

УДК 378.016:004]:159.955.4

Микола Головін

кандидат фізико-математичних наук,
доцент кафедри комп'ютерних наук та кібербезпеки,
Волинський національний університет імені Лесі Українки, м. Луцьк, Україна
ORCID ID 0000-0003-4516-4677
Golovin_Mykola@vnu.edu.ua

Ніна Головіна

кандидат фізико-математичних наук,
доцент кафедри експериментальної фізики, інформаційних та освітніх технологій,
Волинський національний університет імені Лесі Українки, м. Луцьк, Україна
ORCID ID 0000-0002-1152-1536
ninaholovina@gmail.com

Дмитро Гузачов

здобувач,
Волинський національний університет імені Лесі Українки, м. Луцьк, Україна
ORCID ID 0009-0007-3281-602X
dmytro.huzachov@gmail.com

НАВЧАЛЬНЕ ПРОГРАМУВАННЯ В КОНТЕКСТІ РЕФЛЕКСІЇ РОЗУМОВИХ ДІЙ ТА ЇХ ФОРМАЛІЗАЦІЯ ПРОГРАМНИМ КОДОМ

Анотація. У роботі розглянуто проблему покращення навчання програмуванню через оптимізацію рефлексії, що супроводжує написання програмного коду. Рефлексія як контроль логіки мислення є важливим механізмом критичного мислення в навчальній і в професійній діяльності. Метою цього дослідження є формування оптимізованих, концептуальних, лаконічних схем рефлексії, що дозволять успішно справлятися із задачами навчального практичного програмування. Для реалізації мети проведений аналіз навчальних практичних дій з програмування в контексті їх рефлексії на прикладі створення комп'ютерної моделі простого фізичного явища. Вибір програмування як діяльності для аналізу рефлексії критичного мислення зумовлений тим, що ця діяльність добре формалізована. Сутність рефлексії критичного мислення розкривається тут в рафінованій прозорій формі. Моделювання надає процесу мислення свій контраст, адже в ментальній діяльності, крім алгоритмічних сенсів, з'являються ще й додаткові фізичні. Програмування моделі фізичного процесу тут є своєрідним полігоном дослідження. Рефлексія розглядається через призму циклічних ментальних процесів у термінах формальної логіки з опорою на еволюцію структури декларативних знань. Кінцевою інстанцією цього аналізу стала формалізація дій рефлексії в лаконічних схемах як самих процесів, так і структури знань. Важливу частину роботи займає зв'язок ментальних дій з відповідною матеріалізованою діяльністю. Цей зв'язок також вдалось лаконічно зобразити на відповідній схемі. Представлена в роботі чітка, коротка, виразна формалізація дій критичного мислення, пов'язана з матеріалістичними діями, в лаконічних схемах могла б застосовуватись здобувачами освіти для оптимізації навчання та формування протоколів абстрактно-логічного і причинно-наслідкового мислення. Розроблені схеми рефлексії можуть розглядатись як методичний засіб навчання, який дозволяє уникнути когнітивних пасток і оптимізувати навчальну діяльність як у сфері практичного програмування, так і в природничих галузях знань, де передбачаються практичні дії по створенню або відлагодженню штучних об'єктів високої складності.

Ключові слова: критичне мислення; рефлексія; методика навчання програмування; ієрархія; еволюція структури знань.

1. ВСТУП

Проблема формування критичного мислення є важливою для всіх майбутніх

спеціалістів та фахівців природничої галузі зокрема. Критичне мислення характеризується причинно-наслідковістю, абстрактно-логічністю, матеріалістичністю. Такі модальності мислення неможливі без відповідної досконалої рефлексії розумових дій.

Навчальні дії, що відбуваються впродовж виконання лабораторних робіт з природничих дисциплін, зокрема з програмування та фізики, де передбачається створення штучного об'єкта, мають багато спільних між собою рис. У всіх випадках відбувається поетапне виконання робіт та пов'язані з цим багатократні випробовування об'єкта вивчення на різних етапах його готовності. У сфері програмування ці навчальні дії добре формалізуються, мають рафінований вигляд, доступний для спостереження.

Поетапна технологія побудови програми продукує ланцюги відповідних допоміжних підзадач, розв'язок яких, на кінцевому етапі призводить до успіху. Кожна допоміжна підзадача на лабораторній роботі з програмування уособлюється у відповідному логічно завершеному програмному коді. В аналогічних лабораторних роботах з фізики кожний етап виконання лабораторної роботи пов'язаний з частково побудованою фізичною установкою, яка теж, в сенсі підзадачі, має завершений характер. Найбільшої схожості та інтеграції згадані вище навчальні процеси набувають в лабораторних практикумах з комп'ютерного моделювання фізичних процесів з екранною їх візуалізацією. Тут частина логічно завершених підзадач під час їх програмної еволюції набувають фізичного змісту.

Аналіз останніх досліджень і публікацій. У контексті цього дослідження, особливо цікавим є аналіз наукових робіт, у яких проведено аналіз впливу практичного програмування на вдосконалення рефлексії розумових дій та покращення когнітивних навичок здобувачів освіти. Цей літературний огляд спрямований на дослідження когнітивних процесів у розумовій діяльності, що супроводжує комп'ютерне програмування. Зокрема в [1] автори показують, що використання комп'ютерного програмування в навчанні має позитивний вплив на когнітивні навички, такі як творче мислення, математичні навички та метакогніція. Ці результати свідчать про корисність вивчення програмування для розвитку загальних когнітивних здібностей. Використання програмування в навчанні сприяє розвитку навичок творчого та просторового мислення, що є корисним для розвитку загальних когнітивних здатностей у людей різних спеціальностей. Ці висновки були підтверджені в [2]. Тут розглядаються психологічні переваги вивчення програмування. Автори стверджують, що вивчення коду містить такі когнітивні процеси, як математичне моделювання та творче розв'язання проблем, що сприяє розвитку навичок, які корисні не лише в програмуванні, але й в обробці інформації, критичному мисленні та математиці. Своєю чергою, робота [3] присвячена огляду літератури з метою висвітлення сучасного стану досліджень теорії когнітивного навантаження в контексті використання комп'ютерного програмування в навчанні. Після ознайомлення з роботою стають більш зрозумілими підходи до вимірювання когнітивного навантаження, педагогічні стратегії та концепти теорії когнітивного навантаження.

У фундаментальному огляді літератури [4] розглядається розвиток здатності здобувачів освіти до програмування. Автори проводять аналіз 86 досліджень, які відбувались з 1980 по 2020 рік. За результатами цього огляду, здатність до програмування визначається як стійка особистісна та психологічна характеристика здобувача освіти, що містить метакогніцію, когніцію, операції та комунікацію разом з 17 піднавичками. На цій основі автори розробляють модель навчання програмуванню, яка враховує рівні освіти, методи навчання, засоби навчання та підходи до оцінки. Автори, також підтверджують, що управління пізнавальними процесами та рефлексія розумових дій у студентів покращуються при долученні програмування як дисципліни

в процес навчання.

Щобільше, дослідження показують, що рефлексія в навчанні є важливим компонентом, який сприяє розвитку когнітивних процесів учнів [5], [6], [7]. Насамперед рефлексія допомагає їм глибше розуміти вивчений матеріал, аналізувати його та знаходити нові зв'язки. По-друге, уміння рефлексувати допомагає учням контролювати свої дії, виявляти помилки та вносити корективи. По-третє, рефлексія сприяє розвитку навичок планування, моніторингу та оцінки власної діяльності.

Відкритий університет у співпраці з SRI International, підготував узагальнюючий звіт про десять нововведень, які є концептуальними та актуальними в сучасній освіті, однак ще не мали значного впливу на освіту. Тут розглянуто стратегії навчання з детальними поясненнями та прикладами. TeachThought Staff 5 лютого 2021 р. запропонував уривок зі звіту про «10 інноваційних стратегій навчання для сучасної педагогіки» [8], де серед інших виділяють «Обчислювальне мислення» та «Аналітика емоцій», як потужні підходи до вирішення проблем. У першому випадку «передбачається розбиття великих проблем на менші (декомпозиція), визначення того, як вони пов'язані з проблемами, які були вирішені в минулому (розпізнавання шаблонів), відкидання неважливих деталей (абстракція), визначення та розробку кроків, які будуть необхідні для досягнення рішення (алгоритми) і вдосконалення цих кроків (налагодження)». У другому випадку зазначається, що «студенти мають мислення (наприклад, бачать свій мозок фіксованим або пластичним), стратегії (наприклад, розмірковування про навчання, пошук допомоги та планування того, як вчитися), і якості залучення (наприклад, наполегливість), які глибоко впливають на те, як вони навчаються». Багатообіцяючим підходом є «поєднання комп'ютерних систем когнітивного навчання з досвідом вчителів» [8].

Авторами цієї роботи в статті [9] була розглянута оригінальна психологічна модель ментальних пізнавальних процесів, що детермінують навчальне програмування. Структура комп'ютерної програми, що формується в процесі навчальних дій, представлена там у вигляді ієрархічної схеми, яка розростається невеликими, логічно завершеними фрагментами. Модельний механізм формування цієї структури в процесі навчання окреслено лаконічною функціональною схемою, у якій задіяно три циклічні процеси. Їх ментальний механізм пояснено термінами формальної логіки. Особливості критичного мислення в контексті програмування розглянуто в роботі [10]. У цьому сенсі цікавим є зміст відповідної рефлексії, що є оптимізуючим механізмом такої складної розумової діяльності, як програмування.

Актуальним завданням, на погляд авторів, є розгляд розумових і матеріалізованих дій, що відбуваються в процесі реалізації задачі з програмування моделі простого фізичного явища. У цьому випадку, завдання на програмування набуває лаконічного змісту з хорошими обмежувачами рамками при реалізації. Виконання завдання базується на витонченій внутрішній структурі, що задає фізичний процес. У задачі з'являється, крім алгоритмічного сенсу, ще й фізичний. Спростуються і стають рафінованими зміст думок і відповідних дій рефлексії, адже навіть проста задача з фізики має математичну основу і структурне наповнення.

Для розгляду ментальних дій з програмування, які до того ж детермінують відповідні матеріалізовані дії, цікавою є концепція когнітивної психології про те, що інтелектуальна діяльність детермінується структурною організацією пізнавальної сфери. Жодна з цих пізнавальних структур не є абсолютно новою. Кожна є трансформацією або модифікацією попередньої [11]. Тому сукупність тематичних задач і підзадач формують деревоподібну ієрархічну структуру. У цих умовах основний принцип навчання є принцип диференціації або деталізації ментальних пізнавальних структур стосовно програмних конструкцій.

Подібні навчальні дії мають циклічний характер і відбуваються стосовно - - кожної логічно завершеної підзадачі. У цих циклах в межах свідомості людини відбуваються дії, які можна описати в термінах формальної логіки [12]. Усвідомлення та контроль логіки дій в свідомості називають рефлексією. Саме в цьому сенсі в цій статті використовується термін рефлексія. Фокусування на змісті дій у власній свідомості та осмислення і контроль логіки цих дій в навчальній і професійній діяльності, дозволяє уникнути когнітивних пасток і оптимізувати цю складну діяльність.

Метою дослідження є аналіз навчальних практичних дій з програмування шляхом їх розгляду через призму пізнавальних (когнітивних) процесів, еволюції структури знань у термінах формальної логіки. Кінцева інстанція цього аналізу – лаконічна формалізація дій рефлексії, що може застосовуватись суб'єктами навчання як методичний прийом для оптимізації навчання у сфері навчального практичного програмування.

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

1. Формалізувати відповідні циклічні навчальні пізнавальні дії в термінах формальної логіки та співвіднести ці дії з динамікою еволюції структури знань упродовж створення простої моделюючої програми.
2. Формалізувати моменти, коли ментальна діяльність з програмування детермінує відповідні супутні матеріалізовані дії з написання, апробування та відлагодження програми.
3. Сформулювати та лаконічно формалізувати у вигляді схем ментальні дії рефлексії, що можуть застосовуватись здобувачами освіти для оптимізації процесу у сфері навчального практичного програмування.

2. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Автори роботи відстоюють думку, що покращення методичних підходів до практичної навчальної діяльності з програмування і з фізики неможливе без детального уявлення про супутні психологічні пізнавальні процеси.

Об'єктом дослідження є особливості ментальних спроможностей людини пов'язані з рефлексією, що можуть бути цікавими для покращення комп'ютерноорієнтованого навчання практичному програмуванню. Пам'ять людини може спрощено бути представлена у вигляді трикомпонентної моделі [12]. Ця модель пам'яті складається з таких її видів: короткочасної, сенсорної і довготривалої. У короткочасній пам'яті уособлюється свідомість і поле уваги. Вона працює в режимі вербальної петлі. Поле уваги (свідомість) обмежено магічним числом Міллера (7 ± 2) [13]. Тому будь-яка людина завжди змушена розбивати поставлену задачу на дрібніші допоміжні логічно завершені підзадачі. Кожна з таких підзадач не може виходити за межі 7 ± 2 взаємодіючих компоненти, що мають своє вербальне відображення. Це важливий **методичний** момент навчальної роботи з будь-яким складним об'єктом.

Предметом дослідження є еволюція розв'язку простої навчальної аплікаційної фізичної задачі з моделювання фізичного процесу. У цій статті такою задачею є задача про рух тіла в полі тяжіння. У межах цієї моделі має бути реалізований рівноприскорений та рівносповільнений рух і відбиття від перешкоди. Ця модель проста і має коротку еволюцію програмного коду. Ця еволюція вкладається в технологію написання програмного коду **методами низхідної деталізації** алгоритму та **модульного програмування**. Ці методи є основою практики структурного програмування [14].

У процесі розв'язку задачі про рух тіла в полі тяжіння утворюється одна проміжна логічно завершена задача. Це підзадача про відбивання тіла від перешкоди і зміни руху з рівноприскореного на рівносповільнений. Останнє дає можливість детально зосередитись на процесах переведення уваги з основної задачі на допоміжну та з допоміжної на основну, відслідковуванні циклічних ментальних процесів стосовно реалізації цієї підзадачі в межах основного програмного коду. Останнє положення є цікавим у контексті методики проведення лабораторних робіт з програмування, з фізики, з моделювання фізичних процесів.

Початкові умови. Нехай на початковому етапі студенти вміють рухати по екрану тіло в будь-якому напрямку з однаковою швидкістю, зокрема і паралельно осі Y .

Прямолінійний рівномірний рух. На рис.1 поданий текст базової задачі, що реалізована з використанням бібліотеки Visual. На цьому ж рисунку зображена структура знань, яка буде обговорюватись нижче.

Модельне представлення мислення в полі уваги. Ментальна схема програмного об'єкта, на якій кружечками означені знання, які відповідають початковому етапу розв'язку задачі (рівномірний прямолінійний рух), охоплені трикутником, представлена на рис.1. Кожен кружечок на схемі відповідає окремому оператору. Виняток є в рядку вводу початкових параметрів. Тут одним кружечком позначена група операторів, що представляються в мисленні цілісно словами «початкові значення».

Конструктом у цій роботі будемо називати компоненти мислення, що одночасно знаходяться в полі уваги (одночасно усвідомлюються) і логікою мислення можуть бути зв'язані в єдиний логічно завершений алгоритмічний механізм. Такі сукупності компонентів окреслюються трикутниками на схемах.

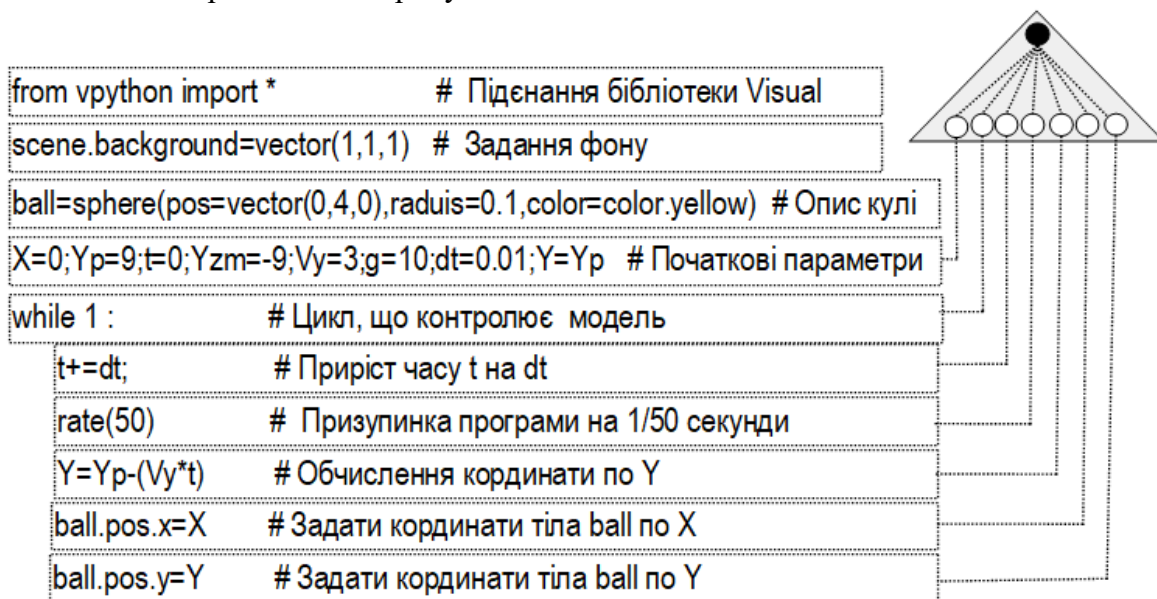


Рис.1 Початкова фаза навчання. Текст програми фізичної моделі та графічне зображення пізнавальної ментальної схеми програмного об'єкта (трикутник)

Процес мислення базується на циклічному симультанному аналізі і синтезі [12]. Тобто, об'єкт мислення у вербальній петлі свідомості пульсує від цілісності (назви конструкта) до диференційованості (компонентів конструкта) і знов до цілісності, і так багато разів з невеликими модифікаціями, поки задача буде розв'язана. Цілісність на рисунках позначена кружечком чорного кольору, що знаходиться у вершині трикутника. Алгоритмічний механізм, що реалізує цю цілісність – кружечки в основі трикутника.

Прямолінійний рівноприскорений рух. Зрозуміло, що перетворення рівномірного руху на рівноприскорений можна здійснити заміною формули $Y=Y_p-(V_y*t)$ формулою $Y=Y_p-(V_y*t)-(g*(t)**2/2)$. Цим реалізується завдання моделювання рівноприскореного руху в полі тяжіння вздовж осі Y вниз (рис.2).

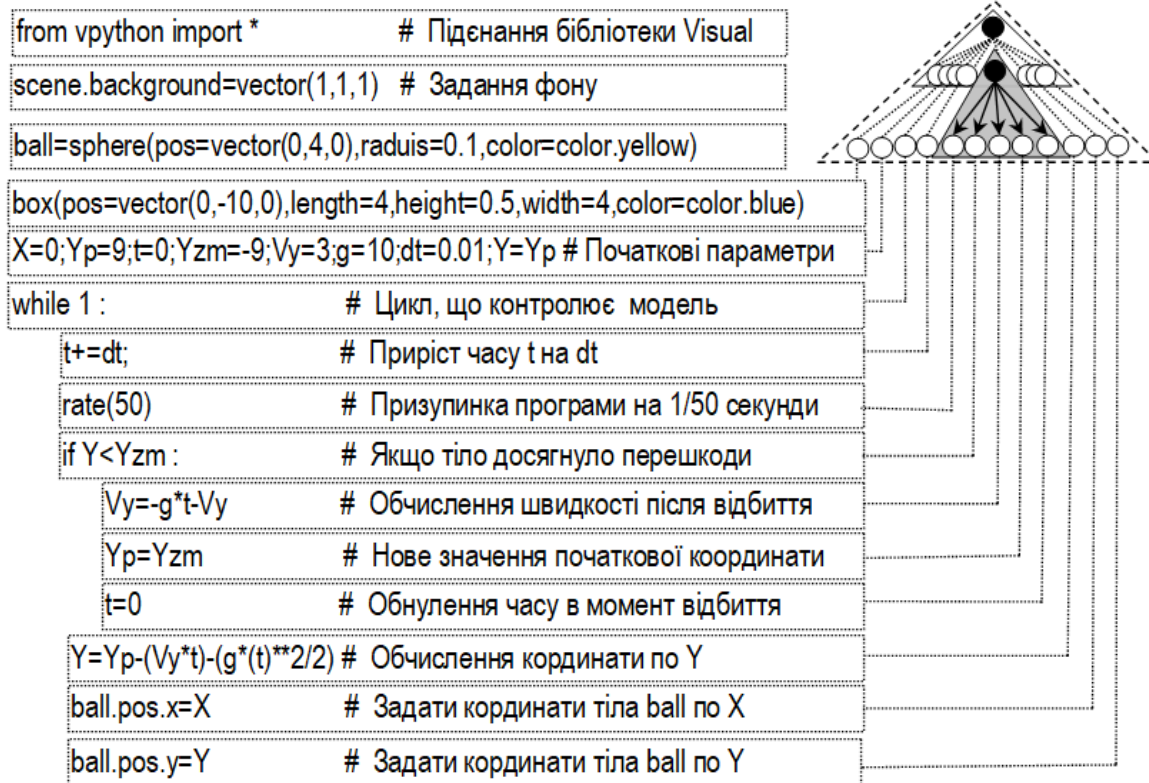


Рис. 2. Кінцева фаза навчання. Текст програми фізичної моделі та графічне зображення пізнавальної ментальної схеми програмного об'єкта

Генерування нового конструкту (фаза I-II на рис.3) пов'язано в цій задачі з реалізацією відскоку. Породження відповідного нового коду в програмі перетворює рис.1 в рис.2. Новий код несе в собі ускладнену логіку фізичного процесу. Психологічний аспект програмної реалізації коду, відповідного новому конструкту, представлено на рис.3 у вигляді еволюції формально-логічних фаз цього процесу. А саме, від початкової фази I до кінцевої фази V. (Термін фаза, в цій статті розуміється, як частина процесу, що має ознаки завершеності).

Процес креативного мислення (фаза II) стосовно реалізації відбиття починається з генерування нового осередку мислення в конструкті. У цій аплікаційній задачі це слово «відскок». Далі, відбувається *абстрагування* від всіх інших компонентів конструкту і включаються *аналітико-дедуктивні* дії (рис.3а), які формують уявлення про компоненти нового конструкту. *Аналіз* розчленовує новий компонент на кілька понятійних утворень в контексті орієнтовної основи дій (знань з програмування та фізики). Далі, ці нові, щойно згенеровані компоненти мислення зв'язуються *дедуктивно*, в контексті повернення цілісності. Цей етап можна назвати алгоритмічним, адже тут відбувається генерування ідеології програмних дій і не завжди досягається рівень деталізації, коли можна написати програмний код. Відмітимо, що в цій простій аплікаційній програмі, такий код написати можна. Однак, не факт, що щойно придуманий код запрацює.

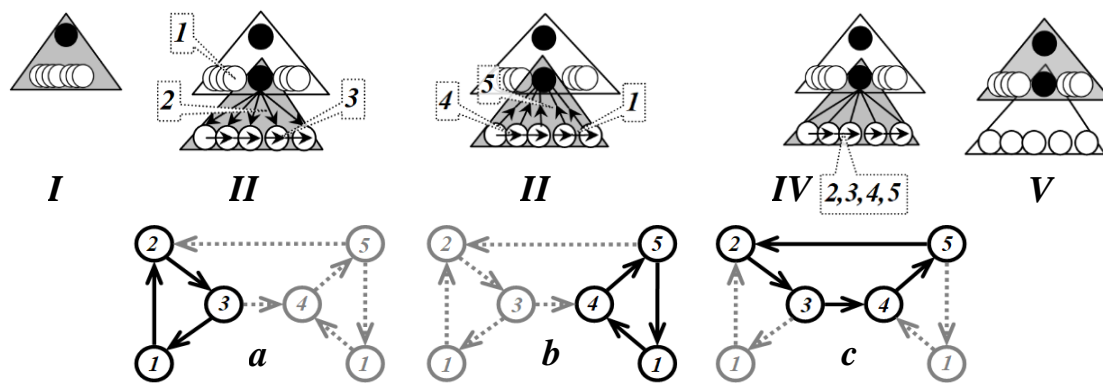


Рис.3 Схема логічних дій стосовно функціональних вузлів об'єкта в полі уваги в режимі: **a** - конкретизації; **b** - узагальнення; **c** - підбору операторів. Чорні стрілки показують домінуючі дії, а сірі допоміжні. Дії формальної логіки: 1.Абстрагування; 2.Аналіз; 3.Дедукція; 4.Індукція; 5.Синтез. Римськими цифрами позначені етапи еволюції структури знань

Написання нового фрагменту програми (фаза III на рис.3). Тут відбувається програмна реалізація уявлень про відбивання і рух тіла в полі тяжіння. Ментальні дії, що супроводжують ці матеріалізовані дії з написання програмного коду мають *індуктивно-синтетичний* характер (рис.3б). Логічна дія *індукція* зв'яже нові компоненти нижнього конструктору в цілісність. Це зв'язування завершується *синтезом*, створенням цілого з компонентів.

Індуктивні дії мають характер передбачення. Звісно, передбачення може і не справджуватись. Тому, фіналізація синтетичних дій, стосовно конструктору, що поточно знаходиться в полі уваги, реалізується матеріалізованими діями при випробовуваннях відповідного програмного фрагменту.

Відлагодження програми (фаза IV). Якщо ж програма не працює правильно, тобто мета не досягнута, то починається фаза відлагодження програми. Психологічний аспект процесу відлагодження представлений на рис.3с. Це процес впродовж якого абстрагування не відбувається, однак компоненти конструктору та їх зв'язки, що поточно знаходяться в полі уваги, можуть модифікуватись та трансформуватись в процесі кожного нового прояву аналізу, дедукції, індукції та синтезу. Процес відлагодження триває до досягнення дієздатності відповідного програмного коду.

Так, наприклад, у програмному фрагменті $if Y < Yzm: Vy = -g \cdot t - Vy; Yp = Yzm; t = 0$ можна поплутати знак менше на більше або мінус на плюс. У цих випадках, матеріалізовані дії з перевірки програми на дієздатність приведуть до неправильної роботи моделі. Зокрема, у першому випадку тіло буде пролітати повз перешкоди. Тоді, цикл у схемі рис.3с запусниться на повторну ітерацію без зміни поля уваги. Аналіз перетворить щойно цілісне уявлення про фрагмент програми, що має забезпечити відбиття, на відповідні компоненти. Буде проведена дедуктивна перевірка вигляду компонентів та зв'язків між ними. Можливо, ця перевірка буде успішною, знак більше буде замінений на знак менше. Далі, за схемою, відбудеться ментальна індукційна перевірка зв'язків на предмет відновлення цілісності фрагмента. Із компонентів синтезується оновлена цілісність, що має забезпечити відбиття. Необхідно відзначити, що помилка може бути знайдена і на індукційному етапі ментальних дій. Якщо подальша перевірка на дієздатність виявиться успішною, то відбудеться вихід зі схеми рис.3с. Концентрація поля уваги на програмному фрагменті буде припинена, а увага

буде переведена на інші об'єкти. Аналітико-синтетичні дії у циклі, зображеному на рис.3с можуть бути швидкими багатократними і без відповідних матеріалізованих перевірок у режимі симультанного аналізу і синтезу.

Фіналізація програмних дій (фаза V на рис.3). У разі успішного випробування, уявлення про конструкт згортаються до назви – функції (в нашому випадку це «відскок»). Далі, увага переводиться від компонентів нижнього конструкту на компоненти верхнього. У межах верхнього програмного конструкту нижній конструкт виглядає, як поодинокий компонент синтетичного типу. У результаті індуктивних дій стосовно компонентів верхнього конструкта весь програмний код набуває цілісності. Якщо випробування коду показало, що мета вже досягнута, то уявлення про верхній конструкт, а потім і про всю програму цілому, теж згортаються, тепер вже до назви — функції всієї програми. У нашому випадку це «рух тіла в полі тяжіння». Далі поле уваги переводиться на іншу діяльність, тобто відбувається *абстрагування* від компонентів і зв'язків верхнього конструкту.

Матеріалізовані дії, що супроводжують утворення структури знань, розглянемо через призму поетапного формування розумових дій та понять. Тут можна виділити наступні якісні моменти в освоєнні навчального матеріалу: мотиваційний етап; складання власної схеми орієнтуючої основи дій; виконання матеріалізованих дій; перехід від матеріалізованих до зовнішньо-мовних дій; перехід до внутрішньо-розумової діяльності; здатність до масштабування матеріалом.

На рис.4 зображено пізнавальний процес стосовно матеріалу, який складається з двох конструктів, в контексті практичної діяльності з навчального програмування. Матеріалізовані дії тут строго детермінуються ментальними. На схемі представлений зв'язок ментальних та матеріалізованих дій.



Рис.4 Розумові та матеріалізовані дії в контексті практичного програмування та виконання лабораторних з фізики

Цікавою особливістю цієї схеми є наглядна демонстрація особливості ментальних і матеріалізованих дій людини в процесі навчання. Ці дії мають циклічний характер.

Програмна реалізація верхнього конструкту відповідає поверхневому освоєнню матеріалу, а реалізація двох конструктів відповідає освоєнню матеріалу в повному обсязі.

Освоєння матеріалу в повному об'ємі передбачає дії, які можна прослідкувати слідуючи за стрілками лінією чорного кольору за схемою на рис.4. Видно, що ця лінія чорного кольору спочатку має вигляд спіралі «всередину», а потім спіралі «назовні». Спіраль «всередину» відповідає діям за схемою, зображеною на рис.3а, а спіраль «назовні» - діям за схемою рис.3б.

Цікавою особливістю цієї схеми на рис.4 є те, що в ній врахована ситуація, коли сукупність ментальних і матеріалізованих дій не завершуються успішним випробуванням програмного коду. Сірі стрілки можуть замкнути хід ментального процесу. І тоді, спіралі перетворюються в замкнені траєкторії. Зовнішня замкнена лінія тоді, вказує на послідовність ментальних і матеріалізованих дій стосовно верхнього конструкту, а замкнена внутрішня – на послідовність ментальних і матеріалізованих дій стосовно нижнього конструкту. В обох випадках це дії за схемою рис.3с з відлагодження програми без абстрагування. Вихід з цих циклічних дій можливий після успішного випробування відповідного конструкту програмного коду.

Іншою важливою особливістю схеми на рис.4, яка якісно і точно описує характер ментальних і матеріалізованих дій, є те, що дії стосовно нижнього конструкту можливі тільки після успішних дій стосовно верхнього.

Аналіз навчання в контексті мовної активності. Вербальні пояснення суб'єктів навчального програмування на початковому етапі роботи стосовно дії програми можливі тільки в межах верхнього конструкту. Специфіка коду відбивання ще не проявилась і не може бути деталізована, а тому позначається ключовим словом «відскок». Це можна вважати описом програми без масштабування. Масштабування матеріалом в напрямку його деталізації відбувається тоді, коли пояснення дії програми передбачає перезавантаження поля уваги і деталізацію матеріалу в напрямку пояснення модельної реалізації «відскоку». Останнє ілюструється переходом від пояснення роботи програмного коду, відповідного верхньому конструкту, до пояснення роботи програмного коду нижнього конструкту. Зрозуміло, що цілісність розповіді в її фіналі вимагає повернення від пояснення коду нижнього конструкту до пояснення коду верхнього, адже сенс нижнього конструкту має значення тільки в контексті верхнього. Останнє, теж є масштабуванням матеріалу в зворотньому порядку і теж є проявом абстрактно-логічного мислення. У результаті сукупної дії цих механізмів відбувається поступова диференціація структури знань.

Аналіз навчання в сенсі пізнавальних процесів людини. Дослідження роботи студентів через призму пізнавальних (когнітивних) процесів, еволюції структури знань в термінах формальної логіки показали, що становлення фахівця у галузі природничих наук проходить через кілька рівнів гатунку в сенсі структурно-організаційного протоколу розумової роботи.

На першому рівні, суб'єкт навчання може зв'язати невелику кількість компонентів (фізичних чи інформаційних), як правило, в межах одного конструкту, і розв'язати відповідне завдання.

На другому рівні, формується вміння переносити увагу між двома конструктами, що охоплюють масив знань який перевищує об'єм короткочасної пам'яті (максимум в два рази).

На третьому рівні передбачається вміння робити далекі транзитні переходи уваги структурою знань і дає можливість створювати і відлагоджувати пристрої або програми високої складності і великого розміру.

Між першим, другим і третім протоколом мислення існують якісні бар'єри, які не можливо ефективно подолати не усвідомлюючи їх наявність. Перехід між кожним з цих бар'єрів утруднений і вимагає відповідних пакетів задач. Протокол мислення, що характеризується далекими транзитними переходами уваги структурою знань відповідає розвинутому абстрактно-логічному, причинно-наслідковому, критичному мисленню.

Необхідно зауважити, що згадані протоколи мислення часто не носять вузько-професійної специфіки, а мають сенс загально-освітнього розвитку. Практика навчального програмування, зокрема, в напрямку моделювання фізичних процесів, потужно формує, як абстрактно-логічне, так і причинно-наслідкове критичне мислення.

Якісні зміни специфіки мислення, вимагають значних зусиль, які стосуються не стільки накопичення знань, скільки переформатування протоколу мислення. Це підтверджується спостереженнями за здобувачами освіти, що вивчають другу мову програмування. Освоєння другої мови на етапі структурного програмування, як правило, відбувається без подолання згаданих вище якісних бар'єрів. Професійний протокол мислення вже досягнутий при роботі з першою мовою програмування.

Автори роботи на рис.3 і рис.4, на прикладі створення програми модельного процесу простого фізичного явища, намагались лаконічно та наочно формалізувати основні аспекти ментальних дій, що відбуваються в процесі створення штучного об'єкта програмними засобами інформаційних технологій. Суть рефлексії, що повинна спрацьовувати в процесі подібних навчальних дій, полягає в свідомому відслідковуванні фаз ментальних дій, зображених на рис.3а, 3b, 3с. Ці фази мають відслідковуватись стосовно кожного конструктору структурної схеми об'єкта навчальних дій, що реалізується на лабораторному занятті. Важливим моментом є і координація матеріалізованих навчальних дій, що детермінуються відповідними ментальними діями (рис.4).

Основною методичною рекомендацією, що базується на матеріалах цієї роботи є наступне. У процесі виконання лабораторних робіт з програмування (фізики) необхідно розглядати діяльність, як послідовність добре сформульованих підзадач. Стосовно кожної необхідно виконувати повний цикл дій, зображених на рис.3а, 3b, 3с. та рис.4. Не дотримання протоколу згаданих вище ментальних і матеріалізованих дій, що контролюються рефлексією, веде до хаотизації процесу розв'язку локальної навчальної задачі, зокрема, написання конкретної моделюючої програми.

Комплекс умінь, необхідних для створення великих програмних проєктів або складно організованих фізичних моделей, формується тривалою практичною навчальною діяльністю. Уміння, що формуються усвідомлено, під керівництвом викладача, не проходять тривалої фази спорадичних пошуків оптимальних протоколів роботи. Тому, на думку авторів, при підготовці висококваліфікованих фахівців у цій сфері необхідно значний час приділяти рефлексії мислення – усвідомленому цілеспрямованому формуванню протоколів абстрактно-логічного, причинно-наслідкового, критичного мислення.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У роботі розглянуто проблему покращення навчання практичному програмуванню через оптимізацію рефлексії критичного мислення, що супроводжує написання програмного коду. Розглянуті в роботі обставини ментальних спроможностей людини пов'язані з рефлексією, можуть бути цікаві для покращення комп'ютерноорієнтованого навчання практичному програмуванню. Зокрема це стосується порційності завдань, їх структурної спрямованості в контексті дерева задач

конкретної теми програмування, фази роботи суб'єкта навчання над текстом програми. Розглянуті в роботі фази з успіхом можуть бути трансформовані в три типи завдань де акценти зроблені: на конкретизації коду програмного блоку; на узагальненні уявлень про логічно завершений програмний блок; на відлагодженні програмного коду.

Впродовж створення моделюючої програми простого фізичного явища *проведено* аналіз навчальних практичних дій з програмування шляхом їх розгляду через призму пізнавальних процесів, еволюції структури знань в термінах формальної логіки. Саме такий синтетичний підхід є оригінальним підходом авторів до оптимізації ментальних дій у процесі практичного навчального програмування.

Вибір програмування, як діяльності для аналізу рефлексії критичного мислення, зумовлений тим, що ця діяльність добре формалізована у всіх її проявах. Сутність рефлексії критичного мислення розкривається тут у рафінованій прозорій формі. Моделювання, в цьому контексті, теж надає процесу мислення свій контраст, адже в ментальній діяльності крім алгоритмічних сенсів з'являються ще й додаткові фізичні. Програмування моделі фізичного процесу, в цій роботі, є своєрідним полігоном для дослідження рефлексії критичного мислення. Сформульовано та лаконічно формалізовано у вигляді схем ментальні дії рефлексії, що мають застосовуватись здобувачами освіти для оптимізації навчання практичному програмуванню. Загалом, ці алгоритми ментальної діяльності сприяють досягненню мети покращення навчання шляхом оптимізації системного мислення, відповідної рефлексії.

Автори пропонують в процесі навчання фокусуватись не тільки на змісті навчальних дій, а і на контролі логіки цих дій, тобто фокусуватись на рефлексії дій. Представлена у вигляді лаконічних схем, рефлексія може розглядатись як методичний засіб навчання, який дозволяє уникнути когнітивних пасток, прискорити і оптимізувати навчальну діяльність, як в сфері практичного програмування, так і в природничих галузях знань, де передбачаються практичні дії зі створення або відлагодження штучних об'єктів високої складності. Структурне представлення навчальних програм у вигляді сукупності логічно завершених конструктів, кожен з яких може бути досягнутий полем уваги (свідомістю), може дозволити перейти від якісного опису пізнавальних процесів та відповідних рефлексій до їх кількісних оцінок в майбутніх дослідженнях авторів. Відмітимо також, що структурування складної проблеми дозволяє вирішити її більш ефективно.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. R. Scherer, F. Siddiq та B. Sánchez Viveros, "The cognitive benefits of learning computer programming: A meta-analysis of transfer effects." *J. Educational Psychol.*, т. 111, № 5, с. 764–792, лип. 2019. doi: <https://doi.org/10.1037/edu0000314>.
2. R. Scherer, F. Siddiq та B. Sánchez Viveros, "Some Evidence on the Cognitive Benefits of Learning to Code", *Frontiers Psychol.*, т. 12, верес. 2021. doi: <https://doi.org/10.3389/fpsyg.2021.559424>.
3. J. H. Berssantette та A. Francisco, "Cognitive load theory in the context of teaching and learning computer programming: A systematic literature review", *IEEE Trans. Educ.*, т. 65, № 3, с. 440–449, 2022. doi: <https://doi.org/10.1109/TE.2021.3127215>.
4. L. Sun, Z. Guo та D. Zhou, "Developing K-12 students' programming ability: A systematic literature review", *Educ. Inf. Technol.*, т. 27, с. 7059–7097, 2022. doi: <https://doi.org/10.1007/s10639-022-10891-2>.
5. Л. Філіппова, "Рефлексивний підхід у професійній підготовці магістрів фармації", *Укр. пед. журн.*, № 2, с. 123–130, 2020. doi: <https://doi.org/10.32405/2411-1317-2020-2-123-130>.
6. А. Лозенко, "Формування рефлексивних умінь у майбутніх учителів початкової школи в процесі дидактичної підготовки", Дис. на здобуття наук. ступен. канд. пед. наук. Нац. Пед. Ун-т ім. М.П.Драгоман., Київ, 2010.
7. Л. Маляр, В. Староста та М. Кухта, *Основи психолого-педагогічної підготовки у вищій школі*. Ужгород: ДВНЗ «УжНУ», 2019.

8. TeachThought Staff. “10 інноваційних стратегій навчання для сучасної педагогіки”. [Електронний ресурс]. Доступно: <https://www.teachthought.com/the-future-of-learning/innovative-strategies/>. Дата звернення: 1 трав. 2024.
9. М. Головін, Н. Головіна та Н. Головіна, “Модельний розгляд пізнавальних процесів, супутніх навчальному програмуванню”, *Психол. перспективи*, № 31, с. 57–70, 2018.
10. М. Головін та Н. Головіна, “Механізми критичного мислення та навчання фізики і програмування”, *Фізика та освітні технології*, № 1, с. 15–26, 2022. [Електронний ресурс]. Доступно: <http://journals.vnu.volyn.ua/index.php/physics/article/view/734/675>. Дата звернення: 15 квіт. 2024.
11. J. Piaget, “Intellectual evolution from adolescence to adulthood”, *Human Develop.*, т. 15, № 1, с. 1–12, 1972. doi: <https://doi.org/doi:10.1159/000271225>.
12. R. Solso, *Cognitive Psychology*. Boston • London • Toronto • Sydney • Tokyo • Singapore: Allyn Bacon, 2001.
13. G. Miller, “The magical number seven, plus or minus two”, *Psycholog. Rev.*, т. 63, № 2, с. 81–97, 1956. [Електронний ресурс]. Доступно: <https://img3.reoveme.com/m/049764c53e25268b.pdf>. Дата звернення: 15 квіт. 2024.
14. O. Dahl, E. Dijkstra та C. Hoare, *Structured Programming*. London and New York: Academic, 1972. [Електронний ресурс]. Доступно: <https://pdfcoffee.com/structured-programming-dahl-dijkstra-hoare-academic-press-1972-pdf-free.html>. Дата звернення: 15 квіт. 2024.

Матеріал надійшов до редакції 14.05.2024 р.

EDUCATIONAL PROGRAMMING IN THE CONTEXT OF REFLECTION OF COGNITIVE ACTIONS AND THEIR FORMALIZATION WITH PROGRAM CODE

Mykola Holovin

Candidate of Physical and Mathematical Sciences,
Associate Professor of the Department of Computer Science and Cybersecurity,
Lesya Ukrainka Volyn National University, Lutsk, Ukraine
ORCID ID 0000-0003-4516-4677
Golovin_Mykola@vnu.edu.ua

Nina Holovina

Candidate of Physical and Mathematical Sciences, Associate Professor of the Department of Experimental Physics, Information and Educational Technologies,
Lesya Ukrainka Volyn National University, Lutsk, Ukraine
ORCID ID 0000-0002-1152-1536
ninaholovina@gmail.com

Dmytro Huzachov

Education applicant,
Lesya Ukrainka Volyn National University, Lutsk, Ukraine
ORCID ID 0009-0007-3281-602X
dmytro.huzachov@gmail.com

Abstract. The paper examines the problem of improving programming learning through optimization of reflection accompanying the writing of software code. Reflection, as control of the logic of thinking, is an important mechanism of critical thinking in educational and professional activities. The purpose of this research is the formation of optimized, conceptual, concise schemes of reflection, which will allow to successfully cope with the tasks of educational practical programming. To realize the goal, an analysis of educational practical activities in programming was carried out in the context of their reflection on the example of creating a computer model of a simple physical phenomenon. The choice of programming as an activity for analyzing critical thinking reflection is due to the fact that this activity is well formalized. The essence of critical thinking reflection is revealed here in a refined and transparent form. Modeling gives the thinking process its own contrast, because in mental activity, in addition to algorithmic senses, additional physical ones also appear. Programming a model of a physical process here is a kind of research ground. Reflection is considered through the prism of cyclical mental processes in terms of formal logic with reference to the evolution of the structure of declarative knowledge. The final instance

of this analysis was the formalization of reflection actions in laconic schemes, both the processes themselves and the structure of knowledge. An important part of the work is the connection of mental actions with the corresponding materialized activity. This relationship was also succinctly depicted on the corresponding diagram. The clear, short, expressive formalization of critical thinking actions presented in the work, in connection with materialistic actions, in laconic schemes could be used by education seekers to optimize learning and form protocols of abstract-logical and cause-and-effect thinking. The developed reflection schemes can be considered as a methodical means of learning that allows you to avoid cognitive traps and optimize educational activities both in the field of practical programming and in natural fields of knowledge, where practical actions to create or debug artificial objects of high complexity are expected.

Keywords: critical thinking; reflection; methodology of teaching programming; hierarchy; evolution of knowledge structure.

REFERENCES (TRANSLATED AND TRANSLITERATED)

- [1] R. Scherer, F. Siddiq ta B. Sánchez Viveros, “The cognitive benefits of learning computer programming: A meta-analysis of transfer effects.” *J. Educational Psychol.*, t. 111, № 5, s. 764–792, lyp. 2019. doi: <https://doi.org/10.1037/edu0000314>. (in English)
- [2] R. Scherer, F. Siddiq ta B. Sánchez Viveros, “Some Evidence on the Cognitive Benefits of Learning to Code”, *Frontiers Psychol.*, t. 12, veres. 2021. doi: <https://doi.org/10.3389/fpsyg.2021.559424>. (in English)
- [3] J. H. Berssantette ta A. Francisco, “Cognitive load theory in the context of teaching and learning computer programming: A systematic literature review”, *IEEE Trans. Educ.*, t. 65, № 3, s. 440–449, 2022. doi: <https://doi.org/10.1109/TE.2021.3127215>. (in English)
- [4] L. Sun, Z. Guo ta D. Zhou, “Developing K-12 students programming ability: A systematic literature review”, *Educ. Inf. Technol.*, t. 27, s. 7059–7097, 2022. doi: <https://doi.org/10.1007/s10639-022-10891-2>. (in English)
- [5] L. Filippova, “A reflexive approach in the professional training of masters of pharmacy”, *Ukr. ped. zhurn.*, № 2, s. 123–130, 2020. doi: <https://doi.org/10.32405/2411-1317-2020-2-123-130>. (in Ukrainian)
- [6] A. Lozenko, “Formation of reflective skills in future primary school teachers in the process of didactic training”, *Dys. na zdobuttia nauk. stupey. kand. ped. nauk. Nats. Ped. Un-t im. M.P.Drahoman.*, Kyiv, 2010. (in Ukrainian)
- [7] L. Maliar, V. Starosta ta M. Kukhta, *Basics of psychological and pedagogical training in higher education*. Uzhhorod: DVNZ «UzhNU», 2019. (in Ukrainian)
- [8] TeachThought Staff. “10 innovative learning strategies for modern pedagogy”. Application date: 1 trav. 2024. [Online]. Available: <https://www.teachthought.com/the-future-of-learning/innovative-strategies/>. (in Ukrainian)
- [9] M. Holovin, N. Holovina ta N. Holovina, “Model consideration of cognitive processes accompanying educational programming”, *Psykholog. perspektyvy*, № 31, s. 57–70, 2018. (in Ukrainian)
- [10] M. Holovin ta N. Holovina, “Mechanisms of critical thinking and learning physics and programming”, *Fizyka ta osvitchni tekhnolohii*, № 1, s. 15–26, 2022. [Online]. Available: <http://journals.vnu.volyn.ua/index.php/physics/article/view/734/675>. Application date: 15 apr. 2024. (in Ukrainian)
- [11] J. Piaget, “Intellectual evolution from adolescence to adulthood”, *Human Develop.*, t. 15, № 1, s. 1–12, 1972. doi: <https://doi.org/doi:10.1159/000271225>. (in English)
- [12] R. Solso, *Cognitive Psychology*. Boston • London • Toronto • Sydney • Tokyo • Singapore: Allyn Bacon, 2001. (in English)
- [13] G. Miller, “The magical number seven, plus or minus two”, *Psycholog. Rev.*, t. 63, № 2, s. 81–97, 1956. [Online]. Available: <https://img3.reovome.com/m/049764c53e25268b.pdf> (in English)
- [14] O. Dahl, E. Dijkstra ta C. Hoare, *Structured Programming*. London and New York: Academic, 1972. [Online]. Available: <https://pdfcoffee.com/structured-programming-dahl-dijkstra-hoare-academic-press-1972-pdf-free.html>. (in English)

