Vladyslav Bilous
Manager of cabinet of laboratory of the embedded systems and 3D design
Borys Grinchenko Kyiv University, Kyiv, Ukraine
v.bilous@kubg.edu.ua
ORCID: 0000-0001-6915-433X

PROTOTYPING OF MOBILE APPLICATIONS OF ANDROID IN ENVIRONMENT OF MIT APP INVENTOR AS MEANS ARE INTRODUCTION TO BASES OF PROGRAMMING ON THE LESSONS OF INFORMATICS OF STUDENTS OF MIDDLE SCHOOL

Abstract. In the article the instrument of mobile development of MIT App Inventor, as means of educating of elements of programming of students of high school, is presented on the lessons of computer science. The technological, methodical and vocational orientation aspects of the use of environment of planning of mobile appendixes of MIT are presented. App Inventor’s intuitive programming metaphor and incremental development capabilities allow the developer to focus on the logic for programming an app rather than the syntax of the coding language, fostering digital literacy for all. Since it was moved from Google to MIT, a number of improvements have been added, and research projects are underway. A requirement in new teaching methods that facilitate and accelerate the transmission of knowledge strengthens the process of assimilation of knowledge. Such methods of study can come true through the use of information technology in education. A main problem is a necessity to develop methods and resources for support and educating in schools, using mobile technology. Using new economic feasibilities, a student creatively goes near implementation of tasks, thus explaining to realization of self-development.

Keywords: mobile development; prototyping applications; lesson of computer science; middle school; programming

Introduction. We live in the century of rapid development of technologies and large height of information in different industries of knowledge. Certainly, this fact was represented and on the system of education on the whole, and on the educational article of "computer science" in particular. Obviously, that the sharp necessity of modernization of maintenance came to a head, search of pedagogical innovations that must bring quality over of preparation of graduating students of school in accordance with the queries of modern society. In respect of school course of informatics, it will demand mastering of new concepts, introduction of the terminological vehicle related to new possibilities of ICT, in particular, mobile technologies. It is possible that it will demand modernization and didactic principles tested by time, realization of them at qualitatively new level.

It is very important to understand that knowledge left off to be something single. Divided into "sciences" and "disciplines", it loses slender integrity and appears before student, figuratively speaking, as elements of "puzzle". Universal education is not that another, as a study of separate details of this "puzzle". Many researchers suppose that a problem exists from organization of the school educating. For example, American scientist in area of theory of the computer systems, one of pioneers of areas of the object-oriented programming and graphic interface Alan Key, that attentively watches after educating of children, considers "that it is needed as possible before to give to the child a powerful "instrument for thinking". The basic setting of this instrument is cognition of new and creation of connections between well-known, development of not only analytical but also synthetic thinking".

Thus, it registers in the position papers of the national educational program that his accordance must become the "main result of school education to the aims of passing ahead development" and "studying in schools is necessary methods and technologies that will be useful in the future". It follows from this that the every student must master key to the competence, to purchase experience of independent activity and personal responsibility, that, in turn, and determine quality of modern education. One of educational results in accordance with the requirements of state educational standard at all levels of universal education is ICT-competence of student. In this connection the role of school course of informatics increases considerably.

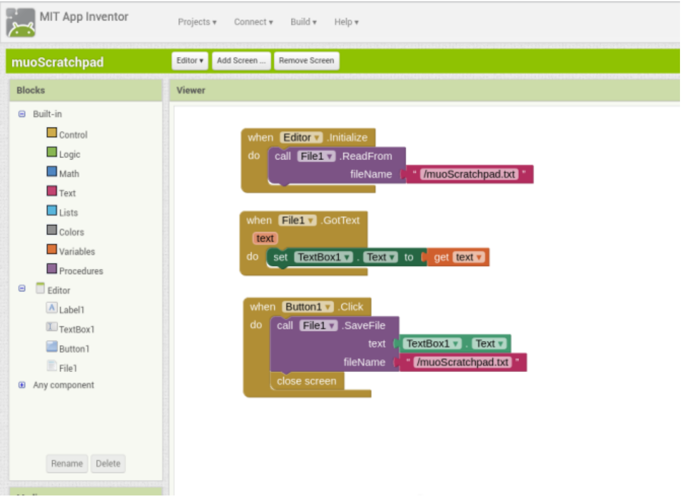
Working at the computer, a student constantly runs into the "window" programs, in that there are the buttons, inscriptions, images, sounds, therefore in his understanding, program that he makes must be the same. Instead, him one of programming languages is offered to: Basic or Pascal. Going near their study is approximately identical in all textbooks: in an order to write the program it is needed to consider a programming environment, program structure, syntax of language, after students proceed to writing of the simplest program

Today the teacher of informatics has unique possibilities of forming of cognitive interest of schoolchildren in the process of educating to programming, foremost, due to the wide spectrum of the used software environments. So, for example, on the lessons of informatics at the study of programming teachers are actively use visual environments programming (Scratch) oriented to development of "desktop" applications that, undoubtedly, causes large interest for schoolchildren, but dissatisfies him fully. It can be explained by that, besides computers, the student use mobile devices all anymore. Does a question get up before the teacher of informatics, how to convert a favorite toy into the means of educating.

Most schoolchildren perceive the division of programming as difficult and boring, and looks forward to, when he will make off. At the same time, educating to programming is very important, and it would be desirable to organize this process so that he took not two-three persons from a class, and would be interesting to all. One of ways of decision of this problem is the use of editor MIT App Inventor. MIT App Inventor is the programming environment worked out by scientists from Massachusetts Technological Institute (MIT). Program is intended for development of appendixes for mobile devices (smartphones and plane-table computers) working on the operating system Android.

MIT App Inventor is an environment of visual development of Android- of appendixes. For development of appendixes knowledge of programming of Java and Android SDK language is not required in MIT App Inventor, enough knowledge of elementary bases of algorithmization is at the level of high school. Dignity of this environment is that with her help just for a few minutes it is possible to create the elementary application, and for a hour - to build the difficult enough program with a few screens. (Fig.1)

That he does not need to be set on a computer behaves to basic advantages of editor MIT App Inventor, he works straight from a browser, it is needed only to have Google account.



```
when Editor .Initialize
do  call File1 .ReadFrom
              fileName " /muoScratchpad.txt "

when File1 .GotText
  text
do  set TextBox1 . Text to get text

when Button1 .Click
do  call File1 .SaveFile
                  text TextBox1 . Text
              fileName " /muoScratchpad.txt "
    close screen
```

Fig.1 Start screen application MIT App Inventor

A programming of environment of MIT App Inventor language is very simple, as he from the beginning was created in order that he was used by schoolchildren. At programming on him it is not needed to write code lines, as it takes place on other programming languages. To create appendixes in MIT App Inventor, simply enough to drag blocks, collecting them in the program, as puzzle.

Developing in the environment of App Inventor, students can be taught to work in a command. Working in groups with App Inventor is not an easy task. Students can work together to plan their apps and share ideas, but when it is time to design and implement their apps, the tool has some limitations that make group work challenging. For example, blocks cannot be copied from one project to another, and two or more people cannot work on the same project at the same time. Even with the limitations of App Inventor to support collaborative work, 68 percent of the respondents agreed/strongly agreed they enjoyed working in teams to develop mobile apps. The course included several group assignments with group sizes of 2 to 5 students. Besides teamwork, many assignments also included time and scope constraints to challenge and persuade students to manage their tasks and progress. Students had to come up with their own approach to make a group assignment work. For example, groups spent more time planning the app and creating mockup screens to define the components and functionalities needed before starting any code. On some occasions, they even discussed how to name the components so that other group members could easily find them. Some groups decided to separate their activities and each member would work on their own to complete their respective tasks. After that, they would create a shared account to access App Inventor and then each member would take turns implementing their part of the project. One group tried to get all members logged in on App Inventor at the same time to work on the same project using the same user account. However, they quickly learned that the current version of App Inventor does not support synchronous collaboration. For some groups, each member would work individually on their tasks and send their work to a member that was responsible to combine all parts into one project. Finally, for other groups, the approach was one member working on the computer and the other members around him or her discussing the project and providing support during the implementation.

The visual programming approach of App Inventor helps to hide some of the complexities of programming, providing students with opportunities to concentrate on the design of the application, its features, and how users will interact with it. That means, students should learn and practice the design of user interfaces, user input and output, and input validation as they play an important role in the user’s experience with mobile applications. More often than not, students will detect some problems with their apps that can be a result of poor user interface design. In particular, user inputs are overlooked which will make apps misbehave or crash when users enter unexpected data or do not provide any data. Of course, App Inventor provides several properties to the components to allow developers to set up the application as needed. For example, the component TextBox can be set to number only in order to restrict the type of data entered. However, it is the developer’s responsibility to define a range of acceptable numbers and to create the appropriate code to validate it. Other properties such as enable/disable and visible/hidden provide ways for developers to customize their apps.

The process of app development is not so different from the traditional software development process. However, if we develop Android apps more quickly with a project team having developers who deficient in Android SDK and Java programming experience, it will greatly contribute to improving apps development productivity. The App Inventor is used by many departments relating computer science and engineering in college and many students interested in smart apps in the school. Everyone insufficient to programming knowledge can create Android apps by using App Inventor. You can design Android app on a web page, put a few of logic blocks together on the same page, and test Android app on an emulator or on your phone at the same time. App Inventor is the advantage of component visual programming, where one can drag and drop visual components, and then give programmable behaviors to logic blocks to develop mobile apps easy. Therefore, App Inventor using for app prototyping is highly efficient because App Inventor is easy to use and intuitive for app programming and App Inventor is effective on connectivity with other process of app development processes.

With the help of App Inventor, a mobile development course should be fun and packed with several computing concepts besides programming, such as database, data communication, software development, project management, mobile applications development, web services and more. Now that students have built a background on app development, instructors teaching more advanced courses can illustrate the concepts of their specific courses with the support of App Inventor. For example, in a database course, students could create forms to insert data into tables or display data from the tables using both static and dynamic queries. In a software engineering or systems analysis and design course, students could benefit from App Inventor's support for rapid development to plan, design, implement and test mobile apps as part of course assignments or projects. In particular, instructors could explore principles and techniques for user interface design.

Unique possibility of MIT App Inventor is testing of the developed applications on a mobile device real-time without the preliminary compiling and setting on a mobile device. For this purpose it is enough to set the special appendix of MIT AI2 Companion on a mobile device. In addition, testing of appendixes is possible and in the emulator of Android for PC. For creation of appendix on a screen it is necessary to take away the necessary elements of interface. Programming is executed with a help block - charts. It is necessary to connect blocks so that an appendix executed the required actions. Upon completion of development it is possible to get either the prepared APK- file (format of the archived executable files-applications for Android) for setting on a device or QR-cod with reference to downloading.

The prospect of educating of schoolchildren to development of mobile applications and process of forming of professional interest have vocational orientation character, as it is a dynamically developing area of professional activity. It is enough to give a few well-known facts as arguments. It is possible to say with a confidence, that at the market of mobile applications a transition is fixed from the market of suggestion of appendixes to the market of demand. The consumption of mobile services grows on the whole, as sales of smartphones grow, a consumable mobile traffic grows, the sales of plane-tables grow. There is an active height of mobile advertisement.

Conclusion. Thus, the professional sphere related to programming of mobile applications is an actively developing sector of IT-industry. However in senior classes to explain to the receipt of knowledge-intensive professions already late. To the creatively working teacher obviously, that a mortgage of success of preparation of high quality programmer is steady motivation already on the stage of basic universal education and as a cooperant instrument, MIT App Inventor is one of facilities of forming and support of this motivation.

The study of programming built thus assists an achievement not only of subject results, similarly during a lesson there is forming:

- responsible attitude toward studies, readiness and ability of student to self-development and self-education on the basis of motivation to educating and cognition;
- integral world view, corresponding to the modern level development of science;
- abilities independently to plan the ways of achievement of aims;
- abilities to correlate the actions with the planned results, to carry out control of the activity in the process of achievement of result;
- abilities to estimate the rightness of performance of educational objective.

REFERENCES

The MIT App Inventor Library: Documentation & Support (2018). March 20, 2018. http://appinventor.mit.edu/explore/library.html

Soares, A. & Martin, N. (2015). Teaching non-beginner programmers with app. inventor: survey results and implications. Proceedings of the Information Systems Educators Conference.

Gestwicki, P. & Ahmad, K. (2011). App inventor for Android with studio-based learning. Journal of Computer Science in Colleges, 27(1), 55-63.

Gray, J., Abelson, H., Wolber, D. & Friend, M. (2012). Teaching CS principles with app inventor. Paper presented at the Proceedings of the 50th Annual Southeast Regional Conference, Tuscaloosa, Alabama.

Wolber, D. (2011). App Inventor and Real-World Motivation. Proceedings of the 42nd ACM technical symposium on Computer science education (SIGCSE ‘11).

Text of the article was accepted by Editorial Team 23.03.2018

ПРОТОТИПИЗАЦІЯ МОБІЛЬНИХ ДОДАТКІВ ANDROID В СЕРЕДОВИЩІ MIT APP INVENTOR ЯК ЗАСІБ ВСТУПУ ДО ПРОГРАМУВАННЯ НА УРОКАХ ІНФОРМАТИКИ УЧНІВ СЕРЕДНЬОЇ ШКОЛИ

Білоус Владислав Володимирович
Завідувач кабінету лабораторії вбудованих систем та 3D-дизайну
Київський університет імені Бориса Грінченка, м. Київ, Україна
v.bilous@kubg.edu.ua
ORCID: 0000-0001-6915-433X

Анотація. У статті представлений інструмент мобільної розробки MIT App Inventor, як засіб викладання елементів програмування учнів середніх шкіл на уроках інформатики. Представлені технологічні, методичні та професійно - орієнтовні аспекти використання середовища прототипування мобільних додатків MIT. Потреба в нових методах навчання, які полегшують та прискорюють передачу знань, зміцнюють процес асиміляції знань. Інтуїтивне середовище програмування Inventor і додаткові можливості розробки дозволяють розробнику зосередити увагу на логіці програмування додатка, а не на синтаксисі мови кодування, сприяючи цифровій грамотності для всіх. Оскільки він був переміщений з Google на MIT, було внесено ряд покращень, і проводяться дослідницькі проекти. Завдяки орієнтованим блокам в середовищі розробки, увага акцентується на побудові схеми та логіці проекту з мінімальним написанням коду. Для більш детального редагування є можливість продивитися повний код розробки. Такі методи навчання можуть здійснитися завдяки використанню інформаційних технологій в освіті. Головною проблемою є необхідність розробки методів та ресурсів для підтримки та навчання в школах, використовуючи мобільні технології. Використовуючи нові економічні можливості, учень творчо підходить до виконання завдань, тим самим пояснюючи реалізацію саморозвитку.

Ключові слова: мобільна розробка; прототипування додатків; урок інформатики; середня школа; програмування