Propósitos y Representaciones
Set. – Dic. 2020, Vol. 8, N° 3
http://dx.doi.org/10.20511/pyr2020.v8n3.484

ISSN 2307-7999
e-ISSN 2310-4635

Development of a Programming Course for Students of a Teacher Training Higher Education Institution Using the Programming Language Python

Desarrollo de un curso de programación para estudiantes de una institución de educación superior de formación docente utilizando el lenguaje de programación Python

Mikhail Semenovich Prokopyev*
Ammosov North-Eastern Federal University, Yakutsk, Rusia
ORCID: https://orcid.org/0000-0002-7565-8902

Elena Zotikovna Vlasova
Russian State Teacher Training University named after A.I. Herzen, St. Petersburg, Rusia
ORCID: https://orcid.org/0000-0001-7356-5019

Tatyana Vasilyevna Tretyakova
Training Institute of Ammosov NEFU, Yakutsk, Rusia
ORCID: http://orcid.org/0000-0002-4391-5556

Maksim Anatolyevich Sorochinsky
Ammosov North-Eastern Federal University, Yakutsk, Rusia
ORCID: http://orcid.org/0000-0002-4651-3384

Rimma Alekseyevna Solovyeva
Yakutsk State Agricultural Academy, Yakutsk, Rusia
ORCID: http://orcid.org/0000-0002-3616-7932

Recibido 12-11-19 **Revisado** 15-12-19 **Aprobado** 14-03-20 **En línea** 16-03-20

Correspondencia

Email: MikhailSProkopyev@yandex.ru

Citar como:

Prokopyev, M.S., Vlasova, E.Z., Tretyakova, T.V., Sorochinsky, M.A., Solovyeva, R.A. (2020). Development of a Programming Course for Students of a Teacher Training Higher Education Institution Using the Programming Language Python. *Propósitos y Representaciones, 8*(3). doi: http://dx.doi.org/10.20511/pyr2020.v8n3.484

Summary

At present, teaching programming to future teachers is superficial or is excluded from computer science curricular. Only programming fundamentals are taught during computer studies, and skills acquired during the training are not applied in practice. The aims of this article are 1) to develop a programming course for teacher training disciplines that are taught in training units of the Ammosov North-Eastern Federal University (NEFU), and 2) give recommendations to apply features of the programming language Python in the professional activities of future teachers. This work applies the iterative methodological approach to undertake the study in various groups of students of NEFU from 2016 to 2019. Quantitative and objective indicators are used. Results show which modules of Python are the most appropriate for students to study for their future professional teaching activities. The training course "Use of the language Python by future teachers" is developed, and will be taught in all training units of the university, that have teacher training disciplines. The article determines a direction of future research related to teaching programming, which is important for all computer studies.

Keywords: Learning Programming; Future Teacher; Programming Languages; E-Learning; Python; Education.

Resumen

En la actualidad, la programación de enseñanza para futuros maestros es superficial o está excluida del plan de estudios de informática. Solo se enseñan los fundamentos de programación durante los estudios de computación, y las habilidades adquiridas durante la capacitación no se aplican en la práctica. Los objetivos de este artículo son 1) desarrollar un curso de programación para disciplinas de capacitación docente que se imparten en unidades de capacitación de la Universidad Federal del Nordeste de Ammosov (NEFU), y 2) dar recomendaciones para aplicar características del lenguaje de programación Python en el Actividades profesionales de futuros docentes. Este trabajo aplica el enfoque metodológico iterativo para llevar a cabo el estudio en varios grupos de estudiantes de NEFU de 2016 a 2019. Se utilizan indicadores cuantitativos y objetivos. Los resultados muestran qué módulos de Python son los más apropiados para que los estudiantes estudien para sus futuras actividades de enseñanza profesional. El curso de capacitación "Uso del lenguaje Python por futuros maestros" se desarrolla y se impartirá en todas las unidades de capacitación de la universidad que tengan disciplinas de capacitación docente. El artículo determina una dirección de investigación futura relacionada con la programación de enseñanza, que es importante para todos los estudios de computación.

Palabras clave: Programación de aprendizaje; Futuro maestro; Lenguajes de programación; E-learning; Python; Educación.

Introduction

A large number of people have recently become interested in programming, and self-study services, such as Coursera, and Stepik. Programming courses are being introduced in educational institutions, new educational programs are being created and existing programs are being supplemented. Thus, software products (applications, websites, programs), that realize various ideas can be created using programming competencies. The more skills a future specialist has, the more expensive he/she becomes in the labor market. For example, more than half of the journalists in The New York Times have programming skills. It is necessary to learn programming languages, study codes written in them, and practice constantly, creating various programs to develop these skills. In general, having these skills disciplines an individual and helps him/her to think structurally and strategically. However, many people encounter a number of problems when studying programming, such as searching for the optimal amount of time for programming, understanding the progress achieved, the burnout due to the difficulty of choosing solutions to problems, and the lack of motivation for learning.

The contradiction between the need of modern society for qualified specialists who can program, and students' reluctance to learn programming, which is demonstrated by many bachelors and masters, makes it difficult to use modern opportunities in professional activities. The basis of any computer science education is the ability to analyze and develop program codes and everything that supports this activity. Given these contradictions, higher education institutions face the need to redefine their curricula, paying particular attention to the development of programming courses, the knowledge of which should be further applied in the professional activities of future specialists. Most higher education institutions choose the programming language Python (PL/ Python), which has an open source code. In 2018 it won the "Programming Language of the Year" award (TIOBE Index, 2019), which is given to programming languages that have the highest rating growth per year. Programs written in PL/Python appear considerably shorter than equivalent programs written in other popular programming languages, such as Java, C, C ++, Visual Basic, and .NET, due to intrinsic high-level data types and dynamic typing. John Guttag, a professor at the Massachusetts Institute of Technology, believes that much more educational institutions are using Python as their main programming language, because of its large set of useful libraries and modules, which have been created for many years (Guttag, 2013). Shriram Krishnamurthi, a professor at Brown University, agrees that PL/Python is easy to understand, and thus opens programming to a wide audience of students. However, although Krishnamurthi notes that Python is popular, he also believes that it lacks capability (Shein, 2015). Many authors note that students experience boredom and misunderstanding while being taught programming, and many of them perceive programming as a highly difficult subject; consequently, they rethink the way by which this subject will be useful for their future profession (Giannakos *et al.*, 2017).

Thus, this study aims to create a programming course for students in teacher training higher education institutions using the programming language Python. On the basis of the defined research purpose, the following tasks are formulated: to perform a theoretical analysis of the literature and experience with issues in teaching programming in teacher training higher education institutions, to choose PL/Python modules and libraries necessary for the course "Use of the language Python by future teachers", and to develop and test the course.

Literature review

The contradiction between the demand of modern society for qualified specialists in the digital economy and the reluctance of students to learn programming as a main ICT competency has led to the need to develop a programming course for students that would show possibilities of using programming in the professional activities of future teachers (Barakhsanova, Varlamova, 2015).

First, we analyze studies and sources related to problems in teaching programming. Most sources substantiate the relevance of training information technology (IT) personnel and the labor market advantage of specialists who possess fundamental programming knowledge and can solve applied problems using programming languages (Lanskykh, 2017). New educational object-oriented programs (OOP) in programming languages are being developed (Zablotsky, 2016). Students should obtain sufficient learning skills in algorithmization and programming, and modern curricula need correction both in content and methodology (Potapova, Aleynikova, 2017). In many higher education institutions, there is an open access to some methodological materials, which shows that the transition to information-oriented society requires further development of teaching disciplines related to programming (Rychkova, Smirnov, 2016). Some special aspects of teaching the theory of algorithms are considered, which describe new features of developed software products related to modeling and parallel programming (Pushkarev, Ogorodov, 2018). Such foundations of OOP, as encapsulation, inheritance, polymorphism and abstraction, are defined as a universal algebraic system (Ivanova, 2017). Teaching programming is facing serious challenges worldwide. Many students refuse programming courses during the first years of their studies because they are not able to understand basic concepts of programming. This problem can be solved using the problem-based learning (PBL) which, for example, is associated with the development of training games (Martins *et al.*, 2018). The training of future school teachers in the regional education system of Russia is considered by a number of authors (Barakhsanova *et al.*, 2018), who show the effectiveness of teaching methods for future school teachers in Yakutia under conditions of bilingualism; the study is confirmed by results of its implementation, which have shown a higher level of professional and communicative skills of students in mathematics, physics, computer science and chemistry.

Also, the following authors note problems of students, due to a large number of programming concepts. As students gradually become better programmers, both within the course and their profession, they are constantly provided with new and complex programming concepts, regardless of their experience (Giacaman, & De Ruvo, 2018).

Based on the works related to the definition of programming languages, which will be in demand in the future, we have concluded that there are many programming languages that provide one or another function applicable to data analysis tasks, where Python holds the leading position, R and Matlab are slightly behind (Shkarban, 2016). Classification types of programming languages are also provided. Advantages of the most popular programming languages are analysed. Data paradigms, typifications, types and structures are shown. Recommendations are given on how to choose a programming language depending on the purpose and tasks (Dedov *et al.*, 2018). Methods to teach disciplines related to programming are discussed, including a good study of didactic material, which involves studying a wide range of issues in laboratory classes - from fundamentals of a particular programming language to its specific features (Smolyanov, Pantileykin, 2018). Issues of creating a cross-platform work environment with the cloud storage of data, where features of the arrangement of the cloud storage of data are studied, have substantiated the choice of operating systems for servers. The application and purpose, advantages and disadvantages of the main web servers are considered in the work of these authors (Kogutenko, Plotnikova, 2018).

Teaching practice shows that the effectiveness of training depends on different methods, for example, traditional teaching of the programming language C ++ is not effective, it is better to use the game training method and ACM / ICPC. Some principles and characteristics for tasks in the programming language C ++ are provided. This methodology is highly appreciated by students, which improves teaching programming (Zheng, & Sarem, 2018). One of these methods is to use computer games as a new approach to teaching programming. Results of the study have shown significant improvement both in solving problems and in algorithmic thinking. Experimental groups have shown better results in average grades than control groups. The experimental group of students applied more correct computational concepts and practice in programming (Pellas, & Vosinakis, 2018).

A number of works are devoted to the discussion of the influence of thinking styles on students' attitude to programming through online learning, which shows academic achievements of the distance learning system (DLS). This study uses a single-factor, pre-test and post-test single group and a semi-empirical model. To implement problem-based educational activities, a course in the DLS Moodle has been developed. After the study, it has turned out that active students have a more positive opinion about a web-based learning than ordinary students (Yağcı, 2018).

The reform of the content of education programs, teaching methods, and the exam grading system is an urgent problem. Today Python is elegant and easy to learn. There is more than a thousand of third-party libraries for this language. At present, students need to learn more and master more skills in a limited time. It is important for teachers to think about improving their professional level. A new method to teach PL/Python is proposed, based on minimal sets of knowledge combined with characteristics of students; this method can help students to understand basic laws and programming paradigms, and will also provide practical skills, innovative thinking, and then effectively improve the quality of education understanding (Kui *et al.*, 2018). Requesting feedback is aimed at helping students improve their learning. Nowadays, a lot of programming tools, that automate feedback requests for students are offered. The review of the literature has clarified which kind of feedback focuses on identifying errors and, to a lesser extent, on eliminating problems, while moving on to the next step. However, the variety of feedback types has increased over the past decade, and new methods are being applied to create feedback that is becoming increasingly useful to students (Keuning *et al.*, 2018).

For example, in object-oriented programming tasks, students were given unit tests to evaluate their programs, and their editing and testing activities were recorded in the problem-solving environment. It was necessary to discover what programming strategies were being used by students for exercises with different levels of complexity, and to find out if there was any connection between these strategies and success in solving programming problems. Results of the study show a clear difference in strategies used by students at different levels of the complexity of tasks. Results also emphasize ways of providing information to solve tasks in a specified time period (Sharma *et al.*, 2018).

The following information is available about PL/Python and its libraries. For example, new algorithms for collecting and analyzing data may offer a different understanding of complex processes in machine learning, especially those tasks in which students are able to create unique databases and programs. These approaches are especially useful because there is a need for students of different ages (Blikstein *et al.*, 2014). The use of PL/Python in some biology issues related to data processing improves the performance of students who have used Python programs to solve tasks during laboratory classes in biology (Sarkar, & Sarkar, 2017). Thus, we have outlined our interests in teaching programming, what problems researchers face around the world, and what methods we have to deal with in the future.

Method

The study was conducted from 2017 to 2019 in training units of the Ammosov North-Eastern Federal University. The sample set of surveys among teachers in such disciplines as computer science and programming consisted of 15 respondents. The sample set of surveys among students in the teacher training field consisted of 113 respondents, out of three training units: the Road Transport Faculty, the Teacher Training Institute, and the Institute of Natural Sciences.

The first stage of the study was conducted in the academic year 2016 - 2017, and a programming language was selected; the survey was conducted among university teachers who teach such disciplines as computer science and programming. Then, modules and libraries were selected that would be necessary for future teachers; the survey was conducted among students. Based on the sample data, a course was developed. A programming language and libraries were chosen with the help of surveys among teachers and students using Google Forms, and their surveys were distributed through social media.

The second stage of the study was conducted in the academic year 2017 - 2018, and the course was tested in three groups as an elective. The groups were formed from the Teacher Training Institute of NEFU and the Institute of Natural Sciences of NEFU. The first group prepares future teachers of computer science (Gr. 1), and the second one prepares future teachers of biology (Gr. 2). The control group was also selected, in which simple computer science with programming elements was taught (Gr. 3); this group was formed from the Road Transport Faculty of NEFU.

The third stage of the study was conducted in the academic year 2018 – 2019; the course was refined taking into account the wishes of students and teachers. The course is presented as an elective in all training units of NEFU where future teachers are trained.

Results

The study was conducted from 2017 to 2019 on the basis of training units of the Ammosov North-East Federal University. The sample set of surveys among teachers in such disciplines as computer science and programming consisted of 15 respondents. The sample set of surveys among students of the teacher training field consisted of 113 respondents.

During the first stage, programming languages were analyzed, and the choice was made according to the following criteria: easy to understand, free-of-charge, freeware distribution, and popularity. The information on programming languages was analyzed according to three popular international indices TIOBE, GitHub, and RedMonk.

The TIOBE index determines the popularity of programming languages, and is updated once a month. Ratings are based on the number of requests from experts worldwide, through popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu. It should be noted that the TIOBE index does not determine the best programming language, but instead defines the language in which the highest number of code lines have been written (TIOBE Index, 2017). According to the rating, the most popular 10 languages in ascending order are: Java, C, C++, Python, C#, PHP, JavaScript, Perl, Visual Basic .NET and Ruby.

The resource, GitHub, is the largest web service for hosting IT projects and their joint development. This resource generates its own popularity index based on user requests: user visits, contacting repositories, arranging and solving problems in public projects, entering and editing codes, writing comments, and so on (GitHub Index, 2017). According to the rating, the most popular 10 languages in ascending order are: JavaScript (1, 604, 219), Java (763, 783), Python (744, 045), Ruby (740, 610), PHP (478, 153), C++ (330, 259), CSS (271, 782), C# (229, 985), C (202, 295), Go (188, 121).

The analytical company, RedMonk, creates an index according to two indicators: the number of code lines posted in the GitHub service, and the number of questions regarding the language on the question and answer site, Stack Overflow. These two resources demonstrate the real interest of programmers, and allow you to make strategic decisions as to which languages to use for your projects (RedMonk, 2017). According to the rating, the most popular 10 languages in ascending order are: JavaScript, Java, PHP, Python, C#, C++, Ruby, CSS, C, Objective-C.

Having considered these indexes, we see that the most popular programming languages are the same, the top of them being: JavaScript, Java, Python, PHP, and C ++. Then we will choose one according to our criteria: ease of use, beginner friendliness, flexibility, web developments, artificial intelligence and machine learning, working with data including Big

Data (Table 1. Choosing a programming language); according to these criteria we will give 3 scores, where 3 is the maximum score for the criterion. Teachers of computer science and programming of NEFU collected scores through Google Forms, then the scores were reduced to average valuesand rounded. The teachers chose PL/Python.

Table 1.
Choosing a programming language

Criteria	JavaScript	Java	Python	PHP	C++
Ease of use, beginner friendliness	3	2	3	2	1
Flexibility	2	2	3	1	3
Web developments	3	2	3	3	1
Artificial intelligence and machine learning	2	2	3	1	2
Working with data including in Big Data	1	1	3	1	1
Total:	11	9	15	8	8

After choosing a programming language, the question arose: Which libraries would be of interest to future teachers? To find an answer to this question, we developed a survey on the topic: “What would you like to learn to program?”. The survey was conducted among students of three units of the university (Figure 1. Results of the student survey). The questions are related to intentions to use the knowledge gained while learning a programming language.

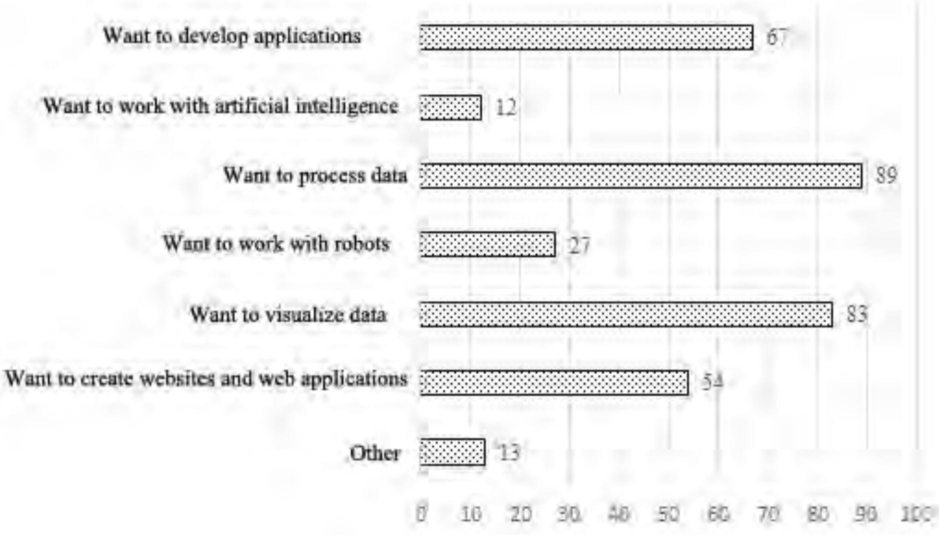

Figure 1. Results of the student survey.

According to the data obtained from the survey, we can conclude that the majority of respondents have a desire to learn how to process data (89%), and to visualize data (83%); many answers have been received about the development of applications (67%) and web sites (54%); among promising trends, one should note the work with artificial intelligence and machine learning (12%) and with robots (27%).

The results of the survey of students of NEFU set the task to develop a course to use programming in their future professional teaching activities. Since students were interested in working with data, processing it visually and creating applications, the following libraries and projects have been chosen, created for PL/Python. To work with data, the library, Pandas, has been selected to be used for data analysis (Pandas, 2017), and it is convenient to use Matplotlib to construct graphs that will give users a visual representation (Matplotlib, 2017). Since web applications currently prevail over simple applications, we have made the choice to create web applications, and the web framework Django has been chosen, which allows to quickly create secure and supported websites, and to focus on creating one's own applications (Django, 2017). After choosing the libraries, the elective course, "Use of the language Python by future teachers", was developed, consisting of three modules: "Fundamentals of the language Python", "Data processing and visualization", and "Creating a web application in Django".

During the second stage, three groups were formed: Gr.1, on the basis of the Teacher Training Institute of NEFU, prepares future teachers of computer science; Gr.2, on the basis of the Institute of Natural Sciences of NEFU, prepares future teachers of biology; Gr.3, on the basis of the Road Transport Faculty of NEFU, prepares future teachers in the field of transport.

The students were distributed in units. The Teacher Training Institute (Gr.1): group ICT-17 (Professional education (by industries). Computer science and computer engineering) consisting of 13 bachelors, group ICT-16 (Professional education (by industries). Computer science and computer technology) consisting of 17 bachelors. The Institute of Natural Sciences (Gr.2): group BO-17 (Pedagogical education. Biologist, teacher of biology) consisting of 24 bachelors, group BO-16 (Pedagogical education. Biologist, teacher of biology) consisting of 22 bachelors. The Road Transport Faculty (Gr. 3): group PO-17 (Professional education (by industries). Transport) consisting of 16 bachelors, group PO-16 (Professional education (by industries). Transport) consisting of 21 bachelors, these two groups of the Road Transport Faculty make up the control group.

Since the course was elective, it was held in extra time, lectures on modules were continuous, and the practical work was extended for the entire training period. The course was arranged through a distance learning system at NEFU (yagu.s-vfu.ru). Projects were developed according to modules. There were three projects according to the first module (M.1): "Credit calculator", "Notebook", and "Schedule of a training unit". It was necessary to process the selected database and visualize it according to the second module. In the third module, it was necessary to create a web application that would advertise the training unit.

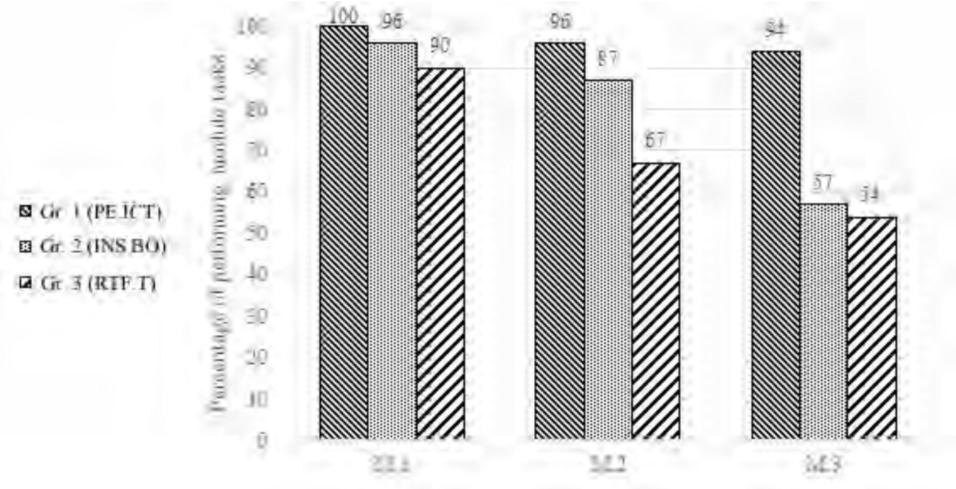

Figure 2. Results of performing tasks by modules

According to the data presented in Figure 2, we can conclude that all the groups coped completely with the first module (M.1), but only the first group (Gr.1) coped completely with the second (M.2) and third (M.3) modules; the second and third groups fell behind because they did not have special subjects related to programming; but we should note that most of the tasks were completed by all groups. Taking into account the fact that this course was elective, we can state that one half of students mastered the skills of using programming in their professional activities. After finishing the courses, their comments were collected, and the work was carried out according to the third stage of the study, which referred to refining the course and making it an elective discipline for all teacher training disciplines of the university.

Discussion

The study was conducted from 2017 to 2019 on the basis of training units of the Ammosov North-Eastern Federal University. A training course was developed; to do this, surveys were conducted to select a programming language, modules and libraries necessary for future teachers. The survey was conducted using Google Forms, and it was distributed through social media. The training course was tested in three groups as an elective, and refined, taking into account the wishes of students and teachers. The course is provided as an elective in all training units of NEFU where future teachers are trained.

A brief description of the most significant results, which have been defined in the Results section, and their comparison with other studies, devoted to similar topics, are provided; problem areas and the absence of some aspects are shown.

The choice of a programming language was analyzed according to three popular international indices: TIOBE, GitHub, and RedMonk. Having considered these indexes, we see that the most popular programming languages are the same, the top of them being: JavaScript, Java, Python, PHP, C ++. This was followed by the survey of university teachers, who chose PL/Python.

Having conducted the survey of students of NEFU, we found out that it was interesting for students to work with data, to process it visually, and to create applications. Based on these preferences, we have developed a course consisting of three modules “Fundamentals of the language Python, “Data Processing and Visualizing”, “Creating a Web Application in Django” and tested it in practice.

Conclusions

In the course of the study, we performed a theoretical analysis of the literature and current experience in teaching programming. Based on the analysis of the literature and works of scientists involved in problems of teaching programming, we can conclude that programming that future teachers are using becomes interesting to them.

Programming languages are analyzed, their positive aspects are defined, and, based on this analysis, Python is chosen as the most advanced programming language for our study. Libraries for the course are selected. An elective course has been developed and tested to identify problems associated with teaching programming.

Based on results of conducting the elective course, we have developed an elective discipline that is posted at yagu.s-vfu.ru of NEFU on the topic: "Use of the language Python by future teachers", which will solve the problem associated with the lack of programming skills of future teachers.

The discipline consists of 3 modules: “Fundamentals of the language Python”, “Data Processing and Visualization”, “Creating a web application in Django”. Upon completing the course, students acquire and develop ICT competencies and can use the obtained skills in their professional field.

We can also conclude that modern programming is becoming more accessible for ordinary users; there is a large number of ready-made libraries that help to develop applications, web clients, robots, neural networks, and so on. Python, the programming language that we have chosen for our study, proves this. The study has a practice-oriented nature, as there is a global challenge for the transition of the Russian Federation to the era of digital education.

References

Barakhsanova, E.A., Varlamova, V.A. (2015). The educational and methodological support for the implementation of the principle of regionalization of education in the process of teaching a block of information disciplines. *Modern problems of science and education*, *5*, 522.

Barakhsanova, E.A., Vlasova, E.Z., Varlamova, V.A., Nikitina, E.V., Prokopyev, M.S., Myreeva, A.N. (2018). Vocational training of school teachers in yakutia's universities through the principle of regionalization (case study of the methodology of teaching natural science to prospective teachers). *Espacios, 39*(20).

Blikstein, P., Worsley, M., Piech, C., Sahami, M., Cooper, S., Koller, D. (2014). Programming pluralism: Using learning analytics to detect patterns in the learning of computer programming. *Journal of the Learning Sciences, 23*(4), 561-599.

Dedov, S.V., Kirsanov, O.D., Timoshevskaya, O.Yu. (2018). The analysis of advantages of the most popular modern programming languages. *Actual problems of modern science. The collection of articles on materials of the XVII International research and practice conference*, 63-72.

Django (2017). Retrieved from: https://docs.djangoproject.com/en/3.0/.

Giacaman, N., De Ruvo, G. (2018). Bridging theory and practice in programming lectures with active classroom programmer. *IEEE Transactions on Education, 61*(3), 177-186.

Giannakos, M.N., Pappas, I.O., Jaccheri, L., Sampson, D. G. (2017). Understanding student retention in computer science education: The role of environment, gains, barriers and usefulness. *Education and Information Technologies, 22*(5), 2365-2382.

Guttag, J.V. (2013). *Introduction to Computation and Programming Using Python*. MIT Press.

Halikova, K.Z. (2016). The arrangement of research activities of students of a teacher training higher education institution in the process of teaching the discipline "programming". *Methods of teaching mathematical and natural science disciplines: modern problems and development trends. Materials of the III All-Russian Research and Practice Conference*, 179-184.

Ivanova, N.M. (2017). Teaching modern programming at school. *Computer Mathematics Systems and their Applications, 18*, 243-245.

Keuning, H., Jeuring, J., Heeren, B. (2018). A systematic literature review of automated feedback generation for programming exercises. *ACM Transactions on Computing Education, 19*(1).

Kogutenko, A.A., Plotnikova, S.V. (2018). The study of arranging cross-platform work environment with the cloud data storage. *Computer science and Cybernetics, 2*, 20-24.

Koulouri, T., Lauria, S., Macredie, R.D. (2014). Teaching introductory programming: A quantitative evaluation of different approaches. *ACM Transactions on Computing Education, 14*(4).

Kui, X., Liu, W., Guo, K., Xia, J., Du, H. (2018). Teaching method reform of python language programming course based on minimum knowledge sets. *Mechatronic Systems and Control, 46*(4), 181-186.

Lanskykh, S.F. (2017). The implementation of the certification training of future specialists of IT field in the framework of the taught discipline. *Teacher training education in Russia, 6*, 74-82.

Martins, V.F., de Almeida Souza Concilio, I., de Paiva Guimarães, M. (2018). Problem based learning associated to the development of games for programming teaching. *Computer Applications in Engineering Education, 26*(5), 1577-1589.

Matplotlib (2017). Retrieved from: https://matplotlib.org/contents.html.

Pandas. Python Data Analysis Library (2017). Retrieved from: https://pandas.pydata.org/pandas-docs/stable/.

Pellas, N., Vosinakis, S. (2018). The effect of simulation games on learning computer programming: A comparative study on high school students' learning performance by assessing computational problem-solving strategies. *Education and Information Technologies, 23*(6), 2423-2452.

Potapova, L.E., Aleynikova, T.G. (2017). Problems of teaching programming and training teachers of computer science. *Science for education, production, economy. Materials of the XXII (69) Regional Research and Practice Conference of teachers, researchers and postgraduate students: in 2 volumes*, 77-78.

Pushkarev, I.A., Ogorodov, A.V. (2018). The development of software and methodological support to conduct practical training on the constructive theory of Turing machines. *Advanced science, 1*, 21-27.

Rychkova, A.V., Smirnov, A.A. (2016). Teaching office programming using open education technology. *Open Education, 20*(1), 49-53.

Sarkar, S., Sarkar, S. (2017). A new approach to teach molecular biology in high school using python. *IEEE Region 10 Humanitarian Technology Conference 2016, R10-HTC 2016 – Proceedings.*

Sharma, K., Mangaroska, K., Trætteberg, H., Lee-Cultura, S., Giannakos, M. (2018). *Evidence for programming strategies in university coding exercises*.

Shein, E. (2015). Python for beginners. *Communications of the ACM, 58*(3), 19-21.

Shkarban, A.S. (2016). The choice of a programming language for the problem of data analysis using the hierarchy analysis method. *Nauka-rastudent.ru, 2*, 25.

Sidelov, D.I., Bashirova, Yu.N., Morkovina, E.F. (2015). Methodological features of teaching the discipline "Programming" to students during the first years of their studying at a university/ Modern society, education and science. *The collection of scientific papers on materials of the International research and practice conference: in 16 parts*, 150-151.

Smolyanov, A.G., Pantileykin, N.V. (2018). Methodological aspects of studying scarping using PHP in the course of "Network languages and web-programming". *Research and methodological electronic journal-concept, 4*, 51-58.

The RedMonk Programming Language Rankings: June (2016). Retrieved from: https://redmonk.com/sogrady/2016/07/20/language-rankings-6-16/.

The state of the Octoverse (2016). Retrieved from: https://octoverse.github.com/2016/.

TIOBE Index for November (2019). Retrieved from: https://www.tiobe.com/tiobe-index/.

TIOBE programming community index for June (2016). Retrieved from: http://chartsbin.com/view/41116.

Vlasova, E.Z., Avksentieva, E.Y., Goncharova, S.V., Aksyutin, P.A. (2019). Artificial intelligence - the space for the new possibilities to train teachers. *Espacios, 40*(9).

Vlasova, E.Z., Goncharova, S.V., Luknova, V.A. (2019). Using the EDX platform to arrange corporate training. *Modern Education: Traditions and Innovations*, *1*, 75-77.

Yağcı, M. (2018). Web-mediated problem-based learning and computer programming: Effects of thinking style on academic achievement and attitude. *Computer Applications in Engineering Education, 26*(6), 2012-2025.

Zablotsky, V.R. (2016). C ++ programming for cartographers and geodesists. The training program Bussol with multiple inheritance. *News of higher education institutions. Geodesics and aerial photography, 1*, 105-107.

Zheng, Y., Sarem, M. (2018). A novel C++ teaching method based on game mode and ACM/ICPC. *Proceedings - 9th International Conference on Information Technology in Medicine and Education, ITME 2018,* 348-352.