead allows them to easily modify the code and see what happens when the execution had happen. The Java programming language is the basic one for teaching OOP concepts in most of the countries, including Nazarbayev Intellectual Schools in Kazakhstan. As a result, almost all computer science teachers are familiar with the basics of programming in Java. Therefore, it is Java that should become the primary language in schools for learning object-oriented programming. The choice of Java as the language for teaching is reasonable: firstly, Java is recognized as the best powerful language for teaching the basics of programming. Secondly, Java is an object-oriented programming language that supports main concepts such as class, object, inheritance. Thirdly, nowadays, Java is studied in most of the country's higher education.

However, there are still some problems in teaching and learning OOP concepts at the school level. Researchers suggest that using visual programming environments such as Greenfoot with gamification can be useful for learning object-oriented programming for novices.

Key words: Object-Oriented programming, Teaching Computer Science, programming languages, Educational tools, Greenfoot, Scratch.

Баянбек Амантай

Алматы Ұлттық физика-математика мектебі, Алматы, Қазақстан.

GREENFOOT ИНТЕГРАЦИЯЛАНҒАН ДАМУ ОРТАСЫН ҚОЛДАНА ОТЫРЫП, ОБЪЕКТІГЕ БАҒЫТТАЛҒАН БАҒДАРЛАМАЛАУДЫҢ НЕГІЗГІ ТҰЖЫРЫМДАМАЛАРЫН ОҚЫТУ

Аннотация. Бұл мақалада Greenfoot визуалды бағдарламалау ортасын қолдана отырып, мектепте объектіге бағытталған бағдарламалауды оқыту қарастырылады. Greenfoot - бұл Java-ның интерактивті даму ортасы (IDE), ол негізінен орта мектеп пен бакалавриат деңгейінде білім беру мақсаттарына арналған. Бұл тренажерлер мен интерактивті ойындар сияқты 2D графикасы бар қосымшаларды әзірлеуді жеңілдетеді. Greenfoot білім беру ортасында қолдануға арналған. Дегенмен кез-келген адам оны өз тілін үйрену үшін қолдана алады. Студенттер информатика курсында бағдарламамен танысады және осы уақытқа дейін олар Гринфуттан көп нәрсені үйренеді. Сонымен қатар, бұл бағдарлама Java оқушыларға үйретілмейді, керісінше оларға кодты оңай өзгертуге және орындау кезінде не болып жатқанын көруге мүмкіндік береді. Java бағдарламалау тілі Қазақстандағы Назарбаев Зияткерлік мектептерін қоса алғанда, көптеген елдерде ООР тұжырымдамаларын оқыту үшін негіз болып табылады. Нәтижесінде, информатика мұғалімдерінің барлығы дерлік Java бағдарламалау негіздерімен таныс. Сондықтан Java объектіге бағытталған бағдарламалауды үйрену үшін мектептерде негізгі тіл болуы керек. Java-ны оқыту тілі ретінде таңдау негізделген: біріншіден, Java бағдарламалау негіздерін үйренудің ең жақсы қуатты тілі ретінде танылды. Екіншіден, Java - бұл класс, объект, мұрагерлік сияқты негізгі ұғымдарды қолдайтын объектіге бағытталған бағдарламалау тілі. Үшіншіден, Java қазіргі уақытта елдің көптеген жоғары оқу орындарында оқытылады. Алайда, мектеп деңгейінде ООР тұжырымдамаларын оқыту мен оқытуда әлі де кейбір мәселелер бар. Зерттеушілер Геймификациясы бар Greenfoot сияқты визуалды бағдарламалау ортасын қолдану жаңадан бастаушыларға объектіге бағытталған бағдарламалауды үйрену үшін пайдалы болуы мүмкін деп болжайды.

Түйін сөздер: объектіге бағытталған бағдарламалау, информатиканы оқыту, бағдарламалау тілдері, білім беру құралдары, Greenfoot, Scratch.

Баянбек Амантай

Национальная физико-математическая школа Алматы, Алматы, Казахстан.

ОБУЧЕНИЕ ОСНОВНЫМ КОНЦЕПЦИЯМ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ ИНТЕГРИРОВАННОЙ СРЕДЫ РАЗРАБОТКИ GREENFOOT

Аннотация. В этой статье будет рассмотрено преподавание объектно-ориентированного программирования в школе с использованием среды визуального программирования Greenfoot. Greenfoot - это интерактивная среда разработки Java (IDE), предназначенная в первую очередь для образовательных целей на уровне средней школы и бакалавриата. Это упрощает разработку приложений с 2D-графикой, таких как симуляторы и интерактивные игры. Greenfoot предназначен для использования в образовательных средах, хотя любой желающий может использовать его для изучения языка в своем собственном темпе. Студенты познакомились с программой на курсе информатики, и до сих пор они многому учатся у Гринфута. Кроме того, эта программа на самом деле не обучает учеников Java, а вместо этого позволяет им легко изменять код и видеть, что происходит, когда происходит выполнение. Язык программирования Java является базовым для преподавания концепций ООП в большинстве стран, включая Назарбаев интеллектуальные школы в Казахстане. В результате почти все преподаватели информатики знакомы с основами программирования на Java. Поэтому именно Java должна стать основным языком в школах для изучения объектно-ориентированного программирования. Выбор Java в качестве языка для преподавания обоснован: во-первых, Java признана лучшим мощным языком для обучения основам программирования. Во-вторых, Java - это объектно-ориентированный язык программирования, который поддерживает основные понятия, такие как класс, объект, наследование. В-третьих, в настоящее время Java изучается в большинстве высших учебных заведений страны. Однако все еще существуют некоторые проблемы в преподавании и изучении концепций ООП на школьном уровне. Исследователи предполагают, что использование визуальных сред программирования, таких как Greenfoot с геймификацией, может быть полезно для изучения объектно-ориентированного программирования новичками.

Ключевые слова: объектно-ориентированное программирование, обучение информатике, языки программирования, образовательные инструменты, Greenfoot, Scratch.

Introduction. Computer science is one the rapidly developing field of human life and has become an indispensable attribute of many professions. Therefore, the subject of computer science has a special place in the curriculum of Nazarbayev Intellectual Schools (NIS). The main goal of teaching Computer science mainly based on develop critical thinking and problem-solving skills of the students (Nazarbayev Intellectual Schools, 2016). The school system should guarantee that the graduates have sufficient knowledge and skills to use modern information technologies in their chosen field. The development of computer skills for all students is one of the primary tasks of the contemporary education of NIS.

Recently, the Computer Information Communication Technology curriculum of Kazakhstan schools had several changes. Nazarbayev Intellectual schools were one of the first organization who have started implementing those changes. Currently, the methodology of teaching computer science in NIS for schoolchildren is developing intensively - goals and principles are being specified; teaching methods and tools are being designed (Nazarbayev Intellectual Schools, 2016). That considers not only modern aspects of information technology which need to be taught at school but also new approaches and technics which help to learn some of the challenging topics.

The concept of object-oriented programming (OOP) indicates that the basis for controlling the process of program implementation is the transmission of messages to objects (Horstmann, 2016). It means objects know each other, and they are strongly linked. Consequently, objects must be defined together with messages to which they must respond when the program is running. This is the main difference between OOP and procedural programming, where separately defined data structures are transferred to procedures as parameters (Georgantaki et.al, 2007). Thus, an object-oriented program consists of objects that interact with each other through specific interfaces (Horstmann, 2016). Furthermore, researchers had conducted numerous researches on how to teach object-oriented concepts effectively and to overcome difficulties that students mainly face. Studies conducted by (Georgantaki et.al, 2007) and (Ben-Ari et.al, 2002) indicate that students may have some problems when they start learning the OOP paradigm, which results in poor academic performance on the subject. Moreover, researchers indicate the effectiveness of using educational tools and gamification on

teaching object- oriented concepts. Educational tools such as Greenfoot can be beneficial for students to learn the OOP paradigm; it is also considered as a means of increasing students' motivation and, in general, as a form of building the educational process (Begosso et.al, 2012).

Literature review. The concept of Object-oriented programming first appeared in the 60s of the last century, and it becomes the leading programming technology of recent years. There is because the OOP gives a chance to software developers to increase the efficiency of a sophisticated program. Currently, object-oriented programming is widely used in the creation of software products, and OOP topics are included in the compulsory minimum of the content of computer science education. At the same time, the course materials, methodology for studying the topics of object-oriented programming, and supporting materials in a secondary school has not been sufficiently developed. One of the main points is choosing a suitable programming language and programming environment. Also, to support teaching and learning OOP concepts, there are several educational tools have been established. According to (Georgantaki et.al, 2007), Greenfoot is one of the most widely used tool and it can be used to inspire students to the subject by letting them create their games while learning OOP concepts. In addition, Greenfoot is an integrated development environment that supports learning Java programming language. One of the advantages of this environment is that it was created specifically for training schoolchildren and students in programming, which means that the creators of this product tried to make the interface of the tool comfortable and clear interface for learning programming novices. In this session, theory about object-oriented programming and teaching the OOP concepts, existing computer science curriculum of the Nazarbayev Intellectual Schools, and the Greenfoot programming environment will be discussed.

Object-oriented programming paradigm and teaching. According to (Niemeyer, 2017), the basis of object-oriented programming is the idea of grouping data and the action into one structure, which has the same features. As it said, OOP is a methodology of programming based on Object, where each of them is an exemplar of a class, and classes form a hierarchy of inheritance. The object is an entity that has a specific behaviour and specific way of representing, and Class is a template by which objects are created that is for control the state and behaviour of objects identified by Deitel (Deitel, 2015). As (Niemeyer, 2017) pointed out, OOP is based on three basic principles, such as encapsulation, inheritance, and polymorphism. Encapsulation: a principle that used to hide some of the implementation details of the software components where required. In order to interact with those components, an

interface is required. In this case, the user can interact with an object only through an interface. Inheritance: the principle that to describe a new class based on a parent property and parent class functionality is borrowed by the new class. In other words, the subclass implements the specification of a superclass. As it shown by (Horstmann, 2016), in Inheritance, the fields and methods of the parent class will remain, and the programmer can add more fields to the subclass. Using Inheritance allows avoiding duplicate and redundancy of the code since the same methods and fields do not need to be repeated in subclasses. In consequence, the child class can handle the objects of the parent class in the same way. For example, 'Programming languages' is a parent class, from which the classes 'Java', 'C++', 'Python', etc. are inherited. Polymorphism: the ability of objects with the same specification to have a different implementation. In other words, the ability to use one interface many times for various purposes. Polymorphism is writing the same block of code, and most importantly the name can be the same, but allowing is to take a different type of variable, which enables the programmer to write more abstract programs and increase the ability to reuse code.

The experiences outlined by (Sorva, 2013) demonstrates that begin learning to program was continuously challenging by novices, where pupils confront several difficulties related to methods and programming concepts. Concurring to his idea about, these issues are shared among the learners, and there are not restricted to a specific school or a programming language. In his paper, he also indicates several programming misconceptions among the pupils, such as data type and assigning values to the variables, print statements, and conditionals. He also concludes that assigning a value to the variable, array and loop statements with iteration are common mistakes that students often make.

According to (Yan, 2009), the study of object-oriented programming can be associated not only with learning a new programming language but also with a new object-oriented type of thinking, which involves more problem-solving skills. Learning object-oriented programming requires pre-existing programming skills, logical thinking, and problem-solving skills. Another researcher, (Wing, 2008), also shows that computational thinking is an intelligent approach to solve a complex task, which involves a lot of mathematical calculations and analysing the possible solutions. Furthermore, if students do not have these skills, they will face learning difficulties when it comes to OOP. The study contacted by (Yan, 2009) indicates that, understanding the abstract concepts are most challenging among the students. He emphasises that for the main concepts like Class and Object are often hard to find a real example from everyday life. In consequence, students will

find it more difficult when they learn other complex topics. As pointed by Su and (Hsu, 2017), students always face difficulties when study the main six abstract concepts such as class, object, method, encapsulation, inheritance, and polymorphism. The students found the concepts hard to imagine and remember the consequence and the syntax.

In the period of a rapidly growing number of object languages and the development of object-oriented technologies, the problem of choosing a programming language for acquaintance with the basics of object-oriented programming is acute. Unfortunately, currently, there is no universally recognized language for learning the basics of object-oriented programming, which is widely used to teach the basics of programming. One option may be the Java language (Kölling, 1999; Rosenberg, 2001), created in 1995 by Sun Microsystems, which is more common than other OOP languages and has many advantages. One of its significant benefits is cross-platform since the source code of the program is compiled into unique byte code so that it can be run on any Java virtual machine, regardless of the architecture. But at the same time, a significant drawback of this language is the number of resources consumed (Papastergiou, 2009). Java can be suitable for teaching at a school level because the syntax of this language is similar to other OOP languages such as C and C ++.

Unfortunately, the methodology of teaching object-oriented programming is still under experimenting by researchers. Educators did not come up with a proven way of teaching OOP. Researchers have different opinions about whether to start teaching OOP from primary education or in high school only. Some people believe that teach OOP concepts at primary school can cause students not understanding the concepts very well. Considerable research related to teaching object-oriented programming concepts by novices was conducted by (Kölling, 2001) emphasizes that start teaching from the objects can be beneficial than teaching how to code first. The object-first teaching method is widely used by teachers and given in textbooks and suggested to use. The object-first is one of the essential approaches when students first time with coding, as shown in the research papers, students always find it challenging to start writing code. As pointed by (Sorva, 2013), students often find it difficult to remember commands and the sequence of the code. If students were given objects first, then teach them the interaction between the objects, it will result in a better understanding of the main method and method invoke. According to (Barnes et.al, 2005), the main concepts, such as object and class, should be given at the first lesson with a real example of everyday life. A good case can be school, student, or house, and the primary focus should be on an object and interaction with other objects. Furthermore,

the definition of a class can be given with simple code examples with main points such as variables, data type, and parameter passing.

Dealing with year 9 computer science curriculum of NIS. As shown by (Nazarbayev Intellectual Schools, 2016), the importance of teaching Computer Science includes not only teaching students how to use computers for practical purposes but also to accumulate new knowledge and methods for solving problems that require a computational approach to thinking. Computational thinking by itself includes teaching students how to solve tasks, how to design an intelligent system and to understand the interaction between humans and high technologies. The ultimate purpose of the year nine computer science course is to prepare students for the hight competitive technology century and encourage them to invent a smart software system using the knowledge obtained from other disciplines. (Nazarbayev Intellectual Schools, 2016). They also indicate that the subject is not only theoretical with its concepts and principles but also has an extensive practical base, providing resources and opportunities for studying other disciplines.

According to NIS computer science's actual course and curricula, the course is estimated for 68 hours, and the topic of object-oriented programming is planned to be taught in the third term in section 9.3A in 20 academic hours all together. The primary learning aim of the chapter is to introduce students to the components of an object-oriented programming language, solve simple tasks by using the principal components OOP and create application.

Greenfoot. Greenfoot is a graphical programming environment that is developed at the University of Kent (UK) and Deakin University (Melbourne, Australia), using Oracle. In his 2012 text, Kölling states that 'Greenfoot (http://greenfoot.org) is an educational programming environment designed to learn and teach object-oriented programming concepts (using the Java language) with games and simulations'. Consequently, the Greenfoot environment mainly created for making games using semantics of the Java language. It provides students with the opportunity to concentrate on the algorithmic structure of their written programs that do not require graphics and interfaces. There are several reasons why Greenfoot is chosen as a helper tool for teaching OOP at school. Firstly, the Greenfoot environment is entirely free and is distributed under the open GPL license. Secondly, the program supports the operating systems Windows, Mac OS, Linux, or any other Java virtual machine. Third, the environment is visually oriented and can be easily used by students.

According to (Kölling, 2012), the Greenfoot development environment helps in a more accessible way to present the student with the basic principles of object-oriented programming by visualizing the states and behaviour of

objects. Using Greenfoot allows visualizing the model of the program with executing methods of classes and objects in the process of writing them. It can be done by creating an instance of a class of objects by calling a method or direct execution. Also, testing the Class of objects by setting different parameters for methods. Also, the study conducted by (Chandrashekar, 2018), provides positive feedback from students and teachers about these technologies. At the same time, one of the teachers' comments says that when using Scratch and Greenfoot, students come to understand the programming aspects much faster than when learning other programming languages.

Greenfoot is focused on the creation of computer games, which is designed to stimulate the interest of the students. Every project created in this development environment is called scenarios (Figure 1). Also, the projects created in Greenfoot can be shared on the official website, where developers and users created many scenarios. In the following picture, the Greenfoot environment window will be considered in more detail.

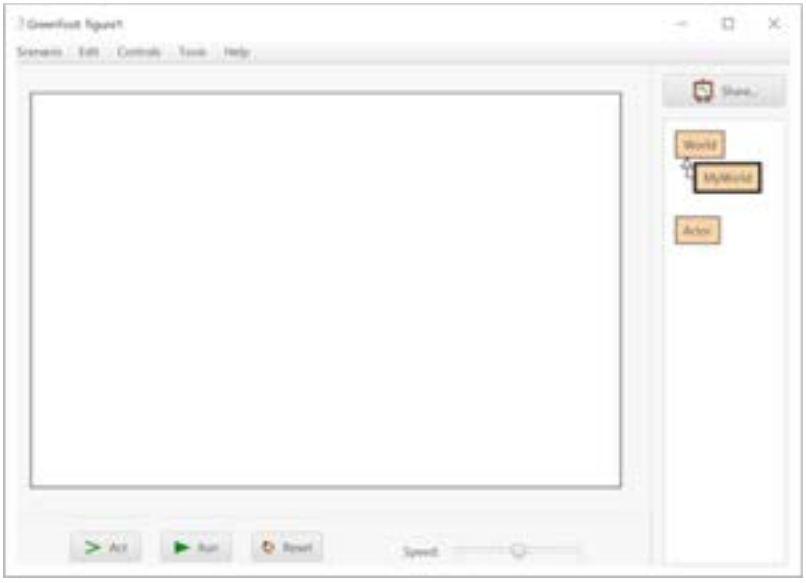

Figure 1. The main window

There are five buttons on the menu bar such as Scenario, Edit, Controls, Tools, Help. On the right side of the window shown scheme inheritance of World and Actor classes. The World and the Actor are the main classes in the Greenfoot development environment.

Besides, Greenfoot has a text editor (Figure 2) for writing program code. To invoke the editor, the user needs to right-click on the class which desired to edit in the class diagram, then click on 'Open editor.

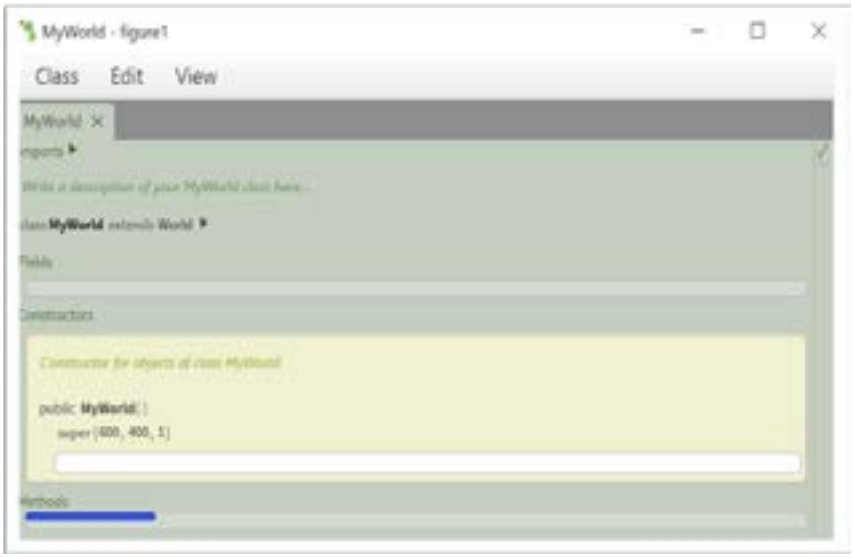

Figure 2. The code editor

Methodology. A systematic review method used to synthesize the qualitative findings of the research. The definition of the systematic review method defined by (Gough et.al, 2012), as a method that aggregates primary qualitative studies and integrates them into an extensive review, which is about individual experience. Furthermore, it comes up with some interpretation of what new knowledge the studies have by reviewing the existing literature. By their definition, a systematic review is to collate existing research on a specific question by synthesizing the result of several studies, and they should be conducted the same principle the is expected from primary research (Sandelowski et.al, 2007). The systematic review aims to summarise and help researchers to understand the findings. Several factors may motivate to undertake a systematic review method such as review can be conducted to resolve conflicting evidence, to explore variation in practice, and to confirm the appropriateness of current practice or highlight a need for future research. They also emphasize that the main advantage of the systematic review method is the ability to provide more comprehensive training about the research topic. Besides, when the systematic review well-conducted should give the best estimate, and the value often depends on what was done, what was found, which affects the clarity of the report.

The inclusion criteria for the scientific papers was that the research must be about teaching computer science, teaching programming to novices, teaching and learning object-oriented concepts, and teaching OOP in Java using educational tools.

Table1 Reviewed studies

Author	Topic	Participants	Method	Results	Focus on teaching OOP/using educational tools?
Ragonis and Ben-Ari, 2005	A Long-Term Investigation of the Comprehension of OOP Concepts by Novices	high school students	constructivist qualitative research methodology based on existing theories	A better understanding of OOP concepts and show the capability to classify methods and classes.	teaching OOP
Yan, 2009	Teaching Object-Oriented Programming with Games	First-year students	Case study	The uses of gamification technologies in teaching programming have a positive impact on students' interest in computer science and programming lessons	Greenfoot
Al-Bow et al., 2008	Using Greenfoot and Games to Teach Rising 9th and 10th Grade Novice Programmers	Grade 9 and Grade 10 students	Survey conducted	Deeper understanding of OOP concepts,	Greenfoot
Su and Hsu, 2017	Building a Visualized Learning Tool to Facilitate the Concept Learning of Object-Oriented Programming	Students	A prototypical VLT-OOP tool tested on the Web-based platform	A web-based VLT-OOP visual tool has been developed	Scratch and Blockly
Begosso et al., 2012	An approach for teaching algorithms and computer programming using Greenfoot and Python	Computer Science teachers and undergraduate students	Two Case studies	The result of first and second case study shown 60% of success in the assessment.	Greenfoot Python

Livovský and Porubän, 2014	Learning object-oriented paradigm by playing computer games: concepts first approach	first year students	questi-onnaire	The Object first method can be successfully used in tea-ching OO programming. Gamification has a significant impact on teaching CS in all the edu-cation levels.	teaching OOP
Georgantaki and Retalis, 2007	Using Educational Tools for Teaching Object Oriented Design and Programming Georgantaki	postg-raduate students	Case study	BlueJ programming environment has advantages in teaching the OOP	Educa-tional Program-ming Environ-ments
Kölling and Rosenberg (2001)	Guidelines for teaching object orientation with Java	first year students	summarise	Suggested 8 guidelines for teaching OOP	teaching OOP BlueJ
Khamis et al., 2008	Assessing Object-oriented Programming Skills in the Core Education of Computer Science and Information Technology: Introducing New Possible Approach	Experts and students	questi-onnaire	An assessment model for teaching OOP developed. Possible problems and solutions discussed in teaching and assessing Computer Science.	teaching OOP
Udvarod and Guban, 2016	Demonstration the class, object and inheritance concepts by software	primary and secondary school students	questi-onnaire	CS curriculum at the school level is not enough deve-loped; they do not receive support. The teaching method needs to be developed.	teaching OOP

Milne and Rowe, 2002	Difficulties in Learning and Teaching Programming—Views of Students and Tutors	Second year students	a web-based questi-onnaire	Students found the object-oriented concepts the most difficult topic of coping with	teaching OOP
Rahman, 2018	From App Inventor to Java: Introducing Object-oriented Programming to Middle School Students Through Experiential Learning	middle school students	Qualitative research, survey	Using class activities can enhance the learning of the OOP concepts.	teaching OOP
Kölling, 2008	Greenfoot - A Highly Graphical IDE for Learning Object-Oriented Programming	students	Qualitative research	Greenfoot is a powerful toll for teaching OOP concepts	Greenfoot
Sorva, 2013	Notional Machines and Introductory Programming Education		Qualitative research	Figures out the possible challenges which students face when they first time learn to program and summarised the misconception topics from object-oriented programming content.	teaching OOP

In order to support the main research method, a supplementary quantitative questionnaire was performed. The participants were asked to answer 12 different types of questions using www.surveymonkey.com.

Data for the research were collected using Google Scholar and Birmingham University Library search system using keywords ‘Object-Oriented programming’, ‘Teaching Computer Science,’ ‘programming languages,’ ‘Educational tools,’ ‘Greenfoot,’ ‘Scratch.’ The search strategy organized by using books, online journals, and recent publications using the

keywords written before. The search limited by the condition of seeking in English materials and articles sked to be published after 2000 years.

Findings and discussion . To sum up the reviewed literature, ten studies were chosen, and data represented in Table 1 above. To evaluate the quality and investigate potential findings used a methodology called subject research design. According to the method, all the studies will be chosen based on criteria and will be examined separately with a comparison relevant. According to (Horner et.al, 2005), qualitative one-subject research design is the best way to study the quality of generalized research. We decided to have six different categories, including Author, Topic, the Participants, the Method used for the research and the results of teaching OOP using educational tools.

The first study (Ragonis et.al, 2005) investigated that students showed an ability to learn OOP concepts at the high school level. According to his findings, almost all students demonstrated they understand object-oriented programming concepts such as class, object, inheritance, polymorphism, and encapsulation. Based on the result of the questionnaire, they indicated that students showed a deep knowledge of the main method, calling method, void method, and method with parameters.

(Yan, 2009) investigated that the way of teaching object-oriented programming with gamification technologies has a positive effect on the students learning with encouraging them to learn more with creating interesting games. Also, they choose an approach to teach students the object-oriented concepts first, then programming where students use theoretical knowledge in-game construct examples.

(Al-Bow et.al, 2008) investigated the positive changes on students' attitude towards subject using Greenfoot environment for learning OOP concepts. They also analysed the survey which seventeen students have completed by the end of camp and shown and understanding of main OOP concepts as well as conditional statements and loop statements.

(Begosso et.al, 2012) represented two case study where one is about teaching object-oriented concepts using Greenfoot programming environment to first-year students and the second case study was conducted with high-school students in which they studied Python. The first study has analysed the students' knowledge before and after the course and shown that students have made progress on learning.

The studies (Kölling, 2008) and (Livovský et.al, 2014) figured out that Structured Programming Languages does not disclose in detail the implementation of classes and objects. However, knowledge of the application of classes and objects in a programming system allows the programmer to master and consciously apply the techniques of object-oriented programming

genuinely. Using visual tools such as Scratch and Greenfoot can be the solution for better understanding the OOP concepts.

Before start creating the course program and teaching materials, we decide to have a questionnaire which is for Computer Science teachers. The main goal of the survey was to evaluate the existing CS curriculum and the potential difficulties and limitations which teachers face during teaching object-oriented concepts at school. We have asked teachers to answer 12 questions, which were multiple-choice, Checkboxes with selecting multiple answers from the list of answer choices, and rating scale questions. In order to ask teachers their own opinion, the last question was respondents to write in a text field. The full questionnaire can be found at: *https://www.surveymonkey.com/summary/OrIc8P1ACguM5Gv4Ixg7fRQxsACqxf2BV*2FBNhrJstqX4vZ_2BjvTbO9PMZ5c4s3NH1X.

The main points of the questionnaire are given following:

a. Question asks about teachers' work experience, in particular, how long is he or she is teaching Computer Science. This question is essential to identify whether the difficulties depend on the length of work experience teacher.

b. This type of question asks about to rate the given programming languages in the level of knowledge. This question aims to know what object-oriented programming languages teachers know.

c. The other group of questions is designed to identify the problematic topics of the object-oriented paradigm. Participants asked to rate the difficulty of the topics of learning by students. Also, participants were asked about the reason why some of the OOP concepts are difficult to learn/teach.

d. The fourth group of questions aimed to figure out what teaching support tools and programming environments teachers use. Also, we decide to ask whether they already use Greenfoot on teaching programming.

From question one, we can see the average number of teachers' work experience is nine years, six months, which means most of the teachers have enough experience in teaching Computer Science at school. On question two, teachers asked about their experience with some programming languages, according to the result (Figure 4), teachers feel they are expert on Pascal. Also, they have enough experience in programming languages like C ++, C and C. Only, few respondents respond they are confident in Java. Besides, we can see Python programming language is new for most of the teachers.

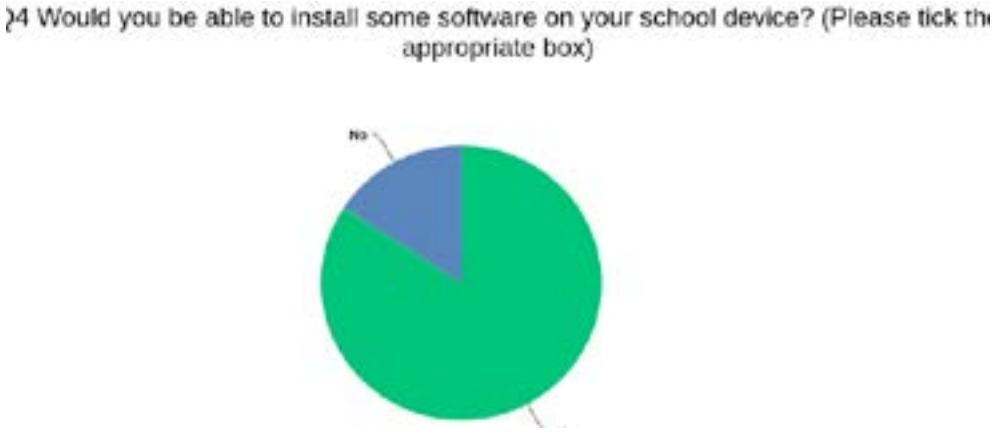

Figure 4. Teaching experience with programming language

The third question asked about what kind of computer do teachers have access to in their school and analysing the responses all the teachers have computers which essential for teaching computer science, especially teaching programming for novices. The following question asked whether teachers have access to install software to the computers, As would be expected(Figure 5), 85% of respondents indicated they could install new software, which is vital for using Greenfoot for teaching Object-oriented concepts.

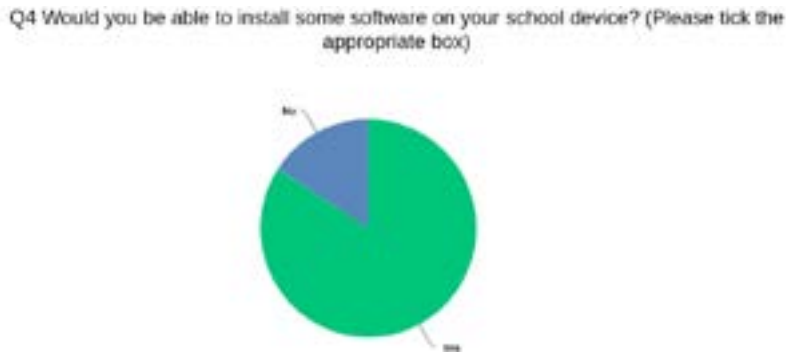

Figure 5. The teacher's response to the question if they can install software on school devices.

The next question asked respondents to answer what the editor and compiler do they mainly use for writing code, interestingly 70% responded they use IDE (Figure 6). Only a few responses were on the visual and online debugger. According to the reactions, we assume that offering using Greenfoot visual programming environment will be a new and effective way of teaching coding.

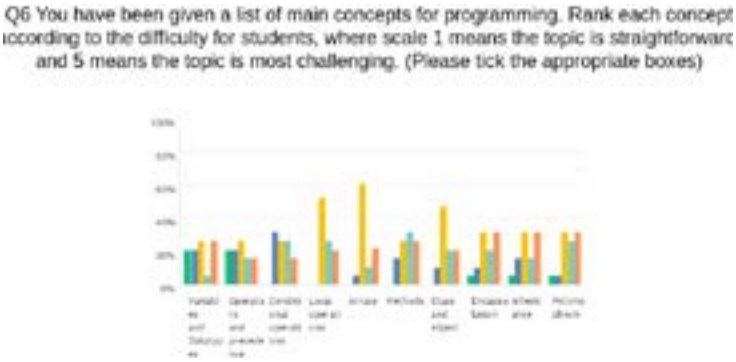

Figure 6. The response to the question of what debugger was used.

In the next question, teachers were given a list of leading concepts for programming. They were asked to rank each idea according to the difficulty for students, where scale 1 meant the topic is straightforward, and five said the question is most challenging. According to Figure7, we can see the items such as variables and data type, operators and precedence were the most accessible topics for teaching and learning for students, where themes like loops, arrays are considered the most difficult topics. Object-oriented concepts such as Class and Object, Encapsulation, Inheritance, Polymorphism are shown the most challenging for teaching and learning. Also, 30% of the responders showed Method one of the complicated topics in the programming courses, which is also indicated by reviewed literature.

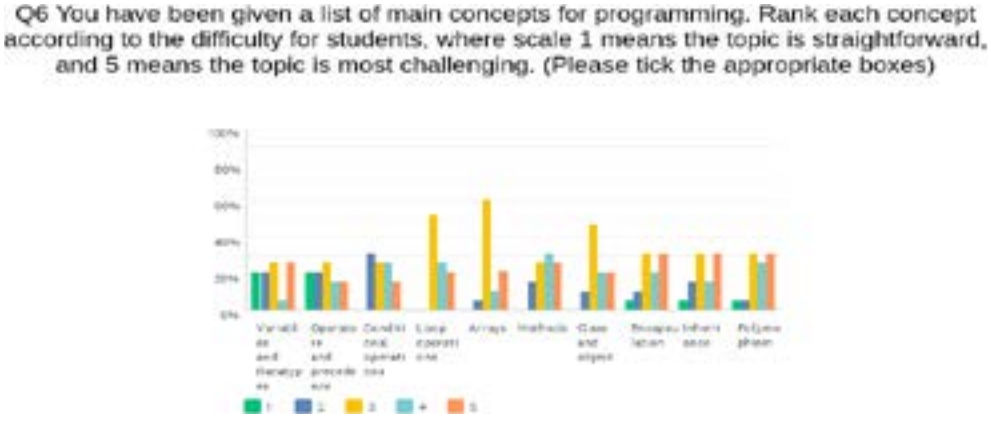

Figure 7. The difficulty of the topics of programming course

The next question was about identifying the reason why some of the themes are difficult for learning, and the possible causes are given to teachers to choose. According to their response (Figure 8), lacking problem-

solving skills can be the main reason followed by not having pre-existing programming language knowledge. Nearly 30% of teachers showed that not all students are enthusiastic about learning programming. Besides, one response was about pupils are not given enough time to domesticate programming concepts.

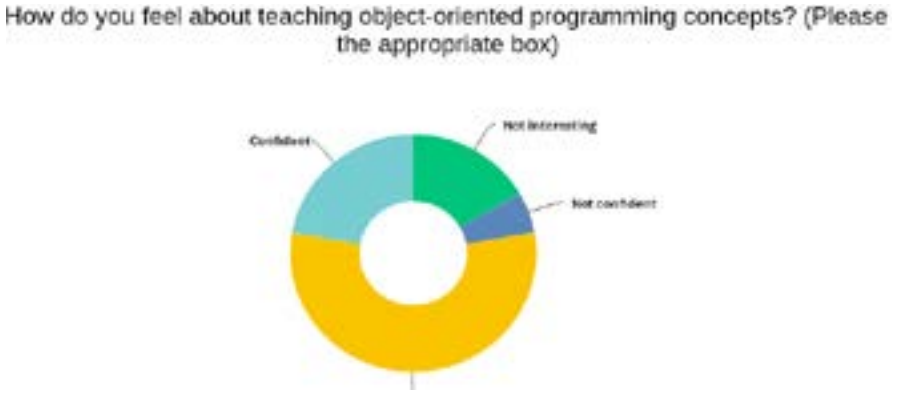

Figure 8. The possible reasons for having problem in learning programming

In the following question, teachers were asked to show how they fell about teaching object-oriented programming concepts. Figure 9 shows that most of the respondents said that OOP is exciting to school, while approximately 15% of the teachers indicated that it is not a fascinating topic to teach. Other responders showed that they are confident about teaching these concepts.

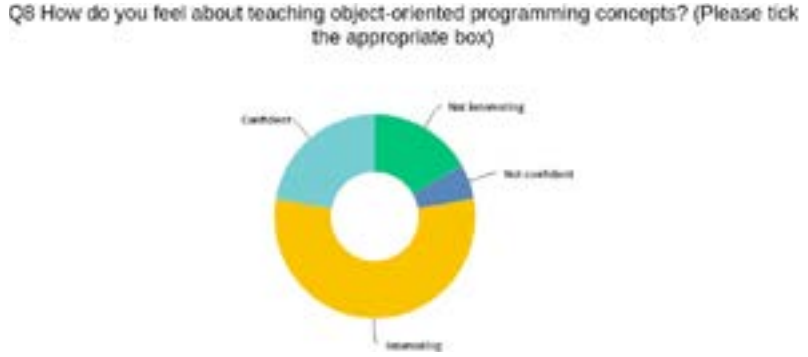

Figure 9. Teachers response about teaching OOP

In the last question, respondents were asked if they use Greenfoot to support learning OOP, and approximately 90% of them answered ‘no.’ We can assume that teaching OOP concepts using Greenfoot visual programming environment will be new for most of the teachers.

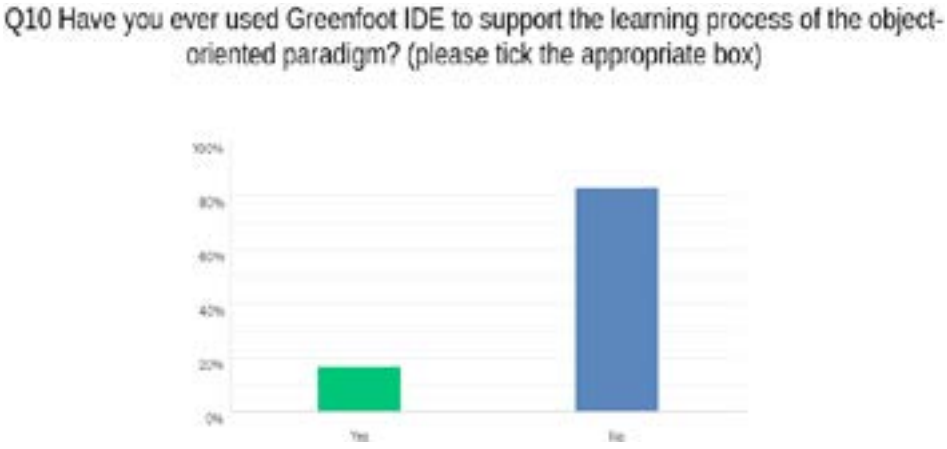

Figure 10. The response to the question if teachers used Greenfoot before

According to the reviewed literature, we can confidently assert the following.

First, the use of gamification technologies in teaching programming has a positive effect on students' interest in computer science and programming lessons. Second, learners from scratch showed slightly higher results, which gives rise to further research into the impact of the transition from procedural programming to object-oriented.

Conclusions. In this project presents an overview of teaching object-oriented programming concepts at the school level. The emphasis is on the existing pedagogical approaches of teaching Computer Science and OOP at school. Also, the emphasis is on the methodology and analysing teaching techniques from existing papers. After evaluating the questionnaire, we found that Greenfoot programming environment is not common in schools. We also figured out that teaching OOP concepts such as Class, Object, Inheritance are the most difficult topics to learn by students.

The results of the study allow us to formulate the following conclusions.

- The expediency of teaching the basics of OOP at school using visual programming tools in the basic course of computer science has been substantiated. Studying the basics of OOP and visual programming technology expands the scientific outlook of students, develops the thinking of students, makes it possible to learn more about programming.

- We can clearly say that using an educational tool like Greenfoot in the framework of the basic course of computer science will expend students programming experience.

REFERENCES:

© 2016 Autonomous Educational Organization 'Nazarbayev Intellectual Schools' (2019) Content Education, Available at: http://nis.edu.kz/en/ (Accessed: 20th June 2020).

Barnes D.J. & Kölling M. (2005). Objects First with Java, A Practical Introduction using Blue J. Prentice Hall, Second edition.

Barnes D.J., Kölling M. (2005): Objects First with Java. A Practical Introduction Using BLUEJ. Prentice Hall.

Ben-Ari M., Ragonis N. & Ben-Bassat Levy R. (2002). A Vision of Visualization in Teaching Object-Oriented Programming. In Proceeding of 2nd Program Visualization Workshop, Hornstrup Centret, Denmark, 83-89.

Bergin J., Stehlik M., Roberts J. & Pattis R. (2005). Karel J. Robot: A Gentle Introduction to the Art of Object-Oriented Programming in Java. USA: Dream Songs Press. D. Gough, S. Oliver, J. Thomas An introduction to systematic reviews.

Deitel H. and Deitel P. (2015). Java How To Program (Early Objects), Global Edition. 10th ed. NOIDA: Pearson Education Limited, pp.10-11.

Flanagan D. (2015). Java in a nutshell. 3rd ed. Sebastopol, CA: O'Reilly, pp.7-8.

Georgantaki S. and Retalis S., 2007. Using Educational Tools for Teaching Object Oriented Design and Programming. Journal of Information Technology Impact, 7(2), pp.111- 130.

Greenfoot. www.greenfoot.org.

Gruenbaum P., 2014. Undergraduates Teach Game Programming Using Scratch. Computer, 47(2), pp.82-84.

Henriksen P. and Kölling M., Greenfoot: Combining object visualisation with interaction. in Companion to the 19th annual ACM SIGPLAN OOPSLA conference, (Vancouver, Canada, 2004), ACM, 73-82.

Horstmann C., 2016. Core Java Vol. 1 Fundamentals, 10th ed. New York: Pearson, pp.130 - 136.

Introducing Programming using Scratch and Greenfoot H.S. Chandrashekar Geetha Kiran, A Uma Sunita P., 2018.

Kolling M. (2015), Introduction to Programming with Greenfoot, 2nd Edition, Pearson Education (US). Available from: VitalSource Bookshelf.

Kölling M. & Rosenberg J. (2001). Guidelines for Teaching Object Orientation with Java. ACM SIGCSE Bulletin 33 (3), 33-36.

M. Papastergiou, Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation, Computers & Education 52(1), 1–12, 2009.

Malan D. and Leitner H., 2007. Scratch for budding computer scientists. ACM SIGCSE Bulletin, [online] 39(1), p.223. Available at: <https://doi.org/10.1145/1227504.1227388> [Accessed 14 July 2020].

Maloney J., Resnick M., Rusk N., Silverman B. and Eastmond E., 2010. The Scratch Programming Language and Environment. ACM Transactions on Computing Education, [online] 10(4), pp.1-15. Available at: <https://dl.acm.org> [Accessed 13 July 2020].

Niemeyer P. (2017). Learning java. 2nd ed. Internal Press: O'Reilly Media, p.5.

SAGE Publications, London (2012).

Scratch. (2017). *https://scratch.mit.edu/.*

Scratch: programming for all / M. Resnick, J. Maloney, A. Monroy-Hern´andez [и др.] // Communications of the ACM, 2009. Т. 52, № 11. С. 60 – 67.

Sorva J. 2013. Notional machines and introductory programming education. ACM Trans. Comput. Educ. 13, 2.

Teif M. & Hazzan O. (2004). Junior High School Students' Perception of Object-Oriented Concepts. Paper presented at 18th European Conference on Object- Oriented Programming, 8th Workshop on Pedagogies and Tools for the Teaching and Learning of Object- Oriented Concepts, Oslo, Norway.

Xinogalos S., Satratzemi M. & Vassilios Dagdilelis V. (2006). An introduction to objectoriented programming with a didactic microworld: objectKarel. Computers & Education, 47(2), 148-171.

Yan L., 2009. Teaching Object-Oriented Programming with Games. Sixth International Conference on Information Technology: New Generations, 13, pp.969-974.

СОДЕРЖАНИЕ

ПЕДАГОГИКА

ЭКОНОМИКА

МАЗМҰНЫ

ПЕДАГОГИКА

ЭКОНОМИКА

CONTENTS

PEDAGOGY

ECONOMICS

Publication Ethics and Publication Malpractice in the journals of the National Academy of Sciences of the Republic of Kazakhstan

For information on Ethics in publishing and Ethical guidelines for journal publication see http://www.elsevier.com/publishingethics and http://www.elsevier.com/journal-authors/ethics.

Submission of an article to the National Academy of Sciences of the Republic of Kazakhstan implies that the work described has not been published previously (except in the form of an abstract or as part of a published lecture or academic thesis or as an electronic preprint, see http://www.elsevier.com/postingpolicy), that it is not under consideration for publication elsewhere, that its publication is approved by all authors and tacitly or explicitly by the responsible authorities where the work was carried out, and that, if accepted, it will not be published elsewhere in the same form, in English or in any other language, including electronically without the written consent of the copyright-holder. In particular, translations into English of papers already published in another language are not accepted.

No other forms of scientific misconduct are allowed, such as plagiarism, falsification, fraudulent data, incorrect interpretation of other works, incorrect citations, etc. The National Academy of Sciences of the Republic of Kazakhstan follows the Code of Conduct of the Committee on Publication Ethics (COPE), and follows the COPE Flowcharts for Resolving Cases of Suspected Misconduct (http://publicationethics.org/files/u2/New_Code.pdf). To verify originality, your article may be checked by the originality detection service Cross Check http://www.elsevier.com/editors/plagdetect.

The authors are obliged to participate in peer review process and be ready to provide corrections, clarifications, retractions and apologies when needed. All authors of a paper should have significantly contributed to the research.

The reviewers should provide objective judgments and should point out relevant published works which are not yet cited. Reviewed articles should be treated confidentially. The reviewers will be chosen in such a way that there is no conflict of interests with respect to the research, the authors and/or the research funders.

The editors have complete responsibility and authority to reject or accept a paper, and they will onh accept a paper when reasonably certain. They will preserve anonymity of reviewers and promote publication of corrections, clarifications, retractions and apologies when needed. The acceptance of a paper automatically implies the copyright transfer to the National Academy of sciences of the Republic of Kazakhstan.

The Editorial Board of the National Academy of sciences of the Republic of Kazakhstan will monitor and safeguard publishing ethics.

Правила оформления статьи для публикации в журнале смотреть на сайте:

www: nauka-nanrk.kz

ISSN 2518–1467 (Online),

ISSN 1991–3494 (Print)

http://www.bulletin–science.kz/index.php/en

Редакторы: *А. Ботанқызы, Р.Жәлиқызы, М.С. Ахметова, Д.С. Аленов*
Верстка на компьютере *Г.Д. Жадырановой*

Подписано в печать 30.04.2022.
Формат 60x881/8. Бумага офсетная. Печать - ризограф.
27,5 п.л. Тираж 300. Заказ 2.

Национальная академия наук РК
050010, Алматы, ул. Шевченко, 28, т. 272-13-19