European Journal of Pedagogical Initiatives and Educational Practices
ISSN (E): 2938-3625
Volume 3, Issue 9, September - 2025

DIGITAL DIDACTIC MODEL FOR DEVELOPING PROGRAMMING COMPETENCE IN ENGINEERS WITHIN A DIGITAL LEARNING ENVIRONMENT

Yelena Kodirova 1
1Senior Lecturer Department of Informatics and Computer Graphics,
Tashkent State Transport University, Tashkent, 100167, Uzbekistan
E-mail: lena.kodirova@mail.ru , Orcid-0009-0008-0994-9307

Abstract
In the context of the digitalization of engineering education, there is an increasing need to develop didactic models that foster sustainable programming competence among students in technical fields. The aim of this study is to design and pilot a digital didactic model implemented on the ITTS platform, which ensures systematic development of algorithmic thinking, programming skills, and readiness for professional activity. The methodological framework of the model comprises four interrelated components: the target component (strategic objectives of engineering education), the organizational-methodological component (competence-based model, principles of systematization, integration, and practice orientation), the content component (ITTS digital platform, theoretical and practical elements, visual learning tools), and the diagnostic component (competence assessment through testing, project defense, and analysis of performance dynamics). The empirical validation of the model was conducted through a pedagogical experiment involving control and experimental groups. The collected data confirm a statistically significant improvement in academic performance, increased student engagement, and enhanced cognitive coherence. The main conclusion affirms the effectiveness of the proposed model, its reproducibility, and its potential for scaling across other engineering disciplines.

Keywords: Digital didactics, programming competence, engineering education, visual learning tools, ITTS platform, algorithmic thinking, pedagogical experiment, digital learning environment.

Introduction

The digital transformation of engineering education necessitates a rethinking of didactic approaches to the development of key professional competencies. One such competency is programming competence—a combination of knowledge, skills, and practical abilities that enables future engineers to design, debug, and apply software solutions in their professional activities [1]. In the context of rapid technological advancement and increasing demands for digital literacy among specialists, there is a growing need for systemic models capable of integrating theoretical and practical components of learning, visual tools, and digital platforms.

Analysis of contemporary research shows that digital learning environments and platform-based solutions contribute to improved academic performance and increased student engagement. The works of Khusainov A.R. and Abdullaeva N.N. [2] substantiate the effectiveness of digital resources in engineering education, particularly when interactive and visual components are employed. Sokolova E.V. [3] emphasizes the role of visualization in developing algorithmic thinking, while Kuznetsova N.E. [4] highlights the importance of a systemic approach in designing didactic models. International reports (OECD, 2021; UNESCO, 2022) confirm the relevance of implementing adaptive technologies and learning analytics systems to personalize educational trajectories [5], [6].

Despite the active development of digital solutions, challenges remain in the didactic organization of programming competence formation. Existing approaches are often fragmented, lack cognitive coherence, fail to account for the specifics of engineering disciplines, and do not offer reproducible assessment methodologies. There is a lack of well-developed models that integrate visual tools, digital platforms, and diagnostic instruments into a unified educational system.

The aim of this study is to develop and pilot a digital didactic model aimed at fostering programming competence among engineering students. The objectives include: – structuring the model into four interrelated components (target, organizational-methodological, content-based, and diagnostic); – integrating visual and interactive elements; – empirically testing the model's effectiveness; – formulating recommendations for its scaling and adaptation to various engineering disciplines.

Preliminary results of the study were presented as part of a pedagogical experiment on the ITTS platform and published in [2].

2. Materials and Methods

This study focuses on piloting a digital didactic model for developing programming competence in engineering students. The overall experimental design included a comparative analysis of control and experimental groups enrolled in the course "Fundamentals of Algorithmization and Programming," delivered in both traditional and digital formats.

2.1. Participants and Conditions. The experiment involved first-year engineering students from Tashkent State Transport University. The control group (n=20) was taught using traditional methods, while the experimental group (n=20) followed a digital didactic model implemented via the ITTS platform. All participants provided informed consent in accordance with ethical standards for educational research [7].

2.2. Structure of the Digital Didactic Model. The model comprises four interrelated components: – **Target component**: defines strategic objectives—development of programming competence, formation of professional skills, and conditions for independent learning. – **Organizational-methodological component**: based on a competence-oriented framework, implemented through the principles of systematization, continuity, integration, and practice orientation [8]. – **Content component**: includes the ITTS digital platform, didactic materials, presentations, assignments for individual and group work, as well as visual diagrams

and flowcharts developed by the instructor. – **Diagnostic component**: encompasses pre- and post-testing, project defense, self-assessment, and expert evaluation of the acquired competence.

2.3. Learning Tools and Procedures. The ITTS platform was used as a digital environment for hosting learning materials, organizing feedback, cumulative assessment, and monitoring student activity. Practical tasks were completed in a local Python programming environment installed on students' personal computers. Assignments included algorithm implementation, working with loops, conditions, functions, and arrays. Completed programs were uploaded to the ITTS platform for review and analysis. Visual tools (flowcharts, algorithmic tables, structural maps) were developed by the instructor and integrated into presentations and assignments. The instructional methodology included:

1. Introductory module with visualization of algorithmic structures;
2. Theoretical sessions using interactive presentations;
3. Practical assignments in Python;
4. Project-based work with defense;
5. Final testing and self-assessment.

2.4. Assessment Methods. The effectiveness of the model was evaluated using: – pre- and post-testing; – analysis of student activity (completed tasks, participation in discussions); – expert evaluation of project solutions; – statistical data processing using Student's t-test and Pearson correlation coefficient [9].

2.5. Reliability and Reproducibility. The instructional and diagnostic methodology was documented and piloted across two academic cohorts, ensuring reproducibility of results. All materials, assignments, and visual diagrams are available via the ITTS platform and the local Python environment, enabling other educators to replicate the experiment under similar conditions.

3. Results

During the pedagogical experiment, quantitative data were obtained reflecting the dynamics of programming competence development among engineering students. The main results are presented in Table 1.

Table 1. Comparative Results of Pre- and Post-Testing in Control and Experimental Groups (in percentages)

Group	Average Pre-Test Score (%)	Average Post-Test Score (%)	Gain (%)
Control (n=20)	42.5	58.0	+15.5
Experimental (n=20)	43.0	76.5	+33.5

Note: The testing covered topics such as algorithmization, loops, conditions, functions, and arrays.

Figure 1. presents a visualization of the score gains in both groups. The graph illustrates a more pronounced positive trend in the experimental group, which was taught using the digital didactic model implemented on the ITTS platform.

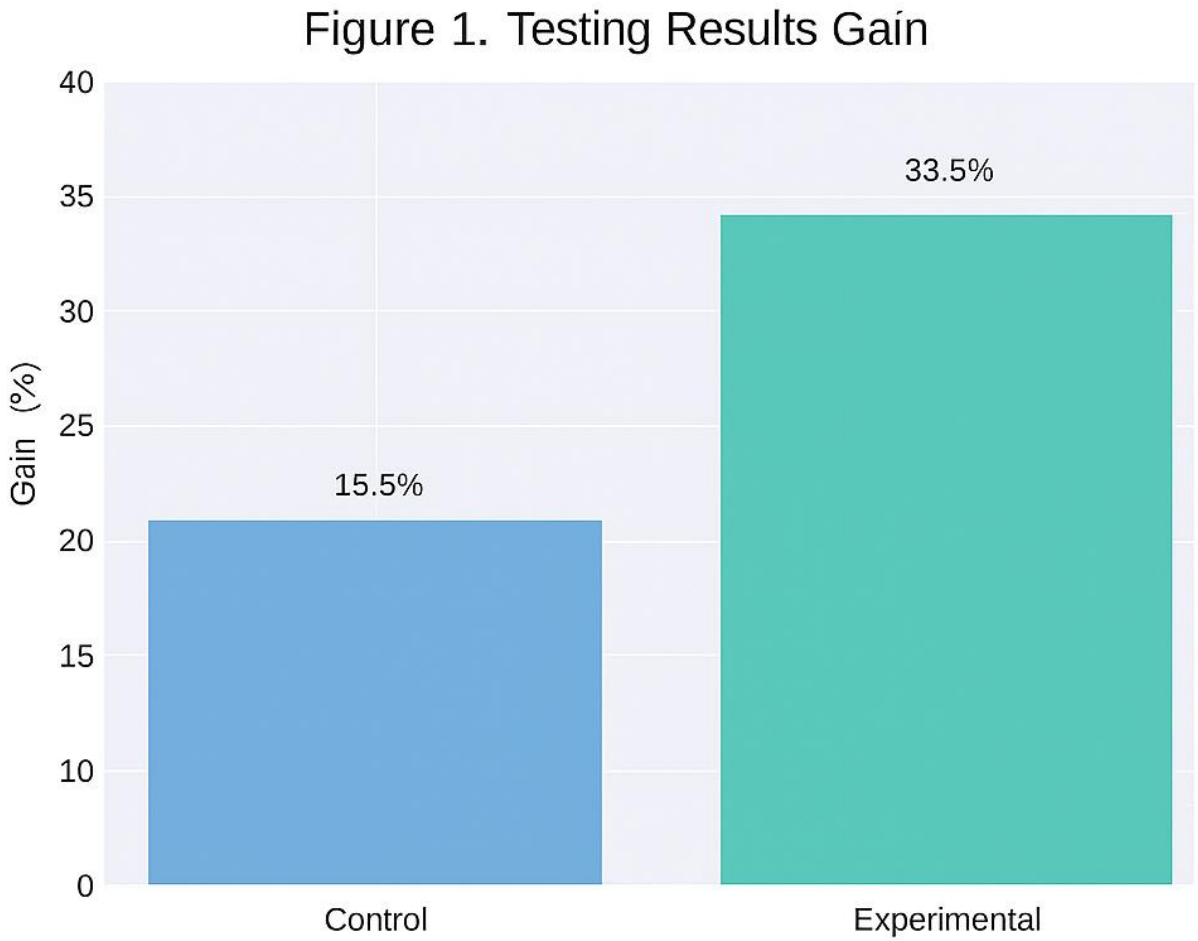

Figure 1. Gain in Test Results in Control and Experimental Groups

Additionally, the following findings were recorded: – In the experimental group, 85% of students completed all practical assignments in Python, compared to 60% in the control group. – The proportion of students who successfully defended their project work was 90% in the experimental group and 65% in the control group. – The average score for the final assignment in the experimental group was 4.6 on a five-point scale, while in the control group it was 3.8. Statistical analysis using Student's t-test confirmed the significance of differences between the groups ($p < 0.05$), which validates the effectiveness of the applied model [10].

4. Discussion

The objective of this study was to develop and pilot a digital didactic model for fostering programming competence among engineering students. The hypothesis posited that the integration of visual tools, the ITTS digital platform, and local practice in Python would contribute to more effective development of algorithmic thinking and applied programming skills.

The results obtained confirm the hypothesis: the experimental group demonstrated a 33.5% gain in test results, significantly exceeding the 15.5% gain observed in the control group. Additionally, increased student engagement, successful task completion, and project defense were recorded. These findings align with the conclusions of Sokolova E.V. [11], who emphasizes the importance of visualization and digital environments in shaping engineering competencies.

Summarizing the results, it can be asserted that the proposed model ensures cognitive coherence, systematization, and practice-oriented learning. Its implementation helps overcome the fragmentation of traditional approaches, enhances student motivation, and ensures reproducibility of the didactic process. This corresponds to international trends in engineering education, as reflected in the works of Crawley et al. [12] and the recommendations of UNESCO [13].

The successful execution of the research tasks was made possible by the clear structuring of the model and the adaptation of the ITTS platform to instructional goals. However, certain limitations were identified during implementation: the absence of built-in code editors and simulators on the platform necessitated the use of external software tools, complicating monitoring and diagnostics. Furthermore, the small sample size (n=20) limits the statistical generalizability of the results.

The practical application of the model is feasible within the modernization of academic courses, development of digital modules, and professional development of engineering educators. It is recommended to integrate the model into platforms with extended visualization and automated diagnostic capabilities, as well as to conduct large-scale studies involving multiple universities. Future plans include expanding the model to other disciplines, developing visual flowcharts for Python, and creating a unified glossary of terms to ensure terminological and methodological consistency. Another relevant task is the development of mechanisms for automated evaluation of project solutions and the integration of the model into adaptive learning systems.

Thus, the findings confirm the effectiveness of the digital didactic model and its significance for developing programming competence within a digital learning environment.

REFERENCES

1. Беспалько В. П. Слагаемые педагогической технологии. — М.: Педагогика, 2021. — 192 с.
2. Хусаинов А. Р., Абдуллаева Н. Н. Цифровые технологии в инженерном образовании: опыт Узбекистана // Вестник ТГПУ. — 2023. — № 2. — С. 45–52.
3. Соколова Е. В. Визуальные средства обучения в цифровой дидактике // Педагогика и цифровизация. — 2022. — № 4. — С. 33–41.
4. Кузнецова Н. Е. Модернизация инженерного образования: методологические подходы // Киберленинка. — 2022. — URL: https://cyberleninka.ru/article/n/modernizatsiya-inzhenernogo-obrazovaniya-metodologicheskie-podhody (дата обращения: 02.10.2025).
5. OECD. Digital Education Outlook: Pushing the Frontiers with AI, Blockchain and Robots. — OECD Publishing, 2021. — 156 p.

6. UNESCO. Reimagining our futures together: A new social contract for education. — Paris: UNESCO, 2022. — 210 p. — URL: https://unesdoc.unesco.org/ark:/48223/pf0000379707 (дата обращения: 02.10.2025).
7. Мухамедова М. А. Этические аспекты педагогических исследований в цифровой среде // Образование и право. — 2022. — № 3. — С. 45–50.
8. Crawley E. F., Malmqvist J., Östlund S., Brodeur D. R. Rethinking engineering education: The CDIO approach. — Springer, 2014. — 286 p.
9. Anderson L. W., Krathwohl D. R. A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. — Longman, 2001. — 352 p.
10. Громов А. В. Статистическая обработка педагогических данных: учебное пособие. — М.: Юрайт, 2021. — 176 с.
11. Соколова Е. В. Визуальные средства обучения в цифровой дидактике // Педагогика и цифровизация. — 2022. — № 4. — С. 33–41.
12. Crawley E. F., Malmqvist J., Östlund S., Brodeur D. R. Rethinking engineering education: The CDIO approach. — Springer, 2014. — 286 p.
13. UNESCO. Reimagining our futures together: A new social contract for education. — Paris: UNESCO, 2022. — 210 p. — URL: https://unesdoc.unesco.org/ark:/48223/pf0000379707 (дата обращения: 02.10.2025).
14. Кодирова Е.В. Использование интерактивных образовательных технологий для повышения компетентности в программировании будущих инженеров. Pedagogik mahorat Ilmiy-nazariy va metodik jurnal 7-son 2025-yil, iyul.
15. Кодирова Е. В. Методика использования программных средств в обучении основам информатики и программирования Pedagogik akmeologiya xalqaro ilmiy-metodik jurnali 2025 г.