Сах Ю. Аналіз сучасних засобів візуального програмування для шкільної інформатики: класифікація, можливості, перспективи використання. *Освіта. Інноватика. Практика,* 2024. Том 12, № 10. С. 133-140. https://doi.org/10.31110/2616-650X-vol12i10-020.

Sakh Yu. Analiz suchasnykh zasobiv vizualnoho prohramuvannia dlia shkilnoi informatyky: klasyfikatsiia, mozhlyvosti, perspektyvy vykorystannia [Analysis of contemporary visual programming tools for school informatics: classification, capabilities, and prospects for use]. *Osvita. Innovatyka. Praktyka – Education. Innovation. Practice,* 2024. Vol. 12, No 10. S. 133-140. https://doi.org/10.31110/2616-650X-vol12i10-020.

УДК 37.018.43:004.91(045)
DOI: 10.31110/2616-650X-vol12i10-020

Юрій САХ
Черкаський національний університет імені Б. Хмельницкого, Україна
https://orcid.org/0000-0003-0708-3497
sah.yuriy@gmail.com

АНАЛІЗ СУЧАСНИХ ЗАСОБІВ ВІЗУАЛЬНОГО ПРОГРАМУВАННЯ ДЛЯ ШКІЛЬНОЇ ІНФОРМАТИКИ: КЛАСИФІКАЦІЯ, МОЖЛИВОСТІ, ПЕРСПЕКТИВИ ВИКОРИСТАННЯ

Анотація*. У статті проаналізовано сучасні тенденції розвитку освіти в Україні, зокрема акцент на формування цифрової грамотності у школярів, зокрема через вивчення інформатики за допомогою візуального програмування. Піднято питання недостатньої опрацьованості методології викладання інформатики для учнів різного віку та рівня підготовки, що призводить до формування "цифрового розриву" — ситуації, коли деякі учні мають обмежений доступ до сучасних технологій і відповідних навчальних матеріалів. Визначено важливість уроків інформатики як ключового компонента цифрового навчання та впливу, який вони можуть мати на формування навичок, необхідних для ефективного використання технологій у різних сферах життя.*

У статті виокремлено необхідність адаптації змістового наповнення уроків інформатики до нових вимог цифрової освіти. Зокрема, акцентовано увагу на важливості інтеграції візуальних мов програмування та блочних середовищ, які можуть значно спростити процес навчання програмуванню. Такі інструменти є доступними й зручними для учнів, що дозволяє формувати основи алгоритмічного мислення і розвивати креативні здібності без необхідності знати складний синтаксис текстових мов програмування.

Досліджено сучасні приклади таких візуальних мов програмування, як Scratch, Blockly, і Code.org, що довели свою ефективність у навчанні дітей основам програмування. Окрім цього, проаналізовано використання блочних середовищ як альтернативи традиційному кодуванню для учнів молодших та середніх класів, що дозволяє створювати інтерактивні проекти, ігри та анімації. Також підкреслено важливість удосконалення програмних засобів для підвищення цифрової грамотності учнів, зокрема через використання адаптованих інструментів для різних рівнів навчання.

Автор наголошує на актуальності розробки спеціалізованих підходів до навчання інформатики, які мають на меті забезпечити усіх учнів необхідними цифровими навичками для успішного функціонування в сучасному світі.

Ключові слова: *цифрова грамотність; уроки інформатики; візуальне програмування; блочні середовища програмування; Scratch; Logo; Blockly; методологія навчання.*

Yurii SAKH
The Bohdan Khmelnytsky National University of Cherkasy, Ukraine
https://orcid.org/0000-0003-0708-3497
sah.yuriy@gmail.com

ANALYSIS OF CONTEMPORARY VISUAL PROGRAMMING TOOLS FOR SCHOOL INFORMATICS: CLASSIFICATION, CAPABILITIES, AND PROSPECTS FOR USE

Abstract. *The article analyzes current trends in the development of education in Ukraine, with a particular focus on the formation of students' digital literacy through the study of informatics using visual programming tools. The issue of insufficiently developed methodologies for teaching informatics to students of different ages and skill levels is raised, as this leads to the formation of a "digital divide" — a situation where some students have limited access to modern technologies and relevant learning resources. The importance of informatics lessons as a key component of digital education is emphasized, as well as their role in developing the skills necessary for the effective use of technologies in various spheres of life.*

The article highlights the need to adapt the content of informatics lessons to the new requirements of digital education. In particular, attention is drawn to the importance of integrating visual programming languages and block-based environments, which can significantly simplify the process of learning programming. Such tools are accessible and user-friendly for students, allowing them to develop algorithmic thinking and creativity without the need to master the complex syntax of text-based programming languages.

Modern examples of such visual programming languages as Scratch, Blockly, and Code.org are analyzed, all of which have proven effective in teaching children the fundamentals of programming. Furthermore, the use of block-based environments as an alternative to traditional coding for primary and secondary school students is examined, as it allows the creation of interactive projects, games, and animations. The paper also emphasizes the importance of improving educational software tools to enhance students' digital literacy, particularly through the use of adaptive instruments designed for different learning levels.

The author underlines the relevance of developing specialized approaches to teaching informatics aimed at ensuring that all students acquire the necessary digital skills for successful functioning in the modern world.

Keywords: *digital literacy; informatics lessons; visual programming; block-based programming environments; Scratch; Logo; Blockly; teaching methodology.*

Постановка проблеми. В умовах трансформації освіти питання пошуку та застосування нових технологій навчання стає особливо актуальним. Використання творчих технологій в освіті в Україні все ще не поширене, багато викладачів та експертів у галузі освіти визнають важливість використання творчих технологій для підтримки розвитку творчих навичок.

Освіта потребує структури, щоб допомогти учням та викладачам розвивати навички творчого мислення, які охоплюють дисципліни та використовують технологічні інструменти для творчих рішень та результатів.

Аналіз останніх досліджень і публікацій. У сучасних дослідженнях візуальне програмування (Visual Programming Languages, VPL) стабільно розглядають як інструмент, що знижує поріг входу у програмування й підтримує розвиток обчислювального мислення (абстракція, декомпозиція, алгоритмізація), особливо на рівні початкової та середньої школи. Узагальнювальні огляди й емпіричні роботи щодо Scratch/Blockly фіксують позитивний вплив на мотивацію та базові алгоритмічні вміння, зокрема у проєктно-орієнтованому навчанні (Stewart, 2022; Putra, 2025).

Окремий напрям демонструє результативність поєднання блокового програмування з освітньою робототехнікою: аналіз цифрової компетентності вчителів у такому контексті вказує на дієвість практик, де візуальні середовища інтегрують із фізичними пристроями, сенсорами та простими алгоритмами керування. (Sánchez-Rivas, 2023).

Водночас, помітним є запит на методики переходу від блокових до текстових мов. Недавні дослідження підкреслюють переваги «подвійної модальності» (паралельні подання блок↔код) над одномодовими середовищами, що підсилює розуміння синтаксису та семантики текстового коду; наявні також систематичні огляди, що окреслюють прогалини й напрями подальших досліджень у підтримці цього переходу. (Iusuf, 2023; Strong, 2025; Seralidou, 2021).

Ефективність впровадження VPL значною мірою залежить від цифрової компетентності педагога та узгодження навчального дизайну з рамками цифрових компетентностей. Європейський фреймворк DigCompEdu пропонує спільну мову й орієнтири для проєктування підвищення кваліфікації вчителів, що прямо релевантно до інтеграції VPL у шкільні курси інформатики. (Redecker & Punie, 2017).

Ще один консенсус напряму — когнітивний дизайн навчальних матеріалів. Принципи мультимедійного навчання (сегментація, сигналізація, близькість, уникнення надлишку) й надалі виступають методологічною основою для конструювання інтерфейсів та завдань у VPL, допомагаючи знижувати екстране когнітивне навантаження. (Mayer, 2019; Mayer & Fiorella, 2022).

Підсумовуючи, наявна література підтверджує: ефективність використання VPL для стартової підготовки та мотивації. Інтеграція з робототехнікою посилює міжпредметність і практичність, також для стійких результатів критичними є фахова підготовка вчителя (DigCompEdu) і когнітивно обґрунтований дизайн.

Метою статті є проведення всебічного аналізу сучасних засобів візуального програмування, їхньої класифікації та методів використання.

Методологія дослідження базується на аналізі сучасних педагогічних підходів до викладання інформатики та вивченні особливостей формування цифрової грамотності в учнів. Порівняно функціональні можливості та дидактичний потенціал різних візуальних середовищ програмування, таких як Scratch, Logo, Blockly, MIT Inventor Code Lab, Zenbo Lab та Tynker. Визначено ключові вимоги до середовищ програмування для навчання школярів, а саме доступність на різних пристроях, вбудована підсистема документування, можливість самостійного навчання та наявність механізмів для обміну кодом. Запропоновано використання сучасних середовищ візуального програмування для підвищення зацікавленості учнів, якості знань та ефективності курсу інформатики.

Виклад основного матеріалу дослідження. Цифрова компетентність громадян, визначена в документі DigComp 2.2 (The European Digital Competence Framework for Citizens), структурується за п'ятьма основними напрямками: технологічний, дослідницький, моделювальний, методологічний та алгоритмічний. Ці навички та компетентності є ключовими для 21 століття (S.Ren, 2022).

На сьогоднішній день методологія формування цифрової грамотності саме для учнів залишається недостатньо опрацьованою. Уроки інформатики, будучи одним із ключових компонентів цифрового навчання, мають значний потенціал у подоланні цифрового розриву. Однак вони вимагають адаптації змістового наповнення, форм і засобів навчання з урахуванням індивідуальних особливостей учнів.

Саме на уроках інформатики учні опановують основи цифрової грамотності, вивчаючи, як користуватися комп'ютером, працювати з текстом та графікою, ефективно шукати інформацію та безпечно спілкуватися в інтернеті.

Застосування ясності на уроках інформатики базується на таких принципах:

- систематичність та послідовність (відповідність певній послідовності у вивченні навчального матеріалу та поступового оволодіння об'єктивним змістом);
- доступність (випливає з вимог вікових характеристик учнів; обсяг та зміст навчального матеріалу вибираються на основі розумового розвитку школярів та існуючих запасів знань, навичок);

- свідомість, діяльність, незалежність та сила асиміляції (організація цілі спрямованого сприйняття досліджуваного матеріалу, його розуміння, обробка та застосування);
- посилення прикладної орієнтації (використання прикладів реальних ситуацій, спілкування, навчання з життям);
- знання (створення умов для розуміння та запам'ятовування в процесі навчання, навчальний матеріал).

Концепція ясності нерозривно пов'язана з методами візуалізації, оскільки вона заснована на сприйнятті інформації через органи чуття.

У науковій та методологічній літературі існують різні визначення видимості: як певний об'єкт (засіб візуалізації); як певна властивість (видимість реальних об'єктів, явищ, мислення); як певна діяльність людини (зорове сприйняття та використання). У педагогічному словнику видимість вважається «дидактичним принципом, згідно з яким навчання побудовано на конкретних зображеннях, які безпосередньо сприймаються студентами» (E.Coronado, 2020).

Важливо розділити поняття «візуального» та «графічного», оскільки розвиток інформаційних технологій відбувається швидко, тому в майбутньому існує можливість розширення об'єктів, з якими програміст буде взаємодіяти. Ми говоримо про введення інших органів чуттів у роботу, крім зору, створення більш складних технічних систем маніпуляцій людини та комп'ютера.

У програмуванні блоку замість набору текстових команд використовуються багатоколінні елементи, при підключенні яких, підтримуються робочі сценарії. Це дозволяє візуально консолідувати змінні, встановлені членом коду, а це означає, що краще вивчити саму суть складеного алгоритму.

Історично візуальні засоби для передачі інформації є однією з давніх прийомів, якими користується людство. Видимість застосовується з примітивних часів, це позначається збереженою касовою картиною. Сьогодні символи також використовуються навколо нас абсолютно різних явищ: система іконок є однією з найбільш універсальних для сучасного суспільства. Так, багато значущих систем розпочалися з візуалізації об'єктів, які входять до нього, через природу зображень. Програмування також має тенденцію до спрощення та впровадження методів видимості за рахунок активного використання органів зору.

Візуальне програмування - головна методика в галузі інформаційних технологій. З моменту розвитку та активної ефективності першої такої мови. Розробка, створена в 2007 році Мітчелом Резником та співробітниками Технологічного університету Массачусетса, встановила новий раунд розвитку галузі навчального програмування. Принцип написання сценаріїв за допомогою маніпуляцій з графічно представленими об'єктами був проривом і встановлений як ефективний інструмент. Дослідження показують, що студенти продемонстрували високий ступінь навчання для складання алгоритмів за допомогою візуальних засобів, що доводить виправдання введення візуалізації в навчальний процес у розглянутому аспекті.

Головна проблема в навчанні програмування полягає в тому, що письмовий код не є матеріальним, його не можна побачити. Для наступного, особливо важко впоратися з новим концептуальним апаратом без можливості повністю реалізувати принципи алгоритму, оскільки без досвіду це може бути очевидним, як це працює: текстові програми дають лише відповідь як результат роботи.

Сікора Я.Б. (2008) зазначає, що візуальне програмування (VPL) – це технологія, яка дозволяє створювати код програми за допомогою графічних елементів, а не тексту.

Ковтанюк М.С., Криворучко І.І. (2024) вказують, що візуальне програмування – це парадигма програмування, яка динамічно розвивається, та використовує візуальні та інтерактивні середовища для створення програмного коду.

За Базуріним В. М. (2017) візуальне програмування розуміється як спосіб створення програми, в якій замість того, щоб писати текстовий код, використовуються графічні об'єкти. Деякі представники наукової спільноти вважають візуальне програмування наступним етапом або генерацією розвитку текстових мов. Це перспективний напрямок, який все частіше привертає увагу розробників, у зв'язку з яким сьогодні існують різноманітні інструменти програмування, які відрізняються залежно від сфери та цілей застосування. Отже, неможливо провести межу між мовами графічного програмування та інструментами візуальної розробки, що використовуються як допоміжні інструменти в розробці інтерфейсів.

На наше переконання, візуалізація - це інструмент, заснований на метафорі, тобто передачі певного зображення для формування концепції нового об'єкта, процесу чи явища, а також для встановлення зв’язку з уже відомим. Ідея такої кореляції необхідна для врахування таких аспектів:

- синтаксичне та семантичне позначення мови;
- прагматик мови;
- навчальний компонент, тобто, як концепція чи навички програмування
- форма внаслідок використання метафори.

На теперішньому етапі розвитку візуальних мов програмування, вибору та формування метафору проводили за схемою:

- вибір графічної моделі;
- вибір відповідності між програмою та графікою, що одночасно визначає поведінку графічної моделі;
- визначення набору дій для взаємодії користувача з графікою.

У візуальних мовах програмування метафори, про які йдеться, завжди базуються на основних парадигмах та програми програмування. Реалізуються поняття змінних, основні алгоритмічні конструкції, функції. Після маніпуляцій з графічними представленнями метафор (блоків) текст програми формується всередині комп'ютера. Ця система управління дозволяє спростити процес налагодження, оскільки внесені зміни легко затримуються, а з'єднання між ними логічно простежуються, що допомагає в процесі формування та розвитку алгоритмічного мислення.

Мови візуального програмування можна класифікувати відповідно до різного розпізнавання. Наприклад, у галузі основного застосування: навчання, галузі, видів дослідження. Відповідно до основного аспекту моделювання: мови опису структури програмних систем; мови опису; мови опису поведінки (процесів); мови для створення графічного інтерфейсу. Такі класифікації також пропонуються:

- мови, засновані на об'єктах, коли середовище візуального програмування забезпечує графічні або символічні елементи, за допомогою яких можна маніпулювати інтерактивними об'єктами відповідно до деяких правил (Scratch, Pictorian, Blockly, MIT Inventor, Kodu Game Lab).
- редактори форм, які дозволяють миші розміщувати частини користувальницького інтерфейсу та налаштувати їх властивості (Visual Basic, Delphi, C ++ Builder).
- мови схем, заснованих на ідеї «фігур і ліній», де фігури (прямокутники, овали тощо) розглядаються як суб'єкти і пов'язані лініями (стрілками, дугами тощо), які є взаємозв'язками (редактори взаємозв'язків у реляційній базі даних).

Щоб покращити шкільну підготовку, першочергове завдання, на думку С. Паперта, полягає у створенні програмних середовищ, що сприяють навчанню та розвитку обчислювального мислення через експерименти та творчість. Цей підхід розвиває креативність, логіку та навички вирішення проблем.

Мазур І.-С. В., Франко Ю. П., Козіброда С. В. (2023) у своєму дослідженні визначили такі ключові критерії для мов програмування:

- ліцензійна чистота та кросплатформність.
- зрозумілий і лаконічний синтаксис.
- відсутність надлишкових елементів, що не впливають на реалізацію алгоритму.
- підтримка різних парадигм програмування.
- придатність для розробки масштабних проєктів.
- наявність функціонального середовища розробки, зокрема візуальних інструментів для графічного інтерфейсу користувача.
- можливість розширення за допомогою бібліотек, модулів та інших засобів.

У статті Яценко О.І.,Чумак Л.М. досліджуються такі інструменти, як Alice, App Inventor, Blockly, Logo, Kodu, Scratch, Snap!, Squeak, Tynker та ДРАКОН, з метою визначення їхньої придатності для формування ІКТ-компетентності. Порівняльний аналіз показав, що Scratch та Blockly є одними з найефективніших систем для цих завдань (Яценко О.І.,Чумак Л.М., 2020).

Щоб досягти цілей у навчанні програмування, важливо розділити етапи, на яких можна інтегрувати візуальні медіа. Отже, найефективніша методика вводиться на початку при формуванні основних понять, після чого проведиться більш плавний перехід до розробки простих програм та вивчення класичних алгоритмів. Завдяки цьому процес навчання може бути організований таким чином, щоб усвідомити основне змістовне завдання - моделювання алгоритмізації.

Підготовчий етап включає:

- вивчення літератури, що містить інформацію про методологію проведення експериментального уроку.
- вивчення літератури, що містить знання про використання програми Scratch.
- підготовка завдань, які учасники уроку можуть виконувати в Інтернеті.
- колекція необхідних матеріалів для проведення уроку (урок).
- створення завдання для учасників уроку: зрозуміло, адаптовано

Розглядаючи освітній напрямок можна відокремити два проекти, а саме Google Blockly (рис. 1) та Scratch (рис. 2).

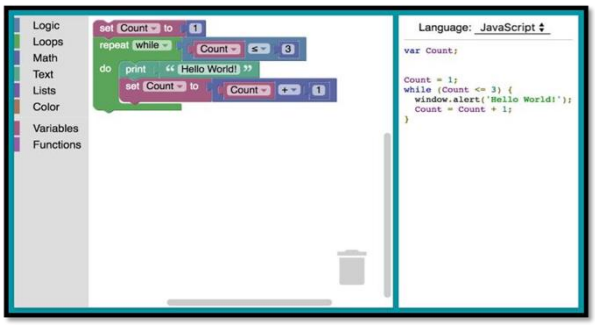

Рис. 1. Середовище Google Blockly

Компанія Asus створила навчального робота Zenbo Junior, який може програмуватися за допомогою веб-середовища Zenbo Lab https://zenbolab.asus.com/editor/?lang=en). Це середовище дозволяє навчати робототехніці та штучному інтелекту через візуальне програмування, а також підтримує програмування мовою Python.

Рис. 2. Середовище Scratch

Користувачі можуть керувати віртуальним прототипом робота, навіть якщо фізичний пристрій відсутній. Zenbo Lab призначена для STEM-навчання, але завдяки своїм можливостям Zenbo Junior також може використовуватися в інших курсах, таких як відпрацювання англійської мови або проведення вікторин. Файли, створені в Zenbo Lab, можна експортувати для роботи на Zenbo Junior без використання комп'ютера чи ноутбука

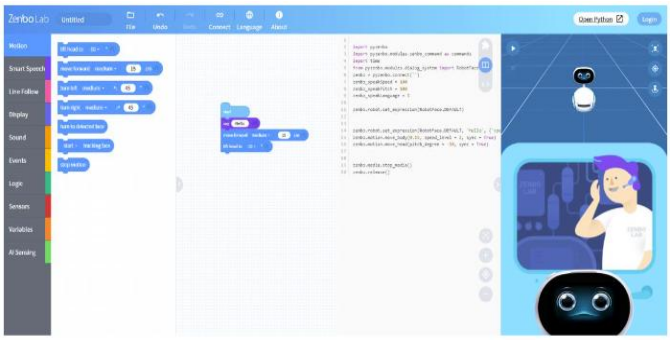

Рис. 3 Робота з Zenbo Lab

Сервіс Tynker (https://tynker.com) пропонує дітям віком від 5 років освоювати програмування за допомогою блочного візуального інтерфейсу. Навчальний процес здійснюється в ігровій формі через виконання проєктних завдань.

Рис. 4 Сервіс Tynker

Схожий підхід використовується і на сервісі CODE (https://studio.code.org), відомому проведенням акції "Час коду".

Рис. 5 Серсвіс CODE

Дацюк Г. М, Лещук С. О. (2024) зазначають, що під час уроків інформатики вчитель може застосовувати програмне середовище ScratchJr, яке забезпечують можливість створення учнями власних мультфільмів, інтерактивних ігор та анімаційних проєктів. Важливо використовувати й сервіси для створення інтерактивного контенту – такі як Wordwall, LearningApps, Kahoot.

На наше переконання існують таки перспективи для середовища програмування в шкільному курсі інформатики:

Доступність. Середовище програмування має бути доступним як на повнорозмірних персональних комп'ютерах, так і на пристроях під керуванням Android та iOS.

Документованість. Мова програмування повинна мати вбудовану підсистему документування програми, яка дозволяла б вести необхідний для розуміння непідготовленим користувачем опис будь-якого блоку програми природною мовою.

Можливість самостійного навчання. Користувачі повинні мати можливість почати програмувати, взявши за основу приклад однієї з конструкцій мови. Такий підхід дозволить зробити навчання самостійним за рахунок того, що програма може бути отримана на основі використання зразка та доведення його до необхідного стану за допомогою методу «проб і помилок» з мінімальною залученістю вчителя.

Соціальність. Мова програмування може стати популярною лише в тому випадку, якщо навколо неї збирається достатньо велика спільнота ентузіастів. З цієї причини описувана мова повинна мати механізми для обміну вихідними «кодами» програм.

Висновки і перспективи подальших досліджень. На завершення необхідно сказати, що розглянуті приклади середовищ демонструють можливість вирішення для кожної із зазначених вимог. Однак поєднання найкращих практик кожної з них є нетривіальним завданням, і, ймовірно, з цієї причини не можна спостерігати настільки ж проривної мови. Проте вже зараз можна застосовувати більш сучасні середовища програмування під час навчання школярів програмуванню, підвищуючи тим самим їхню зацікавленість, якість знань та ефективність курсу в цілому. На наше переконання критично необхідно інтегрувати ці сучасні підходи та інструменти в навчальний процес, щоб підвищити зацікавленість школярів, покращити якість їхніх знань та загальну ефективність курсу програмування. Спільне використання найкращих практик різних середовищ є складним завданням, проте вже існуючі рішення дозволяють значно покращити процес навчання.

Список використаних джерел

1. Badr, H. S. M., El-Sayyed, H., & Shalash, S. (2025). Evaluating the effectiveness of Scratch Jr and Tynker Jr in fostering foundational coding skills among kindergarten children. *International Journal of Early Years Education. Advance online publication.* https://link.springer.com/article/10.1007/s13158-025-00442-4
2. Iusuf, J. A. (2023). Easing the transition from block-based programming in Blockly to text-based programming in Python using BlockPy (Master's thesis). *Mid Sweden University.* https://www.diva-portal.org/smash/get/diva2:1778367/FULLTEXT01.pdf
3. Mayer, R. E. (2019). Multimedia learning (3rd ed.). *Cambridge University Press.* https://assets.cambridge.org/97811071/87504/frontmatter/9781107187504_frontmatter.pdf
4. Mayer, R. E., & Fiorella, L. (2022). The Cambridge handbook of multimedia learning (3rd ed.). *Cambridge University Press.* https://www.researchgate.net/publication/359588183_The_Cambridge_Handbook_of_Multimedia_Learning_3rd_ed
5. Redecker, C., & Punie, Y. (2017). European framework for the digital competence of educators: DigCompEdu. *Publications Office of the European Union.* https://publications.jrc.ec.europa.eu/repository/handle/JRC107466
6. Sánchez-Rivas, E. (2023). Teacher digital competence analysis in block programming applied to educational robotics. *Sustainability*, 16(1), 275. https://www.mdpi.com/2071-1050/16/1/275
7. Seralidou, E., Doukaki, S., & Kalles, D. (2021). Investigating the transition from block-based to text-based programming techniques in secondary education (Greece). *Proceedings/Preprint.* https://www.researchgate.net/publication/358721272_Investigating_the_Transition_from_Block-based_to_Text-based_Programming_Techniques_in_Secondary_Education_in_Greece
8. Stewart, W. H., Hansen, R., & Val, G. (2022). Analyzing computational thinking studies in Scratch programming: A review of elementary education literature. IEEE/ASEE. https://pdfs.semanticscholar.org/7e66/0f803c93386754ddae435b3c1827cf9ca818.pdf
9. Strong, G., Bell, T., & Hermans, F. (2025). Supporting learners in the transition from block-based to text-based programming: A systematic review. *Journal of Computer Languages, 103, 101252*. https://www.sciencedirect.com/science/article/pii/S2590118425000280
10. М Базурін, В. М. (2017). Середовища програмування як засіб навчання учнів основ програмування. *Інформаційні технології і засоби навчання*, *59*(3), 13–27. https://doi.org/10.33407/itlt.v59i3.1601
11. Дацюк, Г. М., & Лещук, С. О. (2024). Використання цифрових ресурсів для організації навчання дітей з особливими освітніми потребами. У *Сучасні інформаційні технології в освіті і науці: матеріали XV Всеукраїнської науково-практичної конференції для молодих учених та здобувачів освіти (25–26 квітня 2024)* (с. 61–63).
12. Ковтанюк, М. С., & Криворучко, І. І. (2024). Переваги та недоліки вебресурсів для створення мобільних застосунків з використанням елементів візуального програмування. У *Збірник тез доповідей VI Міжнародної науково-практичної конференції з економічних та гуманітарних питань. – У 2-х томах. – Т. II* (с. 227-228). ДВНЗ УДХТУ.
13. Мезур, І.-С. В., Франко, Ю. П., & Козіброда, С. В. (2023). *Технології інтернет речей (IoT) в середовищі візуального програмування Node-RED: методичні рекомендації до виконання лабораторних робіт для студентів другого (магістерського) рівня вищої освіти спеціальності 015.39 Професійна освіта (Цифрові технології)*. ТНПУ ім. В. Гнатюка.
14. Офіційний сайт розробників навчального середовища «CODE». (n.d.). Retrieved from https://studio.code.org
15. Офіційний сайт розробників навчального середовища «Scratch». (n.d.). Retrieved from http://scratch.mit.edu/
16. Офіційний сайт розробників навчального середовища «Tynker». (n.d.). Retrieved from https://tynker.com
17. Офіційний сайт розробників навчального середовища Zenbo Lab. (n.d.). Retrieved from https://zenbolab.asus.com/editor/?lang=en
18. Сікора, Я. Б. (2008). Зміст та структура поняття професійна компетентність вчителя інформатики. У *Психолого-педагогічні основи гуманізації навчально-виховного процесу в школі та ВНЗ: зб. наук. праць* (с. 148-156).
19. Coronado, E., et al. (2020). Visual Programming Environments for End-User Development of intelligent and social robots, a systematic review. *Journal of Computer Languages*, *58*, 100970. https://doi.org/10.1016/j.cola.2020.100970
20. Ren, S., et al. (2022). JSPatcher, a visual programming environment for building high-performance web audio applications. *Journal of the Audio Engineering Society*, *70*(11), 938–950. https://doi.org/10.17743/jaes.2022.0056
21. Яценко, О. І., & Чумак, Л. М. (2020). Критерії добору середовища навчання програмування для формування ІКТ-компетентності майбутніх учителів початкової школи. *Інформаційні технології і засоби навчання*, *78*(4), 219-236.

References

1. Badr, H. S. M., El-Sayyed, H., & Shalash, S. (2025). Evaluating the effectiveness of Scratch Jr and Tynker Jr in fostering foundational coding skills among kindergarten children. *International Journal of Early Years Education. Advance online publication.* https://link.springer.com/article/10.1007/s13158-025-00442-4
2. Iusuf, J. A. (2023). Easing the transition from block-based programming in Blockly to text-based programming in Python using BlockPy (Master's thesis). *Mid Sweden University.* https://www.diva-portal.org/smash/get/diva2:1778367/FULLTEXT01.pdf
3. Mayer, R. E. (2019). Multimedia learning (3rd ed.). *Cambridge University Press.* https://assets.cambridge.org/97811071/87504/frontmatter/9781107187504_frontmatter.pdf
4. Mayer, R. E., & Fiorella, L. (2022). The Cambridge handbook of multimedia learning (3rd ed.). *Cambridge University Press.* https://www.researchgate.net/publication/359588183_The_Cambridge_Handbook_of_Multimedia_Learning_3rd_ed

5. Redecker, C., & Punie, Y. (2017). European framework for the digital competence of educators: DigCompEdu. *Publications Office of the European Union.* https://publications.jrc.ec.europa.eu/repository/handle/JRC107466
6. Sánchez-Rivas, E. (2023). Teacher digital competence analysis in block programming applied to educational robotics. *Sustainability*, 16(1), 275. https://www.mdpi.com/2071-1050/16/1/275
7. Seralidou, E., Doukaki, S., & Kalles, D. (2021). Investigating the transition from block-based to text-based programming techniques in secondary education (Greece). *Proceedings/Preprint.* https://www.researchgate.net/publication/358721272_Investigating_the_Transition_from_Block-based_to_Text-based_Programming_Techniques_in_Secondary_Education_in_Greece
8. Stewart, W. H., Hansen, R., & Val, G. (2022). Analyzing computational thinking studies in Scratch programming: A review of elementary education literature. IEEE/ASEE. https://pdfs.semanticscholar.org/7e66/0f803c93386754ddae435b3c1827cf9ca818.pdf
9. Strong, G., Bell, T., & Hermans, F. (2025). Supporting learners in the transition from block-based to text-based programming: A systematic review. *Journal of Computer Languages, 103, 101252*. https://www.sciencedirect.com/science/article/pii/S2590118425000280
10. M. Bazurin, V. M. (2017). Programming environments as a means of teaching students the basics of programming. Information technologies and teaching aids, 59(3), 13–27. https://doi.org/10.33407/itlt.v59i3.1601
11. Datsyuk, G. M., & Leshchuk, S. O. (2024). Using digital resources to organize education for children with special educational needs. In Modern information technologies in education and science: materials of the XV All-Ukrainian scientific and practical conference for young scientists and students (April 25–26, 2024) (pp. 61–63).
12. Kovtanyuk, M. S., & Kryvoruchko, I. I. (2024). Advantages and disadvantages of web resources for creating mobile applications using elements of visual programming. In Collection of abstracts of the VI International Scientific and Practical Conference on Economic and Humanitarian Issues. – In 2 volumes. – Vol. II (pp. 227–228). UDHTU.
13. Mezur, I.-S. V., Franko, Yu. P., & Kozibroda, S. V. (2023). Internet of Things (IoT) technologies in the Node-RED visual programming environment: methodological recommendations for laboratory work for students of the second (master's) level of higher education in the specialty 015.39 Professional Education (Digital Technologies). V. Hnatiuk TNPU.
14. Official website of the developers of the CODE learning environment. (n.d.). Retrieved from https://studio.code.org
15. Official website of the developers of the Scratch learning environment. (n.d.). Retrieved from http://scratch.mit.edu/
16. Official website of the developers of the Tynker learning environment. (n.d.). Retrieved from https://tynker.com
17. Official website of the developers of the Zenbo Lab learning environment. (n.d.). Retrieved from https://zenbolab.asus.com/editor/?lang=en
18. Sikora, Y. B. (2008). Content and structure of the concept of professional competence of a computer science teacher. In Psychological and pedagogical foundations of humanization of the educational process in schools and universities: collection of scientific works (pp. 148-156).
19. Coronado, E., et al. (2020). Visual Programming Environments for End-User Development of intelligent and social robots, a systematic review. Journal of Computer Languages, 58, 100970. https://doi.org/10.1016/j.cola.2020.100970
20. Ren, S., et al. (2022). JSPatcher, a visual programming environment for building high-performance web audio applications. Journal of the Audio Engineering Society, 70(11), 938–950. https://doi.org/10.17743/jaes.2022.0056
21. Yatsenko, O. I., & Chumak, L. M. (2020). Criteria for selecting a programming learning environment for developing the ICT competence of future primary school teachers. Information Technologies and Learning Tools, 78(4), 219-236.