СЕКЦІЯ: ІНСТРУМЕНТИ, МЕТОДИ ДИСТИНЦІЙНОГО ТА ЗМІШАНОГО НАВЧАННЯ В ЗАКЛАДАХ ОСВІТИ

INTEGRATING MOBILE DEVELOPMENT INTO THE SCHOOL COMPUTER SCIENCE CURRICULUM: CURRENT APPROACHES AND CHALLENGES

Oksana Vasylenko
Research Staff of the Electrical Engineering, Mechanical Engineering and Industrial Engineering Department
Anhalt University of Applied Sciences (Koethen, Germany)
Oksana.Vasylenko@hs-anhalt.de

Implementing mobile app development instruction in a school computer science curriculum requires the use of modern pedagogical approaches and appropriate toolkits. One of the leading approaches is project-based learning, which involves students creating their own software products while working on educational projects. This approach helps develop practical programming skills, teamwork, and accountability for results.

Problem-based learning also plays a significant role in the process of teaching mobile development. Students are given practical tasks that simulate real-life situations, which require the development of a mobile app to solve. This stimulates students' cognitive activity and promotes a deeper understanding of the learning material [1].

Visual programming environments have become widely used in school practice, lowering the barrier to entry into programming and making it accessible to students of all ages. Among these environments, MIT App Inventor, Thunkable, and Kodular hold a special place, as they allow students to create mobile apps using block-based programming. Using these environments makes the programming process more visual and helps students achieve practical results quickly [2].

In addition to visual programming environments, it is advisable to introduce students in upper grades to text-based programming languages, such as Java or Kotlin for the Android platform. This ensures continuity between school and professional training and broadens students' understanding of modern software engineering.

The use of cloud services, mobile emulators, and online platforms for teaching programming significantly expands the possibilities for organizing the educational process. Such tools allow for work regardless of the school's technical infrastructure and provide access to educational resources at any time.

Despite the significant pedagogical potential of mobile development, its implementation in school practice is accompanied by a number of problems and challenges that require systematic analysis and well-founded solutions. Identifying these problems is a crucial step toward creating an effective methodology for teaching mobile app development in general secondary education institutions.

One of the main problems is the limited material and technical resources available to many schools. An insufficient number of modern computers and mobile devices, or unstable internet access, complicate the organization of students' practical activities. In

some schools, it is not possible to install the necessary software or use mobile device emulators, which negatively affects the quality of mobile development instruction.

Another significant challenge is the level of training among computer science teachers. As a relatively new field, mobile development requires teachers to possess not only programming knowledge but also an understanding of the specifics of mobile platforms, interface design, and modern development environments. The lack of professional development courses and instructional materials makes it difficult to integrate mobile development into the educational process at an adequate level.

Another problem is the overloaded school curriculum for computer science and the limited number of instructional hours. Under these conditions, it is difficult to allocate sufficient time for a thorough study of mobile development, especially given the need to develop practical skills. This necessitates the integration of mobile development with other course topics or the use of the elective component of the curriculum [3].

A psychological and pedagogical challenge is the varying levels of students' preparation and cognitive abilities. Some students have basic programming skills, while others are just beginning to learn about algorithms. In such conditions, it is important to ensure a differentiated approach to teaching and the use of learning environments that allow the complexity of tasks to be adapted to the students' level of preparation.

At the same time, the integration of mobile development into school practice holds significant promise. First, teaching students how to create mobile apps significantly boosts their motivation to study computer science, as the results of their work have practical value and can be used in everyday life. Second, mobile development fosters interdisciplinary connections, particularly with mathematics, physics, design, and technology, which aligns with the principles of integrated learning.

A promising direction is the use of visual programming environments and cloud technologies, which help minimize technical limitations and make mobile development accessible to most educational institutions. Such environments facilitate a gradual transition from block-based to text-based programming, ensuring continuity in learning and preparing students for further professional education in the field of information technology.

Thus, despite existing problems and challenges, the introduction of mobile development into school practice has significant educational potential. Provided there is adequate methodological support, teacher professional development, and improvements to the material and technical infrastructure, mobile development can become an effective tool for fostering students' information and digital literacy and preparing them for life and professional activities in a digital society.

REFERENCES

1. Bransford, J., Brown, A., and Cocking, R. *How People Learn: Mind, Brain, Experience, and School.* National Academies Press, Washington, D.C., 2000.

2. Kafai, Y., Peppler, K., and Chiu, G. High-tech programmers in low-income communities: Seeding reform in a community technology center. In *Communities and Technologies*, C. Steinfield, B. Pentland, M. Ackerman, and N. Contractor, Eds. Springer, New York, 2007, 545–564.

3. Guzdial, M. Programming environments for novices. In *Computer Science Education Research*, S. Fincher and M. Petre, Eds. Taylor&Francis, Abingdon, U.K., 2004, 127–154.