

Огляд середовищ програмування з метою їх добору для формування інформаційно-цифрової компетентності майбутніх вчителів початкових класів

Оксана Іванівна Яценко¹

| Опубліковано | Секція | УДК |
|--------------|-------------------|----------------|
| 21.08.2023 | Освіта/Педагогіка | 004.43:378.147 |

DOI: <http://dx.doi.org/10.5281/zenodo.8277675>

Ліцензовано за умовами Creative Commons BY 4.0 International license

Анотація. В статті зроблено аналіз середовищ програмування з метою відбору найбільш оптимальних для формування інформаційно-цифрової компетентності вчителів початкових класів. При виборі групи мов програмування для їх аналізу враховано рівень підготовки студентів, рівень когнітивного навантаження, можливість використання в професійній діяльності тощо. Зроблено висновок, що найбільш доцільним є використання візуальних блокових мов програмування.

Сформульовано групи критерії для аналізу середовищ програмування, що найкраще відповідають потребам майбутніх вчителів початкових класів: можливості мови та середовища програмування; можливість використання на початковому етапі вивчення мови програмування; технологічні особливості середовища програмування. За визначеними критеріями проведено системний порівняльний аналіз середовища програмування.

За результатами оцінювання зроблено висновок, що незважаючи на певні недоліки (відсутність можливості автоматичної генерації текстового коду програми та обмежені можливості запису математичних виразів), середовище програмування Scratch є найбільш зручним та доцільним для майбутніх вчителів початкових класів.

Ключові слова: мова програмування; середовище програмування; блокові середовища програмування; критерії добору; вчитель початкових класів.

An overview of the programming environment for the purpose of their selection for the future primary school teachers' information and digital competence formation

Abstract. At the present stage of development of society, it is important to prepare future primary school teachers for the effective use of information and communication technologies in professional activities. At the current stage of society's development, it is important to prepare future primary school teachers for the effective use of information and communication technologies in their professional activities.

The article analyzes programming environments with the aim of selecting the most optimal ones for the formation of informational and digital competence among future primary school teachers. When choosing a group of programming languages for their analysis, such

¹ Яценко Оксана Іванівна, асистент кафедри комп'ютерних наук та інформаційних технологій, Житомирський державний університет імені Івана Франка, Житомир, <https://orcid.org/0000-0003-4983-2775>

features are taken into account: the level of training of students; level of cognitive load when learning the language; the possibility of use in lessons, in extracurricular time and for organizing independent work; the simplicity of the programming language and environment.

As a result of the analysis of the works of Ukrainian and foreign scientists, it is concluded that the most appropriate is the use of visual block programming languages.

A group of criteria for the analysis of programming media, which best meet the needs of future teachers of primary classes, have been formulated. When determining the criteria for the analysis of the selected applications, factors such as: the possibilities of the language and the programming environment (peculiarities of writing formulas, the presence of logical operations and algorithmic constructions, the possibility of generating text code); the possibility of use at the initial stage of learning the programming language (convenience of the interface, availability of instructions and textbooks, availability of the Ukrainian language, price of the application); technological features of the programming environment (multi-platform, type of license, availability of versions for mobile devices, level of support and development of the environment). According to certain criteria, a system comparative analysis of the programming environment NXT-G, Scratch, Snap!, Google Blockly, Alice was carried out.

Based on the results of the evaluation, it was concluded that, despite certain shortcomings (lack of automatic generation of program text code and limited ability to write mathematical expressions), the Scratch programming environment is the most convenient and appropriate for future elementary school teachers.

Keywords: programming language, programming environment, block programming environment; selection criteria; primary school teacher.

Вступ

В наш час професійна компетентність фахівця в будь-якій сфері діяльності визначається не лише об'ємом його знань в якісь конкретній галузі, а й вмінням користуватись програмним забезпеченням загального та спеціального призначення під час виконання фахових завдань та для самоосвіти та саморозвитку, забезпечувати роботу в мережі, розробляти програмні застосунки для їх використання під своєї діяльності тощо. В таких умовах використання інформаційно-комунікаційних технологій вважається одним із важливіших напрямків розвитку інформаційного суспільства.

Освіта не могла не відповісти на виклики сучасності. В умовах нерегламентованості та багатоваріантності навчання, формування та розвитку нестереотипного мислення, котре характеризується здатністю генерувати нові ідеї, а також швидкістю, гнучкістю, оригінальністю та точністю, основним засобом навчання стають цифрові інструменти, розширюється функціонал вчителя, змінюється тип освітнього середовища та характер взаємодії в ньому вчителя та учня. Саме тому одним із основних напрямків розвитку освіти є впровадження в освітній процес сучасних інформаційно-комунікаційних технологій, що забезпечують підвищення рівня навчально-виховного процесу, збільшують рівень доступності та ефективності освіти і готують підростаюче покоління до життя в інформаційному суспільстві.

З метою реалізації Державного стандарту початкової освіти невідкладним завданням закладів вищої освіти має стати забезпечення підготовки вчителя початкових класів до викладання нових предметів та до організації навчання на основі інформаційно-комунікаційних технологій. Вчитель повинен забезпечити розвиток в учнів інформаційно-комунікаційної компетентності (це, означає, що учень повинен оволодіти основами цифрової грамотності для розвитку та комунікації, навчитись безпечно та етично використовувати ІКТ у навчанні та у повсякденному житті),

формувати компетентності в галузі техніки і технологій (здатність до зміни навколишнього світу засобами сучасних технологій без заподіяння йому шкоди, до використання технологій для власної самореалізації, культурного і національного самовираження) [1].

Вимоги до рівня освіти, навчання, професійного розвитку та перелік функцій вчителя початкової школи прописані в професійному стандарті «"Вчитель початкових класів закладу загально середньої освіти", "Вчитель закладу загальної середньої освіти", "Вчитель з початкової освіти (з дипломом молодшого спеціаліста)"» [2]. Відповідно до цього стандарту вчитель повинен бути здатним працювати з цифровими ресурсами та інструментами, а, за потреби, створювати нові цифрові ресурси, вміти шукати та критично оцінювати інформацію, інтегрувати їх у навчальний процес та розвивати інформаційно-технологічну грамотність учнів.

Крім того цей стандарт містить інформацію про предмети та засоби праці вчителя. Серед них велику роль відіграють саме цифрові: персональний комп'ютер, засоби оргтехніки, електронні книги, програмні педагогічні засоби, наочні та віртуальні засоби навчання, освітні платформи, засоби для спілкування, бази даних програм підвищення кваліфікації тощо.

Враховуючи зазначене вище, можна зробити висновок, що однією із основних цілей закладу вищої освіти є формування в майбутнього вчителя початкових класів інформаційно-цифрової компетентності, як необхідної умови його фахової стабільності та затребуваності на сучасному ринку праці, а одним із основних шляхів досягнення зазначеної мети – виважено дібраний програмно-технічний і функціональний інструментарій.

Вибір програмного забезпечення, що можна використати для формування інформаційно-цифрової компетентності майбутнього вчителя початкових класів загальноосвітньої школи та стане в нагоді в процесі роботи за фахом є складною психолого-педагогічною проблемою. Дослідники рекомендують добирати програмні засоби залежно від результатів, яких потрібно досягти. Одні аргументують вибір його простотою, другі – наявністю навчально-методичного забезпечення, треті – актуальністю та поширенням [3].

Дослідженню даного аспекту присвячена значна кількість праць науковців. Висвітленням питань добору мов та середовищ програмування для підготовки вчителів-предметників займаються Вдовичин Т. Я., Лазурчак Л. В. [4], Когут У. П. [5]. Питанням добору та доказам доцільності використання середовищ візуального блокового програмування для вивчення мов програмування на початковому етапі навчання присвячені наукові праці значної кількості закордонних вчених. Серед них Девід Бау, Джеф Грей, Кейтлін Келлехер, Джош Шелдон, Франклін Турбак [7], Д. Бау, Д. А. Бау, М. Доусон та К. С. Пікенс [8], П. Боттоні, М. Черіані [9] та ін.

Недостатньо дослідженим на даний час залишається питання обґрунтованого вибору цифрових засобів, спрямованих на формування інформаційно-цифрової компетентностей вчителів початкових класів, що й визначило мету статті.

Мета статті – дослідити проблему добору блокового середовища програмування як засобу розвитку інформаційно-цифрової компетентності майбутнього вчителя початкових класів.

Завдання цієї статті полягає у наданні об'єктивної інформації, необхідної для розумного вибору середовища програмування для розвитку інформаційно-цифрової компетентності майбутнього вчителя початкових класів.

Результати

Вибір мови програмування для розвитку інформаційно-цифрової компетентності майбутніх вчителів початкових класів має особливості, що визначаються такими чинниками:

- низьким рівнем підготовки майбутніх вчителів в галузі інформаційно-комунікаційних технологій та програмування;
- складністю текстових мов програмування;
- можливістю застосовувати набуті знання.

З моменту появи перших мов програмування викладачі та розробники ждали максимально спростити процес їх вивчення. Створювались прості мови програмування, що давали інструментарій для знайомства з базовими поняттями та засвоєння логічних основ (Basic, Pascal). Наступним кроком в процесі розвитку навчальних мов програмування стала поява інтерактивних візуальних мов та середовищ програмування (Etoys, Squeak, Logo). Як зазначають науковці «саме використання візуалізації в освітньому просторі забезпечує простий і ефективний підхід до отримання результатів, вирішення проблем і відкриття структури моделі в процесі навчання студентів нової інформації. Візуалізація відносин і логічних зв'язків в рамках єдиної моделі дозволяє підтримувати цифрові і базові компетенції студентів в галузі науки і техніки. Моделювання на заняттях з інформатики – це не тільки інструмент, а й сам предмет освіти, коли студенти на основі отриманих знань і з допомогою цифрових інструментів, створюють модель частини реального світу» [9].

Останнім трендом в даній обласні є блокове візуальне програмування. Середовища блокового візуального програмування – це свого роду конструктори з кольорових деталей (блоків), кожна з яких має ім'я. Вони, як правило, використовують функцію перетягування блоків, що можуть містити значки або текстові мітки, а не введення команд з клавіатури. Часто використовуються діалогові вікна, спливаючі підказки, випадаючі списки команд тощо. Вірно підібрані та впорядковані блоки призводять до появи справжнього робочого коду.

Дослідження американських науковців [9] показало, що на початковому етапі вивчення мови програмування основними перевагами блокового візуального програмування є:

1. Словник. Однією з основних проблем в навчанні програмуванню є необхідність запам'ятовувати структуру коду та значну кількість специфічних слів, що описують його. Застосування блоків з текстовими мітками дозволяє зосередитись на процесі розробки програми, а не думати про конкретні слова та особливості їх написання.
2. Когнітивне навантаження. Початківцями досить складно написати текст програми, через те, що потрібно одночасно працювати в декількох напрямках: мислити творчо для того, щоб знайти рішення поставленої задачі та логічно, щоб вірно задати послідовність виконання команд та отримати правильний результат. Використання вже готових блоків зменшує когнітивне навантаження до запам'ятовування конкретної кількості елементів, які потрібно розташувати в певному порядку. Фактично, ціль тут така ж, що і в текстовому програмуванні, а процес досягнення цієї мети є більш простим та зрозумілим. Такий підхід виглядає більш привабливим та цікавим.
3. Помилки. У процесі вивчення програмування на початкових стадіях часто виникають труднощі через неухважність, недостатнє знання синтаксису мови та недостатній рівень володіння іноземними мовами. В результаті наявні помилок, що роблять програму неробочою, процес їх пошуку та виправлення знижує інтерес студента до програмування. У середовищах блокового візуального

програмування зробити синтаксичні чи орфографічні помилки неможливо, помилки логічні є очевидними завдяки обмеженій кількості блоків, а результат, в більшості випадків, той же.

Вибір середовища візуального блокового програмування для підготовки майбутніх вчителів початкових класів є складним завданням через велику кількість умов та обмежень, що на нього впливають. Тому визначити універсальні критерії відбору є складною задачею. На нашу думку при формулюванні критеріїв відбору варто звернути увагу на те, щоб, де це можливо, оцінка була об'єктивною та легко перевірялась [10].

Сформулюємо, на нашу думку, найбільш доцільні:

1. Можливості середовища та мови програмування (дана група критеріїв важлива для переходу на наступну сходинку вивчення програмування: до вивчення текстових мов).
 - 1.1 Підтримка запису формул в математичному вигляді.
 - 1.2 Підтримка основних математичних та логічних операцій.
 - 1.3 Підтримка основних алгоритмічних конструкцій.
 - 1.4 Можливість генерації текстового коду.
2. Можливість використання середовища на початковому етапі вивчення мов програмування.
 - 2.1 Зручність, простота, сучасність та візуальна привабливість інтерфейсу.
 - 2.2 Наявність інструкцій, методичних посібників (посібники та методичні матеріали мають велике значення при впровадженні середовища програмування в освітній процес та для самостійного вивчення мови).
 - 2.3 Багатомовний інтерфейс (не всі студенти володіють іноземними мовами на достатньому рівні, тому відсутність потрібної мови інтерфейсу може вплинути на як процес виконання завдань так і на засвоєння нового матеріалу: студенти почнуть відволікатись на розумову діяльність, що не пов'язана з конкретним завданням).
 - 2.4 Наявність безкоштовної версії застосунку (в сучасних економічних умовах, при виборі певного програмного продукту, ціна відіграє значну роль, адже не всі заклади освіти та не всі студенти можуть придбати дорогі засоби програмування).
3. Технологічні характеристики середовища програмування.
 - 3.1 Мультиплатформність (можливість встановити середовище програмування на комп'ютери з різними операційними системами).
 - 3.2 Здатність працювати на портативних пристроях (наявність мобільної версії середовища програмування дасть користувачам можливість працювати і будь-який момент часу та в будь-якому місці).
 - 3.3 Наявність засобів для програмування популярних робототехнічних конструкторів (даний критерій буде актуальним для організації позаурочної діяльності учнів).
 - 3.4 Наявність відкритої ліцензії (доступ до вихідних кодів середовища дозволить ентузіастам його вдосконалювати та доопрацьовувати).
 - 3.5 Наявність підтримки / розвитку середовища.

Зрозуміло, що за ці критеріями не забезпечать можливість визначити «кращу» мову програмування. Будь-яке середовище, якщо воно використовується та підтримується розробником чи ентузіастами, в чомусь перевершує аналоги і краще підходить для завдання, яке потрібно вирішити.

На сьогодні існує велика кількість блокових візуальних середовищ та мов програмування (Flow-chart, UML, Workflow, SDL, ДРАКОН, Google BLOCKLY, Scratch, Alice

та ін). Проаналізуємо найбільш поширені мови блокового (візуального) програмування за цими критеріями.

Середовище NXT-G. Середовище програмування NXT-G – засіб програмування, що поставляється в комплекті з конструктором Lego Mindstorms NXT (<https://www.lego.com/>). NXT-G опирається на систему LabVIEW, тому «успадковує» мову потоків даних G. Однією з переваг NXT-G є можливість створювати підпрограми та використовувати її як нові блоки в розроблюваних програмах.

Основний недолік NXT-G – обмежена підтримка математичних виразів: вони, як і вся програма, складаються з блоків. Наявні блоки читання і запису значення в змінну, блок, що зчитує значення константи, блоки, для зчитування показів сенсорів та блоки елементарних арифметичних операцій та ін.. Це означає, що для того, щоб запрограмувати нескладну формулу, потрібно зобразити блоками дерево розбору виразу, що задає цю формулу. Написана на текстовій мові програмування (наприклад, C чи C#) така програма міститиме кілька рядків коду, тоді як на NXT-G – займе декілька екранів та міститиме безліч потокових зав'язків та буде досить складною для сприйняття. Таким чином, одному із зазначених вище критеріїв – придатності використанні на початковому етапі вивчення програмування – NXT-G не відповідає. В основному через це NXT-G і не отримала широкого поширення.

До мінусів програми можна віднести: відсутність засобів налагодження; відсутність можливості генерації текстового коду; відсутність україномовного інтерфейсу не існує. Однією з переваг є те, що при купівлі конструктора, розробник надає можливість завантажити безкоштовну версію.

Scratch. Впродовж останніх років досить популярною є мова та середовище програмування Scratch. В рейтингу мов програмування, що включає такі популярні мови як Python, JavaScript, Java, PHP, SQL Scratch вже кілька років поспіль займає одинадцяте місце [11]. Це можна пояснити потребою як педагогічної спільноти в цілому, так і самих дітей в засобі для «думання», дослідження і самовираження.

Scratch (<https://scratch.mit.edu/>) – мультиплатформне візуальне середовище програмування з відкритим вихідним кодом. Створене в Масачусетському технологічному інституті, США в 2007 році, для навчання школярів основам інформатики.

«Написання» програми здійснюється за допомогою з'єднання блоків, що нагадують елементи мозаїки. Крім того Scratch дозволяє малювати та програмувати графічні об'єкти (спрайти).

Scratch перекладений на 70 мов світу в тому числі і на українську. В мережі Інтернет є безліч прикладів розроблених проектів та обговорень як цих проектів так і особливостей використання проектів та самого середовища в освітньому процесі та для самостійного вивчення; існує світове співтовариство Scratch, що дає можливість подивитися вже готові проекти та залишити свій відгук про них, викласти на обговорення власні розробки, поділитися своїм баченням та ідеями щодо подальшого розвитку середовища.

До недоліків можна віднести: відсутність можливості згенерувати текстовий код за візуальною моделлю; відсутність підтримка масивів; відсутність засобів автоматичної перевірки коректності рішення завдань.

Враховуючи зазначене вище можна зробити висновок, що Scratch добре підходить для вивчення програмування загалом та програмування роботів в молодших і середніх класах та може бути використаний не лише на уроках інформатики в початкових класах, а й для підготовки вчителів початкової школи до професійної діяльності [12] та вчителями на уроках математики, іноземної мови, природничих дисциплін [13], [14] та ін.

Існує широкий спектр додатків і середовищ, що базуються Scratch та дозволяють програмувати різноманітні робототехнічні системи. Спільними для додатків є: легкість освоєння, візуально привабливий інтерфейс, відкритість вихідного коду, наявність безкоштовних версій, можливість налагодження віддаленого управління роботом з комп'ютера та завантаження коду для автономного виконання.

Середовище «Snap!». Найбільш популярною модифікацією середовища програмування SCRATCH є «Snap!» (<https://www.snap.com>). «Snap!» це виключно браузерний додаток, що не потребує встановлення на комп'ютер користувача. Середовище «Snap!» написане мовою JavaScript, тому може впливати на комп'ютер користувача, але цей вплив досить обмежений.

Основною характеристикою та значною перевагою цього середовища перед Scratch, є розширені можливості створення власних блоків. Крім того до позитивних особливостей варто віднести те, що є можливість створювати двовимірні масиви, функції є об'єктами першого класу (це дозволяє передавати функцію як аргумент для іншої функції), спрайт (головний герой і виконавець алгоритму) не є об'єктом першого класу та може бути переданий в якості аргументу. Модифікації «Snap!» надають можливості для програмування Arduino, моделювання графів та малювання складних фігур.

Графічний редактор, вбудований в «Snap!», поступається своїм функціоналом Scratch: в ньому відсутня можливість обробки векторних зображень та обмежені можливості опрацювання растрових (немає таких інструментів як штамп, масштабування, поворот). Редактор звуку відсутній повністю. Ще одним значним недоліком є відсутність україномовної версії продукту.

«Snap!» може бути запущений та працюватиме практично у всіх браузерах.

Проекти створені в Scratch, можна імпортувати в «Snap!» за допомогою онлайн-сервісу <http://djdolphin.github.io/Snapin8r2/>.

Google Blockly. Google Blockly (<https://developers.google.com/blockly>) – відкрите середовище програмування, розроблене компанією Google під ліцензією Apache License 2.0. Blockly з самого початку орієнтований на інтернаціоналізацію, його перекладено на десятки мов, в тому числі і українську.

Для створення програми використовуються вбудовані леґо-блоки, що впорядковуються в логічній послідовності, яка забезпечує можливість реалізувати потрібну алгоритмічну конструкцію програми. Blockly оперує двома видами об'єктів: блоками та величинами. Блоки-оператори, з яких складаються конструкції, процедури, функції. Вставки призначені для оформлення констант, виразів і виклику функцій. Процес створення програми дуже схожий на складання головоломки. Крім графічних блоків, середовище надає можливість генерації та виконання текстового коду.

Графічний редактор частково контролює типи об'єктів, не дозволяючи використовувати некоректні значення величин та величини невідповідного типу в елементарних ситуаціях. Базовий набір керуючих конструкцій в ядрі Blockly, є стандартним для процедурних мов програмування та включає різні види циклів та умовних операторів.

Особливостями середовища Google Blockly, що вигідно виділяють його серед інших подібних середовищ, є можливість створювати процедури і функції з параметрами, наявність вбудованих алгебраїчних та тригонометричних функцій, наявність підтримки логічних виразів, наявність процедур для роботи з текстом і списками. Всі змінні – глобальні. Blockly працює з такими типами даних як: числа, рядки.

До особливостей цієї мови програмування можна віднести: сильний акцент на візуальній складовій та можливість компіляції програми на текстову мову

програмування на вибір, що створює умови для реалізації різних підходів щодо подальшого програмування. Окремі частини програми можуть швидко створюватися на Google Blockly, потім, після перетворення програми з графічного вигляду в текстовий, вихідний код такої програми можна доопрацювати вже на текстовій мові, або навіть вставити в уже готовий проект з метою його розширення. Логіку програми можна перевести в JavaScript, Dart, Python, XML.

Розробка програми здійснюється безпосередньо в браузері, в процесі роботи її проміжні результати зберігаються і знову завантажуються у потрібному форматі.

Google Blockly підтримує не тільки звичайні цикли і розгалуження, а й рекурсію. Область застосування – невеликі сервісні скрипти, що містять складну логіку та багаторазово вкладені логічні і циклічні конструкції.

Дане середовище може бути корисним не тільки для на початковому етапі вивчення програмування, а й для написання макетів і міні-програм в незнайомому інтерфейсі середовища програмування або на незрозумілій мові [15].

Alice. Alice (<http://www.alice.org>) – вільне і відкрите об'єктно-орієнтоване середовище програмування для навчання з інтегрованим середовищем розробки. Alice призначена для створення комп'ютерної анімації з використанням 3D-моделей. Це інноваційне блочне середовище програмування, що дозволяє легко створювати анімації, інтерактивні розповіді або програмувати прості ігри в 3D.

Основними особливостями є:

- анімація дозволяє користувачам програмувати соціальну взаємодію між персонажами;
- підручник на основі розповіді знайомить користувачів з програмуванням через створення сюжету;
- галерея 3D-персонажів з користувацькою анімацією дозволяє «оживити» ідеї та історії.

Alice містить у собі класи, що складаються із методів, рекурсії, змінних. Це дає можливість розробляти розробку програмного забезпечення, із графікою й візуалізацією, дизайн інтерфейсу, паралелізму. Об'єктно орієнтовані механізми дизайну й програмування можуть стати важливим елементом у таких курсах. Багатий зміст може бути представлений за допомогою візуалізації й анімації [16].

Alice використовує візуальні методи навчання програмуванню, які взаємодіють з інтерфейсом користувача. Інтегроване середовище розробки (IDE) підтримує дві мови програмування: Alice і Java та дає можливості для ознайомлення з основами об'єктно-орієнтованого програмування та базовими алгоритмічними структурами (слідування, розгалуженнями, циклічними). Процес та результат програмування в середовищі Alice має наочний характер – після складання програми можна відразу спостерігати результати виконання у вигляді тривимірного анімованого об'єкту.

Використовуючи середовище програмування Alice, можна не лише створювати окремі тривимірні об'єкти, але й розробляти віртуальні світи, що відтворюють задані поведінкові сценарії. Крім того, розробники передбачили можливість додавання нових 3D-об'єктів до колекції, створюючи їх за допомогою вбудованого редактора тривимірної графіки. Це відкриває шлях до створення складних комп'ютерних ігор та анімаційних фільмів-додатків в середовищі Alice. Крім того, програми можуть бути конвертовані в мову програмування Java, що дозволяє при потребі перейти від блокового інтерфейсу до текстового програмування.

Таблиця 1.

Порівняння середовищ програмування

| Назва середовища програмування / Критерії | NXT-G | Scratch | Snap! | Blockly | Alice |
|--|----------------|--|-----------------------------|-------------------------------|--|
| Можливості середовища та мови програмування | | | | | |
| Підтримка запису формул в математичному вигляді | ± | ± | ± | ± | ± |
| Підтримка основних математичних та логічних операцій | + | + | + | + | + |
| Підтримка алгоритмічних конструкцій | + | + | + | + | + |
| Можливість генерації текстового коду | - | - | - | JavaScript, Dart, Python, XML | Java |
| Можливість використання середовища на початковому етапі вивчення мов програмування | | | | | |
| Зручність, простота, сучасність та візуальна привабливість інтерфейсу | ± | + | + | + | ± |
| Багатомовний інтерфейс (наявність україномовного інтерфейсу) | - | + | - | + | - |
| Наявність інструкцій, методичних посібників | + | + | ± | ± | ± |
| Наявність безкоштовної версії застосунку | + | + | + | + | + |
| Технологічні характеристики середовища програмування | | | | | |
| Наявність онлайн / офлайн версії | - / + | + / + | + / - | + / + | - / + |
| Мультиплатформність | Windows, MacOS | Windows, Linux, MacOS | - | - | Windows, Linux, MacOS |
| Здатність працювати на портативних пристроях | - | + | - | - | - |
| Наявність засобів для програмування популярних робототехнічних конструкторів | Lego NXT | Lego NXT, Lego EV3, Lego WeDo, Arduino | Lego NXT, Lego EV3, Arduino | Arduino | Arduino |
| Наявність відкритої ліцензії | - | + | + | + | + |
| Наявність підтримки / розвитку середовища | - / - | + / + | + / + | + / + | + / - (остання версія вийшла в 2013 році) |

Висновки

Після детального аналізу середовищ візуального блокового програмування, відповідно до визначених критеріїв, можна з впевненістю сказати, що їх використання під час підготовки майбутніх вчителів початкових класів в галузі ІКТ, а також на перших етапах вивчення мов програмування, є доцільним та ефективним. Їх

впровадження в освітній процес дозволить піднятися на вищий рівень абстракції і сприятиме уникненню кастовості та ізольованості в галузі програмування, а це, в свою чергу, зробить процес входження в дану галузь більш легким та доступним для широкого кола користувачів. Впровадження візуальних блокових середовищ програмування в процес підготовки майбутніх вчителів початкових класів сприятиме створенню умов для підвищення якості освіти шляхом впровадження нових педагогічних методик та сучасних інформаційних технологій.

В дослідженні представлено порівняльний аналіз найбільш популярних на цей час навчальних середовищ блокового візуального програмування. Запропонована система критеріїв для їх оцінювання, а результати порівняння відібраних середовищ програмування за цими критеріями представлені в таблиці 1. З таблиці видно, що, незважаючи на певні обмеження (відсутність можливості автоматичної генерації текстового коду програми та обмежені можливості запису математичних виразів), середовище програмування Scratch задовольняє більшість критеріїв та є найбільш зручним та адаптованим для навчання майбутніх вчителів початкових класів.

До його основних переваг можна віднести:

- наявність україномовного інтерфейсу;
- підтримка базових алгоритмічних конструкцій, арифметичних та логічних операцій;
- підтримка та розвиток середовища (остання версія програми Scratch презентована в серпні поточного року);
- можливість встановлення на комп'ютер з будь-якою операційною системою, наявність online-версії, наявність мобільної версії;
- наявність відкритої ліцензії, безкоштовність та ін.;
- велика кількість посібників та методичних матеріалів як від самого розробника так і від користувачів.

Крім зазначених вище переваг варто також відмітити наявність спільноти Scratch, що надає можливості для обміну думками та ідеями, наявність вбудованого графічного редактора, систематичне проведення конференцій тощо.

Використання Scratch на лабораторних заняттях зі студентами спеціальності 016 Початкова освіта відіграє важливу роль в їх професійній підготовці. Результати лабораторної роботи за темою, присвяченою вивченню середовища програмування Scratch та можливостей його використання в освітньому процесі початкової школи, свідчать про хороший рівень оволодіння майбутніми вчителями досліджуваним засобом навчання та індивідуальними педагогічними технологіями, зокрема технологіями групової діяльності та проблемного навчання.

Враховуючі зазначене вище Scratch може бути рекомендований закладам вищої освіти як зручний та доцільний засіб розвитку інформаційно-цифрової компетентності майбутніх вчителів початкових класів.

Список використаних джерел

1. Про затвердження Державного стандарту початкової освіти : Постанова Каб. Міністрів України від 21.02.2018 р. № 87 : станом на 6 жовт. 2020 р. URL: <https://zakon.rada.gov.ua/laws/show/87-2018-п>.
2. Про затвердження професійного стандарту за професіями "Вчитель початкових класів закладу загальної середньої освіти", "Вчитель закладу загальної середньої освіти", "Вчитель з початкової освіти (з дипломом молодшого спеціаліста)" : Наказ М-ва розвитку економіки, торгівлі та сіл. госп-ва України від 23.12.2020 р. № 2736-20. URL: <https://zakon.rada.gov.ua/rada/show/v2736915-20>.

3. Базурін В. М. Середовища програмування як засіб навчання учнів основ програмування. *Інформаційні технології і засоби навчання*. 2017. Т. 59, № 3. С. 13–27. URL: <https://doi.org/10.33407/itlt.v59i3.1601>.
4. Вдовичин Т. Я., Лазурчак Л. В. Навчання основ програмування студентів фізико-математичного профілю. *Інформаційні технології в освіті*. 2017. № 31. С. 32–45. URL: <http://ite.kspu.edu/index.php/ite/article/view/60>.
5. Когут У. П. Класифікація та критерії вибору програмних засобів для фундаменталізації підготовки бакалаврів інформатики з інформатичних дисциплін. *Інформаційні технології в освіті*. 2012. № 11. С. 88–97. URL: <http://ite.kspu.edu/index.php/ite/article/view/408>.
6. Learnable programming: Blocks and beyond / D. Bau et al. *Communications of the ACM*. 2017. Vol. 60, no. 6. P. 72–80. URL: <https://doi.org/10.1145/3015455>
7. Pencil code: block code for a text world / D. Bau et al. *IDC '15: Proceedings of the 14th International Conference on Interaction Design and Children*, Boston Massachusetts. New York, NY, USA, 2015. P. 445–448. URL: <https://doi.org/10.1145/2771839.2771875>
8. Bottoni P., Ceriani M. Using blocks to get more blocks: Exploring linked data through integration of queries and result sets in block programming. *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*, Atlanta, GA, USA, 22 October 2015. 2015. P. 99–102. URL: <https://doi.org/10.1109/blocks.2015.7369012>
9. Developing Educational Computer Animation Based on Human Personality Types / S. Musa et al. *European Journal of Contemporary Education*. 2015. Vol. 11, no. 1. P. 52–71. URL: <https://doi.org/10.13187/ejced.2015.11.52>
10. Яценко О. І., Яценко О. С. Критерії та показники добору середовища програмування для розвитку інформаційно-комунікаційної компетентності майбутнього вчителя початкової школи. *Актуальні питання сучасної інформатики* : матеріали доп. IV Всеукр. науково-практ. конф. з міжнар. участю «Сучасні інформаційні технології в освіті та науці», вип. VII, м. Житомир, 7–8 листоп. 2019 р. / ред. Я. Б. Сікора. Житомир, 2019. С. 105–108. URL: <http://eprints.zu.edu.ua/31178/>
11. TIOBE Index for August 2023. TIOBE. URL: <https://www.tiobe.com/tiobe-index/>
12. Корецька В., Шлянчак С. Використання середовища Scratch у процесі підготовки майбутніх вчителів. *Vzdelávanie a spoločnosť II* : збірник матеріалів конференції, Prešov. 2017. С. 161–172. URL: <https://www.pulib.sk/web/kniznica/elpub/dokument/Bernatova9>.
13. Гриценко М. О. Скретч-технології для створення навчальних проєктів з фізики. *Комп'ютер у школі та сім'ї*. 2010. № 7. С. 38–41.
14. Мар'єнко М. В., Борисюк І. Ю. Гейміфікація освітнього процесу під час вивчення дисциплін природничо-математичного циклу учнями ЗЗСО. *Фізико-математична освіта*. 2020. № 4(26). С. 72–78. URL: <https://doi.org/10.31110/2413-1571-2020-026-4-013>.
15. Герасименко Н. В. Візуальне програмування з Google Blockly як основа знань для майбутньої професії програміста. *Електронні інформаційні ресурси: створення, використання, доступ* : збірник матеріалів Міжнар. науково-практ. інтернет конф., м. Вінниця, 24–25 жовт. 2016 р. 2016. С. 142–148.
16. Переяславська С. О. Застосування 3D-середовища програмування Alice при викладанні Java-технологій у вищому навчальному закладі. *Інформатика та системні науки (ІСН-2016)* : збірник матеріалів VII Всеукр. науково-практ. конф. за міжнар. участю, м. Полтава, 10–12 берез. 2016 р. 2016. С. 231–233.