

Scientific journal  
**PHYSICAL AND MATHEMATICAL EDUCATION**  
Has been issued since 2013.

ISSN 2413-158X (online)  
ISSN 2413-1571 (print)

Науковий журнал  
**ФІЗИКО-МАТЕМАТИЧНА ОСВІТА**  
Видається з 2013.

<http://fmo-journal.fizmatsspu.sumy.ua/>



Медведева М.О., Жмурко О.І., Криворучко І.І., Ковтанюк М.С. Елементи підготовки майбутніх учителів інформатики до застосування технології формування Computational Thinking. Фізико-математична освіта. 2021. Випуск 1(27). С. 67-75.

Medvedieva M., Zhmurko O., Kryvoruchko I., Kovtaniuk M. Elements of training future teachers of informatics for the application of Computational Thinking technology. Physical and Mathematical Education. 2021. Issue 1(27). P. 67-75.

DOI 10.31110/2413-1571-2021-027-1-011

УДК 378.018.8:373.5.011.3-051:004]:004.81:159.955-057.874

**М.О. Медведева**

Уманський державний педагогічний університет імені Павла Тичини, Україна  
medvedeva-masha25@ukr.net  
ORCID: 0000-0001-9330-5185

**О.І. Жмурко**

Уманський державний педагогічний університет імені Павла Тичини, Україна  
o.i.zhmurko@udpu.edu.ua  
ORCID: 0000-0003-1362-0623

**І.І. Криворучко**

Уманський державний педагогічний університет імені Павла Тичини, Україна  
krivoruchkoi43@udpu.edu.ua  
ORCID: 0000-0002-9886-9315

**М.С. Ковтанюк**

Уманський державний педагогічний університет імені Павла Тичини, Україна  
covtaniuk@gmail.com  
ORCID: 0000-0001-7059-6784

## ЕЛЕМЕНТИ ПІДГОТОВКИ МАЙБУТНІХ УЧИТЕЛІВ ІНФОРМАТИКИ ДО ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ ФОРМУВАННЯ COMPUTATIONAL THINKING

### АНОТАЦІЯ

Формування Computational Thinking у здобувачів освіти є одним з основних завдань сучасного педагога. Застосування технології формування Computational Thinking у навчанні майбутніх учителів інформатики дозволить на практиці опанувати дану технологію та в майбутній професійній діяльності впроваджувати в освітній процес.

**Формулювання проблеми.** Навички Computational Thinking та комп'ютерної грамотності є незамінними для будь-якої людини незалежно від сфери її діяльності. Розуміння студентами принципів та підходів до формування Computational Thinking, а також правильно підібрані форми, методи та засоби навчання дозволять підготувати майбутнього фахівця до впровадження технології формування Computational Thinking учнів під час освітнього процесу, зокрема на уроках інформатики.

**Матеріали і методи.** Матеріалом дослідження є процес проведення серії тренінгів «Технології формування Computational Thinking» в межах вивчення дисципліни «Методика навчання інформатики» для студентів, що навчаються за освітньо-професійними програмами Середня освіта (Інформатика), Середня освіта (Математика. Інформатика) та Середня освіта (Фізика. Інформатика). У дослідженні використані такі методи як аналіз результатів досліджень, узагальнення досвіду дослідників, опитування, анкетування, педагогічне спостереження, синтез, індукція, дедукція, математико-статистичні методи.

**Результати.** В статті окреслено зміст основних принципів та підходів до формування Computational Thinking. Описано досвід впровадження технології формування Computational Thinking за допомогою використання навчальної гри MOON, що була розроблена університетом Деусто (Іспанія) під час проведення навчальних тренінгів.

**Висновки.** Узагальнюючи результати дослідження можна стверджувати, що використання технології формування Computational Thinking, зокрема навчальної гри MOON, у форматі тренінгів сприяють формуванню зазначених навичок, позитивно впливають на атмосферу навчання та готують майбутнього педагога до професійної діяльності.

**КЛЮЧОВІ СЛОВА:** Computational Thinking, методика навчання інформатики, комп'ютерне мислення, обчислювальне мислення, комп'ютаційне мислення, вчитель інформатики.

**ВСТУП**

**Постановка проблеми.** Відповідно до Концепції розвитку природничо-математичної освіти (STEM-освіти) основними завданнями є: формування навичок розв'язування складних (комплексних) практичних проблем, критичного мислення, креативних якостей та когнітивної гнучкості, організаційних та комунікаційних здібностей, вміння оцінювати проблеми та приймати рішення, готовності до свідомого вибору та оволодіння майбутньою професією, фінансової грамотності, цілісного наукового світогляду, ціннісних орієнтирів, загальнокультурної, технологічної, комунікативної і соціальної компетентностей та математичної і природничої грамотності (Концепція розвитку природничо-математичної освіти (STEM-освіти), 2020).

Навчальні методики та навчальні програми природничо-математичної освіти (STEM-освіти) спрямовані на задоволення попиту на наукоємну освіту, формування актуальних на ринку праці компетентностей, а саме: когнітивних навичок; навичок оброблення інформації, інтерпретації та аналізу даних; інженерного мислення; науково-дослідницьких навичок; алгоритмічного мислення та цифрової грамотності; креативних якостей та інноваційності; технологічних навичок; навичок комунікації (Про схвалення Концепції, 2020). Розвитку цих компетентностей має сприяти формування у здобувачів освіти навичок Computational Thinking, а сформувати ці навички зможе висококваліфікований фахівець, зокрема вчитель інформатики. Отже, підготовка такого фахівця має стати в пріоритеті освітньої діяльності педагогічного закладу вищої освіти.

На сьогодні застосування інформаційних технологій, зокрема комп'ютерної техніки дозволяє розв'язувати велике коло завдань: від побутових до надскладних обчислень. При розв'язуванні задач за допомогою комп'ютерів істотно змінюється концепція самої задачі, адже в її розв'язуванні бере участь не лише студент, а й персональний комп'ютер. Новий стиль мислення передбачає існування алгоритмічного виконавця на всіх етапах від постановки задачі до оформлення її розв'язання. Отож, навички Computational Thinking та комп'ютерної грамотності є незамінними для будь-якої людини незалежно від сфери її діяльності. Процес формування цих навичок повинен починатися на рівні дошкільної освіти і продовжуватися неперервно протягом усього життя людини.

**Аналіз актуальних досліджень.** Дослідженням поняття Computational Thinking та його розвитку присвячені праці зарубіжних та вітчизняних учених, серед яких слід відзначити: Жанетт Вінг, Пола Керзона та Пітера Макоуена, які описували Computational Thinking, як спосіб розв'язання проблем людьми, а не намаганням трактувати мислення людини як комп'ютера; Мяхей Суло та Анну Квятковську, які відносять Computational Thinking до основних навичок сучасної дитини, таких як вміння читати, писати та рахувати; Марину Хрипунову та Анну Балджи, які розглядали практику формування Computational Thinking при професійному навчанні, на прикладі вивчення вищої математики; Оксану Пасічник, що досліджувала поняття Computational Thinking, як складову шкільного курсу інформатики; Євгенія Хеннера, який обґрунтував теоретичну значимість Computational Thinking; Майкла Г. Воскоглу, який досліджував Computational Thinking паралельно із статистичним та критичним мисленням та інші.

У сучасній педагогічній науці «Computational Thinking» різними науковцями перекладається по-різному і при цьому іноді втрачається сутність та сенс поняття. Україномовні та російськомовні автори вживають терміни «комп'ютерне мислення» (Sysło&Kwiatkowska, 2013; Хрипунова&Балджи, 2018), «обчислювальне мислення» (Пасічник, 2014; Хеннер, 2016; Воскоглу, 2020), «комп'ютаційне мислення» (Тухчонова&Koshkina, 2018) тощо. Ми вважаємо, що будь-який переклад не повністю передає сутність даного поняття, тому пропонуємо вживати термін в англійській формі.

**Мета статті.** Метою статті є описати елементи підготовки майбутніх учителів інформатики до застосування технології формування Computational Thinking.

**МЕТОДИ ДОСЛІДЖЕННЯ**

Матеріалом дослідження є процес проведення серії тренінгів «Технології формування Computational Thinking» в межах вивчення дисципліни «Методика навчання інформатики» для студентів, що навчаються за освітньо-професійними програмами Середня освіта (Інформатика), Середня освіта (Математика. Інформатика) та Середня освіта (Фізика. Інформатика). У дослідженні використані такі методи як аналіз результатів досліджень, узагальнення досвіду дослідників, опитування, анкетування, педагогічне спостереження, синтез, індукція, дедукція, математико-статистичні методи.

**РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ**

Існують різні підходи до тлумачення поняття Computational Thinking.

Дослідження можливостей використання науки програмування в якості інструменту для навчання започаткував математик Сеймур Пейперт у другій половині 60-х років ХХ століття. Він трактував визначення Computational Thinking як підхід до розв'язування задач, що використовує інформатичні методи (Пейперт, 1989).

Жанетт Вінг стверджувала, що Computational Thinking є найважливішою властивістю, яка необхідна для існування у сучасному світі. Computational Thinking означає мислити способами, які допомогли б програмувати комп'ютер. Вона визначає це, як «процеси мислення, що беруть участь у формулюванні проблем та їх вирішення, щоб рішення були представлені у формі, яка може ефективно здійснюватися агентом обробки інформації». «Computational Thinking передбачає вирішення проблем, проектування систем та розуміння поведінки людини, спираючись на концепції, фундаментальні для інформатики. Computational Thinking включає цілий ряд розумових засобів, які відображають широту галузі інформатики» (Wing, 2006).

Пол Керзон та Пітер Макоуен під Computational Thinking розуміють не лише пошук розв'язків у вигляді алгоритмів, а й набір прийомів, які забезпечують ефективний засіб покращення життєвих умов та осмислення світу (Керзон&Макоуен, 2018).

Мяхей Суло та Анна Квятковська наголошують, що незважаючи на походження, Computational Thinking не зводиться до вивчення комп'ютерів та комп'ютерних програм, хоча вони відіграють важливу роль у розв'язуванні задач. Computational Thinking – це засіб обмірковування задач в комп'ютерних термінах. Вони відмічають важливість

Computational Thinking, яке корисне майже у всіх шкільних предметах, так як воно дає розуміння того, що може бути або не може бути обчислено (Sysło&Kwiatkowska, 2013).

Марина Хрипунова та Анна Балджи під Computational Thinking розуміють когнітивний процес, що включає етапи: декомпозиція – аналіз даних – узагальнення результатів – алгоритм розв’язування – кінцевий результат (Хрипунова&Балджи, 2018).

Отже, під Computational Thinking ми будемо розуміти процеси мислення, що беруть участь у формулюванні проблеми, її аналізі та знаходженні шляхів її вирішення таким чином, щоб рішення можна було представити, зокрема і за допомогою комп’ютера, а спосіб вирішення був застосовний до всіх завдань подібного класу.

Формування Computational Thinking включає шість принципів: логічне міркування та аналіз, алгоритми створені з інструкцій та правил, декомпозиція, закономірність, абстрагування, оцінювання (рис. 1) та п’ять підходів: дослідження, створення, налагодження, наполегливість, співпраця (рис. 2) (Barefoot).

Зупинимося докладніше на принципах формування Computational Thinking, які можна застосовувати під час освітнього процесу, зокрема на уроках інформатики.

**Логічне міркування та аналіз** допомагають прийти до більш повного розуміння предмета; в основі лежить можливість пояснити, чому щось є таким, яким воно є. Наприклад, в комп’ютері кожна виконана дія зводиться до логічних операцій на основі електричних сигналів; інженери-програмісти постійно використовують логічні міркування (розробляючи новий ефективний код, вони спираються на моделі функціонування комп’ютерного обладнання, операційних систем та мов програмування; під час тестування нового програмного забезпечення, пошуку помилок та їх виправлення вони також будуть покладатися на логічні міркування).



Рис. 1. Принципи формування Computational Thinking

**Алгоритми створені з інструкцій та правил.** Алгоритм – це послідовність інструкцій або набір правил для того, щоб щось зробити; алгоритми написані для людини, а не для розуміння комп’ютером (цим алгоритми відрізняються від програм); інформатики прагнуть до алгоритмів, які вирішують проблеми найбільш ефективно – отримуючи максимально точні результати в найкоротший час, з найменшою кількістю ресурсів (пам’яті чи часу).

**Декомпозиція** в обчисленнях – це процес розбиття завдання на менші, більш керовані частини; допомагає нам керувати великими проектами і робить процес вирішення складної проблеми менш важким і набагато легшим для сприйняття; завдання може вирішувати кілька людей, які працюють разом у команді, і кожен учасник вносить свої власні ідеї та навички в окремі аспекти проекту; для обчислювальної техніки декомпозиція не є унікальною, це цілком стандартне рішення в техніці, дизайні та управлінні проектами.

**Закономірність:** виявляючи закономірності, ми можемо створювати правила та вирішувати більш загальні проблеми.

Діти помічають закономірності реагування вчителів на їх поведінку. Схеми погоди входять у наші прогнози. У математиці учні можуть виміряти площу прямокутника, намальованого на міліметровому папері, підрахувавши кількість одиничних квадратів всередині нього, але це може бути важко або довго для прямокутників, які є великими. Більш елегантне – помножити довжину прямокутника на ширину – і це добре працює для всіх прямокутників. Коли учні запам’ятають цю формулу, то за її допомогою обчислити площу буде набагато швидше, ніж підрахувати квадрати. В обчисленнях метод пошуку загального підходу до класу задач називається узагальненням.

Інформатики прагнуть швидко і ефективно вирішувати проблеми, і вони шукають методи, застосовні в інших місцях. Якщо вони бачать шаблон в алгоритмі, вони намагатимуться створити єдиний модуль повторюваного коду, який іноді називають функцією або процедурою – багато мов програмування мають спільні бібліотеки загальних функцій. Розпізнавання закономірностей у вхідних даних відіграє важливу роль у машинному навчанні. Це важливе застосування інформатики, яке відіграє важливу роль у системах, серед іншого, алгоритмічної торгівлі на фондовому ринку та розпізнавання облич та номерних знаків транспортних засобів.

**Абстрагування.** Абстракція полягає у спрощенні речей – визначенні важливого, не надто турбуючись про деталі. Шкільний розклад – це абстракція того, що відбувається за типовий тиждень. Він показує ключову інформацію про класи, вчителів, аудиторії та час, але ігнорує подальші рівні деталей, як-от цілі та діяльність.

Абстракція дозволяє нам думати про речі з різним ступенем деталізації. Це потужний інструмент в галузі інформатики, де він використовується для управління складністю більшості того, що розроблено та створено.

Ми можемо сприймати абстракції як кулі або коробки всередині коробок, що дозволяють нам ігнорувати те, що відбувається всередині кожної з них. Програмне забезпечення містить шари коду, кожен з яких приховує складність наступного. Ми можемо розглядати елементи обладнання як «чорні ящики», нехтуючи їхньою внутрішньою роботою, якщо не вирішимо заглянути глибше.

Учні також можуть розвивати абстрактне мислення під час гри в комп'ютерні ігри, розуміючи, що ці інтерактивні симуляції базуються на реальному житті, але є простішими.

**Оцінювання.** Оцінка полягає у винесенні суджень, де це можливо, об'єктивно та систематично. Щодня ми приймаємо судження про те, що робити і що думаємо, виходячи з ряду факторів. Розглядаючи новий цифровий пристрій для класної кімнати, існує ряд критеріїв: операційна система, портативність, обсяг пам'яті, розмір екрану, простота використання, гарантія тощо.

Оцінка полягає в оцінюванні якості, результативності та ефективності продукції, рішень, процесів та систем. Ми з'ясовуємо, чи відповідають вони певній меті. Одним із підходів може бути розгляд конкретних критеріїв, наприклад, цілі чи специфікації проєкту або потреб користувачів. В інформатиці оцінка є систематичною та суворою.

Водночас для реалізації цих принципів застосовують відповідні підходи до формування Computational Thinking.

**Дослідження** використовується для того, щоб дізнатися щось нове і як це працює. Це тісно пов'язано з логічними міркуваннями. Учні накопичують причинно-наслідкові зв'язки: «Якщо я зроблю це, то відбудеться це». Це велика частина самостійного навчання без керівництва викладача. Для маленьких дітей це життєво важлива фаза експериментів, заснована на іграх, повна питань та сюрпризів. Ідеї, які здаються неправильними, можна випробувати лише, щоб подивитися, що станеться.



Рис. 2. Підходи до формування Computational Thinking

Для дорослих – це цілеспрямована діяльність, часто шляхом випробувань та вдосконалення. Це допомагає нам бачити, що в нашому використанні технології йдеться про розвиток власного розуміння, а не про отримання «правильної» відповіді; ми можемо робити щось по-різному. Використовуючи технологію, з якою ми поспілкувались, ми, швидше за все, будемо відкриті до нових та інноваційних рішень.

Свобода досліджувати в безризиковому середовищі породжує впевненість і позитивне ставлення. Відкриті запитання та завдання заохочують творчі здібності, різноманітні ідеї, здатність дивитись на речі з багатьох різних сторін. Програмісти часто спочатку досліджують нові технології, щоб ознайомитись з тим, як вони працюють, і отримати уявлення про те, як їх можна використовувати. Програмне та апаратне забезпечення з кожним роком оновлюється все швидше. І користувачі, і розробники повинні бути відкритими для частих і швидких дій. Впевненість у майстерності допомагає розглядати зміни як можливість, а не як небезпеку, дозволяючи нам постійно оновлювати свої навички та використовувати нові технології. Це також формує наполегливість.

Тому потрібно виділяти час для того, щоб учні досліджували. Вони можуть: випробувати програмне забезпечення, перш ніж застосувати його для розв'язання певних завдань; переглянути мову програмування, якою вони не користувались деякий час; погратися новим цифровим пристроєм; поекспериментувати з візуальними мовами програмування, такими як Scratch, Kodu та StarLogo, або з фізичними пристроями, такими як Lego Wedo, Makey Makey, Arduino та Raspberry Pi.

Незалежно від мови програмування чи пристрою, потрібно, щоб учні використовували відкриту технологію, аби зрозуміти, як вона працює та для чого використовується. Викладач може поділитися своїм досвідом (наприклад, як користуватись новим мобільним телефоном) і попросити учнів поділитися власним досвідом. У такий спосіб ми розвиваємо уяву та креативність в учнів, заохочуючи їх випробувати нові ідеї – навіть ті, які в кінцевому результаті не є функціональними або застосовними в їх безпосередньому контексті.

**Створення** – це задумка та створення речей. Програмне забезпечення та цифрові носії дають можливість для творчості, і, опановуючи програмні засоби та цифрові пристрої, ми розвиваємо впевненість, компетентність та незалежність, які ми можемо використовувати граючись, експериментуючи та цілеспрямовано для вираження своїх ідей та думок.

Програмування саме по собі є творчим процесом. У нас є ідеї щодо того, що ми хотіли б зробити чи вирішити, проаналізувати проблему, розробити, написати та налагодити необхідний код та оцінити те, що ми створили.

Інформатика не є лише академічною дисципліною: це практична, прикладна інженерна дисципліна, що створює рішення для реальних проблем та забезпечує можливості у різних сферах життя. Процес створення речей також є потужним засобом навчання.

**Налагодження.** Якщо помилки – це помилки в алгоритмах та коді, то налагодження – це процес їх пошуку та виправлення, який часто може зайняти набагато більше часу, ніж спочатку написання коду. Можуть бути помилки в логіці або синтаксисі. Ми можемо розглядати помилки в логіці як частини історії, де сюжет не має сенсу, а помилки в кодуванні та синтаксисі як погану орфографію, пунктуацію та граматику. Під час налагодження може бути корисним відображення вмісту будь-яких змінних у нашій програмі, щоб допомогти нам зрозуміти, що відбувається під час її роботи.

Ми пропонуємо просту послідовність з чотирьох кроків для налагодження, підкріплену логічними міркуваннями:

1. Передбачте, що має статися.
2. З'ясуйте, що відбувається.
3. Попрацюйте там, де щось пішло не так.
4. Виправте це.

Складний код, який програмісти пишуть, часто працює не за призначенням. Існує дуже різноманітна заява про кількість зусиль, часу та грошей, витрачених на налагодження комерційних проектів. Зазвичай налагодження та тестування становлять 50–75% від вартості проекту.

**Наполегливість.** Наполегливість, рішучість, витривалість, завзятість – синоніми до словосполучення «ніколи не здаватися». Комп'ютерне програмування важке. Це є частиною його привабливості – написання елегантного та ефективного коду є інтелектуальним завданням, що вимагає не лише розуміння ідей кодованих алгоритмів та мови програмування на якій ви працюєте, але й готовності наполегливо ставитися до чогось, що часто буває досить складно, а часом і дуже розчаровує.

Щоб розвивати досвід у чомусь складному, нам потрібно наполегливо практикуватись, тренуватись та репетирувати. Це стосується багатьох сфер людських зусиль: мистецтва, музики, танців, спорту, шахів, досліджень, комп'ютерних ігор, програмування тощо.

В обчислювальній техніці та в інших системах, завдання та проблеми можуть бути складними, їх контекст незнайомим. Можливо, нам доведеться спробувати багато варіантів або використовувати нові для нас технології; нам може навіть знадобитися змінити свій звичний, швидкий та інтуїтивний спосіб мислення на щось більш повільне, більш обдумане та логічне. Інформатики потребують якостей терпіння, витримки та терпимості до розгубленості. Граючи у складну комп'ютерну гру, існує щільний цикл зворотного зв'язку причинно-наслідкових зв'язків, який деякі порівнюють із процесом кодування та налагодження: і геймери, і програмісти повідомляють про стан «поток», в якому вони повністю занурені в свою відповідну діяльність, зосереджені на тому, щоб закінчити гру або вирішити проблему.

**Співпраця** означає співпрацю з іншими, і завдяки їй часто досягаються найкращі результати. Вчителі радяться, планують, діляться досвідом, щоб розвивати найкращі практики. Співпраця спонукає нас наполегливо виконувати завдання, які в іншому випадку можуть здатися занадто заплутаними або складними. Інформатики та інженери програмного забезпечення часто використовують чи вдосконалюють роботу та кодування інших, і це значно простіше в програмному забезпеченні з відкритим кодом.

Під час програмування багато хто розглядає «парне програмування» як особливо ефективний спосіб написання коду, коли два програмісти мають спільний екран та клавіатуру. Як правило, один програміст виступає в ролі «драйвера», маючи справу з деталями кодування, тоді як інший виконує роль «навігатора», розглядаючи загальну картину та пропонуючи вказівки. Обидва регулярно міняються місцями, тому обидва накопичують досвід у кожній ролі.

Задачі та системи розкладаються на окремі завдання. У розробці програмного забезпечення беруть участь різні команди з різних спеціальностей, які працюють разом. На прикладі комп'ютерної гри спеціалізація включає програмування, дизайн гри, мистецтво та анімацію. Команди залежать і відповідають один перед одним, тому ефективне спілкування є життєво важливим. Інтернет-інструменти та вебсайти, такі як Google Docs та GitHub, дозволяють спільно використовувати та одночасно редагувати файли співавторів, які можуть знаходитися у різних куточках світу. У багатьох сучасних розробках програмного забезпечення клієнти вважаються частиною розширеної команди розробників, що допомагає удосконалювати програмні рішення.

Використання цих принципів та підходів спрямоване на формування та розвиток Computational Thinking учнів під час освітнього процесу, а інструментами у цьому слугує створення та/або використання як підручних засобів (папір, ручка і т. п.), так і спеціальних настільних ігор, комп'ютерних ігор, симуляцій тощо.



Рис. 3. Тренінги «Технології формування Computational Thinking» для студентів

В рамках вивчення дисципліни «Методика навчання інформатики» студентам, що навчаються за освітньо-професійними програмами Середня освіта (Інформатика), Середня освіта (Математика. Інформатика) та Середня освіта (Фізика. Інформатика) було проведено серію тренінгів «Технології формування Computational Thinking» та запропоновано розробити фрагмент заняття з інформатики для учнів 10-11 класів при вивченні вибіркового модуля «Математичні основи інформатики», використовуючи навчальну гру MOON, що була розроблена університетом Деусто Іспанія (<http://compus.deusto.es/>). Впровадження технології формування Computational Thinking відбувалося відповідно до прослуханих тренінгів в рамках міжнародного проєкту «Модернізація педагогічної вищої освіти з використання інноваційних інструментів викладання» (MoPED) - №586098-EPP-1-2017-1-UA-EPPKA2-SVNE-JP, що діє в рамках програми ЄС Еразмус + KA2 – Розвиток потенціалу вищої освіти, що були проведені з 11 по 16 березня 2019 року на базі ДВНЗ «Переяслав-Хмельницький державний педагогічний університет імені Григорія Сковороди» від партнерів з Університету Деусто (м. Більбово, Іспанія).

MOON – це навчальна гра, де гравці імітують простий комп’ютер, вчать рахувати у двійковому режимі, виконують логічні операції і дізнаються, як працює комп’ютер. Можна роздрукувати картки, які знаходяться у вільному доступі (рис. 4) та грати в настільну гру, а можна скористатися Інтернет-версією для гри на комп’ютері або на інших гаджетах (рис. 5).

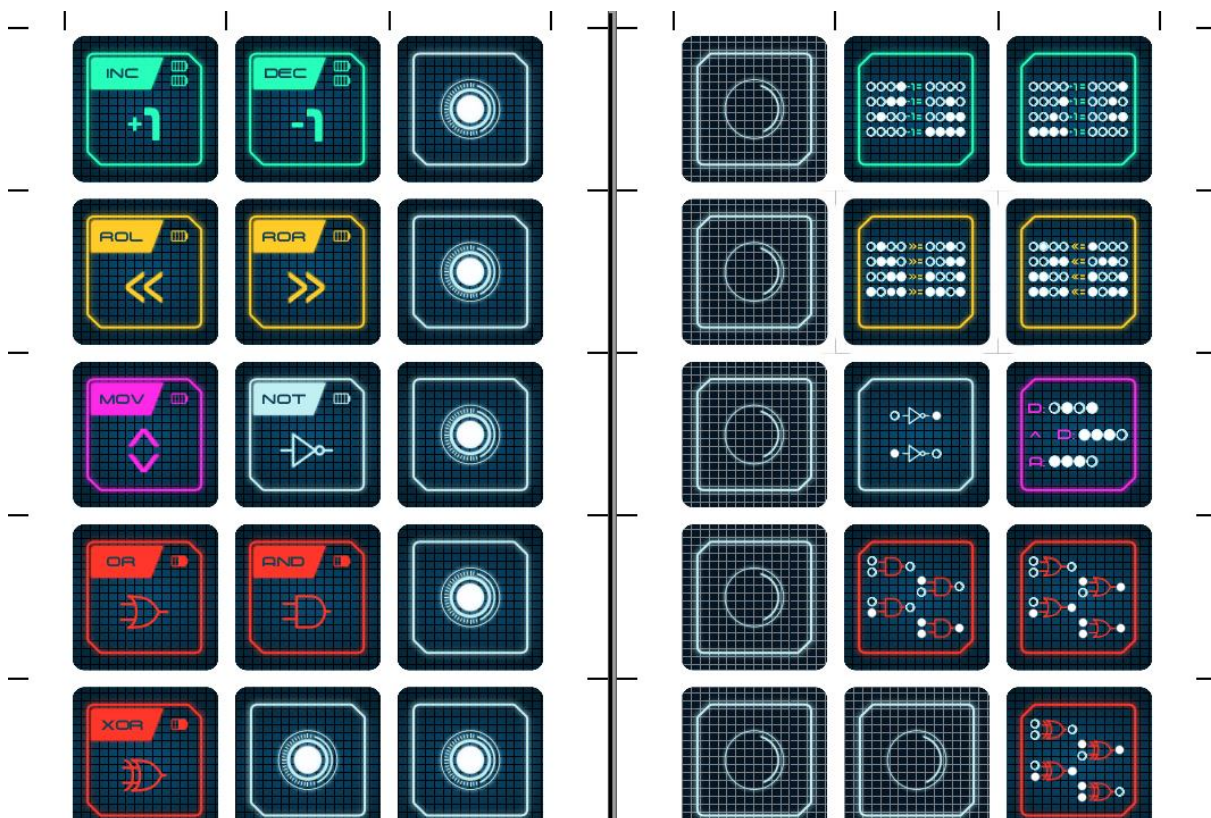


Рис. 4. Приклад карток для гри MOON



Рис. 5. Інтернет-версія гри MOON

Після тренінгів було проведено опитування, в якому брали участь 20 студентів. Студентам було запропоновано оцінити за шкалою від 1 до 10, де 1 – зовсім неважливо, а 10 – дуже важливо, такі ключові моменти використання технології формування Computational Thinking:

- необхідність впровадження та використання таких інноваційних педагогічних технологій як Computational Thinking в освітньому процесі в університеті;

- доцільність їх використання в освітньому процесі у закладах загальної середньої освіти;
- рівень зацікавленості в залежності від використаного засобу навчання;
- рівень задоволеності від застосування такої технології під час вивчення дисципліни «Методика навчання інформатики».

Результати опитування представлені на діаграмах (рис. 6). Загалом відсоток сприйняття даної технології, розуміння її важливості у навчанні майбутнього покоління серед майбутніх учителів інформатики, досить високий. Також дана технологія дозволила студентам удосконалити раніше набуті навички виконання логічних операцій в двійковій системі числення. Позитивний настрій відчувався як у використанні карткової версії гри, так і у проходженні гри на комп'ютері. Проте, у спілкуванні деякі студенти висловили думку, що в наш час, коли майже вся діяльність вчителя тим чи іншим чином пов'язана з комп'ютером, карткова гра здається більш привабливою.

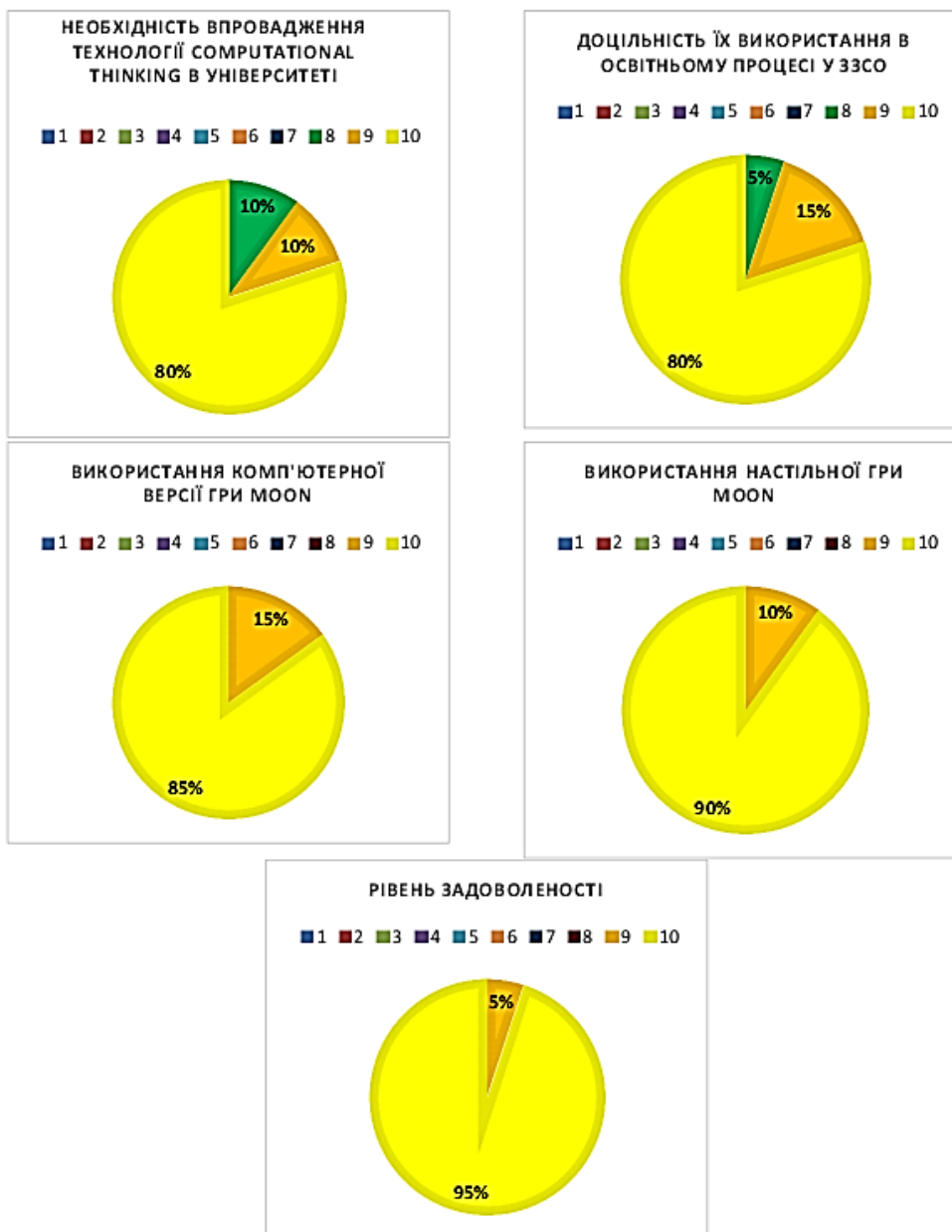


Рис. 6. Результати опитування студентів, що брали участь у тренінгах

**ВИСНОВКИ ТА РЕЗУЛЬТАТИ ПОДАЛЬШОГО ДОСЛІДЖЕННЯ**

Отже, формування та розвиток Computational Thinking учнів під час освітнього процесу можливе зокрема використанню інноваційних технологій. Так, використання навчальної гри MOON, у форматі тренінгів, сприяє формуванню навичок Computational Thinking, позитивно впливає на атмосферу навчання та готує майбутнього педагога до професійної діяльності.

Узагальнюючи вище сказане, ми вважаємо, що лише підготовлений фахівець, який знає суть проблеми із середини, здатний запалити, мотивувати та навчити сучасного учня, який здебільшого інформаційно перенасичений настільки, що немає фізичної та емоційної можливості сприймати корисний навчальний матеріал. Вчитель повинен прилаштовуватися до реалій сьогодення і намагатися допомогти майбутньому поколінню бути готовому до вирішення всіх проблем і завдань, які можуть постати перед ним як зараз, так і в майбутньому.

## Список використаних джерел

1. *Computational Thinking Concepts and Approaches*. Barefoot. URL: <https://www.barefootcomputing.org/concept-approaches/computational-thinking-concepts-and-approaches> (дата звернення: 10.10.2020).
2. Moon. URL: <http://compus.deusto.es/> (дата звернення: 11.10.2020).
3. Sysło M.M., Kwiatkowska A.B. Informatics for All High School Students, A Computational Thinking Approach, in: Diethelm I., Mittermeir R.T. (eds.). *ISSEP 2013*, LNCS 7780, Springer-Verlag, Berlin, Heidelberg, 2013. P. 43–56.
4. Sysło M.M., Kwiatkowska A.B. Преподавание математики с опорой на компьютерное мышление. *Компьютерные инструменты в школе*, 2013. С. 3–14.
5. Тихонова Т. В., Кошкина Г. Л. Computational thinking як сучасний освітній тренд. Електронне наукове фахове видання «Відкрите освітнє е-середовище сучасного університету». 2018. № 5. С. 210–221.
6. Wing J.M. Research notebook: computational thinking – what and why? URL: <http://link.cs.cmu.edu/article.php?a=600>.
7. Wing J.M. Computational thinking. *Comm. ACM* 49(3), 2006. P. 33–35.
8. Керзон П., Макоуэн П. Вычислительное мышление: новый способ решать сложные задачи : пер. с англ. Москва, 2018. 266 с.
9. Про схвалення Концепції розвитку природничо-математичної освіти (STEM-освіти) : Розпорядж. від 05.08.2020 р. № 960-р. URL: <https://zakon.rada.gov.ua/laws/show/960-2020-p#n8> (дата звернення: 15.12.2020).
10. Майкл Г. Воскоглу. Способи мислення при вирішенні проблем. *Фізико-математична освіта*. 2020. Випуск 3(25). Частина 1. С. 11-18.
11. Пасічник О. В. Розвиток алгоритмічного мислення на уроках інформатики. *Комп'ютер у школі та сім'ї*. 2014. № 7. С. 13-18.
12. Пейперт С. Переворот в сознании: Дети, компьютеры и плодотворные идеи: Пер. с англ. / Под ред. А. В. Беляевой, В. В. Леонаса. М.: Педагогика, 1989. 224 с.
13. Хеннер Е. К. Вычислительное мышление. *Образование и наука*. 2016. № 2 (131). С. 18–33.
14. Хрипунова М.Б., Балджи А.С. Формирование компьютерного мышления при профессиональном обучении в вузе. *Современная математика и концепции инновационного математического образования*. 2018. №1. С. 381–388.

## References

1. Computational Thinking Concepts and Approaches. (n.d.). Barefoot. Retrieved from: <https://www.barefootcomputing.org/concept-approaches/computational-thinking-concepts-and-approaches>.
2. Moon. Retrieved from: <http://compus.deusto.es/>
3. Sysło, M.M. & Kwiatkowska, A.B. (2013). Informatics for All High School Students, A Computational Thinking Approach, in: Diethelm I., Mittermeir R.T. (eds.), *ISSEP 2013*, LNCS 7780, Springer-Verlag, Berlin, Heidelberg, 43–56.
4. Sysło, M.M. & Kwiatkowska, A.B. (2013). Prepodavanje matematiki s oporoj na komp'juternoje myshlenie [Teaching mathematics based on computer thinking]. *Komp'juternye instrumenty v shkole - Computer tools at school*, 3–14 [in Russian].
5. Tykhonova, T. V. & Koshkina, H. L. (2018). Computational thinking yak suchasnyi osvitnii trend [Computational thinking as a modern educational trend]. *Elektronne naukove fakhove vydannia «Vidkryte osvitnie e-seredovyshe suchasnoho universytetu»* - Electronic scientific professional publication "Open educational e-environment of a modern university", 5, 210-221. <https://doi.org/10.28925/2414-0325.2018.5.210221/> [in Ukrainian].
6. Wing, J.M. Research notebook: computational thinking – what and why? Retrieved from: <http://link.cs.cmu.edu/article.php?a=600>.
7. Wing, J.M. (2006). Computational thinking, *Comm. ACM* 49(3), 33–35.
8. Kerzon, P., Makoujen, P. (2018). *Vychislitel'noe myshlenie: novyj sposob reshat' slozhnye zadachi [Computational thinking: a new way to solve complex problems]* (per. s angl.). Moskva [in Russian].
9. Pro skhvalennia Kontseptsii rozvytku pryrodnycho-matematychnoi osvity (STEM-osvity): Rozporiadz. 05.08.2020 r. № 960-r. [On approval of the Concept of development of natural and mathematical education (STEM-education): Order of the Cabinet of Ministers of Ukraine of August 5, 2020 № 960-r.]. (n.d.). [zakon.rada.gov.ua](https://zakon.rada.gov.ua/laws/show/960-2020-p#n8). Retrieved from <https://zakon.rada.gov.ua/laws/show/960-2020-p#n8> [in Ukrainian].
10. Voskoglou Michael Gr. (2020). Modes Of Thinking In Problem Solving. *Physical and Mathematical Education*, 3(25), 1, 11-18.
11. Pasichnyk, O. V. (2014). Rozvytok alhorymichnoho myslennia na urokakh informatyky [Development of algorithmic thinking in computer science lessons]. *Kompiuter u shkoli ta simi - Computer at school and family*, 7, 13-18 [in Ukrainian].
12. Pejper, S. (1989). Perevorot v soznanii: Deti, komp'jutery i plodotvornye idei [A Revolution in Consciousness: Children, Computers, and Fruitful Ideas] (per. s angl.) Moskva [in Russian].
13. Henner, E. K. (2016). Vychislitel'noe myshlenie [Computational thinking]. *Obrazovanie i nauka - Education and Science*, 131, 18–33 [in Russian].
14. Hripunova, M.B. & Baldzhy, A.S. (2018). Formirovanie komp'juternogo myshlenija pri professional'nom obuchenii v vuze [Formation of computer thinking in vocational training at a university]. *Sovremennaja matematika i koncepcii innovacionnogo matematicheskogo obrazovanija - Contemporary mathematics and the concepts of innovative mathematics education*, 1, 381–388 [in Russian].



## ELEMENTS OF TRAINING FUTURE TEACHERS OF INFORMATICS FOR THE APPLICATION OF COMPUTATIONAL THINKING TECHNOLOGY

*Mariia Medvedieva, Oleksandr Zhmurko, Inna Kryvoruchko, Maksym Kovtaniuk*  
Pavlo Tychyna Uman State Pedagogical University, Ukraine

**Abstract.** *The formation of Computational Thinking in students is one of the main tasks of a modern teacher. The application of the technology of forming Computational Thinking in the training of future computer science teachers will allow to master this technology in practice and to implement it in the educational process in the future professional activity.*

**Formulation of the problem.** *Computational Thinking and computer literacy skills are indispensable for anyone, regardless of their field of activity. Students' understanding of the principles and approaches to the formation of Computational Thinking, as well as properly selected forms, methods and teaching aids will prepare future professionals to implement the technology of forming Computational Thinking students during the educational process, including computer science lessons.*

**Materials and methods.** *The material of the research is the process of conducting a series of trainings "Technologies of formation of Computational Thinking" within the study of the discipline "Methods of teaching computer science" for students enrolled in educational and professional programs: Secondary education (Informatics), Secondary education (Mathematics. Informatics) and Secondary education (Physics, Informatics). The study used such methods as analysis of research results, generalization of researchers' experience, surveys, questionnaires, pedagogical observation, synthesis, induction, deduction, mathematical and statistical methods.*

**Results.** *The article outlines the content of the basic principles and approaches to the formation of Computational Thinking. The experience of implementing Computational Thinking technology using the MOON educational game, which was developed by the University of Deusto (Spain) during the training, is described.*

**Conclusions.** *Summarizing the results of the study, it can be argued that the use of Computational Thinking technology, in particular the educational game MOON, in the format of training contributes to the formation of these skills, positively affect the learning atmosphere and prepare future teachers for professional activities.*

**Key words:** *computational thinking, methods of teaching computer science, computer thinking, computational thinking, computer science teacher.*

