

Сумський державний педагогічний університет імені А. С. Макаренка

Фізико-математичний факультет

Кафедра інформатики

УДК 378.016:51:004

Марченко Валерій Валерійович

**РОЗВИТОК ОБЧИСЛЮВАЛЬНОГО МИСЛЕННЯ УЧНІВ
НА УРОКАХ ІНФОРМАТИКИ**

Галузь знань: 01 Освіта

Спеціальність 014 Середня освіта (Інформатика)

Кваліфікаційна робота на здобуття освітнього рівня «Магістр»

Науковий керівник:

_____ О.В. СЕМЕНІХІНА,

доктор педагогічних наук,

професор кафедри інформатики

Виконавець:

_____ Валерій МАРЧЕНКО

Суми – 2024

ЗМІСТ

ВСТУП.....	3
Розділ 1. ОБЧИСЛЮВАЛЬНЕ МИСЛЕННЯ ТА ЙОГО РОЗВИТОК В УМОВАХ СУЧАСНОЇ ОСВІТИ	6
1.1. Обчислювальне мислення у психолого-педагогічних розвідках: історія та сучасний стан.....	6
1.2. Способи вимірювання обчислювального мислення.....	16
1.3. Сучасні підходи до розвитку обчислювального мислення молоді в ЗЗСО	28
Висновки до розділу 1.....	38
Розділ 2. ІНФОРМАТИКА ЯК МАЙДАНЧИК РОЗВИТКУ ОБЧИСЛЮВАЛЬНОГО МИСЛЕННЯ УЧНІВ.....	40
2.1. Аналіз навчальних програм з інформатики.....	40
2.2. Успішні практики, орієнтовані на розвиток обчислювального мислення учнів	44
2.3. Типові завдання з інформатики на розвиток обчислювального мислення учнів	47
Висновки до розділу 2.....	57
ВИСНОВКИ	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	61

ВСТУП

Поняття обчислювального мислення як здатності мислити логічно, аналізувати та обчислювати дані є важливою навичкою. Фахівці в галузі педагогіки і психології, а також методик навчання вважають, що учні, які мають розвинене обчислювальне мислення, можуть краще розв'язувати задачі сучасного мінливого світу.

Проблему розвитку обчислювальних навичок вчені-методисти розробляли в кількох напрямках. Ще на межі 20-го і 21-го століття раціоналізації методу обчислень, операціям, що входять до складу методу обчислень, вибору вправ як основного засобу формування усних і письмових обчислень у школі вийшли праці М. Моро [57], М.Фіцули [73], Г. Шульги [74] тощо. Класифікація прийомів обчислень, що вивчаються в початковій школі, досліджена Г. Мартиною [53], С. Скворцовою [69; 70]. Методика формування обчислювальних навичок учнів у процесі виконання арифметичних дій як в усній, так і в письмовій формі розглядається в роботах [44; 50] та ін.

Вирішуючи проблему розвитку обчислювального мислення учнів, науковці Тихонова Т., Кошкіна Г., Резнік К., Liu J., Wang L., Romero M., Lerage A., Lille B. та ін. наслідують кілька ідей, серед яких відзначаємо: використання задач, які потребують для свого вирішення обчислювального мислення, використання ІТ для вироблення навичок обчислювального мислення; підтримка та допомога учням, у яких недостатньо розвинене обчислювальне мислення; використання оцінювань обчислювального мислення. Водночас з розвитком технологій відзначаємо, що проблема розвитку обчислювального мислення учнів залишається актуальною.

Об'єкт дослідження: процес навчання інформатики у ЗЗСО.

Предмет дослідження: розвиток обчислювального мислення учнів на уроках інформатики.

Мета дослідження: обґрунтувати можливість розвитку обчислювального мислення учнів на уроках інформатики та розробити відповідні засоби такого розвитку.

Поставлена мета дослідження обумовила вирішення низки **завдань**:

- 1) охарактеризувати обчислювальне мислення як категорію сучасної освіти, описати методики його вимірювання;
- 2) виявити підходи до розвитку обчислювального мислення молоді в умовах ЗЗСО;
- 3) проаналізувати чинні програми з інформатики та узагальнити успішні освітні практики, орієнтовані на розвиток ОМ учнів;
- 4) розробити типові завдання з інформатики на розвиток ОМ.

Для досягнення мети використано *теоретичні методи дослідження*: аналіз і систематизація наукових джерел для характеристики стану розробленості проблеми дослідження; термінологічний аналіз для уточнення тезаурусу дослідження, ретроспективний аналіз для опису етапів розвитку категорії «обчислювальне мислення» в освіті; систематизація для характеристики складників ОМ; синтез і узагальнення для опису практик розвитку ОМ в навчанні інформатики.

Теоретична і практична значущість дослідження полягає у обґрунтуванні можливості розвитку обчислювального мислення в навчанні інформатики та розробленні банку практичних завдань щодо розвитку обчислювального мислення на уроках інформатики в різних класах.

Апробація матеріалів дослідження здійснювалася на наукових заходах різних рівнів, серед яких: Міжнародна науково-практична конференція студентів, аспірантів та молодих вчених «Молодіжна наука: інновації та глобальні виклики» (м. Полтава, 6 листопада 2024 року) [54].

Структура та обсяг роботи. Кваліфікаційна робота складається зі вступу, двох розділів, загальних висновків та списку використаних джерел.

У першому розділі «Обчислювальне мислення та його розвиток в умовах сучасної освіти» охарактеризовано обчислювальне мислення як категорію

сучасної освіти, описано методики його вимірювання та підходи до розвитку обчислювального мислення молоді в умовах ЗЗСО.

У другому розділі «Інформатика як майданчик розвитку обчислювального мислення учнів» проаналізовано чинні програми з інформатики, узагальнено успішні освітні практики, орієнтовані на розвиток ОМ учнів та запропоновано типові завдання з інформатики на розвиток ОМ.

Загальний обсяг роботи 68 сторінок, з яких 60 сторінок основного тексту. Список використаних джерел включає 74 одиниці. Робота містить 20 рисунків та 2 таблиці.

Робота буде цікавою працюючим і майбутнім учителям інформатики, які цікавляться проблемами розвитку обчислювального мислення на уроках інформатики.

Розділ 1.

ОБЧИСЛЮВАЛЬНЕ МИСЛЕННЯ ТА ЙОГО РОЗВИТОК В УМОВАХ СУЧАСНОЇ ОСВІТИ

1.1. Обчислювальне мислення у психолого-педагогічних розвідках: історія та сучасний стан

Обчислювальне мислення (англ. – computational thinking) є однією із ключових компетентностей сучасної освіти, яка спрямована на підготовку учнів до життя в інформаційному суспільстві. Ця концепція виходить за межі розуміння традиційної освіти і включає в себе навички вирішення проблем, алгоритмічний підхід і структуроване мислення.

Історично передумови розвитку обчислювального мислення можна побачити у працях таких відомих учених, як Алан Тюрінг, який заклав основи теорії алгоритмів і створив модель універсальної машини Тюрінга, що стала фундаментом для сучасної комп'ютерної науки. Важливим кроком у розвитку цього поняття стала також поява мов програмування, які актуалізували оперування структурами та формулювання рішень типових задач через алгоритмічні структури (у вигляді алгоритмів).

Обчислювальне мислення (ОМ) є відносно новою категорією в психолого-педагогічних дослідженнях, яка, однак, швидко набула значної уваги з боку науковців та практиків. Від перших спроб опису цього феномену в 1980-х роках до сучасних підходів, ОМ еволюціонує разом із зростанням значення технологій у нашому житті. Це поняття охоплює не лише розуміння алгоритмів та програмування, але й широкі кола когнітивних процесів, які сприяють розвитку логічного мислення, аналітичних здібностей та здатності до вирішення складних завдань.

Історія розвитку поняття ОМ бере свій початок задовго до офіційного введення терміна. Вона пов'язана з еволюцією комп'ютерної науки та теорії обчислень, які поступово формували підґрунтя для розуміння, як можна використовувати комп'ютери та алгоритми для вирішення складних проблем.

Виділяють шість основних етапів розвитку ОМ (рис. 1.1).

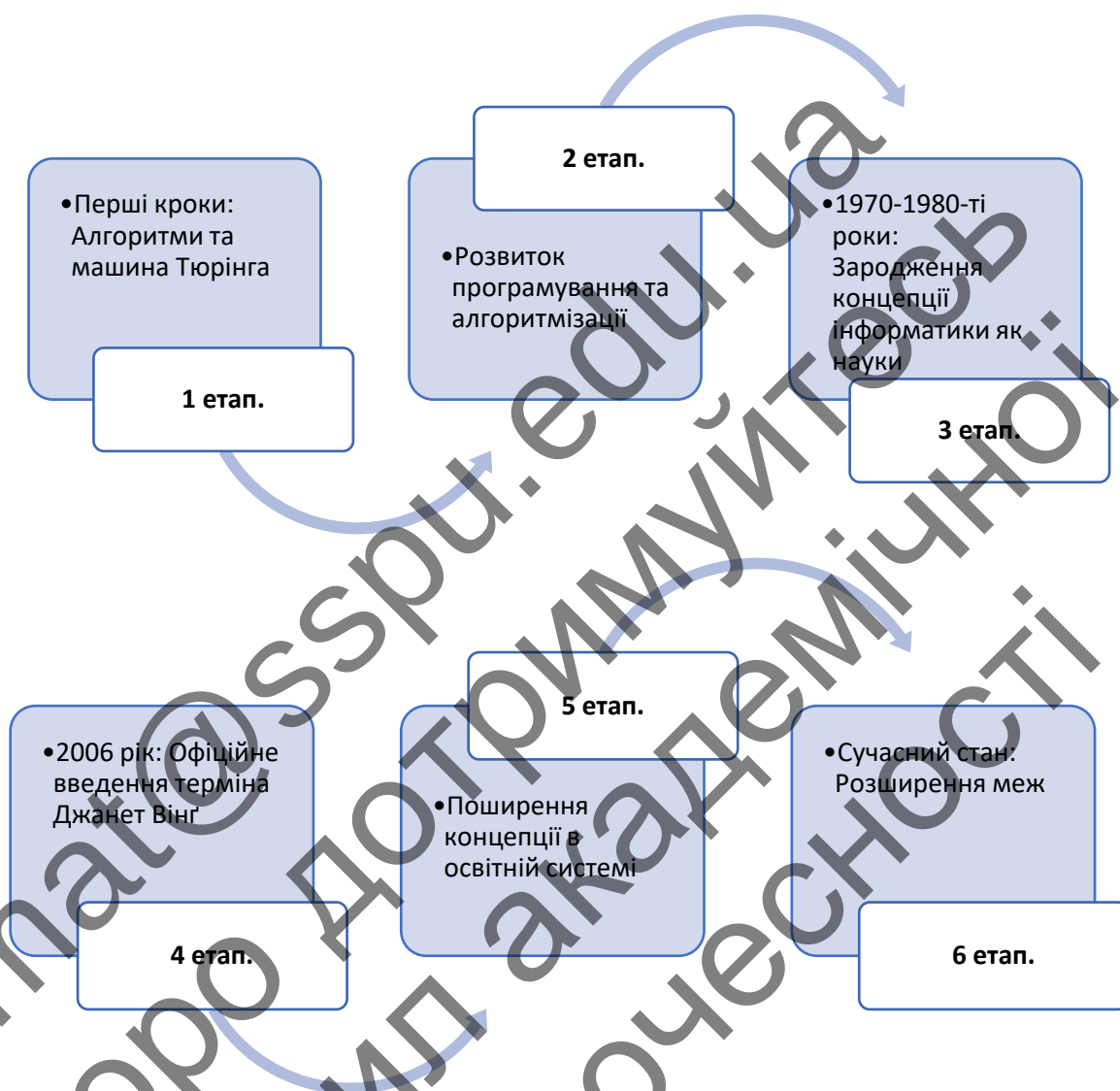


Рис. 1.1. Етапи розвитку ОМ

1 етап. Перші кроки: Алгоритми та машина Тюрінга

Ідеї, які стали основою для ОМ, можна простежити до середини 20-го століття, коли Алан Тюрінг, британський математик і логік, у 1936 році запропонував модель універсальної машини Тюрінга. Його теорія алгоритмів і обчислень стала однією з найважливіших фундаментальних основ сучасної інформатики. Вона показала, що будь-яка проблема, яка може бути формалізована в термінах чітких алгоритмів, може бути вирішена за допомогою обчислень. Машина Тюрінга не лише дала нові горизонти для

розвитку комп'ютерної техніки, але й заклала перші принципи алгоритмічного мислення, що є важливою частиною обчислювального мислення.

2 етап. Розвиток програмування та алгоритмізації

У 1950-1960-х роках із появою перших мов програмування, таких як Fortran і COBOL, зросла потреба в розумінні, як структурувати задачі таким чином, щоб їх можна було реалізувати через комп'ютерні алгоритми. Важливим аспектом у той час стала автоматизація процесів і пошук універсальних підходів до вирішення завдань за допомогою програм. Програмування все більше асоціювалося не тільки з навичкою писати код, але й зі способом мислення, який передбачав здатність декомпонувати проблеми, абстрагувати важливі аспекти і формулювати чіткі алгоритми.

3 етап. 1970-1980-ті роки: Зародження концепції інформатики як науки

У 1970-1980-х роках інформатика як наука почала активно розвиватися. Науковці та освітяни стали усвідомлювати, що вміння вирішувати задачі за допомогою комп'ютерів вимагає особливого способу мислення. Джозеф Вейценбаум та Сеймур Пейперт зробили значний внесок у розвиток освітньої інформатики та розуміння того, як комп'ютерні системи можуть формувати мислення учнів. Сеймур Пейперт, зокрема, в своїй праці "Mindstorms" (1980) розробив ідею "конструкціонізму", який пропонував використовувати програмування як інструмент для розвитку мислення в дітей. Його система програмування Logo дозволяла учням створювати програми для керування графічними зображеннями, сприяючи розвитку алгоритмічного та логічного мислення.

4 етап. 2006 рік: Офіційне введення терміна Джанет Вінг

Офіційно термін «обчислювальне мислення» було введено у науковий обіг завдяки роботі Джанет Вінг (Jeannette Wing), професора комп'ютерних наук в Університеті Карнегі-Меллона, у 2006 році. У своїй статті [41] Вінг підкреслювала, що ОМ є не тільки мисленням програмістів, але й універсальним способом вирішення проблем у будь-яких сферах діяльності. Вона наголошувала, що цей тип мислення включає в себе здатність

формалізувати проблеми таким чином, щоб їх можна було вирішити алгоритмічними методами, незалежно від наявності комп'ютера. Наслідуючи міркування Джанет Вінг, відзначаємо наголос на таких характеристиках [41]:

- Мислення на різних рівнях абстракції, а не програмування;
- Люди набувають базових, а не формульних навичок;
- Створення математичних моделей, що взаємодіють із реальним світом;
- Практичне застосування в повсякденному житті;
- Зв'язки з різними об'єктами і суб'єктами навколишнього світу.

5 етап. Поширення терміну «обчислювальне мислення» в освітній системі

Після публікації Вінг обчислювальне мислення стало швидко інтегруватися в освітні програми багатьох країн світу. Концепція отримала розвиток у рамках освітніх реформ, спрямованих на підготовку учнів до життя у цифровому суспільстві, зокрема у STEM-освіті (наука, технології, інженерія, математика) [40]. В рамках нових підходів до навчання, ОМ стало вважатися важливою навичкою XXI століття, необхідною для успішної діяльності у різних галузях, таких як медицина, бізнес, наука і техніка.

6 етап. Сучасний стан: Розширення меж

Сучасне розуміння ОМ виходить за межі програмування. Воно включає в себе широкий спектр навичок, таких як абстракція, декомпозиція проблем, розробка алгоритмів, моделювання та симуляція, а також здатність адаптувати наявні рішення до нових умов. ОМ зараз є невіддільною частиною навчальних програм багатьох країн, особливо в контексті цифрової трансформації освіти.

Сучасний стан ОМ характеризується його широким застосуванням у різних галузях науки, техніки та освіти. Воно включає в себе такі компоненти, як декомпозиція проблем, абстрагування, пошук шаблонів і створення алгоритмів для автоматизації процесів. Крім того, стрімкий розвиток цифрових технологій та інструментів, таких як штучний інтелект, великі дані і хмарні обчислення, вимагає від учнів нових навичок, пов'язаних з ОМ, що робить його одним із важливих результатів сучасної освіти.

Розглянувши еволюцію ОМ, важливо перейти до його сучасного розуміння в психолого-педагогічній літературі. Сьогодні ОМ розглядається як інтегральний компонент освітнього процесу, що включає в себе не лише технічні навички, але й розвиток критичного мислення та проблемо-орієнтованого підходу до навчання. Приділяючи увагу тлумаченню цього поняття в наукових дослідженнях, науковці [71; 26; 33; 35] намагаються краще зрозуміти його роль у підготовці учнів до викликів сучасного світу.

З точки зору інформатики, обчислювальне мислення – це інструмент для розв'язання багаторівневих проблем, що лежать в основі програмування [15]. Однак таке мислення виходить далеко за межі програмування, і його компоненти постійно використовуються більшістю людей під час розв'язання складних завдань різного рівня. Характеризуючи обчислювальне мислення, К.Резнік і В.Лазарук зазначають, що це розумовий процес, пов'язаний із постановкою проблеми та представленням рішення у формі, яка може бути ефективно реалізована людиною або комп'ютером [63].

Аналіз досліджень ОМ в психолого-педагогічних практиках свідчить, що вчені використовують різноманітні методологічні підходи, включаючи кількісні та якісні методи. Зокрема, програми STEAM демонструють значні покращення в креативності й навичках розв'язання проблем, які підтверджені кількісними методами статистичного аналізу [8]. Якісні методи, включаючи інтерв'ю та спостереження, виявляють особисті переживання учнів, що часто залишається поза увагою кількісних підходів [6].

Проаналізувавши наукові джерела, можна зазначити, що визначення ОМ неоднозначним і різні науковці трактують його по-різному. Наведемо приклади трактувань поняття ОМ з різних ресурсів (табл. 1.1).

Таблиця 1.1

Трактування поняття ОМ

<i>Джерело / ресурс / Автор</i>	<i>Тлумачення ОМ</i>
Джанет Вінг (2006) [41]	Обчислювальне мислення – це універсальний спосіб вирішення проблем, що використовує принципи інформатики. Воно включає в себе абстрагування, декомпозицію завдань і алгоритмічний підхід до вирішення проблем. Це не тільки про програмування, а про те, як формулювати проблеми, щоб їх можна було розв'язати за допомогою комп'ютера або алгоритмів.
Google for Education [17]	Обчислювальне мислення – це процес, який дає можливість людям, незалежно від їхньої сфери діяльності, застосовувати принципи інформатики для вирішення проблем. Воно передбачає розбиття проблем на менші частини (декомпозиція), виявлення закономірностей, абстракцію важливих деталей і розробку алгоритмів для автоматизації вирішення проблем.
Міжнародна асоціація з питань обчислювального мислення (International Society for Technology in Education, ISTE) [22]	Обчислювальне мислення – це навичка, яка допомагає учням зрозуміти, як проблеми можуть бути вирішені за допомогою технологій. Воно полягає в тому, щоб використовувати логічне мислення для розв'язання складних завдань через алгоритми й моделювання.
Національний центр розвитку STEM-освіти	Обчислювальне мислення – це систематичний процес вирішення проблем, який

<i>Джерело / ресурс / Автор</i>	<i>Тлумачення ОМ</i>
(National STEM Centre, Великобританія) [30]	включає декомпозицію, розробку моделей, пошук шаблонів і формування чітких інструкцій або алгоритмів, що дозволяють вирішувати проблеми будь-якого рівня складності.
Концепція Core STEM Skills (США) [13]	Обчислювальне мислення – це застосування методів інформатики до вирішення проблем у різних сферах, включаючи не лише комп'ютерні науки, але й математику, природничі науки, техніку та гуманітарні науки. Воно допомагає формувати завдання в термінах, зрозумілих для комп'ютерів або людей.
Європейська рамка компетентностей цифрового громадянства (DigCompEdu) [49]	Обчислювальне мислення – це здатність ефективно застосовувати принципи інформатики для вирішення реальних життєвих проблем. Це включає алгоритмічне мислення, креативне використання технологій та вміння працювати з великими даними.
Code.org [46]	Обчислювальне мислення – це спосіб мислення, що допомагає учням навчитися вирішувати проблеми, працюючи з ними крок за кроком, використовуючи процеси планування та створення алгоритмів, які можуть бути реалізовані на комп'ютері або іншому пристрої.
Міністерство освіти Фінляндії [56]	Обчислювальне мислення розглядається як здатність формалізувати проблеми так, щоб їх можна було вирішити за допомогою обчислювальних засобів. Важливими складовими

<i>Джерело / ресурс / Автор</i>	<i>Тлумачення ОМ</i>
	є здатність мислити алгоритмічно і розробляти цифрові моделі для аналізу та вирішення проблем.
Computer Science Teachers Association (CSTA, США) [12]	Обчислювальне мислення – це інструмент, що включає в себе алгоритмічне мислення, абстрагування і структурування даних. Це вміння створювати алгоритми та аналізувати дані для вирішення проблем у різних сферах життя.
Освітня платформа Tynker [11]	Обчислювальне мислення – це спосіб вирішення завдань за допомогою алгоритмів та абстракцій, який дозволяє формалізувати будь-яку проблему і створити розв'язок, який можна реалізувати за допомогою комп'ютерних засобів.

Розглядаючи ОМ, вчені Bocconi S., Chioccariello A., Dettori G., Ferrari A., Engelhardt K. [3] вказують на два освітні аспекти (рис. 1.2).

це ментальний процес, який не залежить від технологій

це особливий спосіб формулювання проблем, що містить у собі спосіб аналізу та розв'язання проблем за допомогою інформаційних і цифрових технологій

Рис. 1.2. Тлумачення ОМ за [3]

За термінологічним аналізом слід виділити спільні характеристики, які об'єднують тлумачення та характеризують ОМ (рис.1.3).



Рис. 1.3. Характеристики ОМ

ОМ як вид розумової діяльності людини передбачає основні способи розвитку ОМ [14] (рис.1.4): «декомпозиція», тобто здатність розбивати

складну проблему або систему на дрібніші й керованіші частини; «розпізнавання образів», тобто здатність шукати схожість між проблемами і відповідно схожі рішення; «абстрагування», тобто здатність концентруватися на важливішій інформації, не звертаючи уваги на несуттєві деталі; «алгоритм», тобто процес розроблення покрокового розв'язання проблеми або набору правил, яких треба дотримуватися для розв'язання задачі.

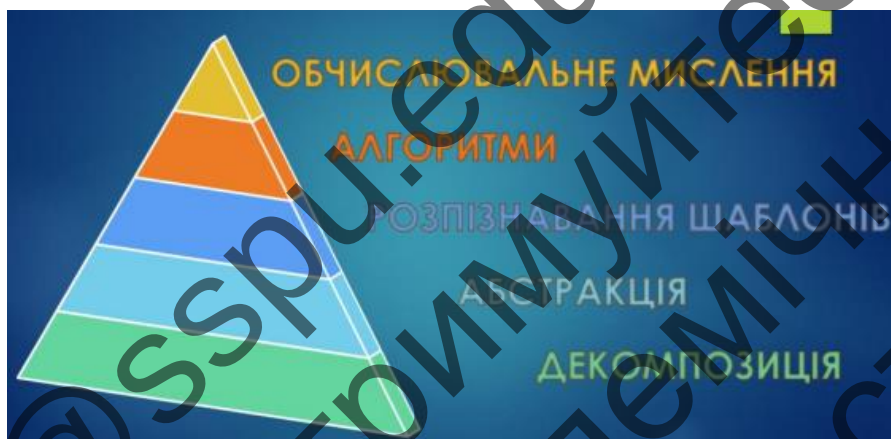


Рис. 1.4. Методи формування обчислювального мислення

Усі ці компоненти інколи вважають еквівалентом обчислювального мислення або рівноправними складниками обчислювального мислення.

Відповідно до цього ми розглядаємо ОМ як:

- процес, що включає формулювання задач та їх розв'язань таким чином, щоб ці розв'язання могли бути ефективно виконані людиною;
- спосіб мислення, що дозволяє людині впорядкувати та перетворити складні завдання в серію зрозумілих кроків;
- вміння мислити в термінах абстракцій, розбивати проблеми на менші підзадачі та шукати ефективні шляхи їх вирішення.

Завершуючи термінологічний аналіз, слід підкреслити значущість обчислювального мислення як інструменту, що сприяє розвитку критичного та аналітичного мислення. Сьогодні обчислювальне мислення не лише входить у навчальні програми, але й формує основу для нових педагогічних підходів, спрямованих на підготовку учнів до викликів XXI століття.

1.2. Способи вимірювання обчислювального мислення

У сучасному світі обчислювальне мислення стає все більш важливою навичкою, необхідною для успішного функціонування в цифровому суспільстві. Відповідно, виникає потреба у надійних та валідних методах його вимірювання. Оцінка обчислювального мислення дозволяє не лише визначити рівень розвитку цієї компетенції в учнів, але й формувати ефективні освітні програми, що сприяють її подальшому вдосконаленню.

Способи вимірювання обчислювального мислення

1. *Інструмент виміру Dr. Scratch*. Dr. Scratch дозволяє автоматично оцінювати навички ОМ за допомогою аналізу програмного коду проєктів серед Scratch (<http://www.drscratch.org>). Оцінюються сім концептів [28], які притаманні ОМ: 1) абстрагування та декомпозиція завдання; 2) паралелізм; 3) логіка; 4) синхронізація; 5) контроль потоку виконання; 6) взаємодія з користувачем; 7) робота з даними.

Для кожної компетенції оцінюється її розвиток до одного з трьох рівнів: "базовий", "такий, що розвивається", "професійний". Рівень розвитку компетентності визначається за допомогою перевірки наявності програмного коду певних конструкцій Scratch. Логічні операції вважаються розвиненими на базовому рівні, якщо у код є конструкція if. Якщо є парна конструкція if — else, рівень вважається таким, що «розвивається», а якщо код містить логічні операції, рівень розвитку вважається «професійним».

Особливістю Dr. Scratch є те, що він оцінює не факт виконання випробуваного проєкту або рішення завдання (хоча формалізовані завдання суперечать ідеям конструкціонізму [19], на яких базується середовище Scratch [4]), а спосіб виконання проєкту. Dr. Scratch може бути корисним педагогам, він дозволяє відокремити учнів, котрі зробили проєкт, використовуючи примітивні конструкції програмування, від учнів, котрі використовували конструкції, які потребують вищого рівня розвитку компетенцій ОМ.

Dr. Scratch використовує визначення ОМ, що відповідає рівню шкільної освіти, але він не підходить для оцінки більш розвинених наборів

компетентностей. Однак його можна використовувати у комбінації з іншими інструментами. Одна з таких комбінацій пропонується у книзі «Навчання дітей програмуванню: запорука розвитку людського капіталу в XXI столітті» [48].

2. Інструмент вимірювання ОМ на платформі Agent Cubes.

Інший приклад інструменту цієї категорії – середовище Agent Cubes, розроблене Університетом Колорадо в Боулдері (<https://agentsheets.com>). Ця платформа комбінує віртуальне середовище навчання та інструмент оцінювання. Навчання організовано так: учень отримує гру-приклад, у якій різним персонажам-агентам заданий певний алгоритм поведінки, а далі створює свою гру, самостійно програмуючи поведінку персонажів-агентів (гра-проект).

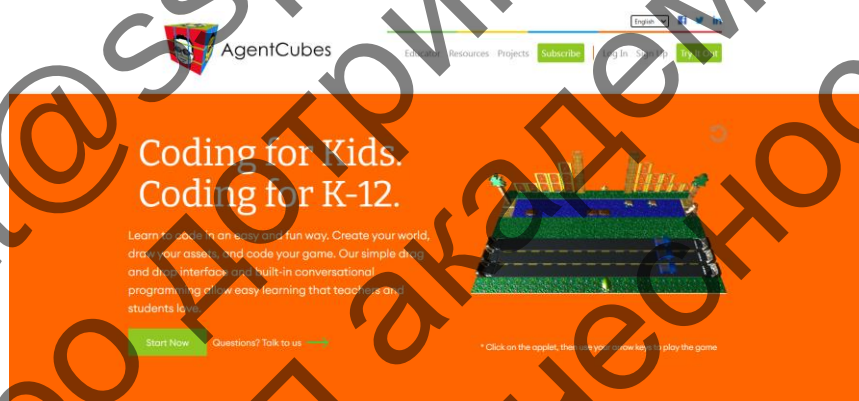


Рис. 1.5. Платформа Agent Cubes

Спосіб оцінювання [2] усередині цього середовища заснований на аналізі патернів ОМ, при цьому для аналізу відповідності конкретному патерну застосовується метрика схожості поведінки програм. У межах оцінки схожості поведінки програм програмний код порівнюється з кодами інших учнів. Для оцінки складності використовується підхід «Латентно-семантичний аналіз», який зазвичай використовується в завданнях порівняння текстів, наприклад при пошуку плагіату. Програмний код перетворюється на векторне уявлення, а далі векторні уявлення різних піддослідних порівнюються один з одним.

У платформі Agent Cubes існують різні види персонажів-агентів. Персонаж діє за алгоритмом, що ґрунтується на одному з дев'яти патернів, які автори інструменту відносять до патерна ОМ: 1) управління курсором; 2) генерація; 3) абсорбція; 4) колізія; 5) транспортування; 6) штовхання; 7) потяг; 8) дифузія; 9) сходження до вершини. Кожному персонажу-агенту відповідає один патерн. Аналіз патернів ОМ будується на застосуванні метрики схожості поведінки програм до реалізації патерну у грі-прикладі та реалізації патерну у грі-проєкті. Для кожного з персонажів-агентів порівнюється схожість поведінки у грі-прикладі та у грі-проєкті. Метрикою є комбінація по кожному з патернів.

Набір інструментів оцінювання, реалізований у платформі, може бути використаний [1] у рамках інших середовищ освоєння ОМ.

3. Тест ОМ, запропонований Роман-Гонсалес та ін.

Тест, який запропонували Román-González M., Pérez-González J. C., Jiménez-Fernández C [34], заснований на визначенні ОМ з роботи К. Бреннана та М. Резніка [5] та стандарті навчання інформатики для 7–8-х класів CSTA [61].

Тест охоплює такі уміння в складі ОМ, як створювати/працювати з: 1) основні напрями та послідовності; 2) цикли із лічильником; 3) цикли з умовою; 4) просте розгалуження if; 5) розгалуження if-else; 6) цикл із передумовою; 7) найпростіші функції. Для кожного уміння оцінюється його розвиток (один із чотирьох рівнів).

Кожне із питань є завданням у одному з двох середовищ — «Полотно» та «Лабіринт», усередині яких задана своя система команд виконавця. Приклад завдання серед «Лабіринт» наведено на рисунку (рис. 1.6). Даний лабіринт 8 x 8 осередків, частина яких заставлена стінами. Пакман знаходиться в комірці з координатами (7, 6). Привид знаходиться у комірці з координатами (1, 3).

Скільки разів Пакман повинен повторити набір команд «ліворуч-ліворуч-вгору», щоб опинитися в одному осередку з Привидом? Варіанти відповіді: (A) 2; (B) 1; (C) 4; (D) 3.

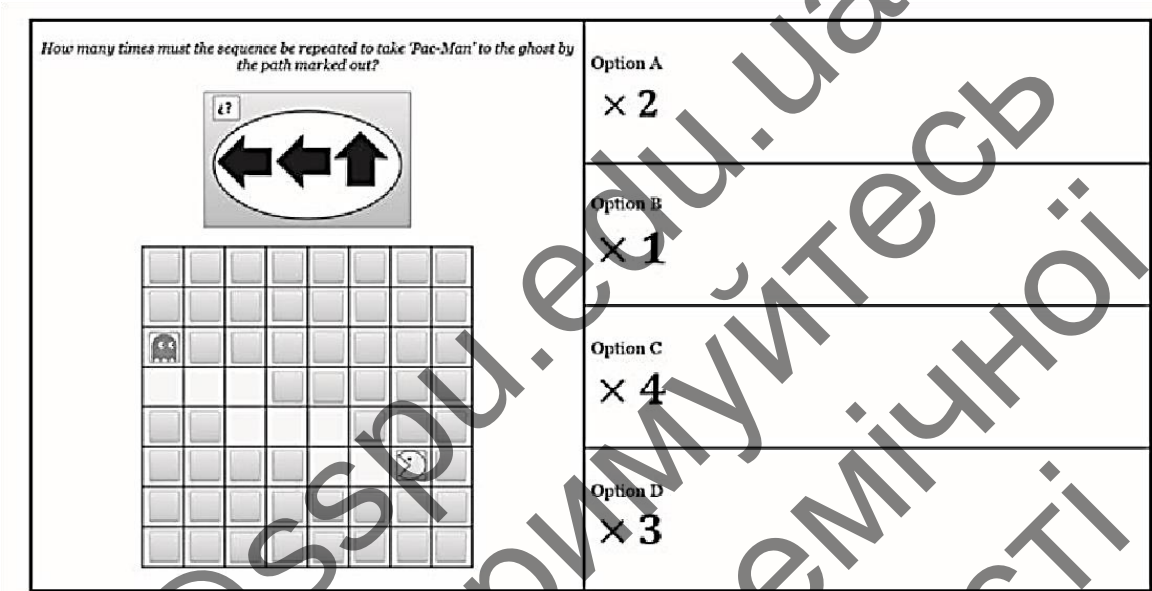
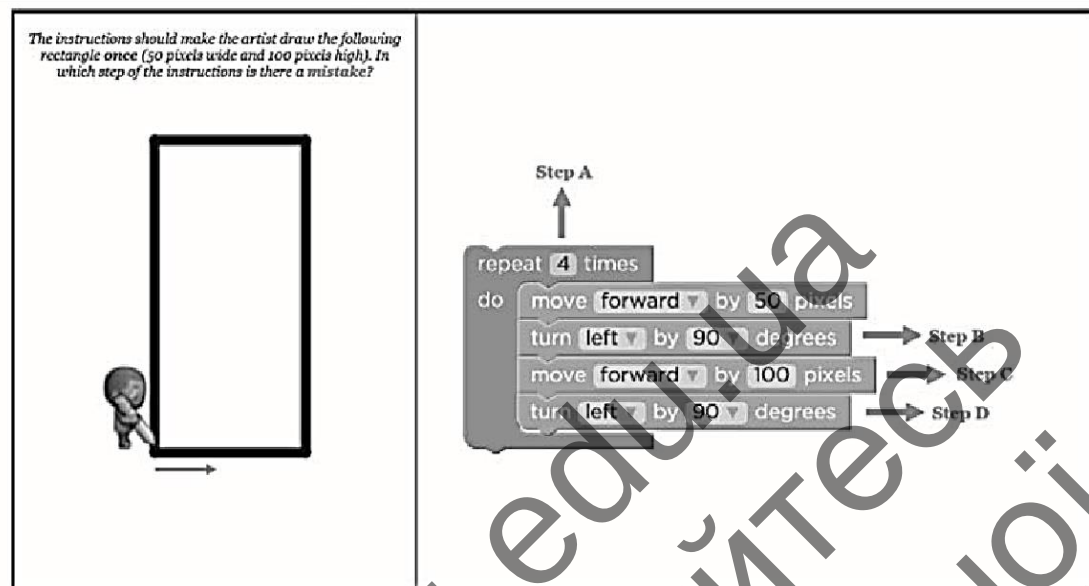


Рис. 1.6. Приклад завдання серед «Лабіринт» з роботи Роман-Гонсалес та інших.

Приклад завдання «Полотно» подано на рисунку (рис. 1.7). Виконавцю потрібно один раз намалювати на полотні прямокутник розміром 50×100 пікселів. Виконавець знаходиться в нижньому лівому куті прямокутника. Даний код у Scrath-подібному середовищі:

- а) повторити 4 рази (крок А);
- б) пройти вперед 50 пікселів;
- в) повернути ліворуч на 90 градусів (крок В);
- г) пройти вперед на 100 пікселів (крок С);
- д) повернути ліворуч на 90 градусів (крок D).

На якому з кроків (А, В, С, D) допущено помилку?



**Рис. 1.7. Приклад завдання серед «Полотно»
з роботи Роман-Гонсалес та інших.**

4. Тест умінь-складників ОМ, запропонований Вейнтропом та Віленські.

У своїй роботі Д. Вейнтроп та У. Віленські [39] пропонують тест умінь, які дослідники відносять до обчислювального мислення.

Завдання містять приклади програм мовами програмування C++ та Scratch. Питання тесту визначають уміння створювати/ працювати з: 1) ітерація; 2) логічні умови; 3) змінні; 4) функції; 5) розуміння роботи алгоритму.

Питання розділу «ітерація» містять приклади циклічних алгоритмів, наприклад, випробуванням потрібно визначити, скільки разів програма виведе той чи інший рядок. Розділ «логічні умови» – приклади алгоритмів з розгалуженням. Розділ «змінні» – приклади алгоритмів із змінними, наприклад, обмін значень. Розділ «функції» – приклади алгоритмів із функціями, наприклад, виклик однієї функції з іншої. Розділ «розуміння роботи алгоритму» – приклади алгоритмів, для яких випробувані повинні визначити результат роботи.

Приклади завдань із тесту Вейнтропа та Віленські:

1. Скільки разів слово here буде виведено після виконання програми?

```

for (var i = 0; i < 2; i++) {
  print(here);
  for (var j = 0; j < 3; j++) {
    print(here);
  } print(here); }

```

2. Що буде виведено після запуску програми?

```

var x, y, z, sentence;
x = Boys; y = Hello; z = Girls;
sentence = y+z+and+x; print(sentence);

```

3. Функція op4 приймає 3 числа. Що опція op4 робить?

```

function op4(a, b, c) {
  var tmp; if (a > b) {tmp = a; } else { tmp = b;
  } if (c > tmp) { tmp = c;
  }
  return tmp;
}

```

5. Тести ОМ на основі завдань міжнародного конкурсу «Бобер».

Конкурс «Бобер» — міжнародне змагання з інформатики та обчислювального мислення для школярів, яке проводиться з 2004 року. Ізу К. та група дослідників [23] використовували завдання конкурсу «Бобер» для вимірювання ОМ. Відібрані завдання конкурсу були співвіднесені з одним із восьми умінь: 1) збір даних; 2) аналіз даних; 3) обробка та подання даних; 4) декомпозиція задачі та абстрагування; 5) алгоритми та процедури; 6) автоматизація; 7) паралелізм; 8) симуляція.

Тест був апробований на групі зі 115 400 учнів. За результатами підтверджено надійність та валідність тесту.

Дослідники В. Дагієни та Г. Футчек [16] також використали завдання конкурсу «Бобер» для вимірювання ОМ. Завдання співвідносилися з однією із п'яти категорій: 1) робота з інформацією; 2) алгоритмічне мислення; 3) використання комп'ютера; 4) структури та патерни; 5) логічні пазли та ігри; 6) ІКТ та суспільство.

Підхід розмітки завдань конкурсів та олімпіад може бути використаний для конструювання інших інструментів виміру на основі, наприклад, завдань Всеукраїнської олімпіади школярів з інформатики.

6. Національний іспит у Ізраїлі як інструмент вимірювання ОМ.

Група дослідників (І. Цур-Баргурі, Б. Парв, Д. Ландберг [43]) проаналізували результати національного тесту Ізраїлю для учнів шкіл з комп'ютерних наук і дійшла висновку про надійність і валідність завдань цього тесту, що являють собою фрагменти алгоритмів, реалізованих в середовищі Scratch із вибором варіантів відповіді.

Завдання тесту чотири вміння ОМ, які стосуються: 1) умови; 2) нескінченний цикл та нескінченний цикл із вкладеною умовою; 3) цикл for; 4) серійне виконання. Кожне завдання співвіднесене також із рівнем мислення з таксономії Блума [20]: запам'ятовування та розуміння; застосування та аналіз; оцінка та створення.

Приклад завдання:

Для яких значень вхідних даних виконавець скаже Hello?

num1 = 5, num2 = 8 num1 = 3, num2 = 9 num1 = 8, num2 = 8 num1 = 7, num2 = 6

Текст програми (у завданні наводиться Scratch-подібною мовою програмування):

вводиться перше число

результат введення записується в змінну num1

вводиться друге число

результат введення записується в змінну num2 якщо num1 > num2: сказати Hello

Завдання національного тесту повністю наведені у роботі [43] та можуть бути використані для вимірювання ОМ. Плюсом є зіставлення завдань тесту з рівнями таксономії Блума, проте рамка конструкту досить вузька і взято лише чотири характеристики, що відносяться до курсу комп'ютерних наук у школах Ізраїлю, але не охоплюють міжпредметні уміння, які багато дослідників включають до ОМ.

7. Тест SCS1.

Тест SCS1, розроблений групою дослідників (М. Паркер, М. Гуздіал, С. Енглман [32]), є покращеною версією тесту FCS1. Автори тесту взяли за основу існуючий тест, перевірили його валідність та скоригували питання з низькою валідністю. Отримана у такий спосіб версія була названа ними SCS1.

Завдання тесту – це фрагменти коду мовою програмування з варіантами відповіді. Завдання належать до десяти характеристик ОМ: 1) основи; 2) логічні операції; 3) умовний оператор; 4) кінцеві цикли; 5) нескінченні цикли; 6) масиви; 7) процедури із параметрами; 8) функції з значеннями, що повертаються; 9) рекурсія; 10) основи об'єктно орієнтованого програмування.

Тест SCS1 є відкритим та досить широко використовується. Він зачіпає високорівневі характеристики ОМ, такі як рекурсія та об'єктно-орієнтоване програмування. Ці характеристики відповідають як теоретичним знанням, так і певному способу мислення. Тест може бути використаний на практиці, оскільки є валідним інструментом оцінювання. Однак, щоб використовувати його в Україні, потрібно адаптувати цей тест українською мовою.

8. Тест Є. Чобан, О. Коркмаз.

Дослідники Е. Чобан та О. Коркмаз [10] розробили набір тестів ОМ, заснованих на результатах діяльності.

У роботі використовується рамка, яка розглядає ОМ як групу навичок, а не як теоретичне знання, продукт чи ставлення/сприйняття. Завдання тесту належать до семи категорій: 1) декомпозиція; 2) абстрагування; 3) критичне мислення; 4) розпізнавання патернів; 5) вирішення завдань; 6) алгоритмічне мислення; 7) креативність.

Тест був апробований на вибірці з 203 учасників, перевірено його надійність та валідність.

Завдання тесту являють собою набори типових завдань, характерних для ОМ, таких як лабіринтні завдання, пошук шляхів у графах, пошук патернів та розпізнавання послідовностей тощо).

У рамках інструменту включені, зокрема, метапредметні характеристики особистості, такі як критичне мислення і креативність, що входять до списку 4К-компетентностей [75]. Варто зазначити, що Дж. М. Вінг [41] визначає ОМ досить широко, включаючи туди всі специфічні характеристики, які допомагають фахівцям з комп'ютерних наук вирішувати профільні для них завдання більш ефективно. Якщо виходити з такого трактування, креативність та критичне мислення можна віднести до обчислювального мислення.

9. Тест ОМ Кентського університету.

Тест придатності до комп'ютерного програмування розміщено на сайті кар'єрної служби (<https://www.kent.ac.uk/ces/tests/computer-test.html>) Кентського університету. Завдання тесту належать до чотирьох категорій, частина яких інші дослідники відносять до обчислювального мислення: 1) логічне мислення; 2) вирішення проблем; 3) розпізнавання патернів; 4) здатність слідувати складним процедурам.

Завдання тесту є питаннями з розділів математики та логіки, кожен з яких перевіряє певну компетенцію ОМ. Випробовуванням не потрібно демонструвати практичні навички програмування при проходженні тесту.

Відсутність доступу до літератури з перевірки надійності та валідності ставлять під сумнів можливість широкого застосування цього інструменту для вимірювання ОМ, проте сам підхід з теоретичними завданнями, що перевіряють результати діяльності, становить інтерес та перегукується з роботою С. Чобан, О. Коркмаз [10].

10. Вимірювання обчислювального мислення через «словник ОМ».

Дослідник Ш. Гровер зі Стенфордського університету у своїй роботі [18] відштовхується від ранніх робіт, які припускають в учнів формування специфічного «словника ОМ» у процесі опанування категорій ОМ. Він описує процес інтерв'ю учнів на тижневому курсі семінарів з робототехніки. Випробовуванням ставляться запитання до та після семінару, і фіксується вживання ними у відповідях специфічних термінів, пов'язаних із компетенціями ОМ.

У своєму дослідженні Гровер використовує п'ять категорій запитань, що належать до словника ОМ: 1) широкі концепції; 2) словник; 3) процедурні/операційні подробиці; 4) технічні терміни; 5) принципи. До широких концепцій належать програмування, автоматизація, зберігання даних. До словника належать оперативна пам'ять, вхідні дані, вихідні дані, програмне забезпечення, завантаження, програма, пам'ять, налагодження. До процедурних/операційних подробиць відноситься опис процесу дій під час виконання випробуваних завдань з робототехніки. До технічних термінів – опис принципів роботи процесора, оперативної пам'яті та ін. До принципів – умовний оператор, декомпозиція завдання, абстрагування, обробка помилок, налагодження. Якщо випробуваний у відповідях використовує специфічні терміни і дає технічно коректне опис своїх дій, вважається, що він відповів питанням правильно. Результати опитування випробуваних фіксуються у таблиці. Підхід представляє інтерес, оскільки дозволяє без письмового тестування перевірити рівень учнів. Важливо уважно слухати змістовну частину відповідей піддослідних, а не просто фіксувати кількість термінів.

У своїй роботі [36] В. Дж. Шут виділяє такі, найбільш типові для визначень ОМ категорії: 1) декомпозиція; 2) абстрагування; 3) алгоритмізація; 4) налагодження; 5) ітерація; 6) узагальнення/розпізнавання патернів. Для кращого розуміння співвіднесемо визначення цих категорій з прийнятими у вітчизняній педагогіці, здебільшого з елементами навчальних програм з інформатики для 7–9-х класів [60].

Декомпозиція – вміння розбивати завдання на підзавдання.

Абстрагування та узагальнення – суміжні категорії. Узагальнення – вміння створювати узагальнення, встановлювати аналогії, класифікувати. Абстрагування варто розуміти як пов'язану категорію, вміння виділяти ту частину характеристик об'єкта, яка важлива для розуміння явища, що розглядається, але без класифікації цих характеристик і без узагальнення результатів.

Алгоритмізація – знання основних алгоритмічних конструкцій та вміння застосовувати ці знання для побудови алгоритмів розв'язання задач.

Налагодження – діалогове налагодження програм: покрокове виконання, перегляд значень величин, виведення налагодження, вибір точки зупинки. У В. Дж. Шут під цим поняттям розуміється деяка окрема категорія пошуку помилок у складних системах, відірвана від конкретної програмування.

Ітерація – повторення процесу проектування доти, доки результату не буде досягнуто. Ітерація не є аналогом вміння застосовувати алгоритмічну конструкцію «повторення», це, швидше, вміння застосовувати ітеративний підхід у розробці програмного забезпечення.

Порівняння інструментів з наведеними категоріями ОМ за В. Дж. Шут подано у табл. 1.2. Компетенція «Декомпозиція» вимірюється у 4-х інструментах з 11-ти розглянутих. «Абстрагування» – у 4-х з 11-ти. «Алгоритмізація» – у всіх 11-ти. «Налагодження» – тільки в одному з 11-ти. «Ітерація» – у 4-х з 11-ти, причому вимірюється, швидше, вміння використовувати алгоритмічну конструкцію «повторення», ніж та інтерпретація, що наводиться у роботі У. Дж. Шут. «Узагальнення» вимірюється в 4-х інструментах з 11-ти.

Таблиця 1.2

**Порівняння інструментів вимірювання ВМ
з компетенціями ОМ В. Дж. Шут**

Назва інструменту	Компетенції ОМ за В. Дж. Шут					
	Декомпозиція	Абстрагування	Алгоритмізація	Налагодження	Ітерація	Узагальнення
Dr. Scratch	+	+	+			
AgentCubes			+			+

Назва інструменту	Компетенції ОМ за В. Дж. Шут					
	Декомпозиція	Абстрагування	Алгоритмізація	Налагодження	Ітерація	Узагальнення
Тест Роман-Гонсалес та ін.			+		+	
Тест Вейнтропа і Віленські			+		+	
Тест на завданнях конкурсу «Бобер» – К. Ізу та ін.	+	+	+			
Тест на завданнях конкурсу «Бобер» – В. Дагієне та Д. Футчік			+			+
Національний екзамєн Ізраїлю			+		+	
SCS1			+		+	
Тест Е. Чобан та О. Коркмаз	+	+	+			+
Тест Кентського університету			+			+
Тест через «словник ОМ»	+	+	+-	+		

Якщо стоїть завдання виміряти весь набір компетенцій ОМ за В. Дж. Шут, то знадобиться щонайменше три інструменти вимірювання: наприклад, тест через «словник ОМ»; тест Е. Чобан, О. Коркмаз; національний іспит Ізраїлю. Причому тест через «словник ОМ» необхідно використовувати у будь-якому разі, оскільки серед розглянутих інструментів є єдиний, котрий вимірює «Налагодження».

Отже, проаналізовано методики вимірювання ОМ: Dr. Scratch, вимірювання ОМ на платформі Agent Cubes, Тест ОМ Романа-Гонсалеса та ін., Тест ОМ Вейнтропа та Віленські, Тест ОМ на основі завдань міжнародного конкурсу «Бобер», Національний іспит у Ізраїлі як інструмент вимірювання ОМ, Тест SCS1, Тест Е. Чобан, О. Коркмаз, Тест ОМ Кентського університету,

Тест через «словник ОМ». Зіставлення методик показало, що кожна з них оперує як спільними, так і розрізненими характеристиками ОМ, проте в усіх методиках досліджується вміння алгоритмізувати розв'язування задач.

1.3. Сучасні підходи до розвитку обчислювального мислення молоді в ЗЗСО

Важливою особливістю нашого часу є стрімка інтеграція інформаційно-комунікаційних технологій (ІКТ) як у роботу, так і в повсякденне життя: очевидно, що ІКТ змінюють людський мозок, змушуючи його по-іншому реагувати на отримання інформації. Учні, ще з початкової школи, належать до цифрового покоління – покоління, яке змалку стикається як із фізичним, так і з віртуальним середовищем.

Згідно з результатами міжнародного проєкту ЄС «Діти онлайн», 80-90 % дітей віком 3-6 років у Європі користуються інтернетом, а кожна друга трирічна дитина – планшетом або смартфоном [66]. У сусідніх країнах третина дітей молодшого шкільного віку користуються інтернетом не менше восьми годин. Аналогічна ситуація спостерігається і в Україні [38]. У результаті, на думку науковців [47], які вивчають вплив ІКТ на особистість сучасних дітей молодшого шкільного віку, до школи сьогодні приходять діти, яких називають «цифровими дітьми», з певними когнітивними особливостями, зумовленими впливом віртуального світу. Цей віртуальний світ діє так само, як і фізичний, тож у дітей необхідно формувати вміння виживати в цьому світі.

У сучасному суспільстві дітям практично неможливо уникнути контакту з віртуальним середовищем. Тому, з огляду на прагнення цифрового покоління занурити дітей у віртуальний світ, необхідно вивчити можливості використання ІКТ в освітньому процесі, особливо в розвитку комп'ютерних компетентностей. Таким чином, в освітньому процесі необхідно враховувати потенціал ІКТ, які надають вчителям потужні та сучасні освітні інструменти для розвитку обчислювального мислення та позитивно впливають на ефективність цього процесу.

Діти, які звикли «сидіти» у віртуальному середовищі (наприклад, грати у відеоігри, дивитися мультфільми/ серіали тощо), отримують високий рівень стимуляції, а тому прагнутимуть такого самого рівня стимуляції мозку в реальному світі (фізичному). Однак фізичний світ не може цього забезпечити, тому традиційна освіта для дітей сьогодні здається учням нудною та нецікавою [37].

Очевидно, що яскраві та динамічні зображення, спецефекти, дії та кліпове представлення інформації, віртуальні середовища спрямовані на специфічні емоційні реакції в школярів. Так, при сприйнятті дітьми цифрового контенту в процесі навчання підвищується їхній емоційний стан, створюється емоційно заряджена та невимушена атмосфера, збільшується швидкість розуміння змісту інформації [72]. А це — спосіб покращити результати навчання, оскільки вища успішність досягається, коли люди отримують емоційно позитивні стимули [68].

Зважаючи на наведені факти, при формуванні обчислювального мислення необхідно враховувати загальні особливості мислення сучасних учнів, серед яких відзначаємо:

- 1) кліпове мислення, що вимагає подання інформації невеликими фрагментами;
- 2) багатозадачність — використання різних видів діяльності, які потребують швидкого перемикання з одного завдання на інше;
- 3) лінгвістичний мінімалізм — надання переваги діаграмному поданню інформації над аудіо на початку навчального процесу;
- 4) цифровий контент — використання цифрового контенту для подання інформації в доступнішій для учнів формі;
- 5) яскраве, динамічне подання навчальної інформації.

Усе перераховане вище може бути враховане під час подання та обробки навчальної інформації з використанням цифрового контенту. Однак, з огляду на негативний вплив ІКТ на мозок сучасних дітей, використання цифрових інструментів має бути дозованим і контрольованим з боку вчителів.

Цифрові засоби навчання надають вчителю широкі можливості для створення навчальних завдань. Як довели психологи [52], у середньому потрібно близько 20 тренувань, щоб учень набув певної навички чи вміння. Ця теза підтверджується нейрофізіологічними даними, які показують, що нейронні зв'язки формуються протягом семи днів після усвідомлення. Для вдосконалення навичок вчителі можуть вважати дуже корисним використання різноманітних онлайн-сервісів, що надають учням велику кількість однотипних завдань з автоматичною перевіркою. Виконуючи такі завдання, учні тренуються протягом певного часу, а функція автоматичної перевірки забезпечує зворотний зв'язок, даючи учням інформацію про правильність виконаних ними вправ.

Процес розвитку обчислювального мислення учнів передбачає виконання певної кількості тренувальних вправ, а також розкриття учнями суті обчислювальних прийомів і виконання обчислень за схемою, яка дає основу для спрямування дій. Сучасним школярам необхідно, щоб навчальна інформація подавалася у вигляді цифрового контенту, що робить проблематичним використання різноманітних інтерактивних вправ та онлайн-тренажерів для розвитку обчислювальних навичок.

Деякі онлайн-тренажери для розвитку обчислювального мислення у школярів доступні на сайтах авторів, зокрема на сайті OnlineMSchool, створеному М. Довжиком (Україна). Деякі з численних математичних тренажерів на цьому сайті включають обчислювальні задачі для різних рівнів шкільної освіти (рис. 1.8).



Рис. 1.8. Каталог обчислювальних тренажерів та вправи на обчислення на сайті OnlineMSchool

Під час онлайн-навчання за допомогою комп'ютерних тренажерів система фіксує час навчання. Це робить роботу багатьох ділянок мозку реальною, що продемонстрував японський дослідник Рей Кавашіма, автор методики розвитку людського мозку за допомогою простих розрахунків за часом. Тренажери для розвитку ОМ учнів надає освітнє онлайн-середовище «Міксіке» [62], яке позиціонується як освітня онлайн-система, що є засобом розвитку усних обчислювальних навичок. Освітній проєкт «Міксіке в Україні» успішно реалізується з 2014 року.

Проєкт реалізується на основі Меморандуму про співпрацю між Міністерством освіти і науки України та ОК MIKSIKE (Естонія). Проєкт надає вчителям, учням та батькам можливість створювати та використовувати як власні навчальні матеріали в режимі онлайн, так і онлайн-колекцію навчальних матеріалів, створених вчителями-учасниками проєкту.

На тлі українських освітніх реформ проєкт покликаний стати практичним інструментом для створення ефективного освітнього середовища з використанням новітніх інформаційних та освітніх технологій.

Підсистема «Pranglimine» – це «Міксік» для розвитку ОМ, що створений у рамках системи. Це тип платформи, рекомендований Міністерством освіти і науки України для розвитку усних обчислювальних навичок учнів середніх шкіл (лист № 1/9-437 від 17.08.2016 «Щодо методичних рекомендацій з викладання предметів у середніх школах»).

Pranglimine також розглядається як потужний тренажер для проведення тренувань і змагань з усних обчислень для різних вікових груп. Учасники проєкту представляють такі європейські країни, як Латвія, Литва, Естонія, Словенія та Україна, які щорічно беруть участь у змаганнях з усного рахування; у 2018/19 році в першому онлайновому етапі, що відбувся в Україні, взяло участь понад 70 000 осіб. Для дітей молодшого шкільного віку змагання поділяються на такі вікові групи: чомучки (1-3 класи) і дослідники (4-6 клас). На навчальній платформі Pranglimine учасникам пропонують завдання на заповнення прогалин і порівняння (додавання та віднімання натуральних і цілих чисел, множення та ділення) (рис. 1.9).



Тренування з натуральними числами		Тренування з цілими числами		Тренування з десятковими числами	
Додавання	Стандарт Спринт	Додавання	Стандарт Спринт	Додавання	Стандарт Спринт
Віднімання	Стандарт Спринт	Віднімання	Стандарт Спринт	Віднімання	Стандарт Спринт
Додавання/Віднімання	Стандарт Спринт	Додавання/Віднімання	Стандарт Спринт	Додавання/Віднімання	Стандарт Спринт
Порівняння	Стандарт Спринт	Множення	Стандарт Спринт	Множення	Стандарт Спринт
Вставити пропущене	Стандарт Спринт	Ділення	Стандарт Спринт	Ділення	Стандарт Спринт
Множення	Стандарт Спринт	Множення/Ділення	Стандарт Спринт	Множення/Ділення	Стандарт Спринт
Ділення	Стандарт Спринт	Випадковий	Стандарт Спринт	Випадковий	Стандарт Спринт
Множення/Ділення	Стандарт Спринт				
Випадковий	Стандарт Спринт				

☒ Десяткові: Улюблений триатлон
☐ Статичний триатлон 1
☐ Статичний триатлон 2

Рис. 1.9. Каталог обчислювальних тренажерів на сайті Міксіке

Про ефективний вплив тренажерів Pranglimine на розвиток обчислювальних навичок можна судити з того, що один із найактивніших учасників змагань, О. Шибецький посів шосте місце на Всеукраїнському фіналі з Pranglimine у Львові (2014/15 навчальний рік) та відтоді продовжує покращувати свої результати. Дані про цього учасника можна знайти на офіційному сайті MIXIKE [62].

Динаміка результатів О. Сівецького повністю узгоджується з теоріями японського дослідника Рьо Кавашими [51] та американських вчених Б. Оклі [59], П. Брауна, Г. Редігера та М. Макденієла [45], які підкреслюють, що значна частина мозку тимчасово активізується, що позначається на якості обчислювальної діяльності. Зокрема, Б. Оклі вказує на ефективність завдань, обмежених у часі, стверджуючи, що рівень стресу різко зростає в міру наближення термінів, а гормони стресу стимулюють мозок для допомоги мисленню [59]. Переваги використання онлайн-обчислювальних тренажерів незаперечні й підтверджуються даними нейронаук. Водночас існує проблема відповідності змісту завдання на конкретному уроці змісту завдання, щоб розвинути в учнів міцні обчислювальні навички.

Проте процесу навчання передуює процес формування обчислювального мислення. Тому вчителям важливо розробляти інтерактивні завдання для виконання завдань із підручників, конспектів з друкованою основою, які, на думку П. Гальперіна, виконують формувальний етап розумових операцій. Тому було проаналізовано онлайн-сервіси, які дають змогу створювати інтерактивні завдання з миттєвим зворотним зв'язком та автоматичним оцінюванням результатів обчислювальної діяльності учнів: LearningApps, Live worksheets, Wizer.me тощо.

Потенціал сервісу LearningApps [25] як засобу навчання дітей шкільного віку було детально проаналізовано С. Скворцовою та Т. Бріцканом [37], тому його загальні особливості ми не розглядаємо. Зазначимо лише, що сервіс є безоплатним міжнародним відкритим освітнім ресурсом як один із хмарних сервісів Web 2.0 для організації та підтримки освітнього процесу.

LearningApps характеризується наявністю готових інтерактивних, доступних вчителю, обчислювальних, готових інтерактивних, банків практичних запитань, що їх можуть використовувати вчителі. Щоб допомогти учням розвинути свої обчислювальні навички, сервіс містить у собі значну кількість вправ.

Вчителі можуть створювати завдання, аналогічні тим, що пропонує цей сервіс, або ж створювати власні інтерактивні вправи, використовуючи завдання з підручників чи друкованих конспектів. Варто зазначити, що онлайн-сервіс автоматично перевіряє розрахунки та відображає правильні/неправильні результати в журналі віртуального класу.

Звісно, не всі шаблони LearningApps підходять для створення комп'ютерних навчальних завдань. Розглянемо деякі з них, які можуть підійти для створення інтерактивних завдань: шаблон «Заповни пропуски» найбільш підходить для таких інтерактивних завдань. Він дозволяє учням вписувати результати числових обчислень і невідомі елементи. Недолік цього шаблону в тому, що він не дає змоги використовувати зображення завдань із підручника або робочого зошита. На рис. 1.10 показано варіант інтерактивного розрахункового завдання, створеного за допомогою шаблону «Заповни порожнє місце».

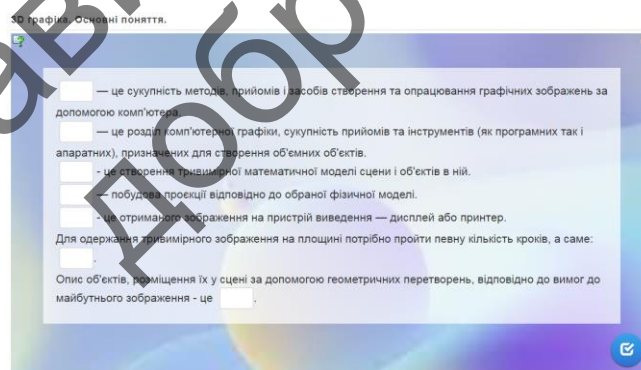


Рис. 1.10. Завдання з шаблону «Заповни порожнє місце» в LearningApps

Шаблон із відкритою відповіддю також може бути використаний для створення обчислювального завдання, але вимагає, щоб учні написали

результат числового виразу під формулою. Однак цей шаблон дозволяє завантажувати як текст, так і зображення. Якщо завантажити зображення, воно відображатиметься так само, як і основний текст, надрукований у підручнику чи зошиті (рис. 1.11). Сервіс негайно перевіряє виконані завдання та повідомляє дитину про результати. Завдання із зеленою рамкою виконані правильно, а з червоною – неправильно.



Рис. 1.11. Завдання можна знайти в шаблоні LearningApps Free Text Response

Можна завантажувати завдання у форматі зображень за допомогою шаблону «Фрагменти зображень». Цей шаблон зручний для обчислювальних завдань у форматі ланцюжка або таблиці. У цьому разі учень не вводить відповідь, а обирає із запропонованих варіантів. Таким чином, під час натискання на шпильку учень відкриває варіанти відповідей і клацає мишею на одному з них (рис. 1.12.).

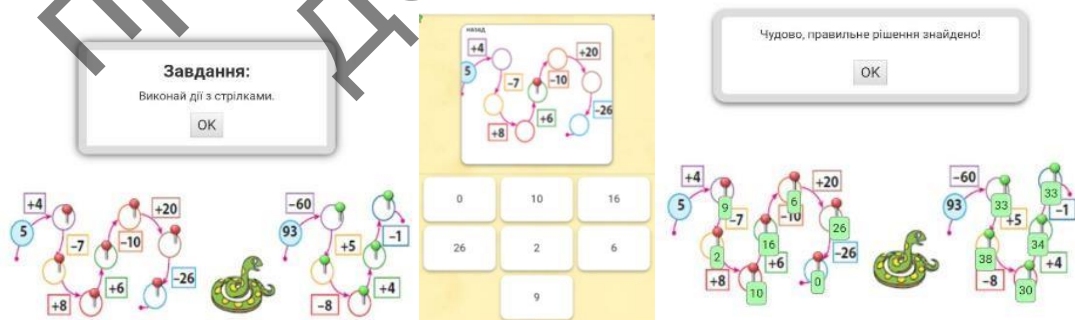


Рис.1.12. Робота над завданням із шаблонем «Фрагменти зображень» у LearningApps

Для створення інтерактивних обчислювальних запитань рекомендуємо шаблон «Знайди пари» (рис. 1.13). Тут завдання може бути представлено як у вигляді тексту, так і у вигляді зображень.

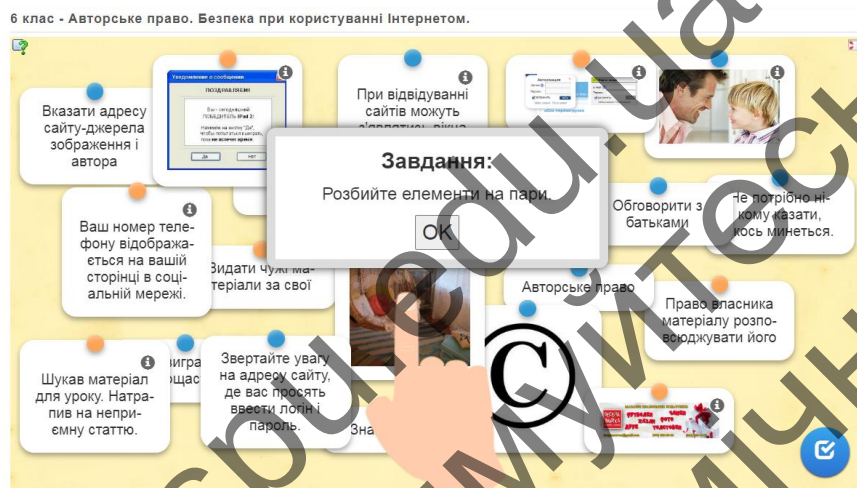


Рис. 1.13. Шаблон LearningApps завдання «Знайди пару».

Перевага LearningApps у тому, що учням можна дати посилання на завдання у вигляді посилання або QR-коду, який можна завантажити на під'єднаний до інтернету гаджет.

Сервіс Liveworksheets [21] – це інтерактивний конструктор робочих аркушів, який має вигляд звичайного робочого аркуша з друкованою основою. Сервіс дає змогу вчителям використовувати готові завдання, створені іншими вчителями, та переглядати шаблони кожного інтерактивного завдання.

Для створення інтерактивних робочих аркушів учителі можуть завантажити сторінки робочого зошита (див. рис. 1.14) і зробити кожне завдання в ньому інтерактивним, використовуючи ключові слова, що відповідають різним шаблонам. Крім того, учні можуть повторити завдання, щоб отримати вищий бал. Після того як учень виконав завдання, сервіс одразу ж перевіряє правильність відповідей. Правильно виконані завдання позначаються зеленим кольором, а неправильно виконані або пропущені – червоним.

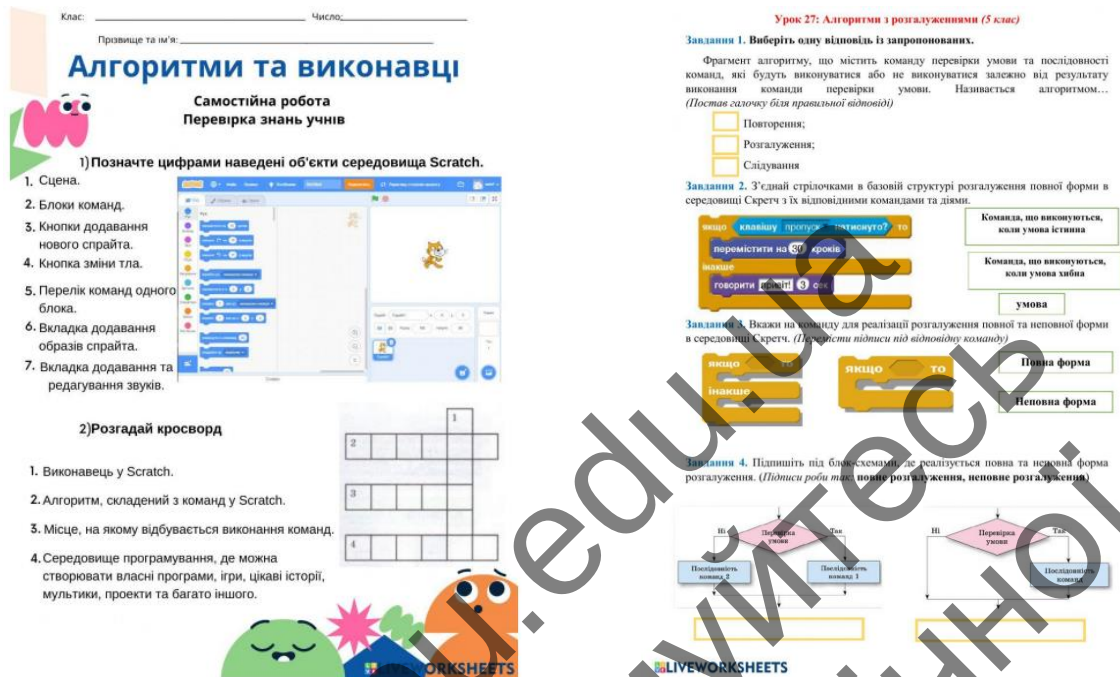


Рис. 1.14. Приклади сторінок робочого зошита

Завдання на робочому аркуші створюються за допомогою шаблону «Текстове поле для запису відповідей», у якому ви пишете числа безпосередньо в комірках зошита. Щоб створити інтерактивне завдання за допомогою цього шаблону, вчителю необхідно виконати кожен крок розв'язання у вигляді текстового поля або узгодити з учнями правила введення. Цей шаблон найбільш привабливий для створення інтерактивних обчислювальних завдань.

Аналогом шаблону LearningApps Pairing у Liveworksheets є шаблон Line Drawing. Цей шаблон можна використовувати в завданнях, де значення виразу обчислюється шляхом вибору відповіді з низки заданих чисел. Однак кількість виразів і кількість відповідей має бути однаковою, і додавати зайві числа не можна, тому цей шаблон дає змогу учням вгадувати відповіді. З цієї причини його не слід використовувати для тренувальних вправ на розвиток навичок обчислень.

Сервіс Wizer.me [42] – це інтерактивний конструктор робочих аркушів, який, як і Liveworksheets, дає змогу користувачам створювати свої власні тренажери для розвитку обчислювальних навичок. Сервіс дає змогу вчителям

переглядати та використовувати готові інтерактивні аркуші, але при їхньому перегляді вони не знають, за яким шаблоном було виконано завдання, як у Liveworksheets.

Завдання завантажується у вигляді зображення з роздрукованого зошита, але, на відміну від Liveworksheets, тут немає текстового поля для написання відповіді. Сервіс LearningApps Free Шаблони «Текстова відповідь» і «Заповни порожнє місце» можна вважати схожими. Однак шаблон LearningApps Fill in the Blanks не дозволяє завантажувати зображення. Також у сервісі Wizer.me є шаблон узгодження «З'єднай пару», який є аналогічним шаблону «Креслення ліній» у Liveworksheets, або шаблону «Утворення пар» LearningApps. Але, як ми зазначали вище, вважаємо недоцільним його використання, оскільки кількість числових виразів і числових результатів є однаковою, і на певному етапі учні можуть вгадувати відповіді.

Таким чином, для ефективного розвитку обчислювального мислення у школярів науковці використовують різні онлайн-тренажери: на сайті OnlineMSchool, в онлайн-середовищі «Міксіке», сервіси Pranglimine, LearningApps, Pairing у Liveworksheets, Line Drawing, Wizer.me. При їх використанні необхідно слідувати теорії поетапного формування розумових дій (П. Гальперін), відповідно до якої важливим етапом є формування дії у матеріальному/матеріалізованому виді з розгорнутим поясненням.

Висновки до розділу 1

У першому розділі «Обчислювальне мислення та його розвиток в умовах сучасної освіти» охарактеризовано обчислювальне мислення як категорію сучасної освіти, описано методики його вимірювання та підходи до розвитку обчислювального мислення молоді в умовах ЗЗСО.

Обчислювальне мислення є однією із ключових компетентностей сучасної освіти, яка спрямована на підготовку учнів до життя в інформаційному суспільстві. Термінологічний аналіз цього поняття засвідчив неоднозначність його тлумачень. В роботі під ОМ розуміється здатність

формалізувати проблеми так, щоб їх можна було вирішити за допомогою обчислювальних засобів. До характеристик ОМ відносять: алгоритмічне мислення; декомпозиція; абстракція; широке застосування; розв'язування проблем; технологічна дотичність; креативний підхід; моделювання та симуляція; універсальність; структуроване вирішення завдань.

Проаналізовано методики вимірювання ОМ: Dr. Scratch, Agent Cubes, Тести Романа-Гонсалеса та ін., Вейнтропа та Віленські, Тест ОМ на основі завдань міжнародного конкурсу «Бобер», Національний іспит у Ізраїлі як інструмент вимірювання ОМ, Тест SCS1, Тест Є. Чобан, О. Коркмаз, Тест Кентського університету, «словник ОМ». Зіставлення методик показало, що кожна з них оперує як спільними, так і окремими характеристиками ОМ, проте в усіх методиках досліджується вміння алгоритмізувати розв'язування задач.

За результатами контент-аналізу показано, що для ефективного розвитку обчислювального мислення у школярів використовуються різні онлайн-тренажери: на сайті OnlineMSchool, в онлайн-середовищі «Міксіке», сервіси Pranglimine, LearningApps, Pairing у Liveworksheets, Line Drawing, Wizer.me. При їх використанні необхідно слідувати теорії поетапного формування розумових дій П. Гальперіна.

Розділ 2.

ІНФОРМАТИКА ЯК МАЙДАНЧИК РОЗВИТКУ ОБЧИСЛЮВАЛЬНОГО МИСЛЕННЯ УЧНІВ

2.1. Аналіз навчальних програм з інформатики

Розглянемо навчальні програми з інформатики на наявність в них тем, пов'язаних з вивченням алгоритмізації та програмування.

Метою навчання інформатики, за модельною навчальною програмою «Інформатика. 5-6 класи» для закладів загальної середньої освіти (автори Ривкінд Й.Я., Лисенко Т.І., Чернікова Л.А., Шакотько В.В.) [65], відповідно до національних стандартів базової середньої освіти є формування характеру учнів, які можуть використовувати цифрові інструменти та технології для розв'язання та розроблення проблем, творчо виражати себе, забезпечувати власний і суспільний добробут, критично мислити та діяти безпечно і відповідально в інформаційному суспільстві.

Основою досягнення результатів навчання, визначених у модельній програмі, є діяльнісний підхід, що ґрунтується на доборі інформаційних об'єктів, у процесі опрацювання яких формуються відповідні ключові компетенції та предметні компетенції. Діяльнісний підхід також включає в себе реалізацію об'єктного підходу. Інформаційні технології використовуються для обробки конкретних інформаційних об'єктів із певними властивостями. Щоб обробити об'єкт, необхідно змінити значення його властивостей. Щоб змінити значення властивості об'єкта, необхідно виконати певну дію над ним. Це означає реалізацію певного алгоритму, що в свою чергу, розвиває у школярів обчислювальне мислення.

Зміст навчального предмету Інформатика в 5-6 класах вибудовується за такими предметними змістовими лініями: інформаційні процеси та системи; комп'ютерні мережі; інформаційні технології; алгоритмізація та програмування. На їх вивчення відводиться 70-140 годин.

Для розвитку ОМ найважливішою змістовною лінією є алгоритмізація та

програмування. Вона представлена темою «Алгоритми та програми» в 5-му та 6-му класах. Зміст даної змістовної лінії відображено на рис. 2.1.

5 клас	6 клас
☐ Команди і виконавці.	☐ Алгоритми і програми з вкладеними циклами.
☐ Система команд виконавця.	☐ Алгоритми і програми з вкладеними розгалуженнями.
☐ Алгоритми. Різні способи подання алгоритмів.	☐ Алгоритми і програми з циклами з передумовою.
☐ Алгоритми і програми. Середовища проектування.	☐ Алгоритми і програми з вкладеними циклами і розгалуженнями
☐ Візуальні середовища проектування.	
☐ Лінійні алгоритми і програми.	
☐ Алгоритми і програми з циклами з лічильником.	
☐ Висловлювання. Істинні та хибні висловлювання.	
☐ Алгоритми і програми з розгалуженнями.	
☐ Тестування алгоритмів і програм.	

Рис. 2.1. Зміст теми «Алгоритми та програми» в 5 та 6 класі

Окремо варто виділити у 6-му класі змістову лінію «Інформаційні технології», яка передбачає вивчення електронних таблиць. Тут передбачається, що учні навчаються визначати і обґрунтовувати прості причинно-наслідкові зв'язки, використовуючи шаблон “якщо, то”, “що треба зробити, щоб”; використовувати ПЗ для простих розрахунків і візуалізації результатів; створювати електронні таблиці з арифметичними обчисленнями тощо. Цей набір пропонування вмінь безпосередньо приймає участь у розвитку обчислювальних вмінь учнів.

У модельній навчальній програмі для 7-9 класу [64] на вивчення інформатики відводиться 140-280 годин і продовжуються ті ж змістовні лінії.

Розвиток ОМ під час вивчення теми «Алгоритми та програми» відбувається за рахунок знайомлення зі змістом навчального предмету, а саме, в 7-9-х класах (рис. 2.2. – 2.4).



Рис. 2.2. Зміст теми «Алгоритми та програми» в 7 класі



Рис. 2.3. Зміст теми «Алгоритми та програми» в 8 класі



Рис. 2.4. Зміст теми «Алгоритми та програми» в 9 класі

У старших класах, за навчальною програмою [58] вивчення програмування передбачено тільки у вибіркового модулі «Креативне програмування» на вивчення якого виділяється 35 годин. Саме в цьому модулі продовжується розвиток ОМ учнів.

Таким чином, розглянувши навчальні програми з інформатики для учнів 5-11-х класів ЗЗСО можна зазначити, що програмування допомагає учням розвивати обчислювальне мислення через:

- Практичні навички. Учні постійно практикуються в алгоритмічному мисленні та знаходженні рішень для реальних проблем.
- Постійний аналіз та оптимізацію. Вони вчаться аналізувати ефективність своїх алгоритмів, що розвиває критичне мислення і здатність до оптимізації.
- Інтеграція з іншими дисциплінами. Програмування дозволяє застосовувати обчислювальне мислення в різних галузях, таких як математика, фізика, економіка, що розширює застосування цих навичок.

Отже, навчання алгоритмізації та програмування є невід'ємним елементом розвитку обчислювального мислення, оскільки саме ці дисципліни формують у школярів здатність системно підходити до вирішення проблем і застосовувати методи обчислень у різних контекстах.

2.2. Успішні практики, орієнтовані на розвиток обчислювального мислення учнів

Аналіз різних навчальних програм, що орієнтовані на розвиток ОМ, свідчать про їхню різноманітність. Наприклад, дослідження, пов'язані з адаптивним навчанням, вказують на важливість технологічних інструментів для підтримки ОМ, що сприяє розвитку навичок критичного мислення та програмування [29]. Це підтверджує, що технологічні ресурси можуть суттєво вплинути на навчальні результати учнів.

Впровадження ОМ у навчальні програми демонструє позитивні результати, особливо в розвитку критичного мислення та навичок розв'язання проблем. Наприклад, учні, які вивчають програми, орієнтовані на розвиток ОМ, зазвичай мають кращі результати в розумінні складних концепцій і їх застосуванні в реальних ситуаціях [9]. ОМ також сприяє розвитку міждисциплінарного підходу, що дозволяє учням бачити зв'язки між різними галузями знань [7].

Дослідження Ningtyas D. P., Setyosari P., Kuswandi D., Ulfa Saidi. [31] вказують на важливість адаптації навчальних стратегій до індивідуальних потреб учнів. Використання ігрових елементів у навчанні ОМ показало значне підвищення ефективності, особливо серед дітей з різними когнітивними стилями. Це підкреслює необхідність врахування різних стилів навчання при впровадженні ОМ у навчальний процес.

Інтерактивні платформи можуть позитивно впливати на навчання, сприяючи розвитку обчислювального мислення. Наприклад, участь у практичних вправах в рамках MOOCs підвищує ефективність навчання [24]. Важливою є також роль навчальних середовищ, які заохочують активну участь учнів, створюючи умови для експериментування та вирішення проблем.

Культурні фактори формують уявлення про ОМ, а також різноманітні підходи до його викладання в різних країнах. Наприклад, у США акцент робиться на інтеграцію ОМ у всі предмети, тоді як в інших країнах можуть використовуватися більш традиційні методи навчання, що обмежує розвиток

ОМ в освітньому процесі [27]. З огляду на це, обчислювальне мислення є важливим елементом сучасної освіти.

Отже, узагальнення успішних практик розвитку ОМ свідчить, що вони передбачають розвиток обчислювального мислення через такі компоненти:

- Основи алгоритмізації та програмування. Учні знайомляться з базовими алгоритмічними структурами, що закладає фундамент для розвитку логічного мислення.
- Розробка програмних проєктів. Проєкти, що передбачають розробку програм, допомагають учням застосовувати отримані знання на практиці та розвивати навички вирішення задач.
- Використання сучасних програмних засобів. Інтеграція сучасних інструментів та технологій у навчальний процес сприяє розвитку обчислювального мислення на практичному рівні.

Розвиток ОМ, в першу чергу, пов'язаний із вивченням в школах основ алгоритмізації та програмування, оскільки ці теми вчать учнів мислити структуровано, логічно й системно. Наведемо деякі обґрунтування взаємозв'язку вивчення алгоритмізації та програмування й розвитку ОМ:

1. Алгоритмізації та програмування вимагає здатності мислити алгоритмічно, тобто розробляти послідовні кроки (алгоритми) для вирішення конкретних задач. Під час написання програм учні навчаються декомпонувати проблеми, розробляти покрокові інструкції та шукати оптимальні шляхи їхнього розв'язання. Це є важливою частиною обчислювального мислення, оскільки дозволяє структурувати завдання таким чином, щоб їх можна було вирішити за допомогою обчислень.

2. Алгоритмізації та програмування вимагає розбиття складних проблем на менші, легші для розуміння та вирішення частини. Цей процес декомпозиції допомагає учням побачити структуру проблеми і підходити до її вирішення з більш системним підходом. Навички декомпозиції, які відпрацьовуються

через програмування, безпосередньо сприяють розвитку обчислювального мислення.

3. Алгоритмізації та програмування передбачає не лише розуміння алгоритмів, але й здатність реалізувати їх у вигляді програмного коду. Учні, які вивчають програмування, вчаться розробляти ефективні алгоритми, аналізувати їхню роботу та оптимізувати рішення. Це сприяє розвитку таких ключових елементів обчислювального мислення, як пошук шаблонів, оптимізація та ефективне управління даними.

4. Алгоритмізації та програмування розвиває здатність до абстрагування, коли учні вчаться виділяти основні аспекти проблеми, ігноруючи несуттєві деталі. Створення програмних модулів або функцій вимагає вміння створювати абстрактні моделі, що дозволяють узагальнювати рішення і застосовувати їх до різних проблем.

5. Під час програмування учні постійно стикаються з потребою перевіряти логічність своїх дій, відстежувати помилки (debugging) та шукати причини їх виникнення. Цей процес розвиває критичне мислення і вміння оцінювати власні рішення з точки зору їхньої правильності та ефективності, що є важливою складовою обчислювального мислення.

6. Алгоритмізації та програмування тісно пов'язані з обробкою та аналізом даних. Учні вчаться працювати з різними типами даних, створювати структури для їх збереження та ефективно використовувати їх у своїх алгоритмах. Це формує здатність до структурованого та систематичного опрацювання інформації, що також є важливою частиною обчислювального мислення.

7. Алгоритмізації та програмування вчить мислити проектно – від початкового формулювання проблеми до остаточного її розв'язку у вигляді програми. Учні проходять через етапи планування, розробки, тестування та впровадження, що розвиває вміння системно підходити до розв'язання будь-яких проблем.

8. Алгоритмізації та програмування також розвиває креативність, оскільки учні можуть знайти кілька можливих шляхів вирішення однієї проблеми. Вибір найбільш оптимального підходу вимагає вміння оцінювати різні варіанти, передбачати можливі наслідки і приймати рішення на основі аналізу, що формує важливі елементи обчислювального мислення.

2.3. Типові завдання з інформатики на розвиток обчислювального мислення учнів

Типові завдання – це завдання, які часто зустрічаються в певній сфері або дисципліні, і які допомагають закріпити навички та знання, необхідні для успішного виконання більш складних завдань [55].

У контексті розвитку обчислювального мислення, типові завдання:

- Мають чітко визначену мету – вони спрямовані на розвиток конкретних навичок, таких як алгоритмізація, аналіз даних, розв’язання проблем.
- Відрізняються рівнем складності – завдання можуть бути простими для початківців і складними для досвідчених користувачів.
- Можуть бути різноманітними за формою –можуть бути текстові задачі, логічні головоломки, програмування, аналіз даних тощо.
- Повторюються з різними даними або умовами – це дозволяє учням закріпити навички і застосовувати їх у різних ситуаціях.

Л. Семко розділяє типові завдання з інформатики на кілька категорій [67] (рис. 2.5).

Алгоритмічні задачі	Ці задачі спрямовані на розвиток навичок алгоритмізації. Учні вчаться розуміти та створювати алгоритми для розв'язання різноманітних задач.
Задачі на програмування	Завдання, що включають написання програмного коду, допомагають учням закріпити свої знання про алгоритми та навчитися перетворювати їх у працюючі програми.
Проектні задачі	Проекти, що потребують розробки та реалізації програмних продуктів, сприяють розвитку навичок планування, управління часом та командної роботи.
Задачі на абстрагування та моделювання	Ці завдання допомагають учням вчитися абстрагувати складні поняття та моделювати їх у вигляді простіших структур.

Рис. 2.5. Типові завдання з інформатики

Алгоритмічні задачі є основою для розвитку обчислювального мислення. Учні вчаться розбивати складні проблеми на простіші підзадачі, формувати послідовність кроків для їх розв'язання та аналізувати ефективність запропонованих рішень. Наприклад, учням може бути запропоновано завдання написати алгоритм для сортування масиву чисел або для пошуку найкоротшого шляху в графі.

Завдання на програмування дозволяють учням застосовувати свої знання алгоритмів на практиці. Такі завдання можуть включати написання програм на різних мовах програмування, створення простих ігор або розробку веб-додатків. Прикладом може бути завдання написати програму, що перевіряє

правильність розв'язання математичного рівняння, або створити веб-сторінку, яка відображає дані з бази даних.

Проектні задачі допомагають учням вчитися працювати у командах, планувати свою роботу та ефективно використовувати час. Учні можуть працювати над проєктами, що включають розробку програмного забезпечення, створення вебсайтів або моделювання фізичних процесів. Проектна діяльність також сприяє розвитку навичок презентації та спілкування, оскільки учні повинні представити свої проєкти перед класом або журі.

Завдання на абстрагування та моделювання допомагають учням навчитися узагальнювати складні поняття та представляти їх у вигляді простіших моделей. Наприклад, учні можуть створювати моделі економічних процесів, моделювати поведінку систем або розробляти симуляції фізичних явищ. Такі завдання сприяють розвитку критичного мислення та здатності до аналізу складних систем.

Для оцінювання ефективності типових завдань з інформатики використовуються різні методи. Вчителі можуть використовувати тести, контрольні роботи, а також спостереження за роботою учнів під час виконання завдань. Важливою частиною оцінювання є також самооцінка учнів, що дозволяє їм аналізувати власні досягнення та визначати напрямки для подальшого вдосконалення.

Типові завдання з інформатики для розвитку ОМ можуть бути різноманітними та адаптованими до віку учнів.

Для молодших школярів:

- Логічні головоломки: Лабіринти, sudoku, шашки, ребуси.
- Сортування предметів за різними ознаками: Колір, форма, розмір.
- Складання простих алгоритмів: Послідовність дій для виконання завдання (наприклад, як приготувати бутерброд).

– Блокові програми: Створення простих анімацій або ігор за допомогою візуальних середовищ програмування (Scratch, Blockly).

Для середньої школи:

- Розв’язання задач на логіку: Задачі на переливання води, розгадування шифрів, математичні головоломки.
- Створення алгоритмів: Розробка алгоритмів для сортування даних, пошуку елементів у списку, розв’язання математичних задач.
- Програмування простих ігор: Створення ігор з використанням елементів штучного інтелекту (наприклад, лабіринти, стратегії).
- Аналіз даних: Збір та аналіз даних з різних джерел (таблиці, графіки), пошук закономірностей.

Для старшої школи:

- Розробка алгоритмів для складних задач: Розв’язання оптимізаційних задач, моделювання реальних процесів.
- Програмування на різних мовах: Вивчення мов програмування Python, Java, C++ для розробки більш складних програм.
- Створення вебдодатків: Розробка простих вебсайтів або вебдодатків.
- Робота з базами даних: Створення та управління базами даних для зберігання та обробки інформації.

Завдання з інформатики для розвитку обчислювального мислення повинні відповідати віковим особливостям учнів, тому важливо враховувати рівень абстрактного мислення та здатність до самостійного вирішення задач.

Нижче наводимо приклади типових завдань для кожної вікової групи з поясненням їхньої важливості для розвитку обчислювального мислення.

Для молодших школярів (1-4 клас)

Завдання для цієї вікової категорії повинні бути максимально простими, зосередженими на базових поняттях, таких як послідовність, повторення, ігрова форма.

1. Завдання: Скласти послідовність дій для казкового героя

Опис: учні створюють алгоритм для героя казки (наприклад, Червона Шапочка), щоб він дійшов до бабусі, уникаючи перешкод.

Розвиток обчислювального мислення: це завдання навчає дітей базовим поняттям послідовності дій, розвиває логіку та планування.

2. Завдання: Побудувати лабіринт і знайти вихід

Опис: використовуючи прості графічні інтерфейси або роботів, учні розробляють шлях через лабіринт.

Розвиток обчислювального мислення: учні вчаться декомпонувати задачу, створюючи прості алгоритми, що розвиває навички аналізу та пошуку рішень.

3. Завдання: Перетягування блоків у програмі Scratch для створення анімації

Опис: учні створюють прості анімації, використовуючи блоки для написання програм у середовищі Scratch.

Розвиток обчислювального мислення: робота з блоковими програмами сприяє розумінню принципів послідовності та циклів у програмуванні.

4. Завдання: Створення алгоритму складання цифр

Опис: учні повинні придумати послідовність дій для складання чисел (наприклад, додати два числа).

Розвиток обчислювального мислення: це завдання допомагає розвинути вміння мислити логічно та структурувати завдання в алгоритмічному вигляді.

5. Завдання: Гра в цифрову версію гри "Саймон"

Опис: гра, де учні повинні повторювати послідовності кольорів, що з'являються на екрані.

Розвиток обчислювального мислення: завдання допомагає розвинути навички запам'ятовування послідовностей та створення шаблонів для вирішення проблем.

Для учнів середньої школи (5-8 клас)

У цьому віці завдання мають бути більш складними і включати базові елементи програмування, логіки та роботи з даними.

1. Завдання: Написання алгоритму для обчислення площі фігур

Опис: учні розробляють алгоритм для обчислення площі простих геометричних фігур (квадрат, трикутник, коло).

Розвиток обчислювального мислення: учні працюють з формулами і логікою програмування, що допомагає розвинути вміння вирішувати математичні завдання за допомогою алгоритмів.

2. Завдання: Створення програми для перевірки парності числа

Опис: написати програму, яка перевіряє, чи є введене число парним.

Розвиток обчислювального мислення: завдання вчить застосовувати умовні оператори і працювати з логічними виразами, розвиваючи навички абстрактного мислення.

3. Завдання: Створення програми для генерації випадкових чисел

Опис: учні розробляють програму, яка генерує випадкові числа та виводить їх на екран.

Розвиток обчислювального мислення: завдання включає роботу з випадковістю і алгоритмами генерації даних, що допомагає розвинути розуміння того, як програми можуть взаємодіяти з непередбачуваними результатами.

4. Завдання: Алгоритм сортування масиву

Опис: створити простий алгоритм сортування чисел (наприклад, бульбашкове сортування).

Розвиток обчислювального мислення: учні навчаються структурувати і оптимізувати дані, що розвиває навички аналізу та ефективного вирішення проблем.

5. Завдання: Написання програми для перетворення одиниць вимірювання

Опис: створити програму, яка перетворює сантиметри в метри або градуси Цельсія у Фаренгейти.

Розвиток обчислювального мислення: це завдання вчить учнів працювати з функціями та математичними операціями, що допомагає розвивати здатність знаходити алгоритмічні рішення реальних завдань.

Для старшої школи (9-11 клас)

Завдання для старшокласників можуть включати більш складні алгоритмічні концепції, роботу з великими обсягами даних і використання елементів штучного інтелекту.

1. Завдання: Написання програми для обчислення факторіала

Опис: створити програму для обчислення факторіала числа.

Розвиток обчислювального мислення: завдання вчить працювати з рекурсивними алгоритмами та декомпозицією, що розвиває системне мислення та розуміння складних математичних операцій.

2. Завдання: Створення чат-бота

Опис: розробити простого чат-бота, який відповідає на запитання користувача на основі попередньо заданих відповідей.

Розвиток обчислювального мислення: учні вчаться моделювати взаємодію людини з комп'ютером, що розвиває логіку, аналіз ситуацій і застосування алгоритмів обробки тексту.

3. Завдання: Написання алгоритму для пошуку найкоротшого шляху (алгоритм Дейкстри)

Опис: реалізувати алгоритм для пошуку найкоротшого шляху в графах (наприклад, на карті).

Розвиток обчислювального мислення: завдання включає роботу з оптимізацією і теорією графів, що розвиває критичне мислення та здатність вирішувати складні задачі.

4. Завдання: Створення програми для аналізу великих наборів даних

Опис: написати програму, яка аналізує великий набір даних і видає основні статистичні показники (середнє значення, медіану, моду).

Розвиток обчислювального мислення: це завдання допомагає учням працювати з великими масивами даних і застосовувати алгоритми для їх обробки, що розвиває навички роботи з інформацією та структурованого мислення.

5. Завдання: Створення алгоритму для автоматичної гри "Хрестики-нулики"

Опис: створити програму, яка грає в "Хрестики-нулики" проти користувача.

Розвиток обчислювального мислення: учні вчаться створювати алгоритми на основі аналізу стану гри та прийняття рішень, що розвиває вміння планувати наперед та мислити стратегічно.

Ці завдання є типовими, оскільки вони охоплюють широкий спектр навичок, необхідних для успішної роботи в сфері інформатики та інших областях, що вимагають аналітичного мислення.

Щоб перевірити, наскільки успішно учні опановують ОМ, можна скористатися низкою підходів, що подані на рис. 2.6 і які пропонує штучний інтелект Gemini.

Ознаками успішного розвитку ОМ є: здатність розбивати складні задачі на простіші підзадачі; вміння виділяти повторювані дії і формалізувати їх у вигляді циклів; вибір оптимального типу циклу для розв'язання задачі; здатність до дебагування і виправлення помилок у програмах; вміння аналізувати ефективність алгоритмів і пропонувати способи їх оптимізації; креативність у підході до розв'язання задач.

1. Спостереження за процесом розв'язання задач:	Аналіз ходу думок: Спостерігайте, як учні підходять до розв'язання задачі. Чи вміють вони правильно виділити повторювані дії, визначити умову циклу та вибрати відповідний тип циклу? Використання блок-схем: Запропонуйте учням візуалізувати алгоритм за допомогою блок-схем. Це допоможе їм краще структурувати свої думки і виявити можливі помилки. Дебагування: Спостерігайте, як учні виправляють помилки у своїх програмах. Чи вміють вони використовувати інструменти для налагодження, ставити експерименти і аналізувати результати?
2. Аналіз готових рішень:	Якість коду: Оцініть, наскільки код учнів є зрозумілим, структурованим і ефективним. Чи використовують вони правильні конструкції мови, дотримуються правил стилю кодування? Правильність результатів: Перевірте, чи програми учнів дають правильні результати на різних вхідних даних. Ефективність алгоритмів: Порівняйте різні рішення однієї задачі і обговоріть їхню ефективність з точки зору швидкодії та використання пам'яті.
3. Тестування знань:	Теоретичні питання: Задавайте учням теоретичні питання про цикли, їх типи, умови циклу, застосування. Практичні завдання: Пропонуйте учнів розв'язати різноманітні задачі на побудову алгоритмів з циклами, що вимагають застосування різних навичок. Проекти: Запропонуйте учням розробити невеликі проекти, в яких вони зможуть застосувати свої знання про цикли для вирішення реальних задач.
4. Відкриті запитання та дискусії:	Запитання про алгоритми: Задавайте учням відкриті запитання, які стимулюють їх до критичного мислення і пошуку альтернативних рішень. Наприклад: "Які ще способи можна використати для розв'язання цієї задачі?", "Як можна оптимізувати цей алгоритм?" Обговорення алгоритмів: Організуйте дискусії, на яких учні зможуть обмінюватися своїми ідеями і порівнювати різні підходи до розв'язання задач.
5. Використання онлайн-платформ:	Інтерактивні завдання: Використовуйте онлайн-платформи з інтерактивними завданнями, які дозволяють відразу перевірити правильність рішення. Конкурси з програмування: Запропонуйте учням взяти участь у конкурсах з програмування, щоб вони могли порівняти свої результати з результатами інших учнів.

Рис. 2.6. Підходи до визначення розвитку ОМ

Інший спосіб для оцінки розвитку ОМ – це проведення тестування «Тестування для оцінки розвитку обчислювального мислення».

Мета тесту: оцінити рівень розуміння учнями основних понять алгоритмізації, здатність будувати алгоритми з використанням циклів та умовних операторів, а також вміння аналізувати і виправляти помилки в алгоритмах.

Тест складається з теоретичних питань та практичних завдань.

Теоретична частина:

- *Запитання з множинним вибором:*
 - Що таке алгоритм? Які його основні властивості?
 - Які типи циклів ви знаєте? Наведіть приклади їх застосування.
 - Які умовні оператори ви знаєте? Для чого вони використовуються?
 - Що таке декомпозиція задачі? Навіщо її застосовують при розробці алгоритмів?
- *Відкриті запитання:*
 - Поясніть, в чому полягає різниця між лінійним алгоритмом і алгоритмом з циклом.
 - Наведіть приклад алгоритму, який розраховує суму перших N натуральних чисел.
 - Опишіть, як би ви побудували алгоритм для сортування масиву чисел за зростанням.

Практична частина

- *Завдання на побудову алгоритмів:*
 - Складіть алгоритм, який виводить на екран всі парні числа від 1 до 20.
 - Напишіть алгоритм, який обчислює факторіал заданого числа.
 - Розробіть алгоритм, який перевіряє, чи є задане число простим.
- *Завдання на виправлення помилок:*
 - У наведеному алгоритмі знайдіть помилки і виправте їх:
 - для i від 1 до 10 з кроком 2
 - вивести i
 - кінець для
- *Завдання на аналіз алгоритмів:*
 - Порівняйте два наведені алгоритми і поясніть, який з них більш ефективний:
 - Алгоритм 1:
 - для i від 1 до N
 - якщо i ділиться на 2 без остачі то
 - вивести i
 - кінець якщо
 - кінець для
 - Алгоритм 2:
 - i := 2
 - поки i <= N
 - вивести i
 - i := i + 2
 - кінець поки

Особливості оцінювання:

- Теоретична частина - оцінюється правильність відповідей на запитання, глибина розуміння понять.
- Практична частина - оцінюється правильність побудованих алгоритмів, ефективність використання циклів та умовних операторів, здатність аналізувати і виправляти помилки.

Критеріями розвитку обчислювального мислення є: розуміння основних понять (учень демонструє чітке розуміння таких понять, як алгоритм, цикл, умова, декомпозиція); уміння алгоритмізації (учень може самостійно будувати алгоритми для розв'язання різних задач, використовуючи цикли та умовні оператори); аналітичні навички (учень вміє аналізувати алгоритми, виявляти в них помилки та пропонувати способи їх виправлення); логічне мислення (учень демонструє здатність до логічних міркувань, послідовності дій при розв'язанні задач); креативність (учень пропонує оригінальні рішення задач, використовує нестандартні підходи).

При цьому важливо забезпечувати: індивідуальний підхід, складнощі завдань слід підбирати з урахуванням рівня підготовки учнів; різноманітність завдань, вони мають бути різноманітними за тематикою та рівнем складності; зворотний зв'язок, надавати учням фідбек-аналіз щодо їхніх результатів.

Типові завдання з інформатики є важливим інструментом для розвитку обчислювального мислення у школярів. Вони допомагають учням навчитися вирішувати складні задачі, розвивати логічне мислення та здатність до аналізу. Завдяки цим завданням учні можуть не лише підготуватися до подальшого навчання у галузі інформаційних технологій, але й отримати навички, необхідні для успішного функціонування у сучасному цифровому суспільстві.

Висновки до розділу 2

У другому розділі «Інформатика як майданчик розвитку обчислювального мислення учнів» проаналізовано чинні програми з

інформатики, узагальнено успішні освітні практики, орієнтовані на розвиток ОМ учнів та запропоновано типові завдання з інформатики на розвиток ОМ.

За аналізом навчальних програм з інформатики встановлено можливість розвитку ОМ учнів при вивченні тем, пов'язаних з алгоритмізацією та програмуванням, оскільки ці теми сприяють розвитку навичок побудови алгоритмів для вирішення різних задач в різних галузях, таких як математика, фізика, економіка, що розширює застосування цих навичок.

За результатами узагальнення успішних практик також доведено, що розвиток ОМ, в першу чергу, пов'язаний із вивченням в школах основ алгоритмізації та програмування, оскільки ці теми вчать учнів мислити структуровано, логічно й системно.

Типові завдання з інформатики для розвитку обчислювального мислення у школярів представлені саме в контексті вивчення алгоритмізації та програмування. Завдання з інформатики для розвитку обчислювального мислення повинні відповідати віковим особливостям учнів, тому важливо враховувати рівень абстрактного мислення та здатність до самостійного вирішення задач. Наведено приклади типових завдань для кожної вікової групи з поясненням їхньої важливості для розвитку обчислювального мислення. Також запропоновано авторські методики для встановлення розвитку ОМ на основі тестів.

ВИСНОВКИ

В роботі подано авторське вирішення проблеми розвитку ОМ в навчанні інформатики. Проведене дослідження засвідчує досягнення мети, вирішення поставлених завдань та уможлиблює такі **висновки**.

1. У сучасній освіті розвиток обчислювального мислення стає все більш важливим завданням, оскільки технології швидко розвиваються, а з ними зростає і потреба в навичках, які дозволяють учням орієнтуватися в цифровому середовищі. Обчислювальне мислення охоплює не лише технічні аспекти освіти, пов'язані з програмуванням, а й широкі інтелектуальні процеси, зокрема логічне мислення, аналіз та креативність здобувачів освіти.

2. Під ОМ розуміється здатність формалізувати проблеми так, щоб їх можна було вирішити за допомогою обчислювальних засобів. До характеристик ОМ відносять: алгоритмічне мислення; декомпозиція; абстракція; широке застосування; розв'язування проблем; технологічна дотичність; креативний підхід; моделювання та симуляція; універсальність; структуроване вирішення завдань.

3. Серед методик, за допомогою яких можна визначити розвиток ОМ, відзначені тести Dr. Scratch, Agent Cubes, Романа-Гонсалеса та ін., Вейнтропа та Віленські, SCS1, Є. Чобан і О. Коркмаз, Кентського університету, «словник ОМ» та інші. За результатами контент-аналізу показано, що для ефективного розвитку обчислювального мислення у школярів використовуються різні онлайн-тренажери: на сайті OnlineMSchool, в онлайн-середовищі «Міксіке», сервіси Pranglimine, LearningApps, Pairing у Liveworksheets, Line Drawing, Wizer.me. Зіставлення методик показало, що кожна з них оперує як спільними, так і окремими характеристиками ОМ, проте в усіх методиках досліджується уміння алгоритмізувати розв'язування задач.

4. Навчання алгоритмізації та програмування є невід'ємним елементом розвитку обчислювального мислення, оскільки саме ці дисципліни формують у школярів здатність системно підходити до вирішення проблем і застосовувати методи обчислень у різних ситуаціях. Сучасні навчальні

програми з інформатики серед іншого орієнтуються на розвиток ОМ. Типові завдання з інформатики грають важливу роль у формуванні цих навичок.

Проведене дослідження довело важливість інтеграції ОМ у навчальні програми, включаючи: інтеграцію ОМ у всі рівні навчальних програм, створення спеціалізованих навчальних матеріалів, оцінку та моніторинг прогресу учнів.

fizmat@sspi.edu.ua
суворо дотримуйтесь
правил академічної
добросовісності

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Basawapatna A., Repenning A., Koh K. H., et al. Recognizing computational thinking patterns / Proceedings of the 42nd ACM technical symposium on Computer science education. ACM, 2011. P. 245–250
2. Basawapatna A., Repenning A., Koh K. H., et al. The zones of proximal flow: guiding students through a space of computational thinking skills and challenges / Proceedings of the ninth annual international ACM conference on International computing education research. ACM, 2013. P. 67–74.
3. Bocconi S., Chiocciariello A., Dettori G., Ferrari A., Engelhardt K. (2016). Developing computational thinking in compulsory education – Implications for policy and practice. URL: http://publications.jrc.ec.europa.eu/repository/bitstream/JRC104188/jrc104188_compu_thinkreport.pdf
4. Boyer J. T. Using scratch for learner-constructed multimedia: A design-based research inquiry of constructionism in practice. University of Florida, 2010.
5. Brennan K., Resnick M. New frameworks for studying and assessing the development of computational thinking // Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada. 2012. Vol. 1. P. 25.
6. Cardoso M., Morgado E., Leonido Levi. "Unleashing Creative Synergies: A Mixed-Method Case Study in Music Education Classrooms." Applied Sciences (2023). 10.3390/app13179842/
7. Cardoso M., Morgado E., Leonido Levi. "Unleashing Creative Synergies: A Mixed-Method Case Study in Music Education Classrooms." Applied Sciences (2023). 10.3390/app13179842
8. Chu-Yang Chang, Zhengyi Du, H. Kuo, Chih-Ching Chang. "Investigating the Impact of Design Thinking-Based STEAM PBL on Students Creativity and Computational Thinking." IEEE Transactions on Education (2023). 10.1109/TE.2023.3297221

9. Chu-Yang Chang, Zhengyi Du, H. Kuo, Chih-Ching Chang. "Investigating the Impact of Design Thinking-Based STEAM PBL on Students Creativity and Computational Thinking." *IEEE Transactions on Education* (2023). 10.1109/TE.2023.3297221
10. Çoban E., Korkmaz Ö. An alternative approach for measuring computational thinking: Performance-based platform // *Thinking Skills and Creativity*. 2021. Vol. 42. P. 25.
11. Coding For Kids and Teens Made Easy. URL: <https://www.tynker.com/>
12. Computer Science Teachers Association (CSTA). URL: <https://csteachers.org/>
13. Core STEM Competencies in STEM Education. URL: <https://weaveproject.ie/printables/corestemcompetencesinstemeducation.pdf>
14. Corradini I., Lodi M., Nardelli E. (2017). Conceptions and Misconceptions about Computational Thinking among Italian Primary School Teachers. URL: <https://hal.inria.fr/hal01636235/document>
15. Curzon P., McOwan P.W. Computational thinking: A method of solving complex problems. 2018. 266 p.
16. Dagienė V., Futschek G. Bebras international contest on informatics and computer literacy: Criteria for good tasks // *International conference on informatics in secondary schools-evolution and perspectives*. Springer, Berlin, Heidelberg, 2008. P. 19–30.
17. Google for Education. URL: <https://edu.google.com/>
18. Grover S. Robotics and engineering for middle and high school students to develop computational thinking // *Annual meeting of the American educational research association*, New Orleans, LA, 2011.
19. Harel I. E., Papert S. E. *Constructionism*. Ablex Publishing, 1991.
20. Huitt W. Bloom et al.'s taxonomy of the cognitive domain // *Educational psychology interactive*. 2011. Vol. 22. P. 1–4.; Rudenko Yu., Proshkin V., Naboka O., Yurchenko A., Semenikhina O. Using Bloom's taxonomy to assess information hygiene skills. In E. Smyrnova-Trybulska (ed.). *E-learning & Artificial*

Intelligence (AI). 2023. „E-learning” Series, 15, Katowice–Cieszyn, pp. 137–148.
<https://doi.org/10.34916/el.2023.15.12>.

21. Interactive Worksheets For all Languages and Subjects. URL:
<https://www.liveworksheets.com/>

22. International Society for Technology in Education, ISTE. URL:
<https://iste.org/>

23. Izu C., et al. Exploring Bebras Tasks Content and Performance: A Multinational Study // Informatics in Education. 2017. Vol. 16, no. 1. P. 39–59.

24. Jitesh V. Understanding the doer effect for computational subjects with MOOCs. Thesis: M. Eng., Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2018. URL:
<http://dspace.mit.edu/handle/1721.1/7582>

25. LearningApps.Org – тепер українською! URL:
<http://dystosvita.blogspot.com/2014/04/learningappsorg.html>.

26. Liu J., Wang L. Computational Thinking in Discrete Mathematics. 2010. Vol. 1. P. 413-416.

27. Matthew Henley. "Thinking about Thinking: Dance Education and 4E Cognition." Journal of Dance Education (2021). 10.1080/15290824.2021.1949185

28. Moreno-León J., et al. Analyze your Scratch projects with Dr. Scratch and assess your computational thinking skills // Scratch conference. 2015. P. 12–15.

29. Mukhasheva M., Ybyraimzhanov K.T., Mamekova A.. "The importance of feedback for the development of computational thinking in robotics educational programs for elementary schools." BULLETIN of the L N Gumilyov Eurasian National University PEDAGOGY PSYCHOLOGY SOCIOLOGY Series (NaN). 10.32523/2616-6895-2024-148-3-122-136

30. National STEM Centre. URL: <https://www.stem.org.uk/>

31. Ningtyas D. P., Setyosari P., Kuswandi D., Ulfa Saida. "Enhancing Early Childhood Problem-Solving Abilities through Game-Based Learning and Computational Thinking: The Impact of Cognitive Styles." Golden Age: Jurnal Ilmiah Tumbuh Kembang Anak Usia Dini (2024). 10.14421/jga.2024.93-04

32. Parker M. C., Guzdial M., Engleman S. Replication, validation, and use of a language independent CS1 knowledge assessment // Proceedings of the 2016 ACM conference on international computing education research. 2016. P. 93–101.
33. Rambally G. Emerging Research, Practice, and Policy on Computational Thinking. 2017. P. 99-119.
34. Román-González M., Pérez-González J. C., Jiménez-Fernández C. Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test // Computers in human behavior, 2017. Vol. 72. P. 678–691.
35. Romero M., Lepage A., Lille B. Computational thinking development through creative programming in higher education. International Journal of Educational Technology in Higher Education. Vol. 14. P. 42. 2017. URL: <https://doi.org/10.1186/s41239-017-0080-z>
36. Shute V. J., Sun C., Asbell-Clarke J. Demystifying computational thinking // Educational Research Review. 2017. Vol. 22. P. 142–158.
37. Skvortsova S., Britskan T. Training of primary school teachers for the use of information technology teaching mathematics. Mathematics, Information Technologies, and Applied Sciences (MITAV2019) / Editor Miroslav Hrubý a Pavlína Račková. 20.06 – 21.06.2019. Brno. Faculty of Military Technology of the University of Defence in Brno. Brno, 2019. 31 p.
38. Skvortsova S., Ishchenko A., Britskan T. Using of information and communication technologies in the primary school teacher's professional activity Theoretical and applied aspects of sustainable development. Series of monographs Faculty of Architecture, Civil Engineering and Applied Arts Katowice School of Technology Monograph 33. Publishing House of Katowice School of Technology. 2020. P.124–136.
39. Weintrop D., Wilensky U. Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs // Proceedings of the eleventh annual international conference on international computing education research. 2015. P. 101–110.

40. Wing J. Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences. 2008. Vol. 366. P. 1881. P. 3717–3725.
41. Wing J. Computational thinking. Communications of the ACM. 2006. Vol 49. No. 3. P. 33–35.
42. Wizer.Me – інструмент для створення інтерактивних робочих аркушів. URL: <http://i-math.com.ua/vsikt/wizer-instrument-dlya-stvorenniya-interaktivnix-robochixarkushiv/>
43. Zur-Bargury I., Pârv B., Lanzberg D. A nationwide exam as a tool for improving a new curriculum // Proceedings of the 18th ACM conference on Innovation and technology in computer science education. 2013. P. 267–272.
44. Бажан З.І., Згоран О.М. Особливості формування обчислювальних навичок табличного множення та поділу у початковій школі. Педагогіка, освіта та психологія: сучасні проблеми та напрямки розвитку: зб. наук. праць за матеріалами І міжнар. наук.-практ. конф. 2017. С. 45-52.
45. Бернетт Д. Наш дивакуватий мозок / пер. з англ. Ю. Маричева. Харків: Віват, 2020. 384 с.
46. Вчіться сьогодні, будуйте світле майбутнє. | Code.org. URL: <https://code.org/>
47. Гончаренко С. А., Ваврик А. Й., Верещак Є. П. Психологічна діагностика особливостей когнітивного розвитку молодших школярів в умовах інформаційного суспільства: колективна монографія / за ред. С. А. Гончаренко, Л. О. Кондратенко. Київ; Кіровоград: Імекс-ЛТД, 2014. 228 с.
48. Група Світового банку, 2018. Навчання дітей програмуванню: запорука розвитку людського капіталу у XXI столітті. Вашингтон, округ Колумбія: Світовий банк.
49. Європейська рамка компетентностей цифрового громадянства (DigCompEdu). URL: https://joint-research-centre.ec.europa.eu/digcompedu_en

50. Івашова О. А. Останіна Є. Є. Формування обчислювальних умінь молодших школярів як предметних навчальних процесів. Герценівські читання. Початкова освіта. 2018. Т. 9. № 2. С. 64-75.
51. Кавашима Р. Японська система розвитку інтелекту та пам'яті. Програма "60 днів" / пер. з яп. І. Лебедева, 2017. 192 с.
52. Костюк Г. С. Навчально-виховний процес і психічний розвиток особистості. Київ : Рад. шк., 1989. 608 с.
53. Мартинова А. І., Скворцова С. А. Формування у молодших школярів обчислювальних навичок. Одеса: ЮУПУ, 1997. 40 с.
54. Марченко В. В. Розвиток обчислювального мислення учнів на уроках інформатики: аналіз публікацій. Міжнародна науково-практична конференція студентів, аспірантів та молодих вчених «Молодіжна наука: інновації та глобальні виклики» (м. Полтава, 6 листопада 2024 року). С. 55-57.
55. Масич В., Лимарева Ю., Білих В. Типові графічні задачі у навчальному процесі з фізики. The 11th International scientific and practical conference "World science: problems, prospects and innovations" (July 14-16, 2021) Perfect Publishing, Toronto, Canada. 2021. Pp. 267-275.
56. Міністерство освіти Фінляндії. URL: <https://okm.fi/etusivu>
57. Моро М.І. Посилити увагу до формування обчислювальних навичок. Початкова школа. 1984. №7. С. 18-28.
58. Навчальна програма вибірково-обов'язкового предмету «Інформатика» для учнів 10-11 класів загальноосвітніх навчальних закладів (рівень стандарту). URL: <https://mon.gov.ua/osvita-2/zagalna-serednya-osvita/osvitni-programi/navchalni-programi-dlya-10-11-klasiv>
59. Оклі Б. Навчитися вчитися. Як запустити свій мозок наповну. 3-тє вид. / пер. з англ. А. Замоцний. Київ : Наш формат, 2020. 272 с.
60. Освітні програми. URL: <https://mon.gov.ua/osvita-2/zagalna-serednya-osvita/osvitni-programi>
61. Освітній стандарт CSTA для 7-8 класів. URL: <https://www.csteachers.org/Page/standards>

62. Освітня онлайн система MIKSIKE як засіб формування навичок математичного рахунку. URL: <http://miksike.net.ua>
63. Резнік К., Лазарук В. Обчислювальне мислення як один із показників високого рівня сформованості цифрової компетентності. The 8 th International scientific and practical conference “Results of modern scientific research and development” (October 17-19, 2021) Barca Academy Publishing, Madrid, Spain. 2021. p. 279-284.
64. Ривкінд Й. Я., Лисенко Т. І., Чернікова Л. А., Шакотько В. В. Модельна навчальна програма «Інформатика. 7–9 класи» для закладів загальної середньої освіти, 2023. 29 с.
65. Ривкінд Й.Я., Лисенко Т.І., Чернікова Л.А., Шакотько В.В. Модельна навчальна програма «Інформатика. 5-6 класи» для закладів загальної середньої освіти. 2021. 22 с.
66. Романишин Р.Я. Методична система формування обчислювальних навичок в учнів початкової школи. Дис. ... док. пед. н.: 13.00.02 – теорія та методика навчання (математика). Черкаси, 2020. 550 с.
67. Семко Л. Прикладні задачі у навчанні інформатики в гімназії. URL: <https://lib.iitta.gov.ua/id/eprint/730111/1/Semko.doc..pdf>
68. Сігел Д., Брайсон Т. Секрети мозку. 12 стратегій розвитку дитини / пер. з англ. І. Борщ. Київ : Наш формат, 2017. 192 с.
69. Скворцова С. О. Обчислювальні навички як складова предметно-математичної компетентності молодшого школяра. Початкова школа. 2011. № 8. С. 48–51.
70. Скворцова С. О. Обчислювальні навички як складова предметноматематичної компетентності молодшого школяра (продовження). Початкова школа. 2011. № 9. С. 43–49.
71. Тихонова Т., Кошкіна Г. Computational thinking як сучасний освітній тренд. Електронне наукове фахове видання «Відкрите освітнє есередовище сучасного університету». 2018. т. 5. С. 210-221.

72. Філер З., Фадєєва Т. Формуємо алгоритмічність мислення молодших школярів на уроках математики. Початкова школа. 2008. № 2. С. 52–56.

73. Фіцула М. М. Педагогіка : навчальний посібник для учнів вищих навчальних педагогічних закладів освіти. 2-е вид. Тернопіль : Навчальна книга-Богдан, 2003. 192 с.

74. Шульга Г. Б. Підготовка майбутнього вчителя до формування математичних уявлень і понять в учнів початкової школи : автореф. дис. ... канд. пед. наук : 13.00.04. Вінниця, 2007. 18 с.

75. Яременко А. Розвиваємо навички 4К: креативність, критичне мислення, комунікацію та командну працю. Освіторія. URL: <https://osvitoria.media/experience/rozvyvayemo-navychky-4k-kreatyvniist-krytychne-myslennya-komunikatsiyu-ta-komandnu-pratsyu/>