Implementation of education for sustainable development principles in the training of future software engineers

Serhii Koniukhov*, and *Kateryna* Osadcha

Bogdan Khmelnitsky Melitopol State Pedagogical University, Department of Informatics and Cybernetics, Melitopol, 72312, Ukraine

Abstract. The article examines the professional training of future software engineers in higher education institutions in the context of the implementation of the Sustainable Development Goals set by the UN General Assembly, as well as the Education for Sustainable Development (ESD) principles. A number of contradictions that arise in the course of this training are mentioned on the example of training of future object-oriented programming engineers. The UNESCO documents on education for sustainable development, scientific sources on the professional training of future software engineers, as well as the integration of sustainable development into relevant educational programs are analyzed. The methodology and results of experimental work carried out with the aim of overcoming existing contradictions, promoting sustainable development of information technology education and implementation of ESD principles in the process of professional training of future software engineers are presented. In particular, the organizational and methodological conditions, that were implemented into the educational process, and experimental data are presented. The effectiveness of the experimental work was proved by statistical verification of the reliability of the obtained data.

1 Introduction

The achievement of the Sustainable Development Goals set in 2015 by the UN General Assembly [1] is linked to the training of highly qualified professionals from all sectors of the economy who are capable of reflection, professional mobility and lifelong learning, aware of responsibility for the results of their activities. Such training is based on Education for Sustainable Development (ESD) principles.

According with UNESCO Roadmap "ESD empowers learners to take informed decisions and responsible actions for environmental integrity, economic viability and a just society, for present and future generations, while respecting cultural diversity. It is about lifelong learning, and is an integral part of quality education. ESD is holistic and transformational education which addresses learning content and outcomes, pedagogy and the learning environment. It achieves its purpose by transforming society." [2, p. 12].

Intensive development of the information technology and the need to overcome the numerous challenges facing humanity, lead to an increase in requirements for professionals whose activities are creation, implementation and maintenance of software. The professional training of such specialists in the bachelor's and master's degrees of higher education in Ukraine is carried out in the specialties of the field of knowledge "Information Technology", in particular "Software Engineering" and "Computer Science". In order to

ensure that the level of professionalism of graduates of these specialties corresponds to the requirements of society, effective procedures of updating the content, forms, methods and means of training should be implemented in higher education institutions on the basis of systematic monitoring of the state of the industry and a balanced combination of fundamental, applied and humanitarian components of higher education.

This is the opinion of Ukrainian researchers V. Kruhlyk and V. Osadchyi [3], A. Striuk and S. Semerikov [4], T. Vakaliuk, V. Kontsedailo, D. Antoniuk, O. Korotun, I. Mintii and A. Pikilnyak [5], O. Markova, S. Semerikov, A. Striuk, H. Shalatska, P. Nechypurenko and V. Tron [6], H. Chemerys, K. Osadcha, V. Osadchyi and V. Kruhlyk [7] and others.

An important area of professional work of software engineers is the software development using an object-oriented approach, so graduates of higher education institutions should understand its fundamental principles, be able to use object-oriented programming languages, to apply existing and make their own decisions, to decompose and compose tasks, to document the process of building an object model, etc. The lack of the necessary capabilities of the developers is one of the reasons for the low quality of software products, which in some cases pose a threat to the sustainable future. Other threats include focus on current tasks and immediate goals, lack of understanding of global economic, environmental and social issues, opportunities to overcome or minimize them at the level of individual IT-professionals or businesses, the impact of IT-products

---

* Corresponding author: konukhov@mdpu.org.ua

on the present and future, and responsibility to future generations.

Therefore, there is a need to create in the higher education institutions the conditions for students to develop appropriate professional competences.

Domestic and foreign scientists have thoroughly developed conceptual foundations and examined some aspects of higher education and professional training of future software engineers. However, insufficient attention is paid to developing their professional competence in the study of object-oriented programming (OOP), as well as to acquire the knowledge and skills necessary to pursue activities with a view to sustainable development goals. Therefore, there are contradictions between: the need to combine the fundamental theoretical and thorough practical training of future software engineers as highly qualified specialists and the limited time of studying the disciplines of the vocational training cycle; the necessity of using abstraction, decomposition and composition in the process of studying OOP and insufficient level of abstract-logical thinking in students; high level of complexity of OOP educational content and insufficient readiness of higher education students to systematic cognitive self-activity; possibility to demonstrate object-oriented approach on the example of large projects and traditional use of educational tasks with limited content; the society requirements for the professional training of software engineers and the students' lack of awareness of these requirements; lack of knowledge on sustainability issues and the need to promote a sustainable future in professional life and daily life.

Purpose of the article: to present the results of experimental work carried out with the purpose of overcoming the mentioned contradictions, promoting sustainable development of IT education and implementation of ESD principles in the process of professional training of future software engineers.

2 Literature review

As I. Mulà et al. emphasise "The world is shaped by an education system that reinforces unsustainable thinking and practice" [8, p. 798]. So, it is very important to transform an education system taking into account aims of sustainability and sustainable future. Education for sustainable development principles are substantiated in numerous UNESCO documents [2; 9; 10; 11] and discussed in scientific papers [8; 12].

Internationally recognized ESD principles are named by D. Tilbury and I. Mulà. They are the next: futures thinking; critical and creative thinking; participation and participatory learning; partnerships; systemic thinking [11, p. 5]. Moreover, there are pointed out key ESD learning themes (e.g., gender equality; biological diversity; ecological principles, ecosystems; natural resources management; health and well-being; consumerism and ethical trade; rural and urban development; corporate social responsibility) that are critically significant to the sustainable development agenda [11, p. 6].

Implementation of ESD principles in the process of professional training of future software engineers in higher education institutions is in different ways, in particular: deep informatization of the educational process [5; 6; 13; 14]; enhancing opportunities for professional mobility and lifelong learning for IT professionals through the development and implementation of intelligent means of recognition of qualifications and competences obtained in different countries and educational institutions [15]; improving educational programs for the short cycle of vocational training [16]; improving the content, forms, methods and tools of object-oriented programming [17]; introduction of sustainability issues to the content of educational programs [18; 19; 20; 21] and others.

A detailed analysis of scientific literature on the problems of OOP training future software engineers was carried out in [17].

Researchers in the training of computer science and software engineers note that sustainable development issues are hardly addressed in relevant university education programs. However, software engineers should take their share of responsibility for sustainability because of growing of IT's productivity in combination with cutting down of life cycles and growing resource problems of our planet [18, p. 454]. In this regard, it is necessary to determine the mechanisms for introducing sustainable development issues into university computer science and software engineering curricula, as well as motivating students and teachers to address them. In foreign pedagogical practice these issues have been investigated sufficiently well.

B. Penzenstadler and A. Fleischmann offer a strategy for integrating the concept of sustainability into a degree course scheme across three stages: 1) to propose a seminar and form a core of interested people; 2) to give a lecture series for broaden the awareness for sustainability; 3) to establish sustainability as topic by integrating it into the syllabus of appropriate software engineering lectures with teach-the-teacher seminars. During a seminar student should examine chosen issue and to present their topic in class. They are offered such seminar topics as: "Climate killer internet? Energy-efficient nets and systems have a notable impact", "Climate change research and software engineering for climate research", "Marketing for sustainability – how can I make it matter for software engineers?" etc. [18, pp. 455-456].

D. H. Fisher, Z. Bian and S. Chen emphasize the nexus between sustainability and computer science and the necessity to integrate sustainability science and engineering into computing education [19, p. 95]. They survey two levels of integration of sustainability into computer science higher education (the course level and the course component level) and give different examples of such combination. Scientists identify the course-level integration as introducing computer science courses that focus on topics at the intersection of computing and sustainability. These are such courses as "Computing, Energy, and the Environment", "Seminar on Computational Sustainability: Algorithms for Ecology and Conservation" etc. Component-level integration is

implemented by introducing lectures, exercises, and projects, with sustainability themes into computer science courses that do not have a sustainability focus, such as courses in computer organization, databases, and artificial intelligence [19, p. 93-94].

Three sustainability integration strategies are offered by Y. Cai: 1) developing a new course named "green computing" covering selected sustainability and green computing topics; 2) designing and developing independent green computing learning modules and projects that can be easily plugged into the existing computer courses; 3) an integrative and transformative approach to completely re-design some computing courses with sustainability as one of the top priorities [20, pp. 525-526].

L. M. Hilty and P. Huber [21] consider that sustainable development is an important part of the curriculum of ICT-related study programs and content is strongly significant to interest students in it. They present results of empirical investigation aimed identifying topics with the greatest potential to motivate students on sustainability. Researchers reveal five clusters of such topics, namely: "ICT impacts on sustainability"; "Material resources for ICT hardware: Informal recycling"; "ICT as an enabler: Saving material and energy: Videoconferencing example"; "Resource consumption: Global distribution"; "Rebound effect: General concept" [21, p. 652].

An example of IT-students participation in project aimed solution one of sustainability tasks – "provide safe, non violent, inclusive and effective learning environments for all" [1] – is given in [22].

3 Research methodology

The ideas of education for sustainable development were realized by us in the process of experimental and experimental work on forming the professional competence of future software engineers in the process of studying object-oriented programming, necessary for further successful activity in the specialty.

The experiment was conducted during 2015-2018 on the basis of higher education institutions of Ukraine, in particular, the Bogdan Khmelnytsky Melitopol State Pedagogical University. 135 second-, third- and fourth-year students of full-time education studying 6.040302 "Informatics" field and 122 "Computer sciences and information technologies" speciality have taken part in the qualifying and forming stages of the pedagogical experiment. The number of the control group (CG) was 69 people, experimental (EG) - 66 people.

We followed this sequence of pedagogical research [23, p. 10]:

1) comparison of the initial state of formation of components of professional competence in students of the control and experimental groups according to certain criteria and indicators, establishing the absence of statistically significant differences;

2) introduction of the developed organizational and methodological conditions for the formation of professional competence of future software engineers in

the process of studying object-oriented programming in the experimental group;

3) comparison of the final state of formation of components of professional competence in students of the control and experimental groups according to certain criteria and indicators, establishing the presence of statistically significant differences.

In the course of this work, the levels of professional competence components of future software engineers identified by us in [17] were diagnosed:

- motivational: a set of motives that encourage higher education students to actively study OOP; their interest in the in-depth study and use of OOP in their further professional activities; readiness for self-development in object-oriented development;

- cognitive: development of abstract-logical thinking; possession of techniques of formalization, abstraction, decomposition and composition; understanding the fundamental basics of OOP and their implementation in different programming languages; set of theoretical knowledge of fundamental concepts and applied aspects of OOP;

- operational: formation of skills in object-oriented programming necessary for effective professional activity;

- reflexive: the ability to self-understand, analyze and evaluate yourself as a specialist and your actions in the current situation, in the past and in the future, as well as yourself as a member of the software development team.

Since the ECTS scale is used in the institutions of higher education of Ukraine to evaluate students' academic achievement, five levels of formation of these components of the professional competence of future software engineers have been identified:

- professional: certifies the formation of a component of professional competence at the level of an experienced software engineer and the ability of the student to enter professional activity as a mid-level specialist without additional training, corresponds to the level "A" of the ECTS scale;

- high: certifies the formation of a component of professional competence at the level of junior software engineer and the ability of the student to start professional activity and independent tasks without additional training, corresponds to the level "B" of the ECTS scale;

- sufficient: certifies the formation of a component of professional competence at the level of the junior software engineer and the ability of the student to start professional activity and independently perform tasks with additional training at the enterprise, corresponds to the level "C" of the ECTS scale;

- low: certifies the formation of a component of professional competence at the level of the novice programmer and the ability of the student to begin professional activity as a junior software engineer only under the direct supervision and with additional training at the enterprise, corresponds to the level "D" of the ECTS scale;

- critical: certifies the extremely low level of professional competence component and the student's lack of ability to take up professional work as a software

engineer, corresponds to the "E" level of the ECTS scale.

In the experimental group, the educational process was organized on the basis of the following organizational and methodological conditions [17]:

1) Formation of positive motivation for students to study and apply in future professional activities of OOP. The implementation of this condition included demonstration of examples of software development practice, meetings with leading specialists of IT enterprises, organization of group implementation of software projects, involvement of students in the discussion of practical aspects of the software engineer.

2) Formation of a cross-cutting content-activity line of studying OOP within the disciplines of the vocational training cycle. Within each successive course, fundamental concepts of object-oriented programming were repeated, and they were considered at a new level of complexity, taking into account the specifics of a particular area of software development.

3) Application of appropriate forms and methods of formation in higher professional qualifications. The implementation of this condition involved the implementation of various types of software projects, the use of training tasks in object-oriented programming, interactive teaching methods and game technologies.

4) Use of modern information and communication technologies in the process of education of students of OOP, namely: software for educational purposes, software development environments, visualization tools, training management systems, distance courses in academic disciplines and additional specialized online resources.

Some ESD ideas were implemented during the development of the students' program projects. In particular, they were offered project topics such as: developing programs for the statistical processing of observation data (demographic economic, meteorological, medical, biological, etc.); development of educational programs for students of general secondary education institutions (simulators, didactic games, etc.); development of programs for automation of separate production processes for enterprises of different industries, etc. In order to create quality software, students had to pre-analyze the problem given the problems of a sustainable future and the goals of sustainable development.

The measures envisaged by the pilot program were implemented within the training disciplines of the cycle of professional training: "Programming", "Object-oriented programming", "Cross-platform programming", "Web application programming and support".

To evaluate the likelihood of the experimental data obtained, a method of testing statistical hypotheses was used.

The hypothesis of the absence of significant differences in the average values of indicators of the formation of components of professional competence in students of control and experimental groups was tested using the Student's t-test (1) [24, p. 185].

$$t_{\text{emp}} = \frac{|\bar{X}_1 - \bar{X}_2|}{\sqrt{\frac{(n_1-1)\times\sigma_1^2 + (n_2-1)\times\sigma_2^2}{n_1+n_2-2}\times\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}}, \qquad (1)$$

where $\bar{X}_1$ and $\bar{X}_2$, σ_1^2 and σ_2^2, n_1 and n_2 – averages, variances, and volumes of the first and second samples, respectively.

Testing the hypothesis about the conformity of the samples to the normal law of distribution of random variables was performed using formulas (2) and (3) [25, p. 231-232].

$$A = \frac{\sum(x_i-\bar{x})^3}{n\times\sigma^3}, \quad m_A = \sqrt{\frac{6}{n}}, \qquad (2)$$

$$E = \frac{\sum(x_i-\bar{x})^4}{n\times\sigma^4} - 3, \quad m_E = 2\times\sqrt{\frac{6}{n}}, \qquad (3)$$

where A – asymmetry indicator, E – excess indicator, m_A and m_E – representativeness errors for each of these indicators, respectively.

Volumes of control and experimental groups $n_{CG} = 69$ and $n_{EG} = 66$ respectively. Number of degrees of freedom $k = n_{CG} + n_{EG} - 2 = 69 + 66 - 2 = 133$. Critical significance of the Student test for 133 degrees of freedom and significance level $\alpha = 0.05$: $t_{\text{cr}} \approx 1.98$.

4 Analysis of experimental results

4.1 Qualifying stage of the research

The empirical data obtained at the ascertaining stage of the pedagogical experiment gave reason to draw the following main conclusions:

1) future software engineers in the control and experimental groups found insufficient level of professional competence development: approximately one third to two thirds of students demonstrated low or critical level of professional competence components;

2) the initial level of professional competence in the two groups practically did not differ (the difference between the percentage of students at each level of education in terms of individual competence components did not exceed 3%), which testified to their homogeneity.

According to the results of estimation of the level of the motivational component of professional competence of future software engineers, 34.78% of students (24 persons) of CG and 34.85% of students (23 persons) of EG revealed low or critical level. At the same time, 46.38% of students (32 persons) of CG and 45.45% of students (30 persons) of EG showed sufficient level. So, after completing their first year, they understood the need for knowledge of object-oriented programming technologies and languages and were partially motivated to further study OOP.

According to the results of assessment of the level of cognitive component of professional competence of future software engineers, 68.12% of students (47 persons) of CG and 68.18% of students (45 persons) of EG have found low or critical level. This situation is

to a certain extent due to the fact that the first year focuses on the study of algorithmization, structural and procedural programming, and the mechanisms of OOP are mostly considered indirectly to the extent necessary for writing programs in development environments.

According to the results of estimation of the level of the operational component of professional competence of future software engineers, 75.36% of students (52 persons) of CG and 71.21% of students (47 persons) of EG revealed low or critical level. The reason for this is that at the end of the first year students have initial experience writing programs with classes and objects, but they do not yet use the full-scale OOP mechanisms.

As a result of assessing the level of reflexive component of the professional competence of future software engineers, 47.83% of students (33 persons) of CG and 48.48% of students (32 persons) of EG revealed low or critical levels. At the same time, 37.68% of students (26 persons) of CG and 34.85% of students (23 persons) of EG showed sufficient level, which testified to partial formation of ability to evaluate themselves as a student, member of academic group and software development team, as well as the results of its activities.

Valuation of the obtained data. The hypothesis about the absence of statistically significant differences in the average values of indicators of the formation of the components of students' professional competence in the control and experimental groups was performed using the Student's t-test, which was calculated by formula (1). Preliminary verification of sampling of the normal law of random distribution according to formulas (2) and (3) was carried out.

The null hypothesis H_0: $x_{CG} = x_{EG}$, there is no statistically significant difference between the samples, the average values of the indicators of the professional competence components of the students of the control and experimental groups are equal.

Alternative hypothesis H_1: $x_{CG} \neq x_{EG}$, there is a statistically significant difference between the samples. the average values of indicators of the formation of components of professional competence of students in the control and experimental groups differ significantly.

The results of testing these statistical hypotheses are given in Table 1.

Table 1. Results of valuation of the obtained data (qualifying stage of the experiment).

Components of professional competence	t_{emp}	Conclusion
motivational	0.47	For all components
cognitive	0.33	$t_{emp} < t_{cr}$.
operational	0.12	Hypothesis H_0 is
reflexive	0.01	accepted.

Thus, it is proved that there is no statistically significant difference in the levels of formation of the components of professional competence in students of the control and experimental groups at the qualifying stage of the pedagogical experiment, that is, the samples are homogeneous.

Based on the analysis of empirical data of the qualifying stage of the pedagogical experiment, it was

concluded that the level of individual components, and therefore the professional competence of future software engineers was generally low and critical. In order to increase this level, the educational process and the study of object-oriented programming had to be organized in such a way as to ensure the acquisition and completeness of students' acquisition of basic and special knowledge of OOP (cognitive component), the effective formation in them of OOP skills and ability to use them in the process of development of various types of software projects (operational component), persistent high motivation to study OOP, its further use in professional activity, and self-improvement in this field in terms of programmatic development (motivational component), formation of ability to evaluate and responsible attitude to the results of their work and role in the team (reflexive component).

4.2 Forming stage of the research

The empirical data obtained during the formative stage of the pedagogical experiment gave reason to make the following generalizations:

1) the overall level of future software engineers' competence has increased: in both groups the increase in the number of students at the professional and high levels, as well as the increase in the average values of the indicators of the components of professional competence, but the changes in the experimental group were more significant;

2) the final levels of the components of students' professional competence in the experimental group exceeded the corresponding indicators in the control group.

On the basis of these data, a preliminary conclusion was made about the effectiveness of implementing the organizational and methodological conditions for forming the professional competence of future software engineers in the process of studying object-oriented programming in higher education institutions.

The results of the assessment of the level of the motivational component of professional competence of future software engineers revealed that in the experimental group the percentage of students with professional and high level was equal to 21.21% (14 people) and 42.42% (28 people) respectively. In the control group there were minor changes and this indicator was 10.14% (7 persons) and 18.84% (13 persons), respectively. A significant difference was observed in the indicators characterizing the number of students with low and critical levels: 6.06% (4 persons) and 3.03% (2 persons) in EG; 15.94% (11 persons) and 5.80% (4 persons) in CG respectively. The obtained data are presented in Fig. 1.

According to the results of assessment of the level of cognitive component of professional competence of future software engineers, it was found that in the experimental group the percentage of students with professional and high level became equal to 21.21% (14 persons) and 36.36% (24 persons), respectively. In the control group there were slight changes and this

indicator was 4.35% (3 persons) and 11.59% (8 persons) respectively. A significant difference was observed in the indicators characterizing the number of students with low and critical levels: 9.09% (6 persons) and 3.03% (2 persons) in EG; 28.99% (20 people) and 8.70% (6 people) in CG, respectively. The obtained data are presented in Fig. 2.
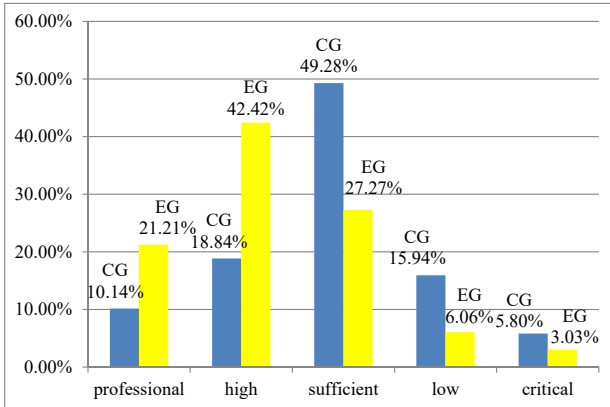

Fig. 1. Formation of the motivational component of the professional competence of future software engineers (forming stage of the experiment).
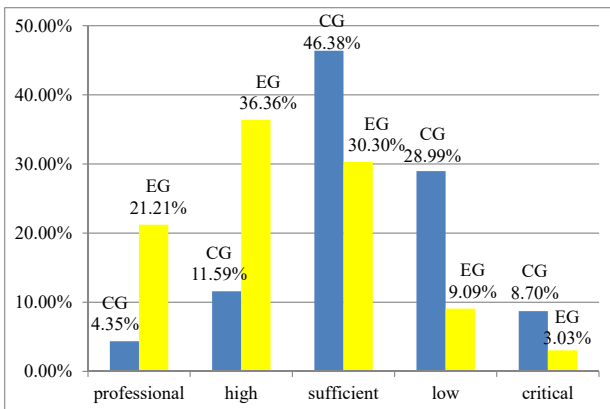

Fig. 2. Formation of the cognitive component of the professional competence of future software engineers (forming stage of the experiment).

According to the results of evaluation of the level of the operational component of professional competence of future engineers-programmers, it was found that in the experimental group the percentage of students with professional and high level was equal to 19.70% (13 persons) and 40.91% (27 persons), respectively. In the control group, there were slight changes and this figure was 5.80% (4 persons) and 15.94% (11 persons), respectively. A significant difference was observed in the indicators characterizing the number of students with low and critical levels: 9.09% (6 persons) and 3.03% (2 persons) in EG; 31.88% (22 persons) and 13.04% (9 persons) in CG, respectively. The obtained data are presented in Fig. 3.

According to the results of the evaluation of the level of the reflexive component of professional competence of future software engineers, it was found that in the experimental group the percentage of students with professional and high level was equal to 22.73% (15

people) and 30.30% (20 people), respectively. In the control group there were slight changes and this indicator was 10.14% (7 people) and 15.94% (11 people), respectively. A significant difference was observed in the indicators characterizing the number of students with low and critical levels: 6.06% (4 persons) and 3.03% (2 persons) in EG; 23.19% (16 people) and 8.70% (6 people) in CG, respectively. The obtained data are presented in Fig. 4.
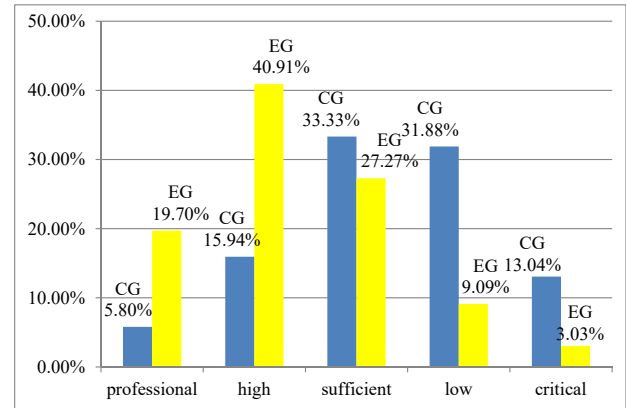

Fig. 3. Formation of the operational component of the professional competence of future software engineers (forming stage of the experiment).
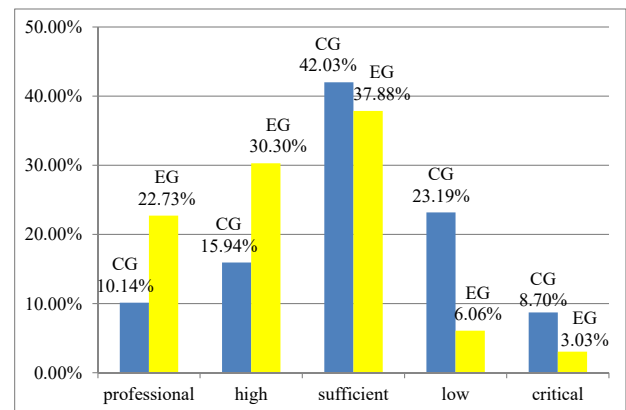

Fig. 4. Formation of the reflexive component of the professional competence of future software engineers (forming stage of the experiment).

Valuation of the obtained data. Testing of the hypothesis about the presence of statistically significant differences in the average values of the formation of components of students' professional competence in control and experimental groups, and therefore different levels of formation of professional competence was generally performed using the Student's t-criterion, which was calculated by formula (1). Preliminary verification of sampling of the normal law of random distribution according to formulas (2) and (3) was carried out.

The null hypothesis $H_0: x_{CG} = x_{EG}$, there is no statistically significant difference between the samples, the average values of the indicators of the professional competence components of the students of the control and experimental groups are equal.

Alternative hypothesis H_1: $x_{CG} \neq x_{EG}$, there is a statistically significant difference between the samples. the average values of indicators of the formation of components of professional competence of students in the control and experimental groups differ significantly.

The results of testing these statistical hypotheses are given in Table 2.

Table 2. Results of valuation of the obtained data (forming stage of the experiment).

Components of professional competence	t_{emp}	Conclusion
motivational	2.85	For all components $t_{emp} > t_{cr}$. Hypothesis H_0 is rejected, hypothesis H_1 is accepted.
cognitive	6.09	
operational	5.32	
reflexive	2.87	

Thus, the existence of statistically significant differences in the levels of formation of components of professional competence in students of control and experimental groups at the formative stage of pedagogical experiment was proved. This leads to the conclusion that the average difference is not accidental, but is the result of the implementation of the proposed organizational and methodological conditions for the formation of professional competence of future software engineers in the process of studying object-oriented programming.

5 Conclusion

Transformation of society to meet the goals of sustainable development is impossible without the active participation of each individual citizen. Training for such activities should be undertaken in education institutions of all levels based on Education for Sustainable Development principles.

The realization of the goals of education for sustainable development in the process of professional training of future software engineers in higher education institutions implies deep informatization of the educational process, enhancement of opportunities for professional mobility and lifelong learning, improvement of educational programs, introduction of issues of sustainability and content.

The contradictions that adversely affect the formation of the professional competence of future software engineers in the process of object-oriented programming study have been identified. In order to overcome the problems, organizational and methodological conditions for the formation of the professional competence of future software engineers in the process of object-oriented programming study at higher education institutions have been implemented.

In order to verify the effectiveness of the organizational and methodological conditions during 2015-2018, the research and experimental work was carried out. 135 second-, third- and fourth-year students of full-time education studying 6.040302 "Informatics" field and 122 "Computer sciences and information technologies" speciality have taken part in the qualifying

and forming stages of the pedagogical experiment. There has been recorded an increase in the level of formation of the professional competence among students of the experimental group at the end of the formative stage of the pedagogical experiment, namely: the number of students with professional and high levels of the formation of the motivational component has increased by 43.94%, the cognitive component – by 50.00%, the operational and activity component – by 51.51%, the reflexive component – by 36.36%; the increase in average values of the indicators of the formation of the motivational component has been 34.95%, the cognitive component – 23.42%, the operational and activity component – by 23.40%, the reflexive component – 27.03%. The reliability of the obtained results has been verified using Student's t-criterion: at the end of the forming stage of the experiment its empirical value (t_{emp}) for all the components of the professional competence exceeded the critical value ($t_{cr} \approx 1.98$). Thus, it was proved that the implementation of the developed organizational and methodological conditions contributed to improving the effectiveness of the formation of the professional competence of future software engineers in the process of the object-oriented programming studying.

In the process of experimental work, attention was paid to acquaint students with the goals and objectives of sustainable development. We believe that it is advisable to direct further research on the implementation of strategies for the introduction of training programs for engineering engineers of disciplines and individual modules aimed at familiarizing students with the problems of a sustainable future, forming in them an understanding of their role in solving these problems.

References

1. General Assembly, *Transforming our world: the 2030 Agenda for Sustainable Development* (United Nations, 2015), https://undocs.org/A/70/L.1. Accessed 25 Jan 2020

2. UNESCO, *UNESCO Roadmap for Implementing the Global Action Programme on Education for Sustainable Development* (UNESCO, Paris, 2014), https://sustainabledevelopment.un.org/content/documents/1674unescoroadmap.pdf. Accessed 30 Jan 2020

3. V.S. Kruglyk, V.V. Osadchyi, Developing competency in programming among future software engineers. Integration of Education **23**(4), 587–606 (2019). doi:10.15507/1991-9468.097.023.201904.587-606

4. A.M. Striuk, S.O. Semerikov, The Dawn of Software Engineering Education. CEUR Workshop Proceedings **2546**, 35–57 (2020), http://ceur-ws.org/Vol-2546/paper02.pdf. Accessed 15 Feb 2020

5. T.A. Vakaliuk, V.V. Kontsedailo, D.S. Antoniuk, O.V. Korotun, I.S. Mintii, A.V. Pikilnyak, Using game simulator Software Inc in the Software

Engineering education. CEUR Workshop Proceedings **2547**, 66–80 (2020), http://ceur-ws.org/Vol-2547/paper05.pdf. Accessed 15 Feb 2020

6. O.M. Markova, S.O. Semerikov, A.M. Striuk, H.M. Shalatska, P.P. Nechypurenko, V.V. Tron, Implementation of cloud service models in training of future information technology specialists. CEUR Workshop Proceedings **2433**, 499–515 (2019), http://ceur-ws.org/Vol-2433/paper34.pdf. Accessed 15 Feb 2020

7. H. Chemerys, K. Osadcha, V. Osadchyi, V. Kruhlyk, CEUR Workshop Proceedings **2393**, 17 (2019), http://ceur-ws.org/Vol-2393/paper_378.pdf. Accessed 15 Feb 2020

8. I. Mulà et al., Catalysing Change in Higher Education for Sustainable Development: A review of professional development initiatives for university educators. International Journal of Sustainability in Higher Education **18**(5), 798–820 (2017). doi:10.1108/IJSHE-03-2017-0043

9. UNESCO, *Bonn Declaration* (UNESCO, 2009), https://unesdoc.unesco.org/ark:/48223/pf0000188799. Accessed 30 Jan 2020

10. UNESCO, *Education for People and Planet: Creating Sustainable Futures for All* (UNESCO, Paris, 2016), http://unesdoc.unesco.org/images/0024/002457/245752e.pdf. Accessed 30 Jan 2020

11. D. Tilbury, I. Mulà, *Review of Education for Sustainable Development Policies from a Cultural Diversity and Intercultural Dialogue: Gaps and Opportunities for Future Action* (UNESCO, Paris, 2009), https://unesdoc.unesco.org/ark:/48223/pf0000211750. Accessed 30 Jan 2020

12. J. Dlouhá, K.V. Mally, J. Dlouhý, ESD principles in higher education from a perspective of Central and Eastern European countries. International Journal of Sustainability in Higher Education **18**(6), 822–840 (2017). doi:10.1108/IJSHE-03-2016-0045

13. E.H. Fedorenko, V.Ye. Velychko, A.V. Stopkin, A.V. Chorna, V.N. Soloviev, Informatization of education as a pledge of the existence and development of a modern higher education. CEUR Workshop Proceedings **2433**, 20–32 (2019), http://ceur-ws.org/Vol-2433/paper01.pdf. Accessed 15 Feb 2020

14. I.M. Tsidylo, H.V. Tereshchuk, S.V. Kozibroda, S.V. Kravets, T.O. Savchyn, I.M. Naumuk, D.A. Kassim, Methodology of designing computer ontology of subject discipline by future teachers-engineers. CEUR Workshop Proceedings **2433**, 217–231 (2019), http://ceur-ws.org/Vol-2433/paper13.pdf. Accessed 15 Feb 2020

15. V. Osadchyi, K. Osadcha, V. Eremeev, The Model of the Intelligence System for the Analysis of Qualifications Frameworks of European Countries. International Journal of Computing **16**(3), 133–142 (2017)

16. V.V. Osadchyi, I.V. Krasheninnik, Formation of short-cycle curricula content for future software engineers training on the basis of the labour market analysis. Information Technologies and Learning Tools **58**(2), 11–25 (2017). doi:10.33407/itlt.v58i2.1637

17. S.L. Koniukhov, Dissertation, Bogdan Khmelnitsky Melitopol State Pedagogical University, 2019

18. B. Penzenstadler, A. Fleischmann, in *Proceedings of the 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T 2011)*, Honolulu, May 2011 (Conference Publishing Solutions, 2011), p. 454. doi:10.1109/CSEET.2011.5876124

19. D.H. Fisher, Z. Bian, S. Chen, Incorporating Sustainability into Computing Education. IEEE Intelligent Systems **31**(5), 93–96 (2016). doi:10.1109/MIS.2016.76

20. Yu Cai, in *Proceedings of the 41st ACM technical symposium on Computer science education (SIGCSE '10)*, Milwaukee, March, 2010 (Association for Computing Machinery, 2010), p. 524. doi:10.1145/1734263.1734439

21. L.M. Hilty, P. Huber, Motivating Students on ICT-Related Study Programs to Engage with the Subject of Sustainable Development. International Journal of Sustainability in Higher Education **19**(3), 642–656 (2018). doi: 10.1108/IJSHE-02-2017-0027

22. A. Kompaniets, H. Chemerys, I. Krasheninnik, Using 3D modelling in design training simulator with augmented reality. CEUR Workshop Proceedings **2546**, 213–223 (2020), http://ceur-ws.org/Vol-2546/paper15.pdf. Accessed 15 Feb 2020

23. D.A. Novikov, *Statisticheskie metody v pedagogicheskih issledovaniyah (tipovye sluchai)* (Statistical methods in pedagogical research (typical cases)). (MZ-Press, Moskva, 2004)

24. V.M. Rudenko, *Matematychna statystyka* (Mathematical statistics). (Tsentr uchbovoi literatury, Kyiv, 2012)

25. E.V. Sidorenko, *Metody matematicheskoj obrabotki v psihologii* (Methods of mathematical processing in psychology). (Rech, Sankt-Peterburg, 2000)