МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
Фізико-математичний факультет
Кафедра інформатики та прикладної математики

«Допущено до захисту»
В.о. завідувача кафедри
_____ Моісеєнко Н. В.
«____» _____ 2023 р.

Реєстраційний № _____
«____» _____ 2023 р.

Програмування мобільних застосунків як засіб навчання основ алгоритмізації учнів ліцеїв

Кваліфікаційна робота студентки
групи Ім-22
ступінь вищої освіти «магістр»
спеціальності 014 Середня освіта (Інформатика)
Зусмановської Юлії Олегівни

Керівник: к.пед.н.
Мерзликін Олександр Володимирович

Оцінка:
Національна шкала _____
Шкала ECTS __ Кількість балів ___

Голова ЕК _____ _____
Члени ЕК _____ _____
_____ _____
_____ _____

Кривий Ріг — 2023

ЗАПЕВНЕННЯ

Я, Зусмановська Юлія Олегівна, розумію і підтримую політику Криворізького державного педагогічного університету з академічної доброчесності. Запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Я не надавала і не одержувала недозволену допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають покликання на відповідне джерело.

Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Криворізького державного педагогічного університету ознайомлена. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

_____

(підпис)

ЗМІСТ

ВСТУП

Алгоритмічне мислення є важливою ключовою компетентністю сучасної людини. Сензитивний період для його розвитку припадає на юнацький вік, а найбільш орієнтованим на його формування шкільним предметом є інформатика, зокрема, ті її розділи, що стосуються основ алгоритмізації та програмування. При цьому навички і прийоми алгоритмічного мислення потребують формування, незалежно від рівня профілізації. Так, на базовий модуль на рівні стандарту в 10-11 класах відведено 35 годин. За такого навантаження може виявитися недоцільним вивчення символьних мов програмування, тому в нашому дослідженні ми сконцентруємося переважно на візуальних мовах програмування.

Актуальність теми. Сучасні тенденції навчання основ алгоритмізації диктують нові вимоги до створення освітнього середовища навчання учнів. Важливим стає поєднання швидкості та якості отримання дидактичних матеріалів, а також їхня гнучкість, зокрема, в тому, що стосується адаптивності до різних форм навчання (як дистанційної, так і очної). Серед сучасних перспективних технологій, що можуть застосовуватися при навчання основ алгоритмізації виділимо програмування мобільних застосунків як таке, що може мотивувати учнів ліцеїв, а також такі, що природним чином демонструють ряд концепцій, що є природними в сучасному програмуванні (об’єктно орієнтований та подієвий підходи), але яким часто приділяють недостатню увагу в шкільному курсі інформатики.

Метою кваліфікаційної роботи є дослідження ефективності застосування програмування мобільних застосунків при вивченні основ алгоритмізації учнів ліцеїв та розробка методичних рекомендацій щодо впровадження цього підходу в педагогічну практику.

Для досягнення мети були поставлені такі **завдання дослідження**:

1. Проаналізувати методику викладання основ алгоритмізації в шкільному курсі інформатики.

2. Дослідити можливості використання мобільних технологій при викладанні основ алгоритмізації в шкільному курсі.
3. Розглянути програмування мобільних застосунків як спосіб покращення рівня викладання основ алгоритмізації в шкільному курсі інформатики.
4. Виконати аналіз досвіду використання програмування мобільних застосунків при викладанні основ алгоритмізації в шкільному курсі інформатики.
5. Зробити добір засобів для використання програмування мобільних застосунків при викладанні основ алгоритмізації учням ліцеїв.
6. Провести експериментальну перевірку ефективності використання розроблених методичних рекомендацій у навчанні ліцеїстів.

Об’єктом дослідження є використання технологій програмування мобільних застосунків при навчанні основ алгоритмізації.

Предметом дослідження є методика використання технологій програмування мобільних застосунків при навчанні основ алгоритмізації та умови ефективності її використання.

Методи дослідження: теоретичний аналіз наукової літератури; методи порівняльного аналізу. Вирішення сформульованих завдань здійснювалося з використанням системного підходу в доборі матеріалу, методів аналізу та синтезу, при опрацюванні результатів експерименту використано статистичні методи аналізу даних.

Практичне значення отриманих результатів полягає у розробці методичних рекомендацій щодо застосування програмування мобільних застосунків при вивченні основ алгоритмізації учнями ліцеїв. Висновки та матеріали дипломної роботи можуть використовуватись у навчальному процесі середньої школи.

Структура роботи. Кваліфікаційна робота складається із вступу, трьох розділів, висновку, списку використаних джерел та додатків.

РОЗДІЛ 1.
ОСНОВИ АЛГОРИТМІЗАЦІЇ В ШКІЛЬНОМУ КУРСІ ІНФОРМАТИКИ

1.1. Методика викладання основ алгоритмізації в шкільному курсі інформатики

Сучасна освітня система орієнтована на формування освічених та інтелектуально розвинених особистостей з цілісним уявленням про світ, глибоким розумінням зв'язків і явищ. Випускники шкіл мають вміти ефективно взаємодіяти з оточуючим середовищем, швидко адаптуватися і виявляти компетентність у ключових сферах життя: праця, держава, сім'я, здоров'я, політика та культура. Функціональна грамотність в цих сферах визначає загальну компетентність особистості [8].

На сьогоднішній день інформатика виступає ключовим предметом, що навчає логічному мисленню та навичкам роботи з інформацією. Інформатика допомагає у триманні кроку з швидким розвитком цифрових технологій. Проте, оскільки цей розвиток швидкий, вчителям та учням іноді важко встигнути за нововведеннями, що призводить до недостатнього вивчення певних аспектів даної тематики.

Головною метою у вивчені дисципліни «інформатика» є розвиток учнівських навичок в галузі інформаційних технологій для створення комп'ютерних інформаційних продуктів. Досягнення цієї мети передбачає впровадження в навчальний процес спеціально розробленої системи завдань, які відтворюють реальні проблеми, з якими стикаються фахівці у різних сферах діяльності. Це включає в себе використання відповідних підходів до навчання. Формування інформаційних навичок серед старшокласників базується на таких принципах як науковість, системність, послідовність, зв'язок навчання з практикою, стимулювання самосвідомості учня, диференціація та індивідуалізація навчання, доступність, сила, навчання на високому рівні складності і наочність [12].

Проблему вивчення інформатики в українських школах розкрито в роботах таких науковців як Гевко І. [24], Копосов Д. [46], Донченко Я. [36], Бурдун О. [19]. Камалов Р. [42], Гурняк І. [31], Березовська Ю. [11], Абушкин Х. [1].

Вітчизняні науковці Андрєєв Д. [4], Бондаренко В. [15], Ащепкова Н. [8], Бугайчук К. [18], Голощапов А. [25] зауважують, що «технології змінюються, техніка вдосконалюється, і потрібно визнати, що найближчим часом нова сучасна техніка масово не прийде до НЗ України.

Розділ «Основи алгоритмізації та програмування» у шкільному курсі інформатики має велике значення. Цей розділ розкриває важливість алгоритмів та їх роль у зв'язку між поняттями інформація, алгоритм, комп'ютер, що визначає процес автоматичного опрацювання даних. Через приклади надається можливість формального виконання алгоритмів, демонструються елементи дій, які виконавець повинен виконати за вказівкою. Це підкреслює можливість передачі виконання формально описаного алгоритму виконавцеві-машині, або, іншими словами, автоматизації діяльності людини на основі алгоритмів. Вивчення алгоритмічної мови дає змогу ознайомити учнів із формалізованим записом алгоритмів, розширюючи їхнє уявлення про засоби описування алгоритмів. Метою вивчення алгоритмізації є формування знань та навичок учнів у сфері основних методів організації операцій і даних, а також використання базових алгоритмічних конструкцій для складання описів алгоритмів розв'язання різноманітних завдань [16].

Під час навчання основ алгоритмізації слід зосередитися на наступних аспектах:

1. Виявлення загальних закономірностей та принципів алгоритмізації. Ознайомлення з основними етапами вирішення задач за допомогою сучасних інформаційних технологій. Аналіз постановки задачі, методів формалізації та моделювання реальних процесів. Вибір виконавця, враховуючи його властивості та допустимі операції. Застосування методів та засобів формалізованих описів дій виконавця за допомогою комп'ютера. Однією з проблем, з якою стикаються викладачі при навчанні цього розділу, є поєднання консервативної алгоритмічної лінії курсу з більш

динамічними та прогресивними напрямками виконавця, формалізації та моделювання, інформаційних технологій.

2. Алгоритмізація, яка вивчає створення алгоритмів, відноситься до теоретичної інформатики через свій фундаментальний характер. З розвитком інформаційних технологій, зокрема технології програмування, виникає можливість в розділі «Основи алгоритмізації» ознайомити учнів з перспективними напрямками [13].
3. Теорія і методика викладання алгоритмізації в шкільному курсі мають охоплювати вивчення процесу навчання інформатики на всіх рівнях та в усіх формах: в школі, різних рівнях середньої освіти, самостійному вивченні інформатики, дистанційному навчанні тощо. Кожна з цих областей створює свої унікальні завдання для сучасної педагогічної науки, але особливий інтерес викликає сфера методології інформатики, яка досліджує вивчення інформатики в середній школі як частину загальної освіти в галузі інформатики [18].

Сучасний підхід до вивчення основ алгоритмізації базується на кількох ключових принципах [18]:

1. Орієнтація на використання комп’ютера: процес вивчення основ алгоритмізації спрямований на використання комп’ютера як ефективного засобу навчання.
2. Акцент на змісті алгоритму: мета вивчення полягає у виділенні змісту алгоритму, та в усвідомленні правил його побудови. Важливим аспектом є невироблення конкретних алгоритмічних мов, а розуміння їхньої ролі як лише одного із засобів формального представлення алгоритмів.
3. Розвиток операційного мислення: поняття алгоритму відіграє ключову роль у розвитку алгоритмічного мислення та усвідомлення учнями можливості автоматизації багатьох видів діяльності. Це розширює їхні уявлення про формальність та механічний характер дій при виконанні алгоритмів, що є основою для автоматизації операцій за допомогою комп’ютера.

4. Поєднання теорії та практики: опанування поняття «алгоритм» є першим етапом у формуванні в учнів уявлень про автоматичне опрацювання даних за допомогою комп'ютера. Це також передбачає врахування внутрішньо предметних зв'язків та наголос на важливості формального опису алгоритмів [24].
5. Вивчення теорії алгоритмів: поняття «алгоритм» належить до фундаментальних математичних понять і є об'єктом дослідження теорії алгоритмів.

Компоненти навчання:

– вивчання відомих алгоритмів та їх застосування;
– вивчання класичних алгоритмів;
– навчитися будувати і описувати алгоритми як з використанням, так і без використання відомих алгоритмів.

Значення алгоритмічної мови: для вирішення поставлених завдань важливо використовувати якусь алгоритмічну мову, що надає можливість описувати алгоритми за єдиним набором правил. Оволодіння цією мовою є важливим етапом навчання основ алгоритмізації.

На сучасному етапі розвитку теорії та методики вивчення основ алгоритмізації в шкільному курсі інформатики спостерігається активний розвиток. Хоча шкільна інформатика існує вже майже два десятиліття, багато завдань, пов'язаних із новітніми напрямками педагогічної науки, виникли нещодавно і ще не пройшли глибокого теоретичного обґрунтування та довгострокових експериментів. В рамках загальних цілей викладання, методика вивчення основ алгоритмізації в шкільному курсі інформатики ставить перед собою кілька основних завдань [15]:

Визначення конкретних мет цього навчання та його місця в навчальному плані середньої школи. Розробка та пропозиція раціональних методів і форм навчання для досягнення поставлених цілей. Розгляд повного спектру навчальних інструментів для основ алгоритмізації та розробка рекомендацій з їх використання в практиці вчителя.

У ряді публікацій відзначено, що на протязі тривалого періоду підготовка майбутнього вчителя інформатики страждає від недостатньої уваги, зокрема в частині методичної підготовки. Визначення змісту предмета включає загальні теоретичні аспекти викладання інформатики, комбінацію програмно-технічних засобів, а також конкретну методологію вивчення окремих тем курсу інформатики на всіх етапах навчання.

Метод викладання основ алгоритмізації є відносно молодою науковою дисципліною, але вона виникла не з нуля. Ця наука, будучи самостійною, вбирає знання з інших галузей, таких як інформатика, педагогіка , фізіологія, філософія, психологія, вікова, узагальнений практичний досвід викладання інших предметів. Викладання основ алгоритмізації в сучасному розумінні базується на інформації з різних наукових галузей, таких як біологія (вивчення біологічних самоврядувальних систем, включаючи людей та інші живі організми), історія та соціальні науки (вивчення соціальних систем), українська мова (граматика, синтаксис, семантика і т.д.), логіка (мислення, формальні операції, істина, хиба), математика (числа, змінні, функції, ознаки, дії), психологія (мислення, комунікація) [11].

У контексті глобальної цифровізації багатьох сфер людської діяльності і впливу інформатики на інші науки, можна стверджувати, що метод викладання основ алгоритмізації має тісний зв’язок з практично будь-якою галуззю науки. Це стосується особливо переходу системи загальної середньої освіти України на спеціалізовану освіту, де факультативні курси з основ алгоритмізації стануть актуальними для всіх профілів і предметів шкільної програми [21]. У цьому контексті, об’єктом навчання при викладанні основ алгоритмізації стають не лише поняття та методи інформатики, а й інші науки та їх розділи, які будуть інтегровані з інформатикою під час факультативних курсів.

Комп’ютерному вченому важливо мати орієнтацію в різних галузях знань, таких як філософія (для ідеологічного підходу до вивчення системно-інформаційної картини світу), філологія та лінгвістика (для розуміння систем програмування, текстових редакторів, систем розпізнавання тексту, комп’ютерного перекладу, систем штучного інтелекту), математика, фізика та

економіка (для комп'ютерного моделювання), мистецтво (для графічних редакторів, дизайну, мультимедійних систем) і так далі. Викладач інформатики повинен мати широкі знання та бути постійно вдосконалювати свої навички і рівень освіти [25].

Дослідники дійшли висновку, що використання мультимедійних технологій дозволяє суттєво поліпшити показники осмисленого розуміння та запам'ятовування навчального матеріалу. Серед причин, частіше за все, називають синкретичне навчання (одночасно зорового і слухового сприйняття матеріалу), активну участь в управлінні подачею матеріалу, невелике повернення в ті розділи, які вимагають додаткового аналізу [25].

1.2. Використання мобільних технологій при викладанні основ алгоритмізації в шкільному курсі інформатики

Використання мобільних пристроїв допомагає учням швидко отримати доступ до різних інформаційних ресурсів, у будь-який час і незалежно від місця знаходження. Зручність їхнього використання на уроках полягає ще й у тому, що відсутня залежність від наявності Інтернету в комп'ютерних класах завдяки можливості скористатися мобільним Інтернетом. Це створює, за словами Ю. В. Триуса і В. М. Франчук, більшу кількість «ступенів вільності» – вищу інтерактивність, більшу свободу руху, більшу кількість технічних засобів для навчання [36].

Головним засобом реалізації мобільного навчання на уроках є смартфон. Але є й інші мобільні пристрої, які можна використовувати на уроках. В першу чергу це ноутбуки і планшети. Їхня спільна ознака – малі розміри, невелика вага і достатня функціональність.

Упровадження мобільних технологій в освітній процес значно розширює можливості вдалого впровадження програмування мобільних застосунків при викладанні основ алгоритмізації в шкільному курсі. До основних переваг мобільного навчання Р. С. Гуревич відносить:

- явну інноваційність;

- можливість використання «легких» переносних пристроїв в освітніх цілях;
- можливість застосування технології в якості додаткового засобу навчання поряд з традиційним процесом;
- підвищену ефективність в навчанні осіб з обмеженнями по здоров'ю;
- широкі можливості в проектуванні неперервного навчання;
- відсутність обмежень за часом, місцем і розкладом навчання;
- відсутність вікових обмежень [27].

Як зазначають вітчизняні науковці, більшість сучасних учнів та випускників ІКТ- грамотні та мотивовані. Базові знання із цифрових технологій вони отримують за межами школи й часто завдяки самоосвіті та взаємному навчанню серед однолітків... Технологічні навички та вміння у них розвиваються й удосконалюються разом із розвитком самих технологій. Для учня не є проблемою, не виходячи з помешкання, завантажити реферат на запропоновану тему, дізнатися про погоду на найближчий час, обмінятися повідомленнями з друзями в соціальній мережі, пограти тощо. Окремі учні можуть знайти інформацію про умови вступу до різних закладів вищої освіти, придбати квиток на потяг або концерт тощо [4].

У моделі мобільного навчання учень з'єднаний з учителем постійно за допомогою Інтернет – на відміну від традиційного навчання, де таке об'єднання можливе лише в межах навчального закладу. Учитель відіграє роль консультуючого керівника, котрий спрямовує діяльність учня на отримання необхідної інформації. Це дозволяє реалізувати в даній моделі проблемне навчання через обговорення дій, які допоможуть учневі оволодіти матеріалом, до усвідомлення необхідного результату та набуття нового знання.

Для результативної роботи в мобільному середовищі як надавачу освіти, так і здобувачу освіти необхідно розуміти соціальну природу навчальної комунікації, що є вирішальним у набутті інформації, знань, досвіду та вмінь. Мобільне навчання проводиться не в межах кабінету , а в конкретному навчальному просторі. На мою думку, навчальна група не зникає, стають динамічнішими на кожному предметі окремо. Час навчання для учня не

обмежується шкільним розкладом . Вони швидко усвідомлюють позитивні переваги в ефективності мобільного навчання та різноманітності комунікацій, доступності до навчальних ресурсів. Навчання із мобільними пристроями сприяє соціалізації учнів старших класів, які мають можливість обговорювати набуті знання в соціальних мережах [9].

Але слід зазначити, що віртуалізація навчання сприяє втраті контактів між учнями та вчителем. Лише об'єднавши традиційне та мобільне навчання можна надати професійних знань і сформувати загальнокультурну особистість. Завдяки цьому пропонуються мобільні технології для дистанційного навчання та позакласної робити.

Зараз існує достатньо велика кількість мобільних застосунків для пристроїв різних типів. Розробники застосунків надають необмежений доступ програм, що є необхідним для застосування на уроках інформатики, завдяки цьому надавач освіти робить навчальний процес цікавим та сучасним . Перевагою користуванням мобільних пристроїв можна вирішити обмеженість комп'ютерної техніки в ліцеях . Опанування знань, умінь та навичок з інформатики та програмування відбуваються в ігровій формі, що популяризує вивчення та зацікавленість предметом у ліцеях.

Серед переваг мобільних технологій навчання можна виділити такі (рис 1.1):

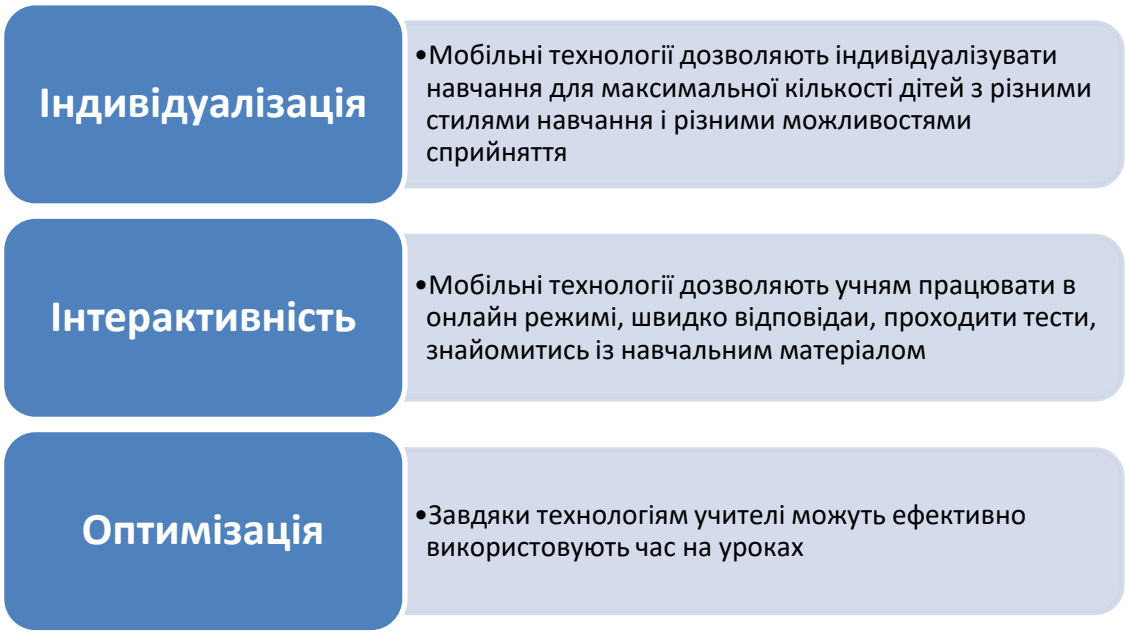


Рис. 1.1. Переваги мобільних технологій навчання

Дослідження ЮНЕСКО виявили, що використання мобільних пристроїв може забезпечити більш ефективне використання часу на уроках. Мобільні технології забезпечують більш продуктивне впровадження діяльнісного підходу до навчання. Основи програмування розпочинаються в 2 класі, де іде ознайомлення із поняттями «команда», «виконавець», а в 3 класі - «алгоритм», де здобувачі освіти оволодівають навичками у створені лінійних програм . У 4 класі діти вдало застосовують у своїх програмах різноманітні алгоритми: лінійний, розгалужений, циклічний [1].

Для того, щоб зробити навчальний процес якомога більш ефективним, треба правильно продумати, які застосунки краще використовувати. Як приклад Керрі Галаахер, ввела цікавий метод: увесь навчальний рік навчатися без паперу, замінивши на хмарне сховище Google Drive; впроваджувати мультимедійні додатки)Animoto, Educreations, Videolicious); застосунки для заміток (Evernote і Skitch, Backchannel), щоб обговорювати завдання, вправи, питання разом з усіма дітьми в позаурочний час; створити електронний ресурс для загальних висловлювань Padlet тощо. Усі ці додатки, допомагають менше витрачати час на отримання знань, умінь та навичок, постійно знаходитись в онлайн-режимі, швидко аналізувати результати своєї роботи, проходити різноманітні опитування, опрацьовувати методичний матеріал, створювати власні закладки та мати доступ до облікової сторінки.

Окрім захопливості, був виявлений ще один важливий аспект – заощадження часу, який не треба було витрачати на марні дії від 5-ти до 15 хвилин, такі як: розгорнути щоденник, пошук сторінок у підручнику, перемалювати графік або записати цитату. З BYOD цей час можна згаяти на важливіші речі – дискусії, особисті консультації, спільну роботу.

Таким чином, мобільне навчання має перевагу в нових можливостях у офлайн та онлайн навчанні . Але, спираючись на думку О. В. Слободняк, то застосування мобільних технологій мають деякі складності в організації навчального процесу та в обізнаності здобувачів освіти. Науковець повідомляє , про те, що мета використання смартфонів на уроках, може здійснюватися надто складно, оскільки є учні з низьким рівнем сформованості навчальних

компетентностей, що систематично відволікаються на ігровий контент та соціальні мережи. У їхньому випадку використання мобільних застосунків на уроках лише знизить їхню ефективність. [7].

Під час використання мобільних пристроїв на уроках вчитель може наштовхнутися на деякі технологічні труднощі, які визначає Р. С. Гуревич у [3].

Серед цих труднощів можна виділити не повну сумісність мобільних пристроїв та програмного забезпечення, труднощі, які виникають при намаганні об’єднати дві мережі з різною архітектурпю для зв’язку учнів через мобільні пристрої з освітнім контентом, обмежені розміри та ємність мобільних пристроїв, обмеження в представленні освітньої інформації візуально, обмеження швидкості передачі даних, а також частотні зміни в моделях, технологіях і функціональних можливостях мобільних пристроїв [3]. Тому вчитель повинен детально спланувати всі аспекти, обдумати, як зменшити вплив зазначених труднощів, за потреби створити власні електронні освітні ресурси, адаптовані як до вікових, так і до особистісних особливостей та освітніх потреб учнів.

1.3. Програмування мобільних застосунків як спосіб покращення рівня викладання основ алгоритмізації в шкільному курсі інформатики

В даний час вчитель інформатики володіє унікальними можливостями формування пізнавального інтересу школярів в процесі навчання основ алгоритмізації в шкільному курсі, перш за все, завдяки широкому спектру використовуваних програмних середовищ. Так, наприклад, на уроках інформатики при вивченні програмування вчителі активно використовують Візуальні середовища програмування (Scratch), орієнтовані на розробку «настільних» додатків, що безсумнівно, викликає великий інтерес у школярів, але не задовольняє його в повній мірі. Це можна пояснити тим, що крім комп’ютерів, учні все частіше користуються мобільними пристроями. Чи постає перед учителем інформатики питання, як перетворити улюблену іграшку в засіб навчання?

Більшість школярів сприймають навчання основ алгоритмізації в шкільному курсі як складний і нудний. У той же час, навчання основ алгоритмізації в шкільному курсі є дуже важливою складовою навчання, і хотілося б організувати цей процес так, щоб воно цікавило не двох-трьох чоловік з класу, а був би цікавий всім [14].

Scratch являється об'єктно-орієнтованим додатком, що допомагає простому засвоєнню базового поняття «програмування», який навчає учнів критичному мисленню та міркувати об'єктно – події, функції, умови, цикли, змінні, команди тощо (рис. 1.2).

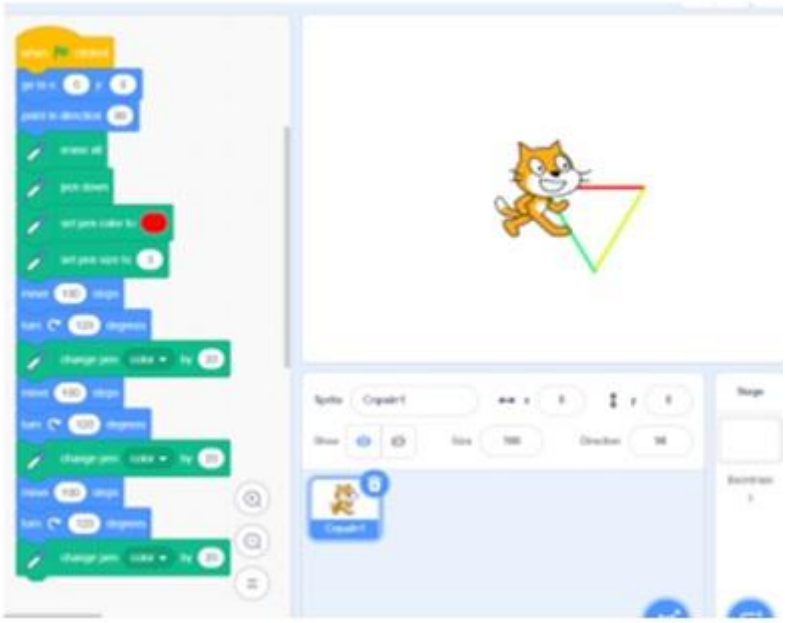


Рис. 1.2. Інтерфейс середовища Scratch

Ще одним із способів вирішення цієї проблеми є використання редактора MIT App Inventor. MIT App Inventor – це середовище програмування, розроблене вченими з Массачусетського технологічного інституту. Програма призначена для розробки додатків для мобільних пристроїв (смартфонів і планшетних комп'ютерів), що працюють на операційній системі Android [17].

MIT App Inventor – це середовище візуальної розробки Android-додатків. Для розробки додатків в MIT App Inventor не потрібно знання програмування на мові Java і Android SDK, досить оволодіння основами алгоритмізації на рівні середньої школи. Перевагою цього середовища є те, що з її допомогою всього за

кілька хвилин можна створити найпростіше додаток, а за годину – скомпонувати досить складну програму з декількома екранами (рис. 1.3).

Рис. 1.3. Робоче меню MIT App Inventor language

Те, що його не потрібно встановлювати на комп'ютер, відноситься до основних переваг редактора MIT App Inventor, він працює прямо з браузера, для цього потрібно тільки мати обліковий запис Google. Застосунок працює у фреймворку GNU, що реалізує мову Scheme (діалект Lisp) для платформи Java. Також, для його роботи в MIT розробили бібліотеку Open Blocks [22].

Програмування середовища MIT App Inventor language дуже просте, так як він з самого початку створювався для того, щоб ним користувалися школярі. При програмуванні на ньому не потрібно писати рядки коду, як це відбувається на інших мовах програмування. Щоб створювати додатки в MIT App Inventor, досить просто перетягувати блоки, збираючи їх в програмі, як пазл. App Inventor – хмарна сфера. Створення та задум проектів здійснюється в онлайн просторі, для роботи з яким потрібно тільки браузера, який сумісний з будь-яким пристроєм – Windows, Android, IOS, Linux тощо. Тестування готового проекту проходить на мобільному пристрої та на емуляторі де є доступ до ОС Windows.

Розробляючи в середовищі App Inventor, учнів можна навчити працювати в команді. Робота в групах з App Inventor – непросте завдання. Учні можуть

спільно планувати свої програми та обмінюватися ідеями, але коли настає час розробляти та впроваджувати свої програми, інструмент має деякі обмеження, які ускладнюють групову роботу. Наприклад, блоки не можуть бути скопійовані з одного проєкту в інший, і двоє або більше людей не можуть працювати над одним проєктом одночасно. Навіть з обмеженнями App Inventor щодо підтримки співпраці, 68 відсотків респондентів погодилися, що їм подобається працювати в команді над розробкою мобільних додатків [33].

Курс включав в себе кілька групових завдань з чисельністю груп від 2 до 5 учнів. Окрім командної роботи, багато завдань також включали обмеження часу та обсягу, щоб кинути виклик учням та переконати їх виконувати свої завдання та прогресувати. Учні повинні були придумати свій власний підхід до виконання групового завдання. Наприклад, групи витратили більше часу на планування програми та створення макетних екранів, щоб визначити компоненти та функціональність, необхідні перед запуском будь-якого коду. У деяких випадках вони навіть обговорювали, як назвати компоненти, щоб інші члени групи могли легко їх знайти. Деякі групи вирішили розділити свою діяльність, і кожен учасник буде працювати самостійно для виконання своїх відповідних завдань [44].

Після цього вони створювали спільний обліковий запис для доступу до App Inventor, а потім кожен учасник по черзі реалізовував свою частину проєкту. Одна група намагалася змусити всіх учасників одночасно увійти в App Inventor для роботи над одним проєктом, використовуючи той самий обліковий запис Користувача. Однак вони швидко дізналися, що поточна версія App Inventor не підтримує синхронну співпрацю. У деяких групах кожен учасник може працювати індивідуально над своїми завданнями і відправляти свою роботу учаснику, який відповідав за об’єднання всіх частин в один проєкт. Нарешті, для інших груп підхід полягає у тому, що один учасник працює за комп’ютером, а інші учасники навколо нього обговорюють проєкт та надають підтримку під час впровадження.

Підхід App Inventor до візуального програмування допомагає приховати деякі складності програмування, надаючи учням можливість зосередитися на

дизайні програми, її функціях і тому, як користувачі будуть взаємодіяти з ним. Це означає, що учні повинні вивчати та практикувати дизайн інтерфейсу користувача, введення та виведення даних користувача та перевірку вхідних даних, оскільки вони відіграють важливу роль у роботі користувача з мобільними додатками. Найчастіше учні виявляють деякі проблеми зі своїми програмами, які можуть бути наслідком поганого дизайну інтерфейсу користувача. Зокрема, вводяться користувачем дані ігноруються, що призводить до неправильної поведінки додатків або збою в роботі, коли користувачі вводять несподівані дані або не надають жодних даних [37].

Звичайно, App Inventor надає компонентам кілька властивостей, що дозволяють розробникам налаштовувати додаток за потребою. Наприклад, текстовому полю компонента можна присвоїти значення number тільки для того, щоб обмежити тип даних, що вводяться. Однак розробник несе відповідальність за визначення діапазону допустимих чисел і створення відповідного коду для його перевірки. Інші властивості, такі як «увімкнути / вимкнути» та «видимий / прихований», надають розробникам можливість налаштувати свої програми [28].

Процес програмування мобільних додатків не так сильно відрізняється від традиційного процесу розробки програмного забезпечення. Однак, якщо ми швидше розробляємо програми для Android за допомогою проєктної команди, до складу якої входять розробники, які не мають досвіду роботи з Android SDK та програмуванням Java, то це значно покращить продуктивність розробки додатків. App Inventor використовується багатьма факультетами, що займаються комп’ютерними науками та інженерією в коледжі, та багатьма учнями, які цікавляться інтелектуальними програмами в школі.

Кожен, у кого недостатньо знань в області програмування, може створювати додатки для Android за допомогою App Inventor. Ви можете створити додаток для Android на веб-сторінці, розмістити кілька логічних блоків разом на одній сторінці та одночасно протестувати додаток для Android на емуляторі або на телефоні. App Inventor – це перевага компонентного візуального програмування, де можна перетягувати візуальні компоненти, а потім задавати

програмовану поведінку логічним блокам, що спрощує розробку мобільних додатків [23].

Таким чином, використання App Inventor для створення прототипів додатків є високоефективним, оскільки App Inventor простий у використанні та інтуїтивно зрозумілий для програмування додатків, а App Inventor ефективний при підключенні до інших процесів розробки додатків.

За допомогою App Inventor курс мобільної розробки повинен бути захоплюючим і включати в себе кілька комп'ютерних концепцій, крім програмування, таких як бази даних, передача даних, розробка програмного забезпечення, управління проєктами, розробка мобільних додатків, веб-сервіси та багато іншого. Тепер, коли учні отримали уявлення про розробку додатків, викладачі, які викладають більш просунуті курси, можуть проілюструвати концепції своїх конкретних курсів за підтримки App Inventor [26]. Наприклад, на курсі баз даних учні можуть створювати форми для вставки даних у таблиці або відображення даних із таблиць, використовуючи як статичні, так і динамічні запити. На курсах розробки програмного забезпечення або системного аналізу та проєктування учні могли б скористатися підтримкою App Inventor для швидкої розробки, щоб планувати, проєктувати, впроваджувати і тестувати мобільні додатки в рамках курсових завдань або проєктів. Зокрема, викладачі могли б вивчити принципи та методи проєктування інтерфейсу користувача.

Ще однією особливістю MIT App Inventor є можливість тестування розроблених застосунків на мобільному пристрої в режимі реального часу без попередньої компіляції та налаштування на мобільному пристрої. Для цього достатньо встановити на смартфон спеціальний застосунок MIT AI2 Companion. Також тестування застосунків можливо і в емуляторі Android для десктопних ОС. Для створення Програми на екрані необхідно відобразити необхідні елементи інтерфейсу. Програмування виконується за допомогою блок-схем. Необхідно з'єднати блоки таким чином, щоб додаток виконувало необхідні дії. По завершенні розробки можна отримати або підготовлений APK-файл (формат архівованих виконуваних файлів-додатків для Android) для установки на пристрій, або QR-код з посиланням на скачування [35].

Перспектива навчання школярів розробці мобільних додатків і процес формування професійного інтересу носять профорієнтаційний характер, так як це динамічно розвивається область професійної діяльності. В якості аргументів досить навести кілька загальновідомих фактів. Можна з упевненістю сказати, що на ринку мобільних додатків фіксується перехід від ринку пропозиції додатків до ринку попиту. Споживання мобільних послуг в цілому зростає, оскільки зростають продажі смартфонів, зростає споживаний мобільний трафік, зростають продажі планшетів. Спостерігається активне зростання мобільної реклами.

Таким чином, професійна сфера, пов’язана з програмуванням мобільних додатків, є активно розвиненим сектором IT-індустрії. Однак в старших класах пояснювати отримання наукомістких професій вже пізно. Творчо працюючому вчителю очевидно, що запорукою успіху підготовки висококласного програміста є стійка мотивація вже на етапі базової загальної освіти, і, як інструмент співпраці, MIT App Inventor є одним із засобів формування та підтримки цієї мотивації [52]. Побудоване таким чином вивчення програмування сприяє досягненню не тільки предметних результатів, аналогічно під час уроку відбувається формування:

- дисциплінованість у навчання , охоче бажання до роботи, саморозвитку та самоосвіти у здобувачів освіти що побудовані на мотивації;
- цілісний світогляд, що відповідає сучасному рівню розвитку науки;
- здатності самостійно планувати шляхи досягнення цілей;
- уміння порівнювати свої результати та дії разом із запланованими, контролювати свою діяльність в процесі досягнення мети та завдання;
- здатності оцінювати правильність виконання освітнього завдання.

Висновки до розділу 1

У першому розділі проаналізовано наявну теорію та практику використання програмування мобільних застосунків при навчанні основ алгоритмізації учнів середньої' школи. Вивчення основ алгоритмізації у шкільному курсі інформатики розкриває важливість алгоритмів та їхню роль у функціональному зв’язку між поняттями «інформація-алгоритм-комп’ютер». що визначає процес автоматичного опрацювання інформації. Метою вивчення алгоритмізації є формування знань та навичок учнів у сфері основних методів організації операцій і даних, а також використання базових алгоритмічних конструкцій для складання описів алгоритмів розв’язання різноманітних завдань.

При цьому слід зазначити, що ефективність застосування програмування мобільних застосунків при вивченні цього розділу не є достатньо висвітленою в педагогічній літературі, що робить кваліфікаційну роботу актуальною.

РОЗДІЛ 2.
РОЗРОБКА МЕТОДИЧНИХ РЕКОМЕНДАЦІЙ З ВИКОРИСТАННЯ ПРОГРАМУВАННЯ МОБІЛЬНИХ ЗАСТОСУНКІВ ПРИ ВИКЛАДАННІ ОСНОВ АЛГОРИТМІЗАЦІЇ В ШКІЛЬНОМУ КУРСІ ІНФОРМАТИКИ

2.1. Аналіз досвіду використання програмування мобільних застосунків при викладанні основ алгоритмізації в шкільному курсі інформатики

Мобільними застосунками називають це програмне забезпечення, призначене для роботи на смартфонах та планшетах. Програмування мобільних застосунків майже ідентичне до створення програм та програмного забезпечення для персональних комп'ютерів. У багатьох сучасних середовищах розробки є спеціальні бібліотеками для створення мобільних застосунків. Існують також середовища розробки і мови програмування, зумисне створені для цієї задачі, наприклад; Android Development; iOS Development; React Native; Kotlin; Ionic Development; Corona; Firebase. Сьогодні більшість здобувачів освіти мають певні навички у створенні мобільних застосунків завдяки спеціально розробленим середовищам (зокрема, Scratch). Незважаючи на різний рівень компетентностей з програмування, здобувачам освіти можна надати на вибір різні програмні середовища для оволодіння основ технології розробки мобільних застосунків [66].

Аналізуючи досвід використання програмування мобільних застосунків при викладанні основ алгоритмізації в шкільному курсі інформатики можна виділити низьку перевагу використання у загальноосвітніх навчальних закладах [55]:

1. Індивідуальний підхід у навчальному процесі.
2. Можливість самостійного розвитку та навчання кожної дитини.
3. Створення ситуації успіху, що сприяє підвищенню мотивації.
4. Більш ефективне використання часу на уроці.
5. Застосування інтерактивних засобів.
6. Розвиток партнерських стосунків між учителями, учнями та батьками.

7. Економія матеріальних, часових та енергетичних ресурсів.
8. Підвищення рівня цифрової грамотності всіх учасників навчального процесу.
9. Залучення дітей до використання Інтернет-ресурсів у навчальних цілях.
10. Візуалізація прогресу учнів.
11. Навчання основам командної роботи в рамках виконання проєктів.
12. Можливість повністю або частково навчатися вдома (зокрема, у випадку примусової ізоляції через пандемію COVID–19).
13. Розширення та інтеграція навчання [65].

Разом з усіма перевагами, використання програмування мобільних застосунків при викладанні основ алгоритмізації в шкільному курсі має і ряд недоліків:

– недостатня наявність якісного цифрового контенту для ефективного використання у навчальному процесі;
– велика ймовірність перерв у навчанні через нестабільність зв’язку;
– недостатній рівень підготовки вчителів до роботи з електронними освітніми ресурсами та інструментами;
– висока втрата у часі вчителя на підготовку уроків;
– збільшення часу, проведеного перед екраном монітора, що може впливати на правильність постави та зір дитини.

О. Коротун та О. Кривонос зазначають, що головна мета надавача освітніх послуг під час використання програмування мобільних застосунків при викладанні основ алгоритмізації в шкільному курсі – методично правильно скомбінувати навчальний матеріал за часом , формою роботи , ресурсами та за темою, метою і завдань. . Вчитель планує, необхідний навчальний матеріал так , щоб здобувачі освіти мали нагоду опрацювати дану тему в комп’ютерному класі в традиційних формах роботи та в онлайн роботі з ІКТ, де домашні та індивідуальні завдання над проектом можна виконати в онлайн режимі . Автори рекомендують планувати урок таким чином, щоб опрацювання основного

матеріалу відбувалося під час уроку, а додатковий матеріал та індивідуальні завдання винести на самостійне опрацювання учнями [54].

При використанні програмування мобільних застосунків у шкільному курсі з основ алгоритмізації важливо зберігати діалогічну форму уроку, стимулювати захист проектів та презентації, впроваджувати дебати між учнями та вчителями . Електронний блок навчальних матеріалів повинен фокусуватися на проектній та групові роботі, містити творчі, практичні та інтерактивні завдання, а також посилання на додаткові інструкційні матеріали в Інтернеті. Для контролю за прогресом у досягненні навчальних результатів використовуються перевірочні та проміжні тести.

Важливою умовою вдалого впровадження програмування мобільних застосунків при викладанні основ алгоритмізації в шкільному курсі у навчальних класах є роль учителя. Звертаючи увагу на психолого-педагогічні особливості учнів, вчитель може працювати з ними у невеликих групах і додатково індивідуалізувати навчання. Він може формувати групи для проєктної діяльності та сприяти обговоренню для розвитку критичного мислення. Такий підхід до диференціації навчання позитивно впливає на якість освіти учнів і покращує їх розуміння навчального матеріалу. Учні виявляють інтерес до навчання, коли використовують спеціально розроблені навчальні матеріали, інтерактивні завдання, головоломки, тести та інше. [49].

І. Тимофєєва, М. Нетреба та Є. Новицька розглядають можливості використання елементів програмування мобільних застосунків при викладанні шкільному курсі для різних типів уроків. Згідно з дослідженнями вчених, у плануванні уроку можна використовувати інформаційні технології для залучення учнів за допомогою аудіального чи візуального матеріалу. Для перевірки розуміння учнями матеріалу можна використовувати ігрові тренажери, пазли, інтерактивні вправи, головоломки, тести та інші види завдань. Для стимулювання початкової діяльності варто користуватися мультимедійними презентаціями, інтерактивними онлайн- матеріалами та демонстраціями. Актуалізацію опорних знань можна провести за допомогою вікторин, пазлів, інтерактивних карт активізації пам’яті. Пояснення нового матеріалу відбувається

за допомогою використання різноманітної наочності (текст, графіки, мультимедіа, анімація) де присутній інтерактивний зворотній зв'язок. Діагностику рівня засвоєння матеріалу пропонується робити, використовуючи інтерактивні завдання, навчальні ігри, вікторини, ребуси, онлайн-тести тощо. Узагальнення та систематизацію знань можна здійснити у формі онлайн-дискусій, форумних обговорень чи чатів. [38].

О. Барановська розглядаючи можливості у використані підручників НУШ для вдалого впровадження програмування мобільних застосунків при викладанні основ алгоритмізації в шкільному курсі, акцентує на важливість сучасних ресурсів, де є повна інтеграція освітніх наскрізних ліній. Авторка наголошує на нових рубриках завдань в підручниках , що зосереджуються на групових та парних формах навчальної діяльності учнів, що використовуються для впровадження програмування мобільних застосунків при викладанні основ алгоритмізації в шкільному курсі. Такі завдання направлені на розвиток важливих наскрізних умінь та навичок здобувачів освіти таких як: командна робота (є головним аспектом для навчання програмування мобільних застосунків при викладанні основ алгоритмізації в ліцеях), критичне мислення, навчатися та розвиватися протягом життя. У підручниках з різних предметів є блоки завдань: «Проведи дослідження», «Перевіряю свої досягнення»; «Завдання на застосування інформаційних умінь»; «Медіавіконце» та ін. [43].

Адаптація таких завдань до електронного формату досить проста і цікава. Екскурсію можна організувати в електронному форматі, відвідавши віртуальний музей, театр або галерею. Велика кількість дидактичних ігор реалізована в електронному вигляді. Для проведення дискусій можна використовувати засоби для онлайн-конференцій. Проблемно-пошукові завдання можна вирішувати онлайн за допомогою інтернет ресурсів з використанням програмування мобільних застосунків і поєднанням вивчення основ алгоритмізації.

Важливим і навіть основоположним фактором вдалого впровадження програмування мобільних застосунків при викладанні основ алгоритмізації в шкільному курсі є підготовка вчителя до уроку. Правильний підбір складності, видів діяльності та інтерактивних завдань позитивно впливає на формування

результатів навчання учнів в кожній змістовій лінії та їхню мотивацію до навчання. [47].

Сучасні дослідження підготовки навчальних матеріалів для впровадження програмування мобільних застосунків при викладанні основ алгоритмізації, акцентують увагу на тому, що це потребує багато часу на опрацювання. Так, за даними eLearningart, для створення однієї години інтерактивного онлайн-навчання, що базується на принципі «натискай і читай» разом із простим тестом, вчителеві потрібно близько 49 годин. У випадку, якщо необхідно розробити інтерактивні елементи для онлайн-курсу, час, витрачений на створення однієї години такого курсу, може становити від 127 до 267 годин. Зазначені цифри не обов'язково означають, що вчителі повинні витрачати стільки часу кожного разу на розробку матеріалів, але вони дійсно відображають обсяг роботи, яку вчителі виконують для створення і впровадження дійсно ефективних та якісних навчальних матеріалів. [50].

Влучне впровадження сучасних інтернет ресурсів дає можливість скоротити час у підготовці, урізноманітнити практичні завдання для навчального процесу з програмування мобільних застосунків де викладають основи алгоритмізації. Відомо лінію «віртуальних навчальних середовищ» (virtual learning environment) – програмна система і платформа, яка підтримує освітній процес. Як зазначив Широков Д., існує два типи сучасних електронних платформ: цифрові навчальні простори на яких заздалегідь підготовлені інтерактивні навчальні матеріали за рекомендаціями навчальних програм і вони доступні вчителю для використання на уроках та «платформи-оболонки» з шаблонами, які учитель заповнює власними матеріалами. [60].

Усі перелічені освітні системи є доступними, постійно вдосконалюються та поповнюються новими освітніми ресурсами. Окрім них широкого поширення набули онлайн-ресурси для виконання окремих видів навчальних завдань. Серед найбільш популярних сервісів, якими користуються вчителі та учні є :

1. Сервіс для проведення онлайн опитувань та тестувань на https://naurok.com.ua. Вчителі можуть створювати свої власні тести або використовувати тести інших педагогів.

2. Сервіс мультимедійних дидактичних вправ на https://learningapps.org для створення навчальних пазлів, ігор, інтерактивних вправ.
3. Онлайн-сервіс для створення інтерактивних плакатів на www.thinglink.com. Цей інструмент надає можливість додавати до зображення інтерактивні мітки (текст, посилання, зображення, відео) [53].

Проаналізувавши основні види електронних освітніх ресурсів, О. Машталір, Н. Дільна, І. Обіход, Н. Закордонець дійшли висновку, що у процесі використання програмування мобільних застосунків для вивчення основ алгоритмізації доцільнішим є застосування блогів та вікі. На думку науковців, ефективний спосіб вивчення інформатики є організація офлайн та онлайн навчання, де надавач освітніх послух веде учбовий блог.

С. Забазна, описуючи свій досвід використання блогу для публікації електронного контенту під час викладання основ алгоритмізації, відзначає сервіс Blogger та наголошує на важливості правильного добору змістового наповнення блогу для здійснення якісного і системного поєднання різних видів діяльності учнів. [39].

У виборі інструмента для створення навчальних мобільних застосунків, треба враховувати складність матеріалу, тому на початку вивчення інформатики важливою є простота інтерфейсу та зрозуміла логіка кодування та алгоритмізації, можливість створювати мобільний застосунок, сумісний з кількома платформами [34].

2.2. Добір засобів для використання програмування мобільних застосунків при викладанні основ алгоритмізації в шкільному курсі інформатики

На початку нашого дослідження назвемо кілька платформ, що використовуються для створення програм.

Android Studio – відкрите і безкоштовне інтегроване середовище розробки для Android [31].

Eclipse – популярна opensource платформа для програмування на Java, зокрема, для створення мобільних застосунків. Особливостями платформи є великий набір API і використання RCP підходу. В Eclipse також є ряд звичних

для великих платформ функцій, таких як автоматичне підсвічування синтаксису коду, навігатор класів, зневаджувач, менеджер файлів і проєктів, рефакторинг коду, система контролю версій.

APPSFERA – конструктор для створення мобільних застосунків для Android та iOS. Цей продукт є закритим та платним, але з безкоштовним демо-періодом тривалістю 1 рік з можливістю розробки і підтримки одного застосунку. В APPSFERA є безкоштовні макети та навчальні проєкти на платформах Android та iOS. Використовуваною тут мовою програмування є Xcode Gradle. Також у демо-версії відсутні push-повідомлення та технічна підтримка.

Thunkable – безкоштовне середовище програмування мобільних застосунків, побудоване на базі MIT App Inventor 2, використовує візуальну мову Scratch. Більшість проєктів, розроблених з використанням подібних платформ (зокрема, і в оригінальному App Inventor) є сумісними зі Thunkable. Спільнота користувачів цієї платформи є достатньо великою і активною, тому час від часу створюються додаткові розширення [29].

MIT App Inventor – це візуальне середовище для розробки android-застосунків.. Він дозволяє створювати повнофункціональні програми для телефонів Android, iPhone і планшетів Android/iOS, з мінімальними знаннями програмування. Платформа побудована для різного вікового складу, від дитини до дорослого та будь-яких технічних можливостей. Тому MIT App Inventor є корисним і цікавим для школярів, вчителів, студентів та викладачів. На платформі, візуальне середовище, має усі необхідні посібники , відео уроки, інструкції у вигляді зображення та підказки під час роботи з робочим середовищем.

Переваги: простота засвоєння, незважаючи на початкові знання з програмування засвоюється легко і достатньо швидко, переглянувши лише декілька навчальних відео; доступність (середовище безкоштовне і відкрите); простота використання, бо дану платформу може засвоїти учень середньої та старшої ланки, що надає можливість використовувати у навчанні; можливість прямого завантаження на смартфон; можливість редагування проєктів інших

користувачів; одна програма підходить як під Android, так і під iOS. Несподіваним недоліком, особливо в часи, коли можливі атаки на критичну інфраструктуру, може стати необхідність стабільного доступу до інтернету.

Таким чином, навчитися створювати програми неважко, якщо розуміти логіку платформи, завдяки допоміжним навчальним посібникам і відео. [30].

Більшість учнів мають початковий рівень сформованості компетентностей з програмування, тому при доборі середовища програмування мобільних застосунків треба звертати увагу на поріг входження, простоту та інтуїтивність інтерфейсу та на доступність.

2.3. Розробка методичних рекомендацій з використання програмування мобільних застосунків при викладанні основ алгоритмізації в шкільному курсі інформатики

Однією з особливостей миттєвого формування ІКТ, перед надавачем освіти з інформатики стоять питання – як стимулювати здобувачів освіти до нових можливостей, коли вони з дитинства із технікою на «ти»? Як оригінально та сучасно підібрати матеріал, при чому дотримуватися навчального плату? В освітній програмі є впровадження в процес навчання учнів ліцею саме профільний рівень «Інформатика», що містить в собі вивчення теми «Алгоритми та програми».

Реалізація Концепції Нової української школи і Державного стандарту початкової освіти ґрунтується на активному, компетентнісному та особистісно зорієнтованому підходах до навчання. Державний стандарт надає широкі можливості для творчості вчителів та впровадження різноманітних технологій сучасного навчання. [19].

О. Власій та М. Винничук, у своїх дослідженнях звертають увагу на те, що вчителю необхідно старанно підбирати сучасні методи, прийоми та засоби у активному формуванні ключових та предметних компетентностей для збереження структури дидактичної системи. Кожен урок повинен починатися від

простого до складного, від теорії до практики, оскільки це спонукає до розвитку їх творчого креативного мислення під час опанування курсу профільного вивчення інформатики.

На основному етапі початкового рівня вивчення алгоритмізації та програмування рекомендують використовувати середовище програмування Scratch. Надалі, при повторенні вивченого матеріалу, або нового, доречно буде поступово впроваджувати інше середовище програмування такий як MIT App Inventor, оскільки він є розвитком ідей блочного програмування, з яким учні знайомляться в Scratch.

Головними формами навчальних занять в комп'ютерних класах з профільним вивченням інформатики залишаються уроки вивчення нового матеріалу; удосконалення знань та формування умінь шляхом розв'язування задач, узагальнення та систематизації вивченого; контролю та корекції знань. [40].

Згідно з рекомендаціями МОН, використовують такі форми організації навчання: уроки-лекції, уроки-семінари, заліки, практичні заняття різного типу, (індивідуальні, робота в парах, у групах тощо), уроки розв'язування задач, лабораторні роботи, роботи над проєктними задачами.

Електронні освітні ресурси використовуються на уроках інформатики з такими цілями [42]:

- підвищення зацікавленості у навчальній діяльності;
- активізація когнітивних процесів;
- інтенсифікація навчання;
- диференціація освітнього процесу;
- індивідуалізація навчання;
- додавання естетичного та емоційного компонентів;
- забезпечення наочності;
- розширення можливостей самостійної роботи;
- збільшення інтерактивності навчання;
- формування дослідницьких компетентностей;
- доступність до вільного користування інформаційними ресурсами.

Кожен урок інформатики в 10-11 класі має бути спрямований на формування таких компетентностей:

- пізнавальної, яка створює середовище для опанування певних понять з програмування;
- практичної, яка залучає використовувати отримані знання, уміння, навички з програмування;
- творчої, яка спрямована на розвиток створення уяви та фантазії, формує художньо-технічний, естетичний смак;
- соціальної , яка мотивує на покращення особистих рис (відповідальність, чесність, працелюбність, комунікабельність, самостійність), виховує трудову культуру.

Це пов’язано зі специфікою освітньої галузі для ефективного застосування навчального часу у вивченні інформатики, рекомендується впроваджувати спарені уроки за умови навантаження у хоча б 3 години на тиждень [41].

Щоурочне повторення правил безпеки базується на дослідженнях учених, які О. Пометун узагальнила у своїй науковій праці «Енциклопедія інтерактивного навчання». Дослідження підтверджує, що сприйняття матеріалу має безпосередній вплив на мозок, проте інформація не може зберігатися тривалий час без механічного запам’ятовування. Навчання не обмежується лише запам’ятовуванням; важливо розуміти, що засвоєння включає кілька підходів до повторення. Повторення матеріалу кілька разів надає учням можливість краще його засвоїти. Ці підходи повинні бути різноманітними та не повторювати попереднє викладення матеріалу, але привносити нові аспекти для кращого розуміння.

Проведення уроку передбачає використання технології проєктного навчання. Кожен урок повинен містити окремий проєкт, розробляючи який, здобувачі освіти самостійно знаходять, аналізують та опрацьовують необхідні знання та навички для вирішення пізнавальних і практичних задач. Знання здобуваються легко, природньо, бо для здобувачів освіти на першому місці стоїть створення програми, а не просто вивчення матеріалу. Отримуючи

остаточний результат, у здобувачів освіти розвивається мотивація щодо удосконалення своїх знань, умінь та навичок у даному напрямку. [45].

Мотиваційна частина визначає перший етап уроку інформатики. Зазвичай цей він проходить у груповій формі за робочим столом у центрі кабінету. Мотиваційний зазвичай триває 15 хв.

Під час мотиваційного етапу рекомендується використовувати ігрові інтерактивні сервіси, такі як https://learningapps.org, які легко актуалізують матеріал попередніх уроків у цікавій формі. Практична частина уроку інформатики включає етап відпрацювання практичних умінь роботи з комп'ютером. Під наглядом учителя здобувачі освіти переходять до індивідуального робочого середовища . Кожен учень самостійно виконує практичне завдання, поставлене на попередньому етапі уроку. Вчитель може демонструвати алгоритм (покрокову схему) на дошці та надавати допомогу учням при виникненні запитань або ускладнень. Етап відпрацювання практичних умінь із застосуванням комп'ютерних засобів зазвичай триває 25 хв.

На етапі практичної роботи з цифровими пристроями у вчителя є можливість обирати будь-які електронні інтерактивні сервіси, програмні пакети та застосунки навчальної спрямованості. На відміну від інших етапів уроку, де вчитель використовує лише демонстраційні електронні матеріали, не залучаючи учнів до безпосередньої роботи з комп'ютерною технікою, на цьому етапі учень активно залучається до власної діяльності.

Використання комп'ютерів у навчанні є дуже ефективним, оскільки це надає детальне уявлення про програмний матеріал. Це дозволяє учням краще розуміти та освоювати абстрактні поняття, такі як моделювання та алгоритмізація, а також формувати практичні вміння та навички. Якісне програмне забезпечення становить основу ефективного використання комп'ютера в освітньому процесі [51].

Підсумковий етап уроку є оцінювальним, рефлексивним та завершальним. Під час цього етапу учні демонструють виконані практичні роботи, презентують проєкти та результати виконаних досліджень. Також, відбувається обговорення

результатів, узагальнення понять, самооцінювання, рефлексія. Цей етап триває близько 10 хв.

Електронні ресурси, які використовуються вчителем на цьому етапі, мають сприяти закріпленню вивченого, розвитку критичного мислення та забезпечувати зворотний зв'язок між учителем і учнями. Для інтерактивного обговорення використовуються мультимедійн

Дослідники електронних освітніх ресурсів для учнів ліцеїв [5, 9, 20–23, 34, 50] зазначають, що під час їх вибору слід приділяти увагу відповідності освітнього електронного ресурсу навчальним цілям, дидактичним і методичним вимогам. Електронне інформаційне і навчальне програмне забезпечення для молодших школярів повинне мати розвивальний характер, бути сучасним і близьким інтересам дитини, забезпечувати можливість експериментування та творчості [56]. При виборі важливо враховувати вікові та психологічні особливості молодших школярів, виражені у поступовому ускладненні навчального матеріалу, використанні яскравих ілюстрацій, ігрових ситуацій, систематичному повторенні [59]. Проаналізувавши дослідження, можна виділити основні дидактичні критерії, які слід враховувати при виборі чи розробці електронних ресурсів для навчання інформатики:

1. Узгодженість з метою уроку.
2. Відповідність змісту ресурсу з програмою .
3. Можливості для розвитку та навчання, включаючи розвивальні та ігрові аспекти.
4. Орієнтація на самостійне засвоєння знань учнями, включаючи дослідницьку діяльність
5. Логічна та структурована організація навчального матеріалу.
6. Можливості для розвитку інформаційно-комунікаційної компетентності.
7. Підтримка розвитку логічного та критичного мислення
8. Впровадження креативності та врахування принципів інклюзивного навчання.
9. Акцент на формуванні вмінь вчитися та самостійно здобувати та обробляти інформацію.

10.Забезпечення безпечного використання електронного ресурсу.

Перед використанням MIT App Inventor в начальній програмі потрібно розібратися з основними особливостями, та нюансами в розробці програм, особливо в частині програмування.

На першому етапі потрібно дізнатися про основні поняття та алгоритмічні структури. Вивчення цього етапу передбачає розробку одноекранних додатків.

На другому етапі вже відбувається поглиблене вивчення App Inventor. Здобувачі освіти займаються створенням проєктів різних рівнів складності. Задля того, щоб продемонструвати, як можуть бути використані засоби MIT App Inventor на традиційному уроці інформатики далі наводимо два конспекти уроків учителя О. Коробки [].

План-конспект уроку №1

Тема: Реалізація базових алгоритмічних конструкцій.

Мета: ознайомити учнів із реалізацією базових алгоритмічних конструкцій, застосувати платформу MIT App Inventor , створення першого проєкту.

Обладнання: Онлайн середовище MIT App Inventor , комп’ютер .

Перебіг уроку

1. **Організаційний момент.** Перед уроком повідомляю учнів про наявність e-mail та готую посилання на сайт MIT App Inventor, розробила проєкт як приклад, який будуть створювати учні; підготувала комп’ютер для трансляції екрану, презентацію з потрібною інформацією [57].
2. **Актуалізація опорних знань.** Опитування учнів:
   - Чи знайома дітям ця тема?
   - Що вони можуть розповісти про неї?

   Створення акаунту в середовищі MIT App. Ознайомлення з основними компонентами простору.
3. **Пояснення нового матеріалу.** В алгоритмах із розгалуженням залежно від умови виконуються ті чи інші інструкції. Існують три типи

розгалуження: одноальтернативне (неповне розгалуження), двоальтернативне (повне розгалуження) і багатоальтернативне (вибір, варіант). Сьогодні і в подальшому будемо працювати з онлайн платформою MIT App Inventor.

В програмному середовище застосовується змінна циклу, значення вибираються послідовно зі значень об’єктів

4. **Формування навичок.**

– Увімкнути комп’ютери;
– Запустити браузер;
– Скопіювати посилання на сайт;
– Зайти, зареєструватися, пояснити здобувачам освіти у разі виникнення труднощів;
– Надати інструкцію у створені проєкту;
– Пояснити головний інструментарій програми.
– Ознайомити учнів зі складовою, де прописується алгоритм самої програми.
– Завчасно підготовленим планом створити з учнями перший проєкт, за допомогою демонстрації свого екрану, де вони за вчителем повторятимуть.
– Показати, як можна змінювати зовнішній вигляд програми.

5. **Рефлексія.**

Підсумкові питання:

6. **Домашнє завдання.**

Змінити зовнішній вигляд, створеної на уроці, програми.

План-конспект уроку №2

Тема: Реалізація алгоритмів з розгалуженням.

Мета: ознайомити з реалізацією алгоритмів з розгалуженням.

Обладнання: онлайн середовище розробки MIT App Inventor, комп’ютери з підключенням до мережі Інтернет, навчальна презентація

Перебіг уроку

1. **Організаційний момент.**

Підготувати проєктор і посилання на середовище розробки MIT App Inventor, сайт з корисно інформацією [58].

2. **Актуалізація опорних знань.** Перевірка знань за попередньою темою.
3. **Пояснення нового матеріалу.**

При складанні алгоритмів із розгалуженням у середовищі MIT App Inventor для того, щоб, наприклад, управляти рухом виконавців на сцені чи визначати відстань до вказаного об'єкта, можна використовувати числові величини, значення яких можуть змінюватись або задаватись.

Числові величини можна використовувати як при формулюванні умов розгалуження, так і наслідків — відповідних дій виконавців алгоритму.

Наприклад, при використанні числових величин в умові 

можна визначити, чи знаходиться об'єкт у «лівій половині сцени», оскільки центр сцени має координати (0,0).

4. **Формування навичок.**

- Вмикаємо комп’ютери, заходимо на сайт, знаходимо свої облікові записи.
- Переходимо на вкладку Blocks.
- Крім того, використовуємо такі вкладки: Math, Text, List, Variables та Logic.
- Спочатку потрібно створити змінну, а саме: переходимо у вкладку Variables, переставляємо блок initialize variable to на основний екран, який дозволяє створювати глобальну зміну, що можна використовувати на інших екранах. Існує два параметри: перша група вибирає місце зберігання даних з цієї змінної в оперативну пам’ять (app), на самому

смартфоні (stored), та в базі даних (cloud); друга група змінює назву змінної.

- Перший параметр залишаємо, а в другий вписуємо просту назву, число, букву. На даний момент змінна не володіє ніяким типом та значенням.
- Робимо 4 змінні, перетягуючи з меню, або копіюючи існуючі.
- Присвоюємо їм значення і тип. Із вкладки Logic витягуємо блок true, із вкладки Math дістаємо блок 0, із вкладки Text блок text (пусті подвійні скобки), а із List блок list.
- Робимо присвоєння, складаючи порожні змінні та блоки, які ми щойно витягли. Тепер змінні мають певний тип та значення, їх можна змінювати, а блок 0 може прийняти як ціле, так і дробові значення [61].
- Для перевірки, робимо просту програму, переходимо в панель Design в розділі Add Components, знаходимо компонент Label (надпис), розташовуємо його на екрані й створюємо подібних 4 штуки, і помічаємо, що в розділі Screens з'являються нові компоненти зі своїми найменуваннями, які можна змінити, лише англійською мовою, щоб не виникли помилки при запуску проєкту.
- Знову переходимо на панель Blocks і починаємо зв'язувати змінні (initialize variable to) з надписами, які ми додали (Label). Зліва в вкладці UI components розташований ваш екран (Screen) та його складові, в нашому випадку назви текстових полів. Клікаємо на них правою кнопкою миші, вилазить панель з додатковими блоками, на потрібні блоки when do та set 's to. Блок має два параметри: перший – це назва компонента, з яким відбувається дія; другий – умова, при якій відбудеться дія, в нашому випадку, при натисканні на текстове поле.
- Блок set 's to теж має дві змінні: заголовок текстового поля і тип даних, які вона буде виводити. Складаємо два цих компонента разом у вкладці, приєднуючи його до set 's to, тепер при натисканні на це текстове поле, з'являється значення змінної, яка присвоєна до цього поля.
- Копіюємо даний фрагмент, клікаючи лівою кнопкою миші на блок when do, а потім на Duplscate, цей блок буде скопійований повністю, робимо

так 3 рази. Заміняємо назви текстового поля на інші так, щоб порівняно з іншим блоком вони не повторювались, те саме робимо із назвами змінних [64].

- Клікаємо тестування (Web Preview), якщо все правильно, то при покажиться певне значення змінної, якої ви присвоїли даному текстовому полю.

5. **Рефлексія.**
6. **Домашнє завдання.**

Створення та робота із власним методом. Проєкт «Калькулятор»

Висновки до розділу 2

В даному розділі розглянуто досвід використання розробки мобільних застосунків при вивченні основ алгоритмізації та розроблені методичні рекомендації щодо впровадження розробки мобільних застосунків у шкільну практику під час вивчення основ алгоритмізації.

Зроблено висновки про те, що програмування мобільних застосунків засобами MIT App Inventor може стати ефективним засобом навчання основ алгоритмізації, адже пропонує інтуїтивно зрозуміле середовище програмування з використанням візуальної мови програмування, що нагадує вже знайоме учням середовище Scratch, вивільняючи навчальний час для вивчення ключових парадигм, понять та концепцій під час вивчення основ алгоритмізації. Окремо слід зазначити, що програмування мобільних застосунків у середовищі MIT App Inventor надає можливість наочно продемонструвати учням об’єктно орієнтований та подіє орієнтований підходи до програмування.

Також у розділі наведено два конспекти уроків з використанням середовища MIT App Inventor при вивченні лінійних алгоритмів та алгоритмів з розгалуженням.

РОЗДІЛ 3.
ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ РОЗРОБЛЕНИХ МЕТОДИЧНИХ РЕКОМЕНДАЦІЙ У НАВЧАННІ ЛІЦЕЇСТІВ

3.1. Методика проведення експерименту

Оскільки наше дослідження стосується сукупність умов, які забезпечують оптимальний підхід до вивчення програмування мобільних додатків як засобу для навчання основам алгоритмізації учнів ліцею, під час експерименту ми будемо акцентуватися на формуванні алгоритмічної компетентності. Оскільки алгоритмічна компетентність є частиною комплексного поняття інформаційно-комунікаційної компетентності, і між ними існує пряма пропорційна залежність, вдосконалення у формуванні алгоритмічної компетентності призведе до покращення загальних показників, зокрема, в цій ключовій компетентності. [62].

Визначення рівнів розвитку алгоритмічної компетентності учнів початкових класів проводилося на етапі констатувального етапу експерименту, під час якого вирішувалися наступні завдання:

- встановлення критеріїв та показників рівня сформованості алгоритмічної компетентності учнів початкових класів;
- відбір та впровадження в експериментальну роботу відповідного діагностичного інструментарію;
- проведення діагностики, аналіз та узагальнення отриманих результатів;
- визначення рівнів розвитку алгоритмічної компетентності учнів.

Констатувальний етап експерименту проводився для здобувачів освіти ліцею (16 учнів).

Враховуючи визначення «алгоритмічної компетентності» як складного особистісного утворення, яке включає в себе:

– розуміння комп’ютерних засобів як виконавців та конструкторів алгоритмів;

- здатність розробляти та реалізувати алгоритми для власної чи групової діяльності використовуючи послідовні дії, умови, повторення;
- навички створення плану дій, подання прикладів виконання завдань у повсякденній діяльності, вказівок для послідовних кроків, умов та повторень [63];
- здатність налаштовувати готові та створювати власні програмні проєкти, складати план для виконавця із точних та однозначних інструкцій, знаходити та виправляти помилки в цьому плані, оцінювати відповідність отриманого результату очікуваному;
- навички створення простих програм та їх налагодження; прогнозування та формулювання очікуваного результату виконання програми.

Складовими сформованості цієї компетентності обрали: когнітивний, мотиваційно-ціннісний, діяльнісний.

Критеріями для визначення рівня розвитку цієї компетентності обрали: когнітивний, мотиваційно-ціннісний та діяльнісний.

Когнітивний критерій описували використовуючи наступні показники: розуміння сутності алгоритмів, їх структури та методів втілення алгоритмічних структур у програмному середовищі; усвідомлення призначення виконавців, їх властивостей та особливостей налаштування.

У мотиваційно-ціннісному критерії використали такі показники: до застосування планування у повсякденній діяльності; бажання постійно удосконалюватися у програмуванні з використанням ІКТ [71].

Щодо діяльнісного критерію, визначили наступні показники: вміння розробляти, виправляти, удосконалювати та реалізовувати алгоритми у повсякденному житті та програмному середовищі; прояв ініціативи, самостійності та творчості під час активної роботи з алгоритмами.

Під час дослідження було визначено три рівні сформованості алгоритмічної компетентності учнів 10-11 класів: ознайомлювальний (низький), репродуктивно-пошуковий (середній), продуктивний (високий) [67].

Ознайомчий (низький) рівень сформованості алгоритмічної компетентності передбачає, що учень має базові уявлення про планування та

послідовність виконання дій, розрізняє способи подання алгоритмів: словесний і графічний; розуміє призначення та функції виконавця алгоритму; відрізняє програмне забезпечення, призначене для реалізації алгоритмів; може складати лінійні алгоритми.

На ознайомчому рівні виявляються такі елементарні вміння: самостійно запускати середовище програмування, складати у ньому програму за зразком та запускати її на виконання; вміти завершувати роботу і завантажувати зроблені проекти в потрібно папку.

Репродуктивно-пошуковий (середній) рівень алгоритмічної компетентності передбачає, що учень знає правила складання алгоритмів та вміє знаходити та виправляти помилки у послідовностях; має знання про основні функції виконавця програми та може змінювати вигляд об’єктів у середовищі; проявляє інтерес до виконання планування із застосуванням головних алгоритмічних конструкцій .

На цьому рівні сформованості алгоритмічної компетентності учень може самостійно або з невеликою допомогою виконувати налаштування об’єктів і короткочасні проекти у середовищі програмування, а також зберігати, завантажувати та повторно запускати власний продукт. Реалізує алгоритми з перевіркою умови та проявляє зацікавленість у створенні власних програмних проектів, розуміючи їхній потенціал. [70].

Продуктивний (високий) рівень передбачає, що учень самостійно використовує середовище програмування для створення в ньому власних алгоритмічних конструкцій та програмних проєктів. Він має знання основних алгоритмічних конструкцій та порядку виконання дій, а також виявляє прагнення до реалізації власної творчості у програмному просторі .

Учень даного рівня може оцінити поставлену задачу, спланувати алгоритм виконання завдання, розробити проєкт у середовищі програмування, знаходити та виправляти помилки, а також налаштовувати властивості об’єктів відповідно до творчого задуму. Він проявляє активність у виконанні як індивідуальних, так і командних творчих проєктів.

Перевірка рівня сформованості в учнів 10-11 класів алгоритмічної компетентності за показниками когнітивного критерію відбувалася під час використання тестових завдань та в середовищах програмування Scratch, де здобувачі освіти відповідали на теоретичні та практичні завдання. Тестування в навчальному матеріалі містить розділ «Алгоритми» 10-го-11-го класу та різноманітні типи запитань. [68].

Для проходження тесту учням відводилося 25 хвилин часу. Максимальна кількість балів за тест – 15. Учні, які набрали від 1 до 5 балів, віднесені до низького (ознайомлювального) рівня; ті, хто отримав від 6 до 10 балів, віднесені до середнього (репродуктивно-пошукового) рівня; та ті, хто набрав від 11 до 15 балів, віднесені до високого (продуктивного) рівня. Діагностика рівня сформованості алгоритмічних навичок обох підгруп учнівського складу, на початок дослідження за показниками когнітивного критерію показала результати, які поданні в табл. 2.1.

Таблиця 2.1

Рівень сформованості алгоритмічної компетентності учнів ліцеїв з інформатики за критеріями ї складової на початок експерименту

Рівні	Контрольна група		Експериментальна група	
	Кількість	%	Кількість	%
Ознайомлювальний	4	31	3	37
Репродуктивно-пошуковий	8	50	9	44
Продуктивний	3	19	4	19
Загальний показник	16	100	16	100

Сформованість алгоритмічної компетентності за показниками мотиваційно-ціннісного критерію перевірялася за допомогою анкетування. За основу було взято анкету для оцінювання рівня шкільної мотивації Н. Г. Лускановой. Анкета містить 10 запитань [69]:

1. Тобі подобається програмувати?

а) так;

б) не дуже;

в) ні.

2. Ти завжди з радістю йдеш на урок інформатики чи тобі часто хочеться залишитися вдома?

а) іду з радістю;

б) буває по-різному;

в) частіше хочеться залишитися вдома.

3. Якби вчитель сказав, що завтра на урок програмування не обов'язково приходити всім учням, ти б пішов до школи чи залишився б вдома?

а) пішов би на урок;

б) не знаю;

в) залишився б удома.

4. Тобі подобається, коли відміняють урок інформатики?

а) не подобається;

б) буває по-різному;

в) подобається.

5. Ти хотів би, щоб тобі не задавали ніяких домашніх завдань з програмування?

а) не хотів би;

б) не знаю;

в) хотів би.

6. Ти хотів би, щоб у школі залишилися лише перерви?

а) ні;

б) не знаю;

в) хотів би.

7. Ти часто розповідаєш про свої проєкти батькам і друзям?

а) часто;

б) рідко;

в) не розповідаю.

8. Ти хотів би, щоб у тебе був інший, менш суворий учитель?

а) мені подобається наш учитель;

точно не знаю;

в) хотів би.

9. У тебе в класі багато друзів, до яких можна звернутися за допомогою у розв'язанні завдання?

а) багато;

б) мало;

в) немає друзів.

10. Тобі подобається виконувати програмні проєкти у команді з однокласниками?

а) подобається;

б) не дуже;

в) не подобається.

Обробка результатів

За кожну відповідь «а» – 3 бали, варіант «б» – 1 бал, варіант «в» – 0 балів.

21–30 балів – ставлення до себе сформовано високо, рівень мотивації позитивний, активна навчальна діяльність.

11–20 балів – ставлення до себе на достатньому рівні. Середній рівень активності та мотивації

1–10 балів – ставлення до себе не сформоване. Відсутня мотивація, низька навчальна діяльність.

За результатами опитування учнів, які отримали від 1 до 10 балів, віднесено до низького (ознайомлювального) рівня; ті, хто набрав від 11 до 20 балів, віднесено до середнього (репродуктивно-пошукового) рівня; а ті, що отримали від 21 до 30 балів, віднесено до продуктивного рівня за мотиваційно-ціннісним критерієм.

Показники сформованості алгоритмічної компетентності обох підгруп на початку експерименту з мотиваційно-ціннісного критерія, внесені в табл. 3.2.

Таблиця 3.2

Рівень сформованості алгоритмічної компетентності учнів старшої ланки з інформатики за критеріями ї складової на початок експерименту

Рівні	Контрольна група		Експериментальна група	
	Кількість	%	Кількість	%
Ознайомлювальний	2	13	3	19
Репродуктивно-пошуковий	9	56	8	50
Продуктивний	5	40	5	31
Загальний показник	16	100	16	100

Рівень сформованості в учнів ліцею алгоритмічної компетентності за критеріями ї складової перевірявся із використанням комплексних практичних завдань, які було розроблено з урахуванням пройденого матеріалу змістової лінії «Алгоритми» відповідно до програми з інформатики для 9 класу.

Результати діагностики рівнів сформованості алгоритмічної компетентності обох підгруп учнів на початок експерименту за показниками діяльнішого критерію показала результати, представлені в табл. 3.3.

Таблиця 3.3

Рівень сформованості алгоритмічної компетентності учнів з інформатики за критеріями ї складової на початок експерименту

Рівні	Контрольна група		Експериментальна група	
	Кількість	%	Кількість	%
Ознайомлювальний	9	31	3	37
Репродуктивно-пошуковий	4	50	6	44
Продуктивний	4	19	7	19
Загальний показник	16	100	16	100

Результати дослідження рівнів сформованості алгоритмічної компетентності за критеріями її складової показав результати аналогічні до

показників рівнів сформованості алгоритмічної компетентності за показниками когнітивного критерію.

Узагальнивши результати діагностики рівнів сформованості алгоритмічної компетентності обох підгруп учнів на констатувальному етапі експерименту, ми отримали результати, представлені в табл. 3.4.

Таблиця 3.4

Зведені результати діагностики рівня сформованості алгоритмічної компетентності учнів 10-11-х класів на констатувальному етапі експерименту

Рівні	Контрольна група		Експериментальна група	
	Кількість	%	Кількість	%
Ознайомлювальний	3	25	4	31
Репродуктивно-пошуковий	8	50	7	44
Продуктивний	5	25	8	25
Загальний показник	16	100	16	100

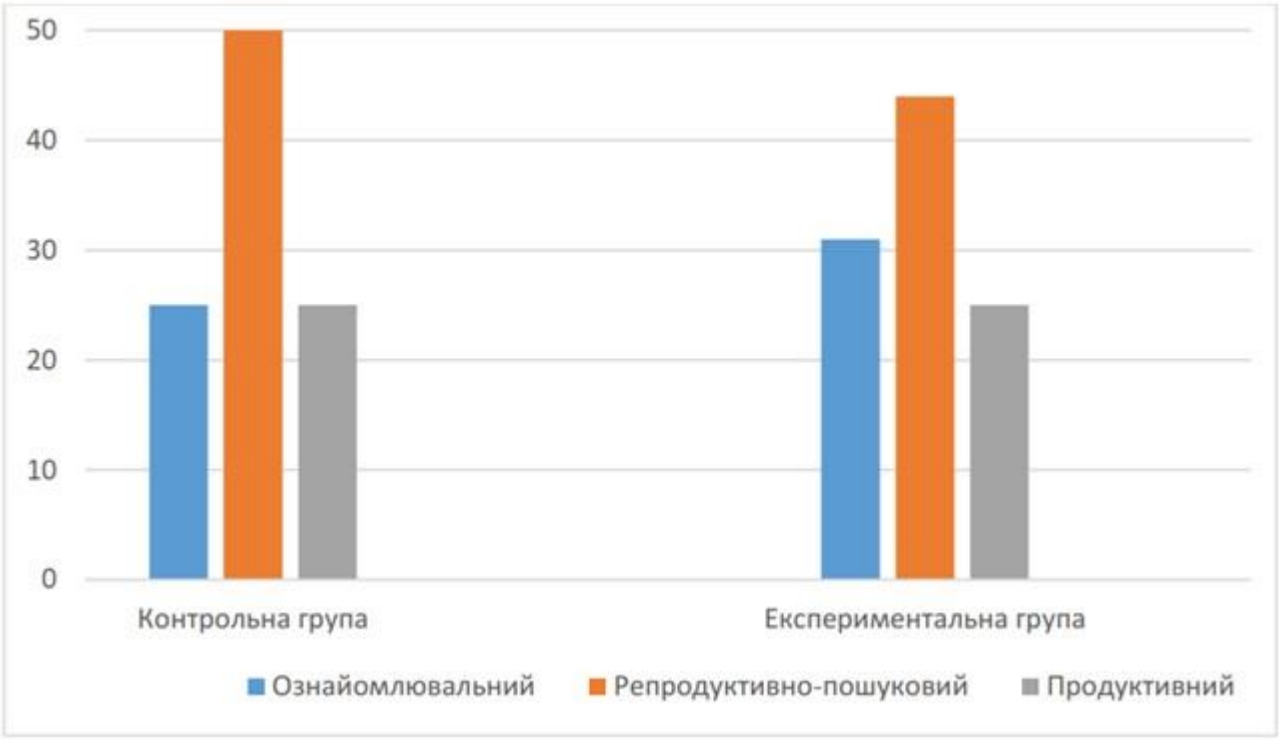


Рис. 3.1. Діаграма зведених результатів діагностики констатувального етапу експерименту

Графічно зведені результати діагностики рівнів сформованості алгоритмічної компетентності обох підгруп учнів на констатувальному етапі експерименту за трьома критеріями, представлені на рис. 2.1.

За результатами дослідження, більшість учнів продемонстрували знання про методи проектування алгоритмів, розуміють базові алгоритми та можуть пояснити структуру алгоритму і реалізувати засобами мови програмування.

Результати першого етапу дослідження вказують на достатній рівень розвитку алгоритмічної компетентності учнів. У дітей відсутні достатні знання для вироблення вмінь та навичок у галузі програмування. Вони можуть самостійно працювати у програмному середовищі, але під наглядом вчителя, володіють навичками планування навчального часу для виконання проектів. Учні виявляють бажання створювати власні ігрові проекти, але досягти цього можуть на постійні практиці. Ця ситуація пов’язана з достатнім цифровим і методичним забезпеченням навчання, візуальний та інтерактивний електронний контент.

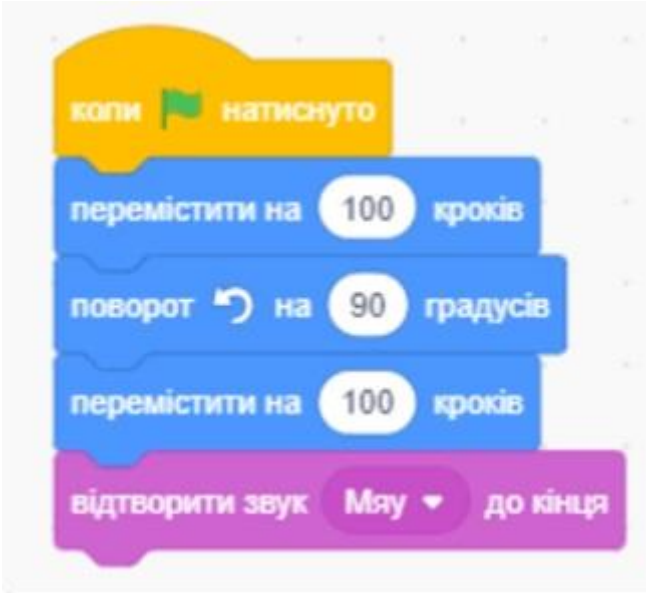
3.2. Результати експерименту

Отримані результати дають змогу зробити висновок, що у сучасному ліцею приділяється достатня увага для формування алгоритмічної компетентності учнів 10-11-х класів. Вони можуть орієнтуватися у навчальному матеріалі нової теми, відповідати на запитання вчителя, самостійно виконувати практичні та лабораторні роботи, ґрунтовно аналізують отримані результати.

Завдання практичної частини уроку розроблені з поступовим нарощуванням складності, від шаблонних завдань на відтворення до творчих індивідуальних завдань.

Прикладом завдання на відтворення послідовностей згідно зразка є наступне завдання: «відтворіть у програмному середовищі алгоритм, що зображено на рис. 3.2.

а)



б)

Команда	Призначення
перемістити на 10 кроків	Перемістити спрайт
наступний образ	Змінити образ спрайта
чекати 1 секунд	Затримати зображення на сцені
якщо на межі, відбити	Відбити від краю сцени
стиль обертання зліва-направо	Забезпечити правильне відбивання образу

Рис. 3.2. Зображення фрагменту коду – а) та набір блоків для виконання завдання – б)

Прикладом завдання репродуктивного рівня є наступне завдання: «складіть із запропонованих блоків програму таким чином, щоб виконавець переміщувався по сцені змінюючи образ» (рис. 3.2б).

Практичні завдання продуктивного рівня містили в собі окреслення бажаного результату без жодних підказок щодо його виконання «Розробіть анімацію діалогу на 4 репліки між двома спрайтами у середовищі програмування»

Для реалізації рефлексії та закріплення вивченого матеріалу використано інтерактивні вправи сервісу LeammgApps. (рис. 3.3 та рис. 3.4.).

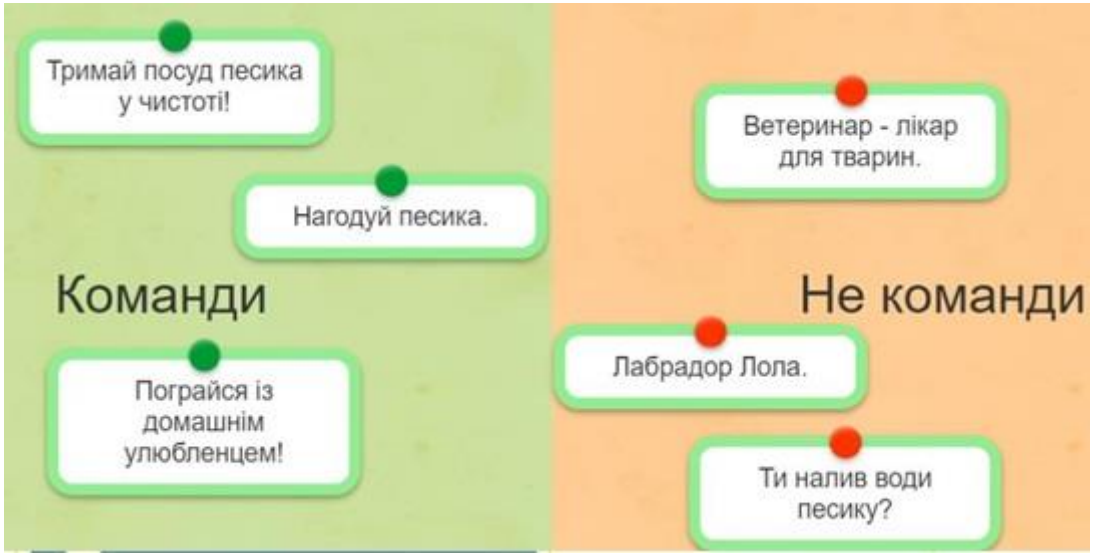


Рис. 3.3. Інтерактивні вправи для закріплення матеріалу уроку «План, інструкція, команди»

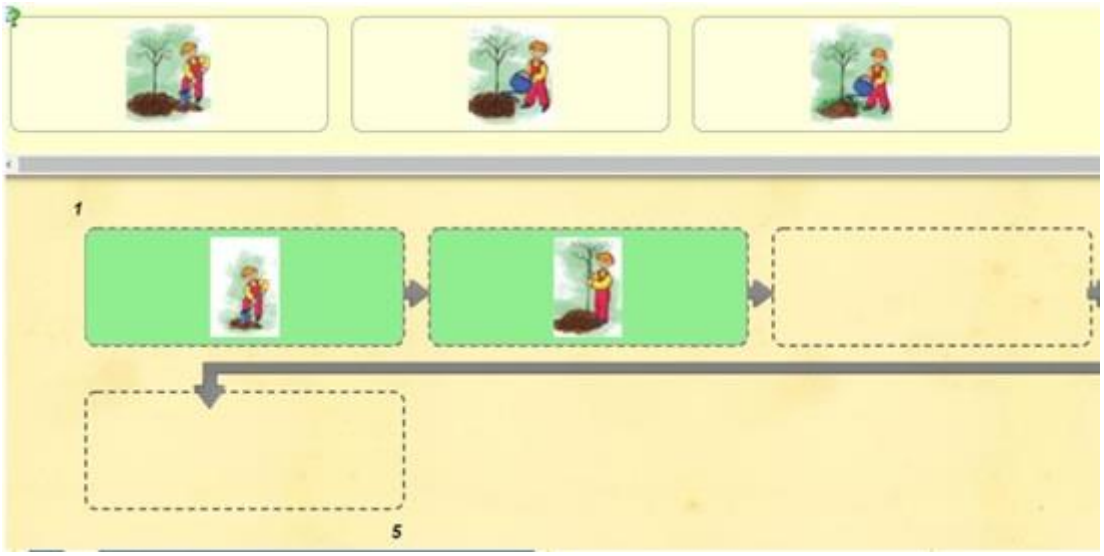


Рис. 3.4. Інтерактивні вправи для закріплення матеріалу уроку «Лінійні алгоритми»

Разом із використанням електронних технологій для навчання «Алгоритмів» активно використовувалось конструювання з Lego та творчими засобами, оскільки алгоритмізація та програмування є абстрактним видом конструювання деяких процесів в онлайн просторі. Також, на розробленому ресурсі розміщено всі необхідні для навчання методичні матеріали. У

відповідних розділах міститься тематичне планування змістової лінії для зазначеного класу та посилання на електронні ресурси.

Для розвитку здібностей до програмування мобільних застосунків як засіб навчання основ алгоритмізації учнів ліцеїв використовувати середовище MIT App Inventor (режим «Блоки») (рис. 3.5). Для цього впровадили режим «Перегляд доступних для вибраної групи блоків. Натиснувши на бажану групу, відкривається список блоків, які доступні в цій групі . Дана область відкривається після вибору одної доступної групи . Режим «Поле» – місце, де можна складати, редагувати та переглядати зібрані конструкції з блоків.

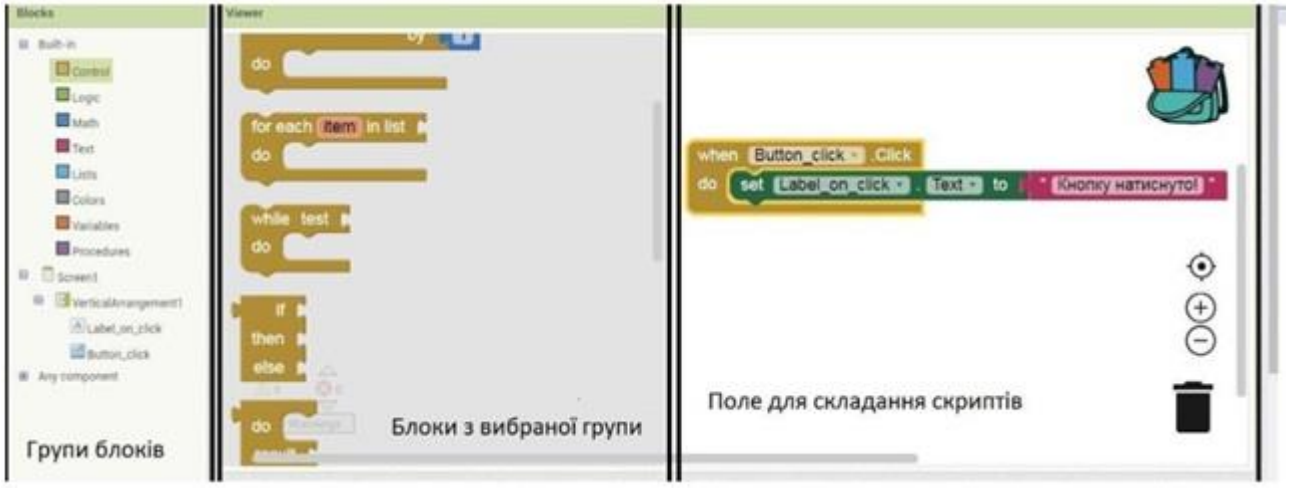


Рис. 3.5. Робоче середовище MIT App Inventor в режимі «Розробник»

Блоки в App Inventor – це засоби для програмування поведінки об’єктів у вигляді пазлів. Режими в цьому конструкторі розроблені для Android і розбиті на дві великі групи за ознакою, на що вони впливають і до чого відносяться, які прив’язані з компонентами та загальними блоками.

Завдяки блокам, пов’язаних із компонентом, можна змінювати властивості, події чи обробники того чи іншого об’єкта.

Розглянемо принцип роботи із розділом Button (Кнопка), який розпізнає натискання. Вигляд кнопки на початковому етапі можна змінити. Це реалізується або в редакторі дизайну, або у кодуванні за допомогою блоків. Наприклад, можна змінити колір фону, тип шрифту, записаний на кнопці текст, колір тексту, форму тощо, щоб надати кнопці більш сучасного вигляду.

Поведінку кнопки можна регулювати за допомогою обробки подій. Головними для кнопки є:

Click – натискання кнопки;

LongClick – довге натискання;

TouchDown (TouchUp) – кнопка натиснута (відпущена).

Існує задача – після натискання користувачем кнопки, програма має відтворити текст, що зберігається у написі. Оскільки це подія, то додаємо блок Click. (див Рис 3.6) Потім даємо команду програмі яка необхідна іншому компоненту – Text To Speech, щоб той озвучив текст. Цього разу, знадобиться блок-команда бузкового кольору. І на останньому етапі дізнаємося, який текст зберігається у компоненті Label, подивитися на його властивість (зелений колір).

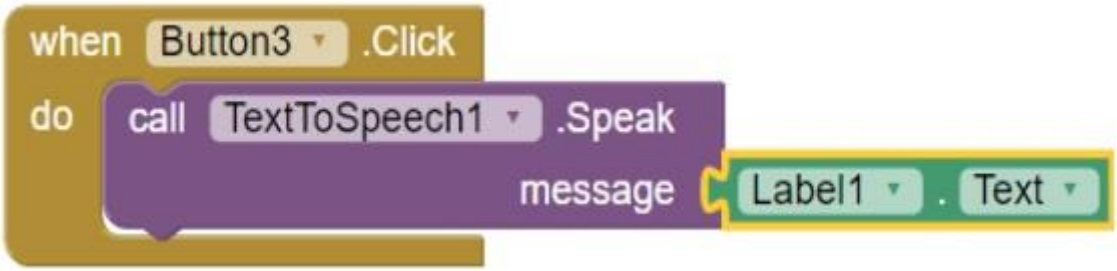


Рис. 3.6. Код в середовищі AI2

Різні кольори блоків допомагають швидко засвоїти значення блоків та легко зрозуміти їх призначення, крім того складання кольорових скриптів посилює зацікавленість учнів до навчання.

Опишемо дії учнів при програмуванні елементів школярами в ліцеї.

Перейди в робочий режим із блоками.

Вибери в групі блоків елемент першої кнопки та перестав на поле цей блок і запусти вкладений скрипт, натиснувши на кнопку.

Прикріпи в середину доданого блоку умовний блок «if-else». Після того, як користувач натисне на першу кнопку, в нього може бути два сценарії розвитку подій:

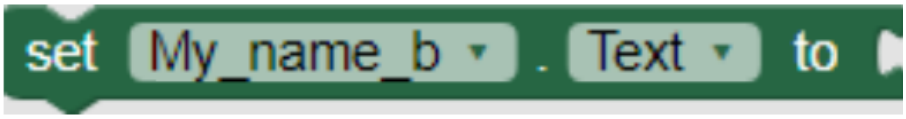
а) якщо інформації на екрані немає, тоді необхідно зробити текст видимим, а текст на кнопці змінити на «Сховати ім’я»;

б) якщо інформація на екрані вже є, тоді можна її сховати, а текст на кнопці змінити на «Як мене звати».

Рис. 3.7. Черговість дій під час програмування елементів

Необхідно буде використати умовний блок , який можна знайти в групі.

Далі буде заміна тексту на кнопці на «Сховати ім’я» або «Як мене звати?», для цього використай такий блок:



В даний блок необхідно вставити інший блок, який зміг би містити в собі інформацію текстового типу. Для таких операцій використовуються блоки з групи **Text** . Натиснувши на дане поле, можна змінити текст на свій. Тобі потрібні два таких блоки, які необхідно вкласти в умовний блок.

Додавання функціоналу для записів.

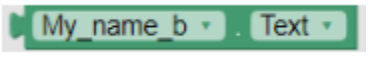
Складений запис з’являється або зникає в залежності від того чи потрібно показати інформацію чи сховати її. Для того, щоб робити текст видимим або ні, потрібно додати такий блок:

Так як видимість може бути лише або true або false (логічна змінна), то попередній блок нам потрібно зв’язати з блоком з групи **Logic**.

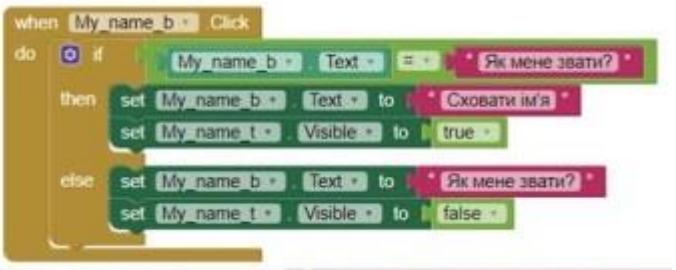
Як і в попередньому кроці, потрібні теж два варіанти:

та

Додавання умови. В умовні блоки потрібно додати відповідну дію. У даному випадку будемо перевіряти текст, що написаний на кнопці. Перевірка здійснюється за допомогою логічного порівняння. Цей блок можна знайти тут.

Вже відомі для школяра блоки 

та  , які йому потрібно вкласти в середину цього блоку такий об’єкт, щоб утворився такий скрипт:



Повторивши натискання попередніх кнопок можна закодувати іншу кнопку та напис.

Можна скопіювати попередній скрипт та змінити тільки потрібну інформацію. Змінюючи текст на кнопці з «Скільки мені років?» на «Сховати вік» або навпаки та відповідно робити другий видимим або невидимим.

Випробування проєкту. Для цього, завантажуємо на смартфон додаток MIT AI2 Companion з Play Market. Відкриваємо його, та натискаємо обери «Scan QR code». Відкриваємо меню Connect в середовищі MIT APP Inventor на комп’ютері та вибираємо зі списку AI Companion:

Відсканувавши за допомогою смартфону код, зачекаємо та завантажимо додаток.

В якості рекомендації по навчанню програмування, як засобу навчання основ алгоритмізації учнів ліцеїв можемо використати конструктор Lego Mindstorms.

Для програмування роботів використовується середовище Mindstorms Education NXT, яке дозволяє використовувати всі базові структури програмування для написання програми керування роботом. Для визначення рівня своїх умінь учасники гуртка постійно беруть участь у різноманітних фестивалях та змаганнях роботів.

Щоб проводити програмування робота, слід розміщувати блоки функціональності в певній послідовності. Залежно від типу блоку, кожен блок може бути налаштований. Наприклад «середній мотор» має 5 режимів роботи: 1 – вимкнути, 2 – включити і обертати, 3 – включити протягом певної кількості секунд, 4 – включити і повернути на певний градус, 5 – включити і повернути фіксоване число разів.

Рис. 3.8. Вікно вибору проєкту

В наборі конструктора є широкий спектр програмних блоків на вибір. Вони згруповані в шість категорій: 1 – дія (зелений), 2 – управління потоком (помаранчевий), 3 – датчики (жовтий), 4 – операції над даними (червоний), 5 – додаткові (синій), 6 – мої блоки (ціановий).

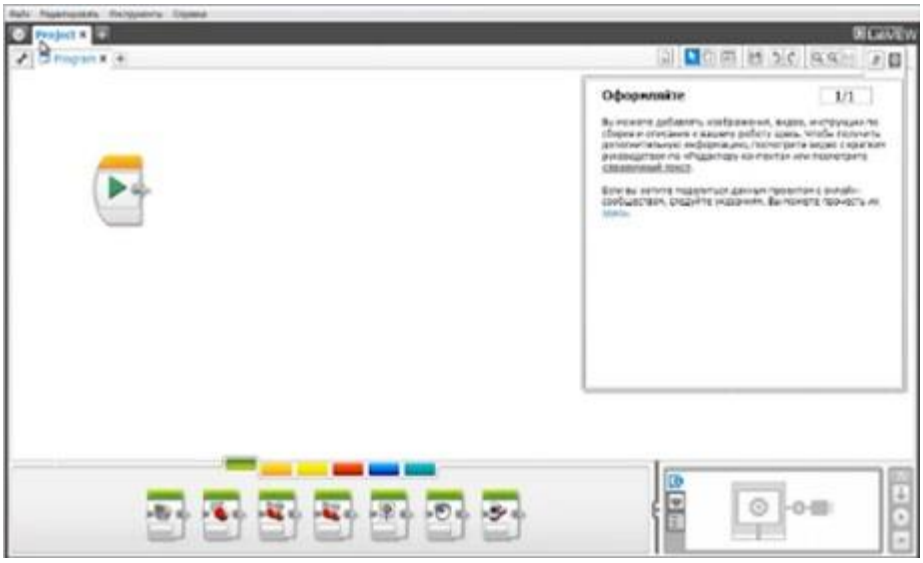


Рис. 3.9. Перший блок програми

Інтуїтивно зрозумілий інтерфейс дозволяє спочатку створювати прості програми, а потім продуктивно розвивати свої навички програмування, роблячи можливим створення складних багаторівневих програм і проведення різної експериментальної роботи.

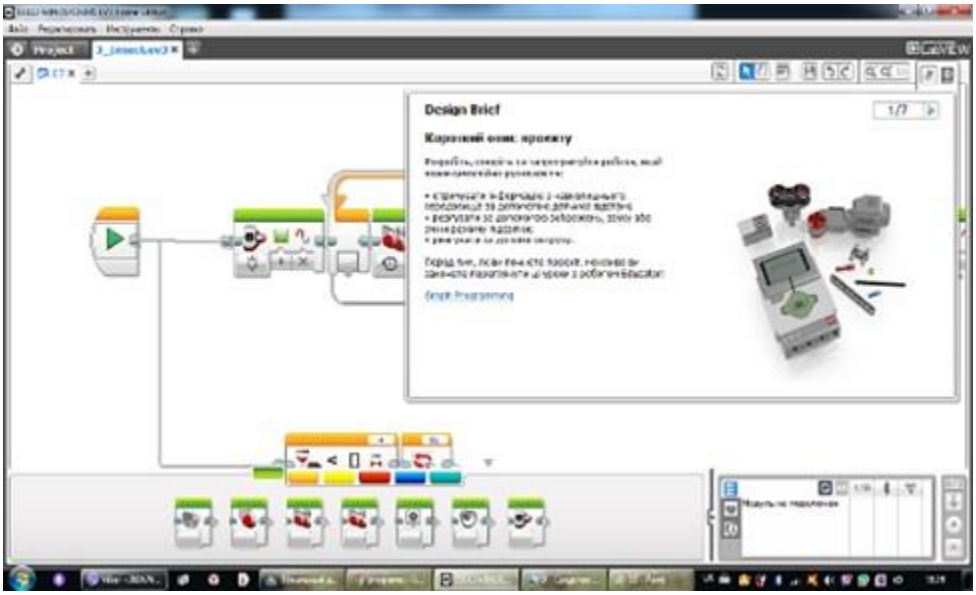


Рис. 3.10. Робота з блоками

На сайті LEGO у вільному доступі є безліч інструкцій. Підключення робота до ПК здійснюється кількома способами: через порт USB, Bluetooth з'єднання або Wi-Fi з'єднання. Використання порту USB є найбільш зручним, адже в цьому випадку робот прив'язаний до комп'ютера і програму на виконання можна запускати прямо з середовища програмування.

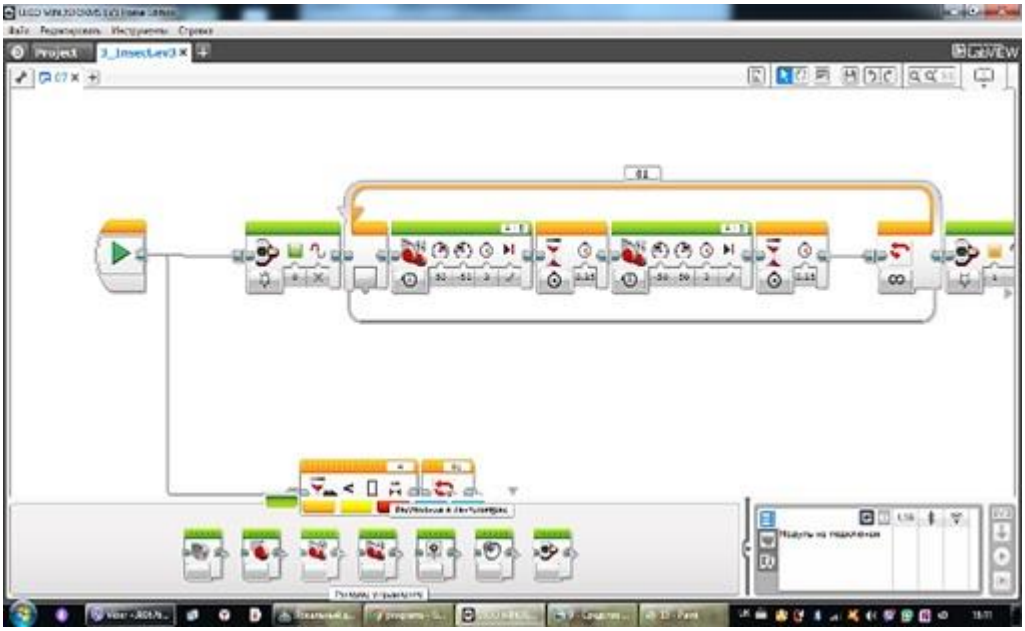


Рис. 3.11. Набір блоків датчиків в програмі Lego Mindstorms HomeEdition

Крім того, під час виконання програми з’являється можливість візуально контролювати хід її виконання (заголовки виконуються в даний момент програмних блоків будуть мерехтіти), можемо відстежувати на комп’ютері. Також можна спостерігати поточні показання датчиків весь час, поки робот залишається підключеним до середовища програмування.

З EV3 в комплекті поставляється Нова графічне середовище розробки на базі LabView, схожа на NXT-G. працювати вона буде, як і NXT-G, на ОС Windows і Mac.

Наведемо приклад написання нами програми прямолінійного руху для проїзду роботом відстані в 1 метр. За один повний оберт сервомотора робот проїжджає відстань, рівну довжині кола колеса. Діаметр колеса з освітнього набору Lego mindstorms дорівнює 56 мм. Якщо переведемо відстань в систему СІ і розділимо на відстань, яку робот проходить за один оберт сервомотора, то дізнаємося: скільки обертів сервомотора необхідно для проїзду заданої відстані.

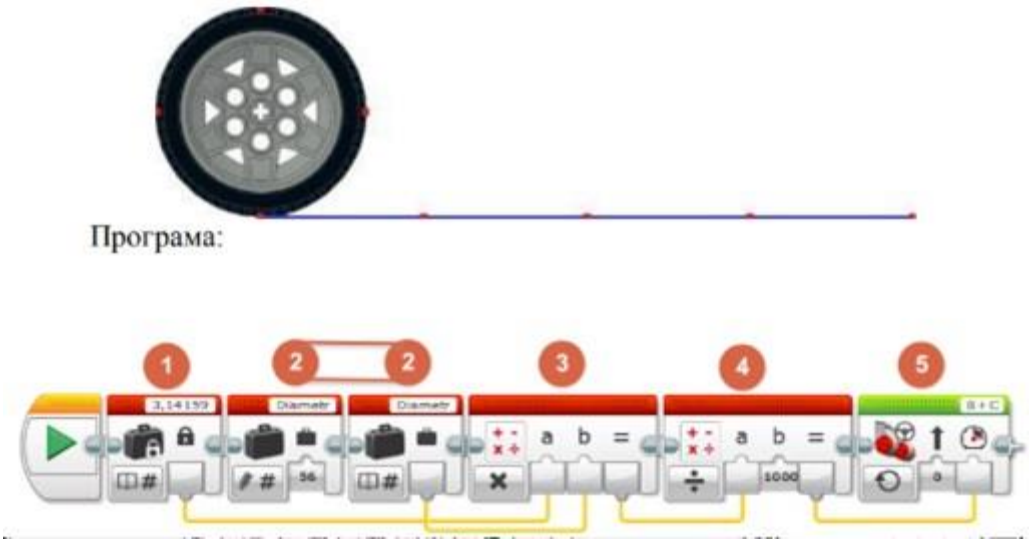


Рис. 3.12. Схематичне зображення порядку формування задачі

Також ми дослідили зміну інтересу у ліцеїстів при вивченні основ алгоритмізації з використанням різних засобів. Так завдання № 1–9 включали використання Scratch, завдання № 10–15 включали використання MIT App Inventor, завдання № 15–20 включали використання Lego Mindstorms HomeEdition. Обробку результатів проводили за допомогою комп'ютера з використанням програми Microsoft Word. Графічну інтерпретацію результатів проводили за допомогою програми – Excel.

Таблиця 3.5

Тестування інтересу контрольної групи ліцеїстів (10 клас) 4.01.2023 р.

	Номер завдання / Прізвище ліцеїста	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	Бобик Іван	+	-	+	-	+	+	-	+	+	+	-	-	+	-	+	-	+	+	+	-
2	Бобич Богдан	-	+	+	+	+	-	-	+	-	+	+	-	+	-	-	+	-	-	+	+
3	Бобота Антон	+	+	+	+	-	+	+	-	+	-	+	-	+	-	+	-	+	+	+	-
4	Бердарь Олена	-	+	+	-	+	-	+	+	-	-	+	+	-	+	+	-	+	+	-	+
5	Ворохта Марина	+	-	+	-	+	+	-	+	-	+	+	+	-	-	+	-	+	+	+	-
6	Гуловко Надія	+	+	-	+	+	+	-	-	+	-	-	+	-	-	+	+	+	-	+	-
7	Гедзур Петро	+	+	-	+	-	+	+	+	+	-	-	+	-	+	-	+	-	-	-	-

8	Глеба Ярослава	-	+	+	-	-	+	-	+	-	-	+	-	-	+	-	-	+	-	+	+
9	Канайло Оксана	+	-	-	+	-	+	+	-	+	-	-	+	-	-	-	+	-	-	-	-
10	Керецман Юрій	-	+	-	+	+	-	-	-	-	+	-	-	+	-	-	-	+	-	-	+
11	Карпухін Ілля	+	+	-	+	+	+	-	+	-	+	-	+	-	+	-	+	-	-	+	-
12	Кинч Іван	-	+	+	+	-	+	+	-	+	-	-	+	-	-	+	-	+	-	+	+
13	Лало Марія	+	+	-	-	+	-	+	-	-	+	+	-	-	+	-	+	-	+	-	+
14	Усик Інна	-	+	-	+	+	-	-	+	+	-	-	+	+	+	-	-	-	-	-	-
15	Ендрик Наталія	+	-	+	-	+	-	-	+	-	+	-	-	+	-	+	-	+	-	+	+
16	Антал Вероніка	+	+	+	-	-	+	+	+	-	-	+	-	-	+	-	+	-	+	-	-

В таб. 3.6 представлені результати тестування експериментальної групи 4.06.2023 р. ліцеїстів.

Таблиця 3.6

Тестування інтересу контрольної групи ліцеїстів (10 клас) 4.06.2023 р.

	Номер питання / Прізвище учня	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	Бобик Іван	+	-	+	-	+	+	+	+	+	+	-	+	+	-	+	-	+	+	+	-
2	Бобич Богдан	-	+	+	+	+	+	-	+	-	+	+	-	+	-	-	+	-	-	+	+
3	Бобота Антон	+	+	+	+	-	+	+	+	+	-	+	-	+	-	+	-	+	+	+	-
4	Бердарь Олена	-	+	+	-	+	-	+	+	-	-	+	+	-	+	+	-	+	+	-	+
5	Ворохта Марина	+	-	+	-	+	+	-	+	-	+	+	+	-	+	+	-	+	+	+	-
6	Гуловко Надія	+	+	-	+	+	+	-	-	+	+	-	+	+	-	+	+	+	-	+	-
7	Гедзур Петро	+	+	-	+	-	+	+	+	+	-	+	+	-	+	-	+	-	-	-	-
8	Глеба Ярослава	-	+	+	-	+	+	-	+	+	-	+	-	-	+	-	-	+	-	+	+
9	Канайло Оксана	+	-	+	+	-	+	+	-	+	-	-	+	-	+	-	+	-	-	-	-
10	Керецман Юрій	-	+	+	+	+	-	-	+	-	+	-	+	+	-	-	-	+	-	+	+

11	Карпухін Ілля	+	+	-	+	+	+	-	+	+	+	-	+	-	+	-	+	-	-	+	-
12	Кинч Іван	+	+	+	+	-	+	+	-	+	-	-	+	-	-	+	-	+	+	+	+
13	Лало Марія	+	+	+	-	+	-	+	-	-	+	+	+	-	+	-	+	+	+	-	+
14	Усик Інна	+	+	-	+	+	-	+	+	+	-	-	+	+	+	-	+	+	-	+	+
15	Ендрик Наталія	+	-	+	-	+	-	-	+	-	+	-	-	+	-	+	-	+	-	+	+
16	Антал Вероніка	+	+	+	-	-	+	+	+	-	+	+	-	-	+	-	+	-	+	+	+
17	Байчер Андріана	-	+	+	+	+	+	-	+	+	+	+	+	-	+	+	-	+	-	+	-

В таб. 3.7 представлено узагальненні результати по виборці позитивних відповідей ліцеїстів на питання, що з’ясовують зміну рівень інтересу до різнопланових завдань (№ 1–20) при вивченні основ алгоритмізації на початку і в кінці навчання.

Таблиця 3.7

Зміна степені інтересу ліцеїстів до різнопланових завдань (№ 1–20) при вивченні основ алгоритмізації на початку і в кінці навчання

	Степені інтересу учнів до предмету	**4.01.2023 р**	**4.06.2023 р**
1	Ситуативний інтерес (питання 1–5)	36	42
2	Навчання за необхідністю	45	40
3	Стійкий інтерес	33	60
4	Підвищений пізнавальний інтерес	40	54

Графічну інтерпретацію отриманих експериментальних результатів представлено на рисунку 3.13.

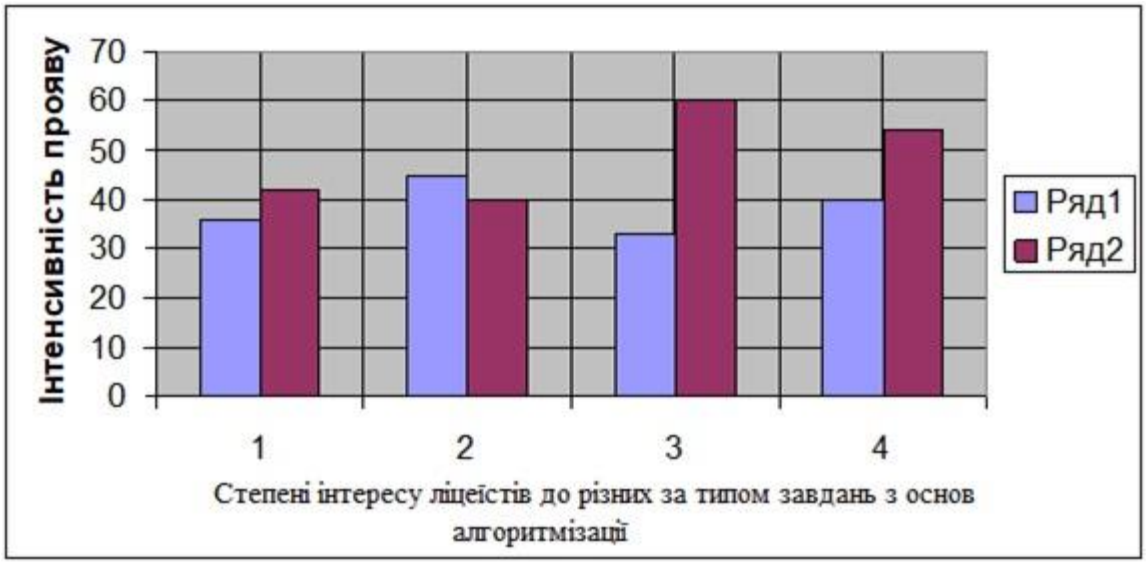


Рис. 3.13. Схематичне зображення зміни степені інтересу ліцеїстів до різнопланових завдань (№ 1–20) при вивченні основ алгоритмізації на початку і в кінці навчання

Ряд 1. Результати тестування 4.01.2023 р.

Ряд 1. Результати тестування 4.06.2023 р.

1 – Ситуативний інтерес; 2 – Навчання за необхідністю; 3 – Стійкий інтерес; 4 – Підвищений пізнавальний інтерес.

Динаміку пізнавального інтересу ліцеїстів наведено в таблиці 2.8.

Таблиця 3.8

Динаміка пізнавального інтересу ліцеїстів на початок та кінець тестування

Форми прояву зацікавленості учнів до предмету	*Результати (учні)*	
	4.01.2023 р	**4.06.2023 р**
1. Відсутність інтересу	**10**	**5**
2. Реакція на новизну	**17**	**19**
3. Цікавість	**17**	**22**
4. Ситуативний навчальний інтерес	**8**	**15**
5. Стійкий навчально-пізнавальний інтерес	**4**	**8**
6. Узагальнений навчально – пізнавальний інтерес	**-**	**3**

Графічну інтерпретацію отриманих експериментальних результатів представлено рис. 3.14–3.18.

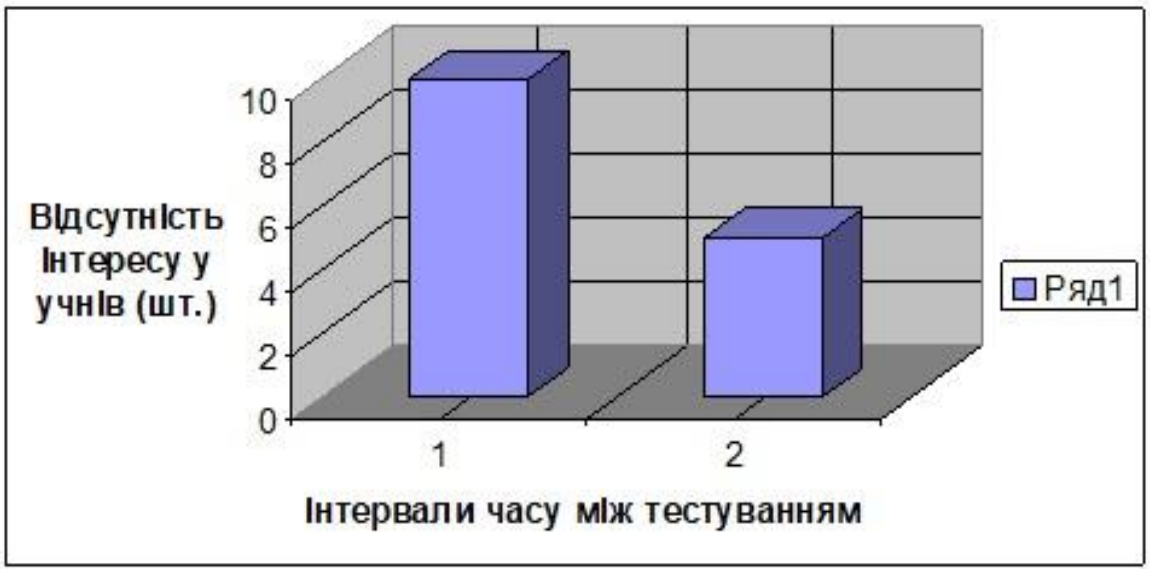


Рис. 3.14. Схематичне зображення рівня відсутності інтересу ліцеїстів до різнопланових завдань (№ 1–20) при вивченні основ алгоритмізації на початку і в кінці навчання (1–4.01.2023 р.; 2–4.06.2023 р.)

Рис. 3.15. Схематичне зображення рівня позитивної реакції на новизну ліцеїстів до різнопланових завдань (№ 1–20) при вивченні основ алгоритмізації на початку і в кінці навчання (1–4.01.2023 р.; 2–4.06.2023 р.)

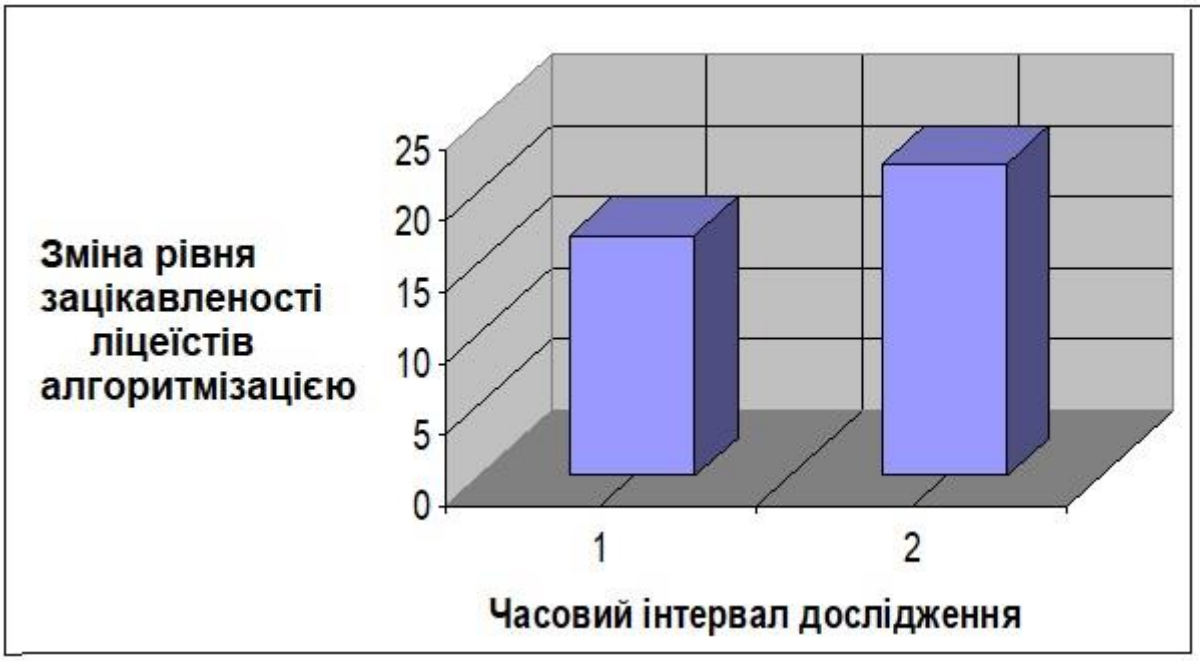


Рис. 3.16. Схематичне зображення зміну рівня зацікавленості у ліцеїстів до різнопланових завдань (№ 1–20) при вивченні основ алгоритмізації на початку і в кінці навчання (1–4.01.2023 р.; 2–4.06.2023 р.)

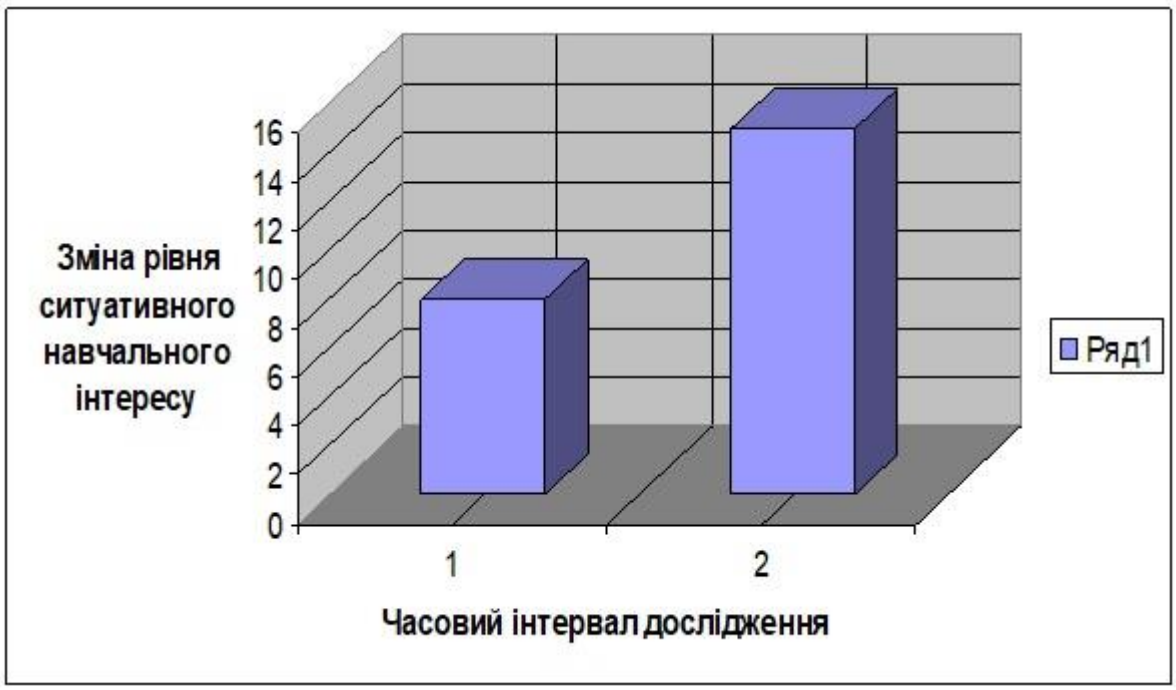


Рис. 3.17. Схематичне зображення зміну рівня ситуативного навчального інтересу ліцеїстів до різнопланових завдань (№ 1–20) при вивченні основ алгоритмізації на початку і в кінці навчання (1–4.01.2023 р.; 2–4.06.2023 р.)

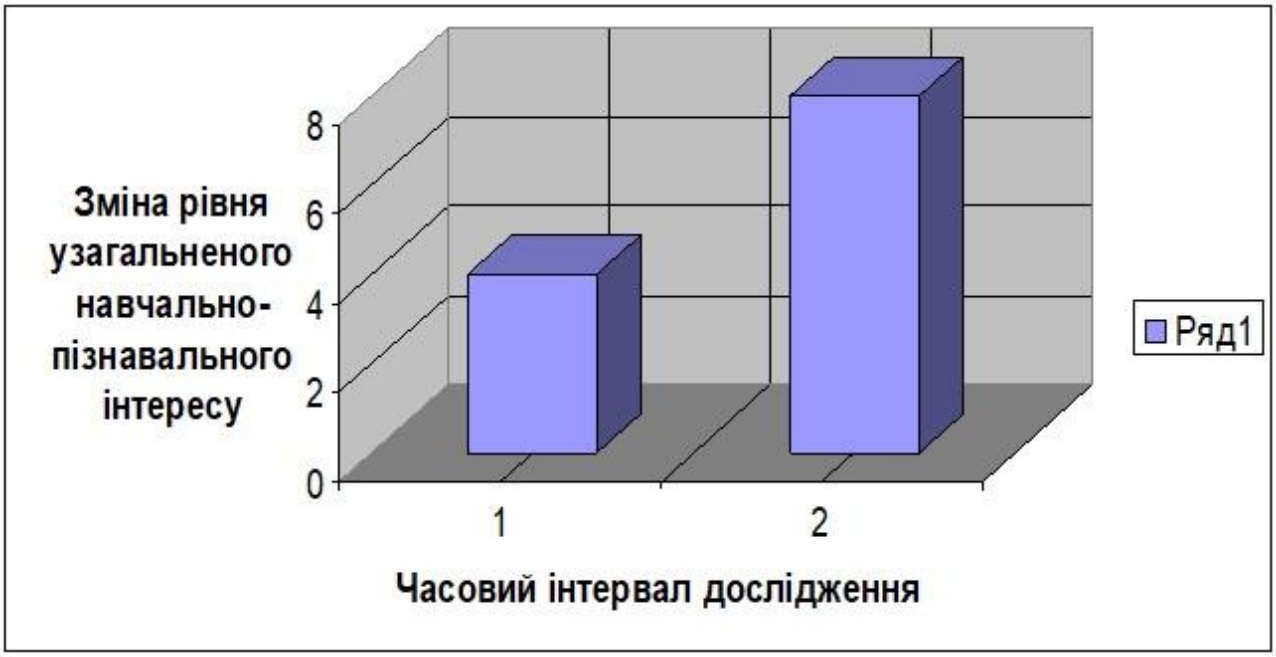


Рис. 3.18. Схематичне зображення зміни рівня узагальненого навчально-пізнавального інтересу ліцеїстів до різнопланових завдань (№ 1–20) при вивченні основ алгоритмізації на початку і в кінці навчання (1–4.01.2023 р.; 2–4.06.2023 р.)

Проведено також експериментальне дослідження виявлення переважаючих мотивів на уроках інформатики де вивчались основи алгоритмізації. Перелік питань наведено нижче, а графічне представлення отриманих результатів зроблено на рисунку 3.19.

Питання для аналізу переважаючих мотивів у навчанні на уроках інформатики.

У навчанні мною рухає:

1. Краще дізнатися свої можливості в даних предметах (інформатика, інформаційні технології)?

2. Інтерес до інтеграції предметів інформатики та робототехніки.

3. Бажання якомога більше дізнатися з областей даних наук?

4. Корисно, так як ці знання стануть в нагоді в майбутній роботі.

5. Ці предмети, міжпредметні зв’язки необхідні для подальшого навчання.

6. Впевненість в успіхах з даних предметів?

7. Легко зрозуміти досліджуваний матеріал на інтегрованих уроках?

8. Цікаво спілкуватися з товаришами на таких уроках.

9. Бажання мати авторитет серед товаришів, тому що ці предмети престижні в даному навчальному колективі.

10. Подобаються вчителі?

11. Бажання бути знаючим, освіченою людиною, цікавим для друзів.

12. Бажання бути готовим до самостійного життя.

13. Бажання бути духовно багатим, культурним і корисним для суспільства.

14. Як би не лаяли вчителі, батьки – це неприємно.

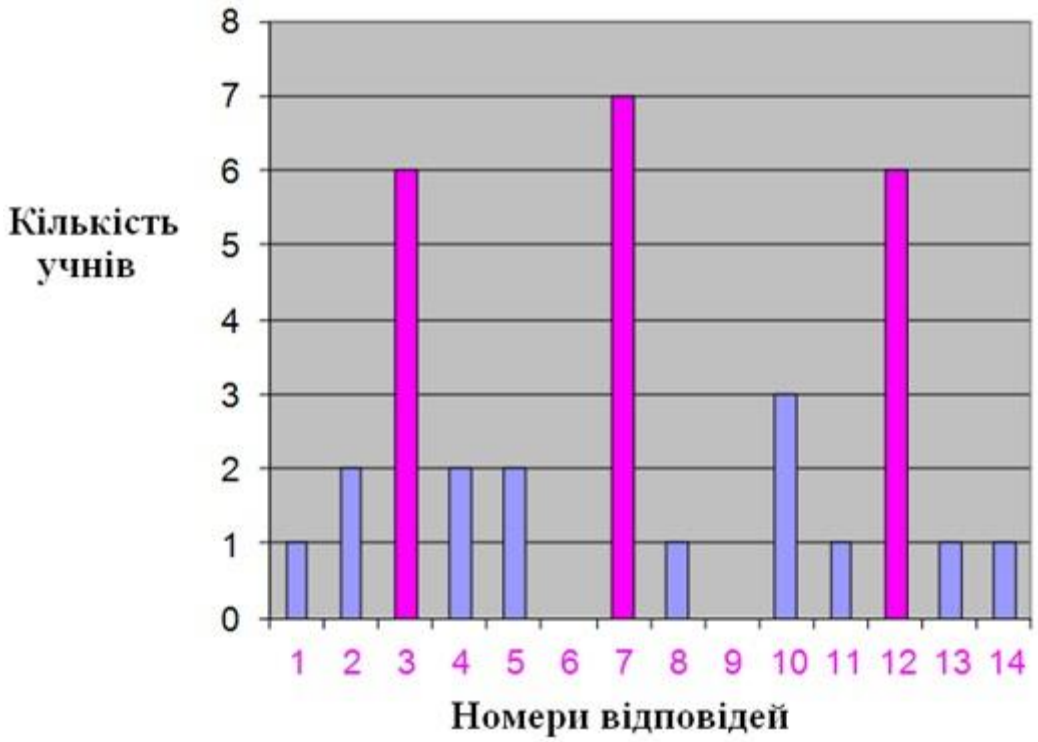


Рис. 3.19. Домінуючі мотиви навчання ліцеїстів основ алгоритмізації

На підставі отриманих результатів експериментального дослідження та аналізу наукової літератури рахуємо за доречне зробити такі пропозиції з покращення навчання основ алгоритмізації:

1. В начальному процесі є доречним використання інтегрованих уроків і в тому числі з інформатики та робототехніки.

2. Раціональним є поділ класу на дві групи, з якими проводяться подальші заняття з блочного програмування. Вчитель має можливість приділити більше часу індивідуальній роботі з кожним учнем.

3. Цікавим і корисним є впровадження на уроці міжпредметних зв’язків. На наших уроках, окрім традиційних підходів використано Scratch, завдання з участю MIT App Inventor та програмування в Lego Mindstorms HomeEdition.

4. Міжпредметний зв’язок дає можливість учням розвивати логічне мислення, вдосконалити знання.

5. Проведене експериментальне дослідження, протягом навчального року, засвідчує збільшення інтересу учнів до предмету інформатики (основи алгоритмізації) та засвідчує розуміння учнями взаємозв’язку даної науки з математикою, фізикою, біологією, медициною.

Висновки до розділу 3

У цьому розділі описано методику проведення педагогічного експерименту з дослідження ефективності впровадження розроблених методичних рекомендацій у навчальний процес Криворізької гімназії №8. Експеримент складався з констатувального, формувального та контрольного етапів. В рамках констатувального етапу експерименту виявлено рівні сформованості алгоритмічної компетентності в учнів 10-11 класів. Перевірка проводилася із використанням тестових завдань теоретичного та практичного характеру. У рамках формувального етапу в експериментальній групі вивчення основ алгоритмізації проводилося із застосуванням програмування мобільних застосунків, а в контрольній – традиційними засобами. Контрольний етап експерименту виявив зростання рівня сформованості когнітивної складової алгоритмічної компетентності в експериментальній групі. Також було досліджено динаміку зміни рівня зацікавленості ліцеїстів при вивченні основ алгоритмізації з використанням різних засобів програмування мобільних застосунків.

ВИСНОВКИ

Основною метою вивчення предмету «інформатика» є розвиток учнівських навичок в галузі інформаційних технологій для створення комп'ютерних інформаційних продуктів. Досягнення цієї мети передбачає впровадження в навчальний процес спеціально розробленої методичної системи завдань, які відтворюють реальні виклики, що виникають у різних сферах людської діяльності.

В роботі розглянуто методику викладання основ алгоритмізації в шкільному курсі інформатики та наголошено на тому, що розділ «Основи алгоритмізації та програмування» у шкільному курсі інформатики відзначається великим методологічним значенням. Цей розділ розкриває важливість алгоритмів та їхню роль у функціональному зв'язку між поняттями «інформація-алгоритм-комп'ютер», що визначає процес автоматичного опрацювання інформації. Метою вивчення алгоритмізації є формування знань та навичок учнів у сфері основних методів організації операцій і даних, а також використання базових алгоритмічних конструкцій для складання описів алгоритмів розв'язання різноманітних завдань.

Проаналізовано досвід використання мобільних технологій при викладанні основ алгоритмізації в шкільному курсі інформатики. Аналіз педагогічної практики та наукової літератури за напрямком дослідження дозволяє нам зробити висновок, що головним засобом реалізації мобільного навчання на уроках є смартфон.

Проведено детальне дослідження стосовно можливостей програмування мобільних застосунків для вивчення основ алгоритмізації на уроці інформатики. Серед усіх засобів програмування мобільних застосунків ми окремо виділили середовище MIT App Inventor через його структурну схожість з широко застосованим у шкільному навчанні середовищем Scratch, через його мобільність та наявність великої і активної спільноти користувачів у всьому світі, а також через те, що розробники постійно працюють над підтриманням актуальності середовища.

В практичні частині роботи зроблено аналіз досвіду використання програмування мобільних застосунків при викладанні основ алгоритмізації в шкільному курсі інформатики. Важливою умовою вдалого впровадження програмування мобільних застосунків при викладанні основ алгоритмізації в ліцеях у 10-11 класах є врахування психолого педагогічних особливостей учнів, які можуть працювати як команді, так й індивідуально. Зроблено добір засобів для використання програмування мобільних застосунків при викладанні основ алгоритмізації в шкільному курсі інформатики. Також проведено педагогічний експеримент на базі Криворізької гімназії №8, в рамках якого проведено перевірку розроблених методичних рекомендацій.

В рамках констатувального етапу експерименту виявлено рівні сформованості алгоритмічної компетентності в учнів 10-11 класів. Перевірка проводилася із використанням тестових завдань теоретичного та практичного характеру. Аналіз результатів показав зростання рівня сформованості когнітивної складової алгоритмічної компетентності в експериментальній групі. Також було досліджено динаміку зміни рівня зацікавленості ліцеїстів при вивченні основ алгоритмізації з використанням різних засобів програмування мобільних застосунків.

Таким чином можна зробити висновок про те, що програмування мобільних застосунків засобами MIT App Inventor може стати ефективним засобом навчання основ алгоритмізації, адже не потребує значних витрат часу на навчання символьних мов програмування, натомість пропонує порівняно знайоме та інтуїтивно зрозуміле середовище програмування, вивільняючи значну кількість навчального часу для формування алгоритмічної компетентності. Також програмування мобільних застосунків надає можливість наочно продемонструвати та надати можливість учням набути досвід у використанні об’єктно орієнтованого та подіє орієнтованого підходів до програмування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Абушкин Х. Х., Дадонова, А. В. Міжпредметні зв’язки в робототехніці як засіб формування ключових компетенцій учнів. *Навчальний експеримент в освіті.* 2014. № 3. С. 32–35.
2. Азарян А. А. , Карабут Н. О., Козикова Т. П., Рибольченко О. Г., Трачек А. А., Шаповалова Н. Н. Основи алгоритмізації та програмування: Навчальний посібник. Кривий Ріг: Вид-во ОктанПринт, 2014. 308 с.
3. Алгоритми та структури даних/уклад. О. В. Щербакова, Ю.Е. Парфьонов, В. М. Федорченко. Харків: ХНЕУ ім. С. Кузнеця, 2017. 58 с.
4. Андрєєв Д. В. Підвищення мотивації до вивчення програмування у молодших школярів в рамках курсу робототехніки. *Педагогічна Інформатика.* 2015. №1. С. 40–49.
5. Архітектура комп’ютерних систем: конспект лекцій для студентів усіх форм навчання з курсу «Архітектура комп’ютерних систем» / Укладачі: Голотенко О. С. Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2016. 120 с.
6. Аршулик Т. В. Дослідження методів інтерполяції зображень при створенні мультимедійного контенту. Тези доповідей Науково-практичної конференції *«Мультимедійні технології в освіті та інших сферах діяльності»,* Київ: НАУ, 2017. С. 13.
7. Ахметов А. К. Операційна система Android: історія створення і розвитку. Розробка додатків для платформи Android. Скіф. 2017. №9. С. 60-68.
8. Ащепкова Н. С. розробка адаптивної системи керування моделі роботанавантажувача на базі Lego Mindstorms NXT. Східно-Європейський журнал передових технологій. 2015. №5 (7) 77. С. 54-63.
9. Балабан Я. Р., Мороз І. О. Сутність мобільного навчання в освітньому процесі. Фізико-математична освіти, 2017. Вип.4 (14). С. 149-155.
10. Барна О. В. Технологія змішаного навчання в курсі методики навчання інформатики. Відкрите освітнє е-середовище сучасного університету: ел. наук. Видан. Київ, 2016, №2. С.24-37.

11. Березовська Ю. Академія Intel: введення в розробку додатків для ОС Android: [Електронний ресурс]. / Березовська Ю., Вологдіна В. НОУ «ІНТУЇТ», 2016. № 2. С. 45-53
12. Білоус В. Мобільні навчальні додатки в сучасній освіті. Освітологічний дискурс. 2018. № 1-2 (20-21). С. 353-362.
13. Биков В. Ю. Мобільний простір мобільно орієнтоване середовище інтернет-користувача: особливості модельного подання та освітнього застосування. Інформаційні технології в освіті. 2013. №17. С. 9-37.
14. Бондаренко О. Безпека web-додатків. Актуальні проблеми та їх аналіз. 2017. № 3. С. 28-36.
15. Бондаренко В. Мобільні застосунки як інструмент у соціокультурних комунікаціях: можливості адаптації в діяльності наукових бібліотек. Наукові праці Національної бібліотеки України ім. В. І. Вернадського. 2017. Вип.46. С. 426-444.
16. Борисенко Д. Використання мобільних додатків при розроблені дизайн-продукту у навчанні майбутніх фахівців з дизайну. Інформаційні технології і засоби навчання. 2018. № 6 (68). С. 47-63.
17. Б. Харді, Б. Філліпс Android. Програмування для професіоналів. 2016. С.640.
18. Бугайчук К. Л. Мобільне навчання: сутність і моделі впровадження в начальний процес вищих навчальних закладів МВС України. Інформаційні технології і засоби навчання. 2012. № 1. С. 154-156.
19. Бурдун О. В. Інформатика як відображення тенденцій інформатизації освіти. Науковий часопис НПУ ім. М. П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання. 2010. №10. С. 58-65.
20. Бут В. , Панченко Г. Впровадження сучасних форм навчання впродовж життя в Україні. Неперервна професійна освіта: теорія і практика (Серія: педагогічні науки). 2016. Вип. 3-4 (48-49). С. 122-126.
21. Варакін М. В. Розробка мобільних додатків під Android. УЦ «Спеціаліст» при МГТУ ім. Н. Е. Баумана, 2012. С. 40-48.

22. Власій О. О., Винничук М. Д. Розробка мобільних додатків засобами блочного програмування: Навчально-методичний посібник. Прикарпатський національний університет ім. Василя Стефаника, 2021. 130 с.
23. Ганжела С. І. Основи інформатики з елементами програмування та сучасні інформаційні технології навчання. Кропивницький: РВВ ЦДПУ ім. В. Винниченка, 2017. 61 с.
24. Гевко І. В. Формування і розвиток професіоналізму вчителя технологій: теорія і методика: монографія. Кам'янець-Подільський: Аксіома, 2017. 392 с.
25. Голощапов А. Л. Google android. Створення додатків для смартфонів і планшетних ПК / А. Голощапов. Вінниця: БХВ, 2013. 832 с.
26. Голян В. В., Кравченко О. К., Порівняння моделей життєвих циклів програмного забезпечення з метою виявлення найефективнішого. Харків: ХНУ. 2019. 8 с.
27. Горбатюк Р. М., Тулашвілі Ю. Й. Мобільне навчання як нова технологія вищої освіти. Науковий вісник Ужгородського національного університету. Серія «Педагогіка, соціальна робота». 2013. Вип. 27. С. 31-34.
28. Гріффітс Д. Head First. Програмування для Android. Київ: Кіберна, 2016. 704 с.
29. Грушева А. А., Філіппова Л. Л. Мобільне навчання: за і проти. Професійна освіта: проблеми і перспективи: ел. наук. видан. 2015. Вип. 8. С. 100-106.
30. Гуревич Р. С. Мобільне навчання – нова технологія професійної освіти XXI ст. Вісник Луганського національного університету ім. Тараса Шевченка. 2012. № 20 (255). С. 113-119.
31. Гурняк І. А. Використання Google Forms І Microsoft Forms в процесі навчання. Фізико-математична освіта, 2018. Вип.2 (16). С. 40-45.
32. Дейтел П. Android для програмістів: створюємо додатки. Вінниця: Кобза, 2013. 560 с.
33. Денисова Л. В. Мова Enchanting для програмування роботів Lego Mindstorms NXT 2.0 (Педагогічний досвід). Інформатика та освіта. 2014. № 7. С. 100–102.
34. Дерсі Л. Розробка додатків для Android-пристроїв. Базові принципи / Л. Дєрси, Ш. Кондер. Том 1. Ексмо, 2014. 598 с.

35.Джамал Х. М. Алгоритм і структура модуля для обчислення квадратного кореня в ПЛІС. Праці міжнародної конференції *«Безпека, відмовостійкість, інтелект».* 2018. С. 74–77.

36. Донченко Я. Розроблення змісту навчання інформатики в загальноосвітніх школах України: ретроспективний аналіз. Професіоналізм педагога: теоретичні й методичні аспекти. 2015. № 1. С. 191-197.

37.Дорошенко Ю. О. Навчання інформатики у структурі 12-річної загальної середньої освіти. Інформатика та інформаційні технології в навчальних закладах. 2006. № 1. С. 55-72.

38. Жвалевський А. Смартфони Android без напрягу. Керівництво користувача. Київ : Кіберна, 2012. 224 с.

39. Іванова Г. І. Використання Thunkable для розробки мобільних додатків при вивчені програмування. V CISP Conference «An integrated approach to science modernization: methods, models and multidisciplinarity». №23 (2022): URL: https: // doi. org / 10.36074 / grail. (дата звернення 15.11.2023).

40. Іващенко М. В., Бикова Т. Б. Особливості використання елементів змішаного навчання в процесі викладання навчальних дисциплін у закладах вищої освіти. Фізико-математична освіта. 2018. Вип. 1 (15). С. 221-226.

41.Калініна Л. М. Латиський В. В., Китайцев О. М., Косик В. М., Мельник О. М. Інформатизація освіти. Стан та перспективи впровадження . Директор школи. 2018. № 9-10 (825-826). С. 7-16.

42.Камалов Р. Р. Використання елементів паралельного програмування для реалізації методичної системи додаткового освіти в галузі інформатики. (Педагогічний досвід). *Інформатика та освіта.* 2014. № 8. С. 65–67.

43. Кармен Делессіо, Лорен Дерсі, Шейн Кондер Створення додатків для Android за 24 години. Київ: Ескмо, 2015. 528 с.

44. Кобильник Т., Когут У., Жидик В. Методичні аспекти вивчення основ алгоритмізації і програмування мовою python у шкільному курсі інформатики у старших класах. Фізико-математична освіта, 2021. № 31 (5). 36-44.

45.Козлов С. В. Особливості навчання школярів інформатики в профільній школі. *Концепт.* 2014. №1. С. 1–7.

46.Копосов Д. Г. Технологія. Робототехніка. 6 клас. Навчальний посібник. Київ Біном. Лабораторія знань, 2017. 128 с.

47.Коробка О. І. Алгоритмічна структура розгалуження : розробка уроку для 5 класу [електронний ресурс]. Режим доступу : https://naurok.com.ua/rozrobka-uroku-5-klas-tema-algoritmichna-struktura-rozgaluzhennya-22510.html (25.11.2023)

48.Коршунова О. В. Удосконалення змісту й структури навчання інформатики в школі відповідно до вимог сучасного суспільства. Комп'ютер у школі та сім'ї. 2015. № 4. С. 20-23.

49. Кривонос О. М., Коротун О. В. Змішане навчання як основна формування ІКТ-компетентності вчителя. Наукові записки [Кіровоградського державного педагогічного університету ім. В. Винниченка]. Серія: Проблеми методики фізико-математичної і технологічної освіти. Кіровоград, 2015. Вип. 8 (II). С. 19-23

50. Кузь М. В., Соловко Я. Т. Методологія формування узагальненого критерію якості програмного забезпечення в умовах невизначеності. Вісник Вінницького політехнічного інституту. 2015. №5. С.1-4-107.

51. Ліщинська Л. Б. Основні аспекти автоматизації роботи з клієнтам засобами CRM-систем. Вісник Хмельницького національного університету. Економічні науки. 2015. № 5 (1). С. 206-209.

52.Луцків А. М. Архітектури високопродуктивних систем опрацювання великих даних. Збірник тез доповідей Науково-технічна VI Науково-технічна конференція *«Інформаційні моделі, системи та технології»,* 12–13 грудня 2018 року. Т.: ТНТУ, 2018. С. 75.

53. Матвієнко М. П. Алгоритми та структури даних: навч. посіб. / М. П. Матвієнко. Київ: Видавництво «Ліра-К», 2014. 340 с.

54. Мінтій І. С. Розробка мобільного додатку для розвитку математичних здібностей на платформі Android. Матеріали міжнародної науково-технічної

конференції «Сталий розвиток промисловості та суспільства». Кривий Ріг, 2015. С. 31.

55. Нетесова О. С. Методичні особливості реалізації елективного курсу з робототехніки на базі комплекту Lego Mindstorms NST 2.0. *Інформатика та освіта.* 2013. № 7. С. 74–76.

56. Оспеннікова Є. В. Освітня робототехніка як інноваційна технологія реалізації політехнічної спрямованості навчання фізики в середній школі. *Педагогічна освіта.* 2015. № 3. С. 33–40.

57. Семіоненков М. Н. Графічне середовище програмування Blockly (Блокли). (Мова програмування). *Інформатика – перше вересня*. 2014. № 3. С. 32–40.

58. Семеріков С. О., Стрюк М. І., Моісеєнко Н. В. Мобільне навчання: історико-технологічний вимір. В кн.: *Теорія і практика організації самостійної роботи студентів вищих навчальних закладів:* монографія / За ред. проф. О. А. Коновала. Кривий Ріг: Книжкове видавництво Кирєєвського, 2012. С. 188–242.

59. Слободяник О. В. Мобільні додатки на уроках фізики. Фізико-математична освіта. Суми, 2017. Вип. 4 (14). С. 293-298.

60. Триус Ю. В., Франчук В. М., Франчук Н. П. Організаційні й технічні аспекти використання систем мобільного навчання. *Науковий часопис Національного педагогічного університету імені М. П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання*. 2012. Вип. 12 (19). С. 53–62.

61. Філіппов С. А. Досвід технологічного навчання школярів на основі робототехніки. (Теорія і методика навчання технології). *Школа і виробництво.* 2015. № 1. С. 21–28.

62. Філліпс Б., Стюарт К., Марсикано К. Android. Програмування для професіоналів. 3-е изд. Київ: Наукова думка, 2017. 688 с.

63. Шакотько В. В. Інформатика в системі освіти України: становлення, перспективи. Інформаційні технології в освіті. 2016. №29. С. 116-130.

64. Шматко О. В., Поляков А. О., Федорченко В. М. Аналіз методів і технологій розробки мобільних додатків для платформи Android: навч. посіб. Харків: НТУ «ХПІ», 2018. 284 с.

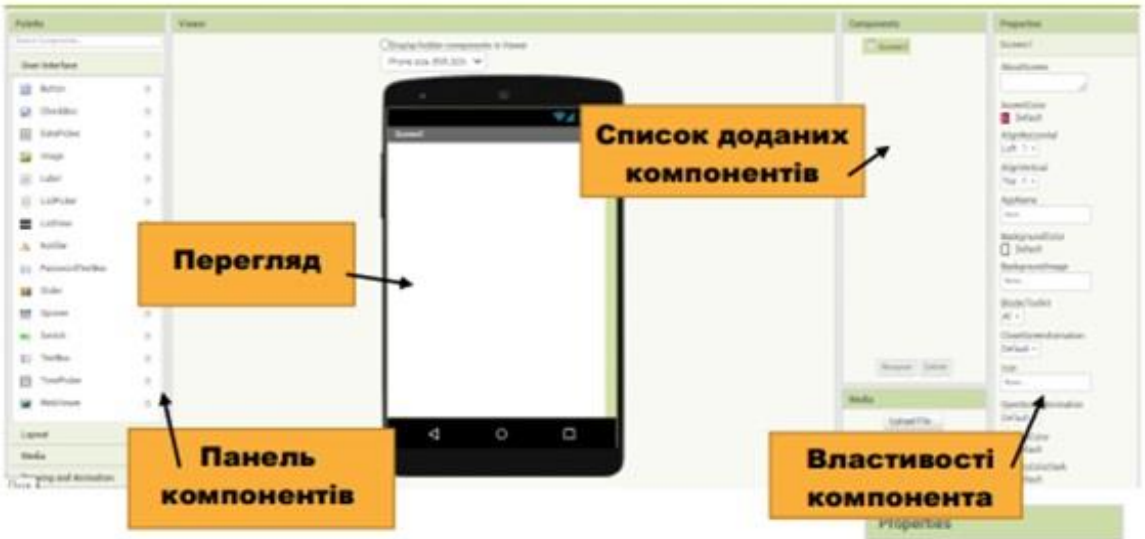
65. Юрченко І. В., Сікора В. С. Інформатика та програмування. Частина 2. Чернівці: Видавець Яворський С. Н., 2015. 210 с.
66. Юрченко А. О., Семеніхіна О. В., Хворостіна Ю. В., Удовиченко О. М., Петренко С. І. Навчання програмувати в старшій школі крізь призму чинних навчальних програм. Фізико-математична освіти, 2019.Вип. 2 (20). Ч.2. С. 48-55.
67. Інструкції по збору роботів Lego MindStorms EV3. LEGO Education. URL: https: // education. lego. com / r uru / educationdownloads2. (дата звернення 16.11.2023).
68. Розробка веб-додатків, мобільних додатків та порталів: URL: http: // ittel. com. ua / informacijni-texnologiyi / rozrobka-mobilnih-dodatkiv / (дата звернення 16.11.2023).
69. Технології створення мобільних додатків. URL: http: // group-global. org / ru / publication / 63343-tehnologiisozdaniya-mobilnyh-prilozheniy. (дата звернення 16.11.2023).
70. Acharjya D. P., Ahmed K. 2016. A survey on big data analytics: challenges, open research issues and tools. International Journal of Advanced Computer Science and Applications. № 7 (2). P. 511-518.
71. Ahmed I, Ahmad M, Jeon G, Piccialli F. A framework for pandemic prediction using big data analytics. Big Data Research 100190. URL: https: // pesquisa. bvsalud. org / global-literature-on-novel-coronavirus – 2019-ncov / resource / pt / covidwho – 1033015. (дата звернення 16.11.2023).
72. Bilous V. Мобільні додатки для навчання математики як засіб підвищення мотивації учнів школи. *Електронне наукове фахове видання «Відкрите освітнє е-середовище сучасного університету»*, 2017. №3. С. 303–309. URL: https: // doi. org / 10.28925 / 2414–0325.2017.3.30309. (дата звернення 16.11.2023)

ДОДАТКИ

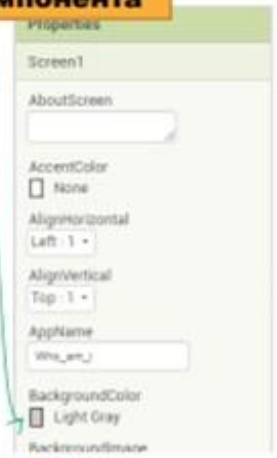
Додаток А

Черговість дій при програмуванні в мобільному додатку

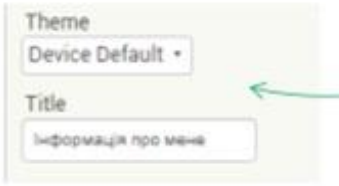
Після попередніх дій тобі відкрився режим дизайнера де ти можеш додавати, змінювати або видаляти компоненти з яких складається твоя програма. Даний режим складається з таких складових:



Зміни колір фону (властивість Background Color) твоєї програми, вибравши його зі списку



В цій же колонці зміни такі властивості Theme та Title так, як це показано на малюнку



Перетягни компонент Vertical Arrangement на екран. Він містить лише один стовпець, що допоможе нам розмістити інші компоненти в рядочок.

Група компонентів Layout допоможе нам гарно розмістити все на нашому екрані, тому ми будемо часто її використовувати. Її елементи - це «таблиці», колонки яких ми заповнюємо іншими компонентами.

Додаток Б

Програмування елементів

Програмуємо елементи

Перейди в блочний режим.

Блочний режим або, як його ще називають - режим розробника, має такий вигляд та структуру.

Вибери в групі блоків елемент першої кнопки та перетягни на поле такий блок.

Даний блок запустить вкладений скрипт після того як користувач натисне на кнопку.

Вклади в середину доданого блоку умовний блок «if-else»

Після того, як користувач натисне на першу кнопку, в нього може бути два сценарії розвитку подій:

1. **Якщо інформації на екрані немає, тоді потрібно зробити текст видимим, а текст на кнопці змінити на «Сховати ім'я»**
2. **Якщо інформація на екрані вже є, тоді потрібно її сховати, а текст на кнопці змінити на «Як мене звати»**

Як ти бачиш, потрібно буде використати умовний блок, який ти можеш знайти в групі

Додаток В

Блочне програмування

Наступним кроком буде заміна тексту на кнопці на «Сховати ім'я» або «Як мене звати?».

Для того, щоб змінити текст на кнопці, використай такий блок:

Але, як ти напевне помітив, в цей блок необхідно вставити інший блок, якій зміг би містити в собі інформацію текстового типу. Для таких операцій використовуються блоки з групи Text.

Тобі необхідний такий блок

Відповідно, тобі знадобляться два таких екземпляри, які потрібно вкласти умовний блок.

В тебе має би вийти такий скрипт

Додай функціонал для написів.

Твій напис повинен з'являтися або зникати в залежності від того чи потрібно показати інформацію чи сховати її.

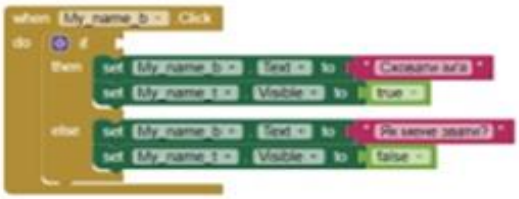
Для того, щоб робити текст видимим або ні, потрібно додати такий блок

Так як видимість може бути лише або true або false (логічна змінна), попередній блок нам потрібно зв'язати з блоком з групи Logic.

Так само як і в попередньому кроці, тобі потрібні два екземпляри

та.

Ось такий скрипт виходить



Додай умову

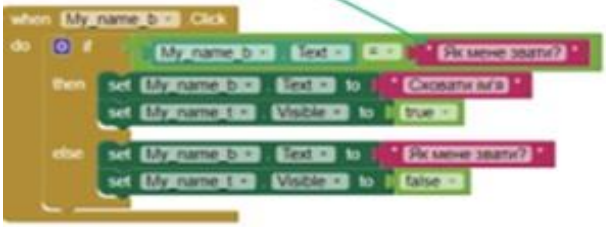
Наскільки ти знаєш, в умовні блоки потрібно добавити відповідно умову. У даному випадку ти будеш перевіряти текст, який написаний на кнопці.

Перевірку ти будеш здійснювати за допомогою логічного порівняння. Цей блок ти можеш знайти тут

Вже знайомі для тебе блоки

та

потрібно вкласти в середину цього блоку так, щоб утворився такий скрипт



Повтори попередні кнопки для того, щоб запрограмувати іншу кнопку та напис.

Примітка. Ти можеш скопіювати попередній скрипт та змінити тільки потрібну інформацію.

Змінюй текст на кнопці з «Скільки мені років?» на «Сховати вік» або навпаки та відповідно роби другий видимим або невидимим.

Додаток Г

Випробування результатів блочного програмування

Наступним кроком буде заміна тексту на кнопці на «Сховати ім'я» або «Як мене звати?».

Для того, щоб змінити текст на кнопці, використай такий блок:

Але, як ти напевне помітив, в цей блок необхідно вставити інший блок, який зміг би містити в собі інформацію текстового типу. Для таких операцій використовуються блоки з групи Text.

Випробуй свій проект

Завантаж на смартфон додаток MIT AI2 Companion з Play Market.

Відкрий завантажений додаток MIT AI2 Companion та натисни кнопку Scan QR code.

Відкрий меню Connect в середовищі MIT APP Inventor на комп'ютері та вибери зі списку

Відскануй смартфоном QR код, який відкрився та зачекай завантаження твого додатку

Випробуй свій додаток. Спробуй натиснути на одну з кнопок. Чи з'являється необхідна інформація? Чи змінився текст на кнопці? А якщо ти знову натиснеш на цю ж кнопку, інформація зникає?

Спробуй виконати такі ж дії з іншою кнопкою.