

<https://doi.org/10.31891/PT-2025-2-8>

УДК 004.4:37.091.3:159.95+[004.42:378.147]:159.923.072

Маргарита ФРОЛОВА

старший викладач,

Херсонський національний технічний університет

<https://orcid.org/0000-0002-6650-053X>

e-mail: frolova.margo63@gmail.com

Євгенія ДРОЗДОВА

старший викладач,

Херсонський національний технічний університет

<https://orcid.org/0000-0003-0276-6387>

e-mail: jennydr@ukr.net

Віктор КОЗЕЛ

кандидат технічних наук, доцент,

Херсонський національний технічний університет

<https://orcid.org/0000-0002-2627-2499>

e-mail: k_vic@ukr.net

РОЗВИТОК МЕТАКОГНІТИВНИХ НАВИЧОК ЯК ЧИННИК ЕФЕКТИВНОГО НАВЧАННЯ ПРОГРАМУВАННЯ У ЗДОБУВАЧІВ ОСВІТИ

У статті розглянуто роль метакогнітивних навичок у підвищенні ефективності навчання програмуванню. Здійснено аналіз понять «метакогніція» та «метакогнітивна компетентність», визначено їхній вплив на саморегуляцію, планування, самоспостереження та рефлексію. Показано, що розвиток цих навичок сприяє кращому засвоєнню алгоритмічного мислення та вирішенню задач. Проаналізовано ефективність таких стратегій, як рефлексивні щоденники, самоперевірка та аналіз помилок. Наведено результати дослідження, що підтверджують зв'язок між рівнем метакогнітивної компетентності та успішністю студентів. Зроблено висновок про доцільність інтеграції метакогнітивного підходу в навчання програмуванню.

Ключові слова: метакогнітивні навички, програмування, саморегуляція, когнітивна активність студентів, самооцінка, саморефлексія.

Margarita FROLOVA, Yevheniia DROZDOVA, Viktor KOZEL

Kherson National Technical University

DEVELOPMENT OF METACOGNITIVE SKILLS AS A FACTOR OF EFFECTIVE PROGRAMMING EDUCATION FOR LEARNERS

The article examines the issue of developing metacognitive skills as a key factor in enhancing the effectiveness of programming education among higher education students. A theoretical analysis of the concepts of "metacognition" and "metacognitive skills" is conducted, outlining their role in the process of self-regulation in learning activities. The importance of developing the ability for self-regulation, self-monitoring, planning, and reflection as components of metacognitive competence is emphasized. The study analyzes scientific approaches to defining the structure of metacognitive processes and reveals the connection between their level of development and academic achievements in programming. It is

demonstrated that the integration of metacognitive strategies into the learning process promotes a deeper understanding of algorithmic thinking, improved problem-solving skills, and higher academic performance. Particular attention is given to methods and tools for developing metacognitive skills in future programmers: the use of reflective journals, self-assessment, conscious planning, and error analysis. The article presents the results of a pedagogical study confirming the positive impact of well-developed metacognitive strategies on the level of mastery of programming knowledge and skills. The conclusion emphasizes the advisability of purposefully developing metacognitive competence in students as a means to enhance their success in the field of programming. Ultimately, the article highlights the need to reconsider traditional approaches to teaching programming with a focus on the development of metacognitive competence as a prerequisite for shaping autonomous, critically thinking IT professionals. The findings can be used to improve programming teaching methods, design metacognition development training, and increase students' overall learning motivation.

Keywords: metacognition, metacognitive skills, programming, self-regulation, students' cognitive activity, self-assessment, self-reflection.

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

У світі, де технології змінюються надзвичайно швидко, здатність до ефективного самонавчання стає ключовим фактором успіху в ІТ-сфері. У сучасній системі вищої освіти, що активно реагує на динаміку розвитку цифрових технологій, все більшої актуальності набуває проблема формування не лише фахових, а й універсальних навчальних умінь. Однією з ключових передумов успішного опанування програмування є наявність у студентів метакогнітивних навичок – умінь усвідомлювати, контролювати та регулювати власні пізнавальні процеси. Метакогнітивні навички дозволяють не лише адаптуватися до нових умов, а й активно впливати на власну траєкторію професійного розвитку, роблячи навчання більш продуктивним та осмисленим. Саме ці навички сприяють більш глибокому засвоєнню складного навчального матеріалу, розвитку алгоритмічного мислення та здатності до самостійного вирішення проблем.

Актуальність дослідження зумовлена потребою в нових підходах до підготовки ІТ-фахівців, які б враховували не лише технічний аспект навчання, але й когнітивні стратегії здобувачів освіти. У центрі уваги – роль метакогнітивної регуляції у процесі навчання програмуванню, зокрема в умовах високої інтенсивності та абстрактності цього предмету.

Аналіз досліджень та публікацій

Загальний аналіз літератури показує, що є великий обсяг уже накопиченої інформації щодо феномену метакогнітивних процесів. Дослідження проблеми метапізнання беруть свій початок із робіт J. Flavell, яким і запропонований сам термін: «Metacognition and

cognitive monitoring: A new area of cognitive-developmental inquiry» (1979). Особливий інтерес викликають роботи G. Schraw «Promoting General Metacognitive Awareness», J. Biggs «The of metalearning in study processes» (1985), D. Hacker, J. Dunlosky, A. Graesser «Handbook of Metacognition in Education. Routledge» (2009) і багатьох інших зарубіжних дослідників. Теоретичні аспекти, пов'язані з визначенням сутності метакогнітивних стратегій в контексті освітнього процесу, висвітлювалися також українськими науковцями: Е. Балашов «Метакогнітивні стратегії в мотивації навчання студентів» (2020), О. Савченко «Формування та вдосконалення когнітивних та метакогнітивних стратегій як напрямок професійної підготовки студентів» (2018), А. Гриньків «Метакогнітивні дослідження в контексті освітніх інновацій» (2016), Т. Доцевич «Метакогнітивна компетентність суб'єкта педагогічної діяльності у вищій школі» (2016), Н. Сидорчук «Формування метакогніцій майбутніх фахівців у вищій школі: досвід та перспективи впровадження у навчальний процес вітчизняних навчальних закладів» (2017).

Незважаючи на велику кількість досліджень психологічних аспектів метакогнітивних стратегій у науковій сфері, залишається недостатньо вивченим їхній безпосередній вплив на ефективність навчання здобувачів освіти у сфері програмування, особливо в умовах інтенсивного опанування алгоритмічного мислення та вирішення комплексних завдань програмування.

Формулювання цілей статті

Метою цього дослідження є теоретичне обґрунтування та емпіричне дослідження впливу метакогнітивних навичок на ефективність навчання у здобувачів вищої освіти під час вивчення програмування. Поставлена мета передбачає вирішення таких завдань:

1. Визначення специфіки навчання програмуванню у контексті когнітивної й метакогнітивної діяльності здобувачів освіти.
2. Проведення емпіричного дослідження рівня сформованості метакогнітивних навичок у студентів, що навчаються програмуванню.
3. Оцінка зв'язку між рівнем метакогнітивної компетентності та результативністю навчання програмуванню.
4. Розробка педагогічних рекомендації щодо формування метакогнітивних навичок у студентів під час вивчення програмування.

Виклад основного матеріалу

У сучасному освітньому середовищі відчувається тенденція на підготовку фахівців, здатних до самостійного опанування нових знань, ефективного вирішення нестандартних задач і швидкої адаптації до змінних умов професійної діяльності. Однією з ключових галузей, що потребує таких компетентностей, є програмування. Опанування цієї дисципліни потребує не лише глибоких технічних знань, але й високого рівня когнітивної активності, критичного мислення, а також здатності до самооцінки і саморефлексії у процесі навчання. На наш погляд, для сучасного фахівця у сфері IT-технологій важливими є когнітивні та метакогнітивні навички.

У контексті нашого дослідження варто розглянути основні поняття. Когнітивні навички – це здатність мозку обробляти нову інформацію, розуміти, запам'ятовувати та використовувати її. Метакогнітивні навички стосуються здатності людини розвивати усвідомлення та розуміння власних процесів мислення. Вони охоплюють знання про те, коли і як використовувати певні стратегії для навчання чи вирішення проблем. «Метакогнітивні стратегії в психології розглядаються як стратегії, що регулюють пізнавальну діяльність студентів вищих закладів освіти. Функціями метакогнітивних стратегій є: цілеполагання, планування, моніторинг, оцінка й контроль пізнавальної діяльності» [5, с.118].

В умовах цифрової трансформації освіти особливої актуальності набуває дослідження ролі метакогнітивних навичок у підвищенні ефективності навчального процесу. Метакогнітивні навички, що охоплюють усвідомлення власного мислення, контроль за пізнавальною діяльністю та її регуляцію, визнано важливим фактором академічної успішності. Як зазначає Е. Балашов, метакогнітивні стратегії необхідні для вивчення, запам'ятування і розуміння матеріалу і поєднують повторення та опрацювання, а також планування, моніторинг та регулювання пізнання студентів для виконання когнітивних стратегій. Застосування метакогнітивних стратегій у навчальній діяльності дає змогу сформувати вміння вчитися, а що, в свою чергу, сприяє виконанню основних освітніх стандартів. Розвиток здатності до самостійного навчання та безперервної самоосвіти студентів – головне надбання використання метакогнітивних стратегій [1, с. 13].

У контексті навчання програмуванню вони сприяють більш глибокому засвоєнню матеріалу, розвитку навичок алгоритмічного мислення та здатності до самостійного вирішення складних завдань. Професійний розвиток у сфері програмування передбачає постійне

засвоєння нових знань і технологій, що вимагає від спеціалістів розвинених метакогнітивних навичок. Для програмістів вони стають особливо значущими через швидкі темпи змін у галузі та необхідність самостійно освоювати складні технічні концепції. Метакогнітивні навички стосуються здатності людини розвивати усвідомлення та розуміння власних процесів мислення. Вони охоплюють знання про те, коли і як використовувати певні стратегії для навчання чи вирішення проблем [3].

Метакогнітивні стратегії починаються з усвідомлення власного стилю навчання. Різні розробники можуть ефективно засвоювати інформацію через різні канали: одні краще сприймають візуальні матеріали, інші віддають перевагу текстовим джерелам, треті потребують практичного застосування знань для їх закріплення. Програміст, який розуміє свої пізнавальні переваги, може вибрати оптимальні джерела інформації та методи навчання, що значно підвищує ефективність процесу. Спираючись на теоретичні дослідження О. Повідайчик, М. Повідайчик, будемо розуміти індивідуальний стиль навчання як сукупність стійких пізнавальних та поведінкових чинників, які визначають особливості сприйняття студентом навчального матеріалу та реакції на різні освітні ситуації [7, с. 105].

Планування навчального процесу є ключовим аспектом метакогнітивних навичок. Замість хаотичного вивчення різноманітних технологій, ефективніше визначати конкретні цілі та послідовні кроки для їх досягнення. Це включає розбиття складних тем на логічні блоки, встановлення реалістичних термінів для їх освоєння та регулярний моніторинг прогресу. Такий підхід дозволяє уникнути перевантаження інформацією та забезпечує систематичність у навчанні. За визначенням Ю. Ватан, до метакогнітивної активності належать такі процеси, як постановка або усвідомлення мети дій із її досягнення, що охоплює і вибір засобів вирішення пізнавального завдання. Така активність (як усвідомлена та опосередкована) може вважатися довільною [4, с. 12].

Моніторинг розуміння матеріалу під час навчання є ще однією складовою метакогнітивних навичок. Досвідчені програмісти часто використовують техніку самоперевірки: після вивчення нового матеріалу вони намагаються пояснити його власними словами або застосувати на практиці. Це допомагає виявити прогалини у розумінні та своєчасно їх усунути. Також корисним виявляється ведення конспектів або створення власних кодових прикладів, які демонструють застосування вивчених концепцій.

Регулювання навчального процесу на основі отриманих результатів є логічним продовженням моніторингу. Якщо певний підхід до навчання виявляється неефективним, програміст може змінити стратегію: знайти альтернативні джерела інформації, звернутися до колег за поясненнями або виділити додатковий час на складні теми. Ця гнучкість у підході до навчання дозволяє адаптуватися до індивідуальних особливостей сприйняття та специфіки матеріалу. Як стверджує Е. Балашов, саморегульовані студенти здатні встановлювати коротко- та довгострокові цілі для свого навчання, планувати заздалегідь задля досягнення своїх цілей, самотивувати себе та зосереджувати увагу на своїх цілях та прогресі. Вони також можуть застосовувати чисельні методи навчання і коригувати їх, коли це необхідно, самостійно контролювати прогрес, звертатися за допомогою до інших і оцінювати свій прогрес на основі власних результатів [3, с. 13].

Розвиток навичок саморефлексії грає значну роль у вдосконаленні метакогнітивних здібностей. Аналіз власних успіхів і невдач у навчанні допомагає виявити ефективні стратегії та проблемні моменти. Наприклад, програміст може помітити, що краще засвоює інформацію вранці, або що певні типи завдань потребують більшого часу для розуміння. Такі спостереження дозволяють оптимізувати процес навчання.

Управління когнітивним навантаженням є критичним аспектом ефективного навчання в програмуванні. Складні технічні концепції часто вимагають одночасної обробки великої кількості інформації, що може призвести до перевантаження. Метакогнітивні стратегії передбачають усвідомлене розподілення навчального матеріалу на менші частини, чергування складних і простих тем, а також регулярні перерви для засвоєння інформації. Це допомагає підтримувати оптимальний рівень концентрації та запобігає інтелектуальній втомі.

Застосування метакогнітивних стратегій у практичній діяльності відрізняється від академічного навчання. У робочому процесі програмісти часто стикаються з необхідністю швидкого засвоєння нових технологій для вирішення конкретних завдань. У таких ситуаціях метакогнітивні навички проявляються у здатності швидко оцінити обсяг необхідних знань, визначити пріоритетні напрямки для вивчення та ефективно фільтрувати інформацію, зосереджуючись на найбільш релевантних аспектах.

Для програміста ці навички критично важливі на всіх етапах роботи. Як показують дослідження, метакогніція безпосередньо

пов'язана з ефективністю навчання і вирішенню складних завдань у програмуванні [8].

Метакогнітивні стратегії при вивченні нових мов програмування та технологій є такими:

- Планування навчального процесу: програміст з високим рівнем метакогнітивної компетентності не просто читає документацію, а формує цілі: «Я хочу вивчити TypeScript для фронтенду, тому спочатку зрозумію типізацію, потім роботу з об'єктами, а потім – інтеграцію з React».

- Оцінка прогресу: самостійне визначення того, чи достатньо знань для вирішення певного класу задач.

- Вибір адекватних стратегій навчання: відео, книги, практичні завдання, участь у спільнотах.

- Усвідомлення прогалин у знаннях – навичка, яка дозволяє уникнути «ілюзії знань».

Метакогніція є також ключовою в кожній фазі життєвого циклу програмного забезпечення:

1. Аналіз вимог – потребує рефлексивного мислення: розуміння того, як поставлені вимоги можна реалізувати технічно, а також чи вони повні, несуперечливі. Важливим тут є уміння задавати правильні питання: «Що мені ще потрібно дізнатися, перш ніж планувати архітектуру?»

2. Проектування – потребує постійної оцінки альтернатив: який підхід буде масштабованішим або читабельнішим? Метакогнітивне планування дозволяє передбачити складні місця та врахувати майбутні зміни в проєкті.

3. Реалізація – самомоніторинг під час кодування включає постійне оцінювання якості та доцільності власних дій: «Чи моє поточне рішення є простим, зрозумілим і підтримуваним?», «Чи не дублюю я код, який вже існує в проєкті?».

Рефлексія над помилками є важливою навичкою, яка дозволяє уникати їх у майбутньому.

4. Тестування передбачає розуміння важливості різних типів тестів, а також контроль когнітивного упередження розробника, який «вірять у власний код».

5. Реліз і супровід – під час цієї фази метакогнітивна гнучкість допомагає адаптуватися до змін. Програміст, який вміє аналізувати свій код після релізу, краще вдосконалює його в майбутньому.

Важливою є також взаємодія з іншими розробниками. Хоча це не технічний аспект, він глибоко пов'язаний із метакогнітивністю. Це стосується, наприклад, розуміння власних обмежень, яке дозволяє

легше просити про допомогу, уміння оцінювати ефективність комунікації, наприклад, при ревію коду. Під час парного програмування це моніторинг власної ролі, регуляція взаємодії, активне слухання [9].

Теоретичний аналіз показує, наскільки важливими є метакогнітивні навички для навчальної діяльності майбутніх фахівців у сфері IT-технологій. Спираючись на роботи Е. Балашова [2], ми розробили та впровадили декілька педагогічних заходів, спрямованих на розвиток метакогнітивних навичок студентів як чинника ефективного навчання програмуванню.

Мета: навчитися усвідомлювати та керувати власним мисленням у процесі розв'язання програмістських задач, а також розвивати навички саморефлексії, планування та контролю за навчальною діяльністю.

Інструкція до виконання:

1. Оберіть невелику програмістську задачу. Наприклад: написати програму, що перевіряє, чи є рядок паліндромом; реалізувати сортування бульбашкою; створити простий калькулятор.

2. Перед початком розв'язання (етап планування): Опишіть словами, що саме потрібно зробити для розв'язання задачі; визначте, які знання вам знадобляться (наприклад, цикли, масиви, функції); напишіть, які можливі труднощі ви очікуєте.

3. Під час розв'язання задачі (етап моніторингу): Ведіть щоденник процесу: коротко фіксуйте, що ви вже зробили, які труднощі виникли та як ви їх подолали; якщо змінили підхід – опишіть, чому саме.

4. Після завершення (етап рефлексії): Проаналізуйте, що вдалося добре, а що можна було зробити інакше; напишіть, як можна покращити свій підхід наступного разу; зробіть висновки: які метакогнітивні навички ви використали (самопланування, моніторинг, рефлексія, корекція дій).

5. Форма звітності: Код задачі. Короткий звіт (до 1 сторінки) за всіма трьома етапами.

Як продемонстрували заняття зі студентами ХНТУ, позитивно себе показала практика роботи з карткою-щоденником та таблицею для самооцінки, які можна використовувати під час або після виконання завдань із програмування. Вони спрямовані на розвиток метакогнітивних навичок.

Шаблон картки-щоденника метакогнітивних роздумів студента

1. Завдання / Тема: _____
2. Що я вже знаю про цю тему? (перерахуйте ключові поняття, які пам'ятаєте)

3. Мій план дій (перед початком): що я маю зробити? Якими етапами? Де можуть бути труднощі?

4. Що я робив(-ла) під час вирішення? (познач галочками або допиши)

☐ Я ставив(-ла) собі запитання

☐ Я намагався(-лася) пояснити собі кожен крок

☐ Я змінював(-ла) стратегію, коли щось не виходило

☐ Я використовував(-ла) приклади або документацію

☐ Інше: _____

5. Що викликало труднощі? Як я з ними впорався(-лася)?

6. Чому мій розв'язок працює (або не працює)?

7. Що я зроблю інакше наступного разу?

З метою вивчення впливу метакогнітивних стратегій (планування, самоконтроль, саморефлексія) на рівень засвоєння програмістських знань та умінь нами було проведено дослідження серед здобувачів освіти 1-4 курсів Херсонського національного технічного університету.

Методологія дослідження: 40 студентів спеціальностей 123 – Комп'ютерна інженерія, 125 – Кібербезпека та захист інформації, 126 – Інформаційні системи та технології випадковим чином розділені на дві групи по 20 осіб: **Експериментальна група:** використовує метакогнітивні стратегії. Експериментальна група проходить короткий тренінг з використання метакогнітивних стратегій: планування навчальної діяльності (встановлення цілей, розподіл часу); самоконтроль (відстеження розуміння, ведення журналу труднощів); саморефлексія (щоденне самооцінювання прогресу). **Контрольна група:** навчається без зазначених стратегій.

Хід експерименту: Обидві групи протягом 10 тижнів вивчають одну і ту саму тему – «Основи об'єктно-орієнтованого програмування на C++». По завершенню курсу всі учасники виконують тест з теорії та практичне завдання (написання програми).

Інструменти оцінювання: Теоретичний тест (макс. 30 балів). Практичне завдання (макс. 70 балів). Загальний бал – 100 балів.

Результати:

Група	Середній бал (загальний)	Теоретичний тест	Практичне завдання
Експериментальна	86,4	26,1	60,3
Контрольна	72,7	21,8	50,9

Статистичний аналіз:

З використанням t-тесту виявлено статистично значущу різницю між групами ($p < 0.05$).

Аналіз результатів дослідження. Результати дослідження засвідчили наявність статистично значущої кореляції між рівнем розвитку метакогнітивних навичок і академічною успішністю студентів. Зокрема, виявлено, що студенти з високими показниками усвідомленого планування, моніторингу та оцінки навчальної діяльності демонстрували кращі результати у вирішенні алгоритмічних задач, виявляли більшу гнучкість мислення та здатність до самостійного усунення помилок у коді.

Висновки з даного дослідження

і перспективи подальших розвідок у даному напрямі

Дані дозволяють зробити висновки про доцільність цілеспрямованої педагогічної підтримки розвитку метакогнітивної компетентності у процесі навчання програмуванню. Зокрема, ефективними виявилися інтерактивні методи навчання, навчальні рефлексивні щоденники, а також впровадження системи формувального оцінювання.

У галузі, де технології швидко змінюються, здатність швидко і якісно засвоювати нові знання стає не менш значущою, ніж технічна експертиза. Програмісти, які володіють розвиненими метакогнітивними навичками, здатні не лише ефективно навчатися, але й точно оцінювати межі власних знань, що є ключовим аспектом професійної майстерності. Практична значущість дослідження полягає у запропонованих педагогічних рекомендаціях щодо формування метакогнітивних навичок у студентів технічних спеціальностей, зокрема через застосування рефлексивних практик, активного навчання та самостійного аналізу результатів діяльності. Подальші дослідження можуть бути спрямовані на розробку спеціалізованих навчальних програм з інтеграцією метакогнітивних стратегій у вивчення програмування та суміжних дисциплін.

Література

1. Балашов Е. Д. Метакогнітивні стратегії в мотивації навчання студентів. *Наукові записки Національного університету «Острозька академія»*. Серія «Психологія»: науковий журнал. Острог: Вид-во НаУОА, 2023. № 16. С. 13-21. DOI: <https://doi.org/10.25264/2415-7384-2023-16-13-21>

2. Балашов Е. М. Метакогнітивні процеси в навчальній діяльності студентів. *Науковий вісник ХДУ. Серія : Психологічні науки*. 2020. № 1. С. 98-105. DOI: <https://doi.org/10.32999/ksu2312-3206/2020-1-11>.
3. Балашов Е. М. Теоретичні підходи до вивчення чинників мотивації саморегульованого навчання студентів. *Теорія і практика сучасної психології*. 2019. № 1, Т. 2. С. 35-41. ISSN 2663-6034 (Online).
4. Ватан Ю. П. Структура та сутність метакогнітивної активності. *Психологічний вісник*. 2021. № 3. С. 10-18. DOI: <https://doi.org/10.32782/psy-visnyk/2021.3.2>.
5. Доцевич Т. Метакогнітивні стратегії розвитку читацької компетентності в студентів закладів вищої освіти. *Psycholinguistics*. 2019. Т. 26, № 1. С. 105-125. DOI: <https://doi.org/10.31470/2309-1797-2019-26-1-105-125>.
6. Муқан Н., Чубінська Н., Сташишин Р. Роль значення когнітивних та метакогнітивних навичок у професійній діяльності правника. *Академічні візії. Секція «Освіта/Педагогіка»*. 2023. Вип. 22. DOI: <https://dx.doi.org/10.5281/zenodo.8177339>.
7. Повідайчик О. С., Повідайчик М. М., Карбованець Е. О. Врахування індивідуально-стильових особливостей у процесі професійної підготовки математиків. *Науковий вісник Ужгородського університету. Серія : Педагогіка. Соціальна робота*. 2022. Вип. 2(51). С. 104-108. DOI: <https://doi.org/10.24144/2524-0609.2022.51.104-108>.
8. Bubnic B., Kovačević Ž., Kosar T. Can metacognition predict your success in solving problems? An exploratory case study in programming [Electronic resource] // arXiv. 2024. October 8. arXiv:2410.06267. Available at: <https://arxiv.org/abs/2410.06267>.
9. Matook S., Wang Y. M., Koeppel N., Guerin S. Metacognitive skills in low-code app development: Work-integrated learning in information systems development. *Journal of Information Technology*. 2023. Vol. 39, No. 1. P. 41-70. DOI: <https://doi.org/10.1177/02683962231170238>.

References

1. Balashov, E. M. (2023). Metacognitive Strategies in Student Learning Motivation. Scientific Notes of the National University "Ostroh Academy". Series "Psychology", (16), 13-21. DOI: <https://doi.org/10.25264/2415-7384-2023-16-13-21>
2. Balashov, E. M. (2020). Metacognitive Processes in Students' Learning Activity. Scientific Bulletin of Kherson State University. Series: Psychological Sciences, No. 1(2020). DOI: <https://doi.org/10.32999/ksu2312-3206/2020-1-11>
3. Balashov, E. M. (2019). Theoretical Approaches to Studying the Factors of Motivation for Self-Regulated Learning in Students. Collected Scientific Works "Theory and Practice of Modern Psychology", No. 1, Vol. 2. ISSN 2663-6034 (Online)
4. Vatan, Y. P. (2021). Structure and Essence of Metacognitive Activity. DOI: <https://doi.org/10.32782/psy-visnyk/2021.3.2>

5. 1. Dotsevykh, T. (2019). Metacognitive Strategies for Developing Reading Competence in University Students. *Psycholinguistics*, 26(1), 105-125. DOI: <https://doi.org/10.31470/2309-1797-2019-26-1-105-125>
6. Mukan, N., Chubinska, N., & Stasyshyn, R. (2023). The Role of Cognitive and Metacognitive Skills in the Professional Activity of a Lawyer. *Academic Visions. Section "Education/Pedagogy"*, Issue 22. DOI: <https://dx.doi.org/10.5281/zenodo.8177339>
7. Povidaychyk, O. S., Povidaychyk, M. M., & Karbovanets, E. O. (2022). Considering Individual-Style Characteristics in the Professional Training of Mathematicians. *Scientific Bulletin of Uzhhorod University. Series: Pedagogy. Social Work*, Issue 2(51). DOI: <https://doi.org/10.24144/2524-0609.2022.51.104-108>
8. Bubnic B., Kovačević Ž., Kosar T. Can metacognition predict your success in solving problems? An exploratory case study in programming [Electronic resource] // arXiv. 2024. October 8. arXiv:2410.06267. Available at: <https://arxiv.org/abs/2410.06267>.
9. Matook, S., Wang, Y. M., Koepfel, N., & Guerin, S. (2023). Metacognitive skills in low-code app development: Work-integrated learning in information systems development. *Journal of Information Technology*, 39(1), 41-70. DOI: <https://doi.org/10.1177/02683962231170238>