

## ЗАВДАННЯ ДЛЯ АВТОМАТИЗОВАНОГО ТРЕНІНГУ СТУДЕНТІВ З ПРОГРАМУВАННЯ

Микола Головін

Розглянуті різні типи завдань для автоматизованого тренінгу з програмування в контексті структурної організації ментального відображення об'єктів цієї діяльності.

The different types of tasks for the automated training on programming in the context of structural organization of this activity objects' mental reflection are considered.

Проблеми традиційних шкільних і вузівських технологій навчання загальновідомі. До них відносяться: поганий зворотній зв'язок між викладачем і кожним суб'єктом навчання, обмеженість ресурсів викладача при роботі з групою, недосконалий індивідуальний підхід у навчальному процесі. Названі проблеми взаємозв'язані і мають свою специфіку. Кожен індивідуум групи, що здійснює навчальну практичну діяльність з програмування, знаходиться у своєму локальному проблемному просторі. Типовий сеанс виконання практичного завдання з програмування (на створення, відлагодження, виправлення, модернізацію програми) передбачає багатокрокові дії. Виконання кожного окремого завдання можливе кількома способами. Викладач не може миттєво зорієнтуватись у кожному з них і надати допомогу. Ненадання вчасно необхідної допомоги суттєво знижує темп навчання і робить його малоефективним. Виконання кожного окремого практичного завдання породжує технологічні проблеми стосовно кожного із суб'єктів навчання. Так циклічно виникають проблеми: пред'явлення наступного навчального завдання, створення умов для його виконання та перевірки правильності; знищення непотрібних файлів, які утворились впродовж практичної роботи; пошуку помилок, їх візуалізації та аналізу; оцінювання результатів виконання кожного завдання; ведення детального протоколу роботи. Все це теж знижує темп навчання.

Комп'ютер, як технічний засіб навчання, дозволяє суттєво покращити ситуацію. Завдання оптимізації, підвищення продуктивності і якості роботи викладача шляхом її автоматизації залишається актуальним.

Автоматизація навчання включає проблеми технологій керування процесом і його інформаційного наповнення. В обох випадках необхідні добре формалізовані коректні теоретичні та емпіричні уявлення про розумові процеси, що супроводжують навчання.

Метою цієї роботи є формування на основі доктрин когнітивної психології теоретичного підґрунтя під методикку створення пакетів завдань для практичного програмування. Теоретичні підходи пізнавальної психології і експериментальна база практичного програмування можуть дати, на погляд автора, цікавий конгломерат можливостей для психолого-педагогічних досліджень в галузі нових інформаційних технологій навчання.

Пізнавальна (cognitive) психологія є цікавою основою для модельних підходів до розумових процесів. Базисною відправною точкою досліджень в чисельних течіях когнітивної психології (Ф. Бартлетт, С. Палмер, У. Найссер, Э. Рош, М. Минський, Б. Величковський і ін. [1]) є думка, що інтелектуальна діяльність детермінується структурною організацією пізнавальної сфери. Ця ідея є ключовою і у даній роботі. Згідно з ідеями відомого генетичного підходу до психічних структур, жодна з них ніколи не є радикально новою, кожна є модифікацією попередньої в часі структури. Найбільш контрастну оцінку дає У. Найссер. Він вважає, що „ті види інформації, для яких у нас нема схем, ми просто не сприймаємо” [2].

Практичне програмування є надзвичайно зручним полігоном для розробки завдань автоматизованого навчання та дослідження їх ефективності. Розглянемо ключові аспекти цієї тези.

У програмуванні зразу після засвоєння порції декларативних знань передбачається використання їх в практичній діяльності, максимально наближеній до професійної. Це вимагає швидкого включення нової порції зовнішньої інформації в систему існуючих у суб'єкта діяльності знань. Не всі навчальні дисципліни надають таку можливість. Слово

„розуміння” набуває конкретного звучання. Якщо суб’єкт навчання може застосовувати новий навчальний об’єкт в діяльності, то він розуміє його сутність. Ця сфера знань абсолютно не приймає навчальних процесів, що будуються на вербальному декларативному „зазубрюванні” догматів.

Інша важлива специфіка програмування в тому, що ця навчальна діяльність, спрямована на складні, добре структуровані об’єкти великого розміру, має достатню тривалість, причинно–наслідковий узгоджений характер та хороший контраст і яскравість. Тут проявляються закономірності розумових процесів, невидимі при діяльності стосовно простих об’єктів. Тому, із всього спектра складних об’єктів, комп’ютерні програми видаються автору найбільш цікавими в контексті дослідження навчальних процесів.

Приваблива особливість програмних об’єктів в тому, що вони добре формалізовані і легко доступні для будь-яких матеріалізованих змін за допомогою текстового редактора. Внутрішні розумові дії, що приводять до радикальної зміни у логіці роботи програми, займають значно більше часу, ніж відповідні зовнішньо-предметні дії редагування тексту. Внутрішні розумові дії дають домінуючий вклад у процеси, що фіксуються. Діапазони складності і розміру програм можна варіювати і вони є достатньо широкими.

Навчальне програмування є зручним для автоматизованого діагностування та керування. Виправлення в тексті програми фіксуються у пам’яті комп’ютера та на його накопичувачі інформації. Логічно завершені етапи роботи суб’єкта діяльності стосовно окремих фрагментів програми приводять до її періодичних збережень та запусків на виконання. Випробування програм на дієздатність дають можливість автоматизовано вимірювати як правильність виконання завдань, так і їх тривалість. Ці випробування є реперними подіями в сенсі спостереження за розумовими процесами. Інформативним, з точки зору автоматизованої діагностики правильності виконання завдання, є файл, що утворюється у результаті роботи програми, яку створює, чи модернізує студент. Сприятливим для такої автоматизованої роботи та моніторингу є вивчення розділів програмування, пов’язаних з масивами, рядками, файлами. Діяльність суб’єкта навчання завжди доступна для безпосереднього спостереження на екрані та для автоматизованого моніторингу за допомогою діагностичної програми, що працює паралельно із засобами програмування в багатозадачному операційному середовищі Windows.

Для експериментального дослідження психолого-педагогічних процесів автором була створена спеціальна програма, яка дозволяє навчати та акумулювати діагностичну інформацію. Ця програма відслідковує навчальні дії практичного програмування в оригінальному програмному середовищі Delphi. Навчальна діяльність максимально наближена до реальної практичної роботи. Перевірка правильності виконання завдання з програмування забезпечується як через сканування ключових фрагментів тексту створеної (виправленої, модернізованої, доповненої) програми, так і через контроль результатів її роботи. Контроль тільки результатів роботи робить тренажерну програму нечутливою до способу та інструменту виконання завдання.

У цілому, алгоритм роботи тренажерної програми для кожного завдання такий: знищення файлів, що з’явилися впродовж виконання попередніх завдань; пред’явлення поточного завдання та завантаження файлів, потрібних для його виконання; хронометрування часу діяльності; завантаження і перевірка результуючих файлів; представлення карти помилок; фіксація результату у протоколі роботи.

Пакет завдань, що використовувався для тренажерної роботи на одному занятті (80 хвилин), містив 20 однотипних завдань з теми. Розрахунковий час виконання завдання 4 хвилини. Більшість з програм, що були покладені в основу завдань, мають описані аналоги в лекційних матеріалах, або інструкціях до лабораторних робіт. Лекційні і лабораторні аплікаційні програми суттєво відрізнялись від програм для тренажерної роботи позначеннями, значеннями параметрів. „Механічний” перенос аплікації під час виконання завдання був неможливий. Синтез нової програми з прив’язкою до завдання

вимагав попереднього аналізу аплікації при підготовці до заняття.. У багатьох випадках одна аплікаційна програма була батьківською для кількох схожих програм для тренінгу.

Для зменшення рутинного набору тексту студентам пропонувався для завантаження попередній стартовий шаблон. Для кожного завдання шаблон був інший. Завдання стосувалось створення, виправлення, дописування, модернізації програм. Вони були однорідними за складністю і трудомісткістю. Частина операторів включалась у шаблон.

Проведемо психолого-педагогічний аналіз процесів виконання завдань різних типів. Здійснимо цей аналіз на основі обговорення ментального відображення об'єкта діяльності в довготривалій пам'яті. Таке відображення формується впродовж виконання завдань будь-якого типу. На погляд автора, найбільш досконале відображення об'єкта утворюється при його створенні. Модернізація відлагодження, виправлення та інші завдання формують відображення об'єкта також. Міра деталізації цього відображення визначається необхідною достатністю в контексті завдання.

Когнітивна психологія оперує широким термінологічним оформленням структурних утворень пізнавальної сфери. Зокрема, використовуються такі поняття, як „ментальний простір”, „ментальна структура”, „ментальний досвід”, „когнітивна структура”, „когнітивна схема”, „когнітивна карта”, „понятійна структура”, „концепт”, „конструкт”, „прототип”, „паттерн”. Деякі з цих термінів дублюють або позначають близькі за значенням поняття. Інші поняття мають різну міру конкретизації. В умовах реальної інтелектуальної діяльності одночасно працюють когнітивні утворення різного ступеня узагальненості. Вони є „вбудованими” одне в одне.

Психолого-педагогічний аналіз процесів виконання різних типів завдань з програмування проведемо на основі оригінальної ментальної схеми, яку пропонує автор для розгляду відображення складного об'єкта (рис.1). Ця схема, як і процес пізнання, що вона описує, не є статичною. Схема здатна до еволюції. Тому аналіз буде стосуватись модельних динамічних змін відображення складного об'єкта в процесі діяльності. Еволюція схеми в процесі створення програми представлена на рис.2.

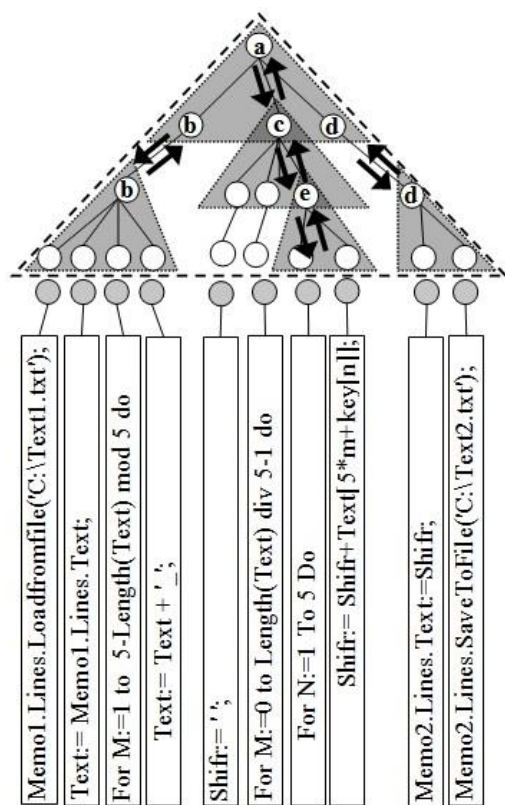


Рис.1 Когнітивна схема навчального об'єкта та відповідний йому текст програми. Стрілками позначені транзити уваги схемою.

Рис.1 є платформою для теоретичного розгляду розумових процесів, супутніх виконанню завдання. Тут зображена графічно формалізована ментальна структура досить простої програми шифрування перестановкою знаків. (Позначення в програмі: Text та Shifr – рядкові змінні для зберігання оригінального та шифрованого тексту, відповідно; Key – числовий масив (ключ), що задає порядок перестановок; N, M – змінні для рахівників у циклах; Text1.txt, Text2.txt - файли з оригінальним і зашифрованим текстом).

Схема має три частини. Ліва відповідає за завантаження текстового файла в строкову змінну, центральна - за шифрування, права - за збереження зашифрованого повідомлення. Ментальна частина структури - когнітивна схема об'єкта, представлена структурою у великому трикутнику, окресленому пунктирною лінією. У вершині трикутника - об'єднуючий вузол. Він позначає функції об'єкта як цілісного утворення. Вузли основи позначають відображення в пам'яті реальних неподільних елементів об'єкта - операторів програми. Реальні оператори зображені як рядок тонованих кружечків зовні трикутника.

Схема має три частини. Ліва відповідає за завантаження текстового файла в строкову змінну, центральна - за шифрування, права - за збереження зашифрованого повідомлення. Ментальна частина структури - когнітивна схема об'єкта, представлена структурою у великому трикутнику, окресленому пунктирною лінією. У вершині трикутника - об'єднуючий вузол. Він позначає функції об'єкта як цілісного утворення. Вузли основи позначають відображення в пам'яті реальних неподільних елементів об'єкта - операторів програми. Реальні оператори зображені як рядок тонованих кружечків зовні трикутника.

Всередині великого трикутника є тоновані трикутники меншого розміру. Це трикутники зосередження уваги. Вони окреслюють функціональні вузли, на яких одночасно сконцентрована увага. У процесі логічних дій поступово відкидаються „зайві” або вводяться „недостаючі” вузли. Межі уваги балансують. Увага завжди концентрується на логічно закінчених структурах. Конструктами будемо називати ментальні структури у довготривалій пам’яті, що є результатами аналітико-синтетичної роботи мозку впродовж зосередження уваги.

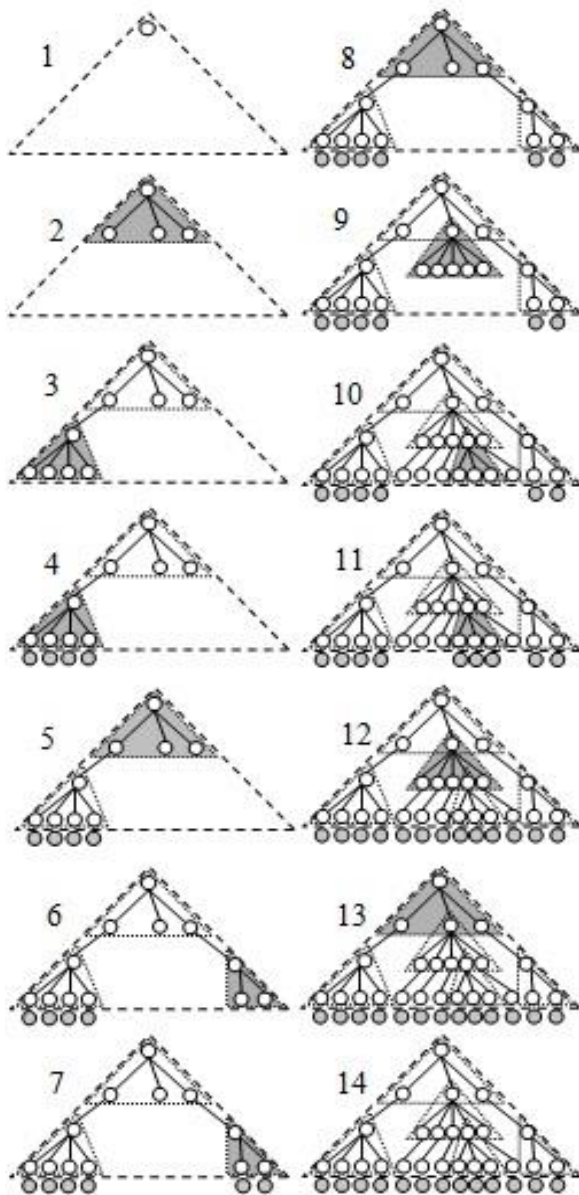


Рис.2 Еволюція когнітивної схеми навчального об’єкта впродовж його створення.

багаточисельними функціональними вузлами складного об’єкта, які вона сприймає, розрізняє, функції яких знає. Суб’єкт навчання змушений у процесі освоєння, створення, модифікації, відлагодження об’єкта переводити увагу і проводити абстрактно – логічні дії для зв’язування функціональних вузлів. У результаті у довготривалій пам’яті поступово формується ментальна структура ієрархічного типу - пізнавальна (когнітивна) схема об’єкта. Ієрархічність схеми реалізується через віртуальні вузли, позначені на рис.1 буквами а, b, с, d, е. Ці вузли дозволяють зв’язати в єдине ціле відображення реальних операторів - вузлів в основі трикутника. Так проблема а шифрування перестановкою тексту, що знаходиться в файлі, представляється ієрархічною структурою у яку в якості складових входять локальні проблеми: b - завантаження файлу і підготовка тексту до

ментальні структури у довготривалій пам’яті, що є результатами аналітико-синтетичної роботи мозку впродовж зосередження уваги. Вузли основи конструкта є складовими частинами вузла у вершині. Конструкти утворюють ієрархічну структуру. Вони частково накладаються. Через спільні вузли суміжних конструктів реалізується логічний зв’язок у розумових діях при переводі уваги. Реалізуються ланцюги абстрактно-логічних розумових дій. Видно, що ментальна структура об’єкта періодична. Конструкти утворюють цю періодичність. Тільки один конструкт у вершині ментальної схеми об’єкта не є вузлом іншого. Будь-які інші два конструкти - або функціональні вузли третього конструкта, або один конструкт є вузлом другого. Пересування подібною структурою має циклічний характер. Кінець одного кроку і початок другого відокремлюється сеансом переведення уваги та абстрагуванням. У сенсі дослідження розумових дій абстрагування є реперною подією.

З рис.1 видно, що поруч розташовані оператори можуть розглядатись у різних сеансах зосередження уваги. Тому актуальним є перенесення відображення оператора схемою з рівня реалізації у відповідний трикутник зосередження уваги. Такий перенесений вузол має зв’язок одинарною лінією з верхньою і нижньою, по відношенню до себе, частиною схеми.

Людина змушена створювати такі ментальні ієрархічні конструкції. Ключовою причиною цього, на думку автора, є те, що людина не може оперувати одночасно всіма

шифрування; **c** - шифрування тексту перестановкою по всій його довжині; **d** - збереження зашифрованого тексту в файлі, **e** - шифрування тексту перестановкою в межах довжини ключа. Ієрархічна схема надає суб'єкту діяльності важливі спроможності створювати об'єкти високої складності та масштабувати уявлення про ці об'єкти у процесі діяльності.

Розглянемо завдання на створення програми в контексті ментальної структури. Виконання цього завдання найбільш повно відображає весь спектр дій, пов'язаних з реальною професійною роботою програміста.

Загальні тенденції наступні. Формування кожної ментальної структури стосовно об'єкта має дві послідовні фази конкретизації і узагальнення [3].

На етапі конкретизації відбувається пересування ментальною структурою вниз, здійснюється аналіз проблеми, яку повинен вирішити програмний продукт. Відбувається аналітичне розумове розчленування проблеми на окремі локальні проблемні ситуації. Наступні дедуктивні дії з'ясовують роль кожної локальної проблеми у контексті загальної.

Сенс конкретизації - у формуванні орієнтовної основи матеріалізованих дій по написанню окремих операторів програми. Складання власної схеми орієнтуючої основи дій є початковим етапом у вирішенні будь-якої проблеми. Специфічність програмування у тому, що оператори програми є послідовністю відкладених кроків подолання проблеми.

Первинними в конкретизації є попередньо набуті базові знання, навички та уміння. Саме вони породжують нову вторинну ментальну структуру - орієнтовну основу для конкретних матеріалізованих дій та самі матеріалізовані дії.

Процес створення орієнтовної основи переходить у матеріалізовані дії тоді, коли виникає можливість подолати кожен з локальних проблем окремих засобом мови програмування. На рис.2.4, 2.7, 2.11, 2.12 показані стани орієнтовної основи дій, достатні для початку практичного програмування, стосовно відповідних фрагментів програм.

Надалі така матеріальна структура, як написаний фрагмент програми з доволі жорстко заданими властивостями, стає джерелом для подальших індуктивно-синтетичних дій. У цьому сенсі матеріалізовані дії тут первинні, а розумові індуктивно-синтетичні - вторинні.

На етапі узагальнення впродовж індуктивно-синтетичних дій відбувається пересування ментальною структурою вгору. Індуктивні розумові дії дозволяють отримати за функціями окремих вузлів (операторів) функцію всього програмного фрагмента, як цілісного утворення. Синтетична дія гармонійно поєднана з ланцюгом індуктивних і є завершальною в формуванні конструкта - цілісних уявлень про відповідний програмний фрагмент. Очевидним є і те, що матеріалізовані дії налаштування операторів через їх параметри продовжуються при індуктивних діях. Таке налаштування написаного в цілому програмного фрагмента можна назвати відлагодженням.

Спостереження за студентами, що знаходяться в діяльності практичного програмування свідчать наступне. Після матеріалізованих дій з написання логічно закінченого фрагмента програми відбувається його випробування. Якщо при випробуванні досягнуто позитивного результату - діяльність закінчується, робота над програмою вважається завершеною.

Можливий і негативний результат перевірки дієздатності готового фрагмента, коли програмне утворення або взагалі не працює, або має функції, відмінні від бажаних. Наприклад, функції вузла у вершині трикутника на рис.2.10 та на рис.2.11 відмінні. У цьому випадку відбуваються повторні індуктивно-синтетичні дії. Якщо у суб'єкта виникають підозри про хибність самої орієнтовної основи відбуваються повторні аналітико-дедуктивні дії, а потім індуктивно-синтетичні.

Матеріалізовані дії часто мають мовне супроводження. У цьому випадку мовну індуктивно-синтетичну діяльність можна спостерігати в рафінованому вигляді під час відлагодження програми. Вона відбувається в термінах назв реальних вузлів об'єкта та їх зв'язків, адже кожний оператор програми має вербально оформлену назву та функціональне призначення.

Спостереження за роботою студентів показали, що мовна підтримка характерна для застосування суб'єктом дій нових для себе операторів. При цьому відбувається перекодування отриманих зовні декларативних відомостей про оператори (лекції, підручники) у власну систему знань.

Перехід до внутрішньо-розумової діяльності стосовно як нової програмної конструкції, так і окремого оператора в цій конструкції, відбувається впродовж багатократного використання відповідного утворення в різних програмах. Розумові дії стосовно часто вживаних конструкцій в значній мірі згортаються та стають скороченими, внутрішніми. У практичній діяльності з програмування, як показують спостереження, можна побачити окремі етапи формування розумових дій та понять теорії Гальперіна [4], однак послідовність цих етапів є більш складною і детермінується особливостями ментального відображення об'єкта діяльності.

Проаналізуємо формування завдань і їх виконання в контексті викладеної вище теорії. Формалізація матеріалу в графічному вигляді дозволяє побачити деякі цікаві закономірності, які погано розрізняються в мовному контексті.

Ментальне відображення завдання на створення програми показано на рис.2.1. Виконання вимагає від суб'єкта навчання багато розумових проміжних дій на шляху до мети - написання програми (рис.2.14). Проміжні стани когнітивної схеми об'єкта на цьому шляху зображені на рис. 2.2-2.13.

Завдання на модернізацію (рис.3.1) відрізняється від завдання на створення тим, що крім опису майбутньої програми в загальних рисах (кружечок у вершині трикутника) у завданні надається текст програми, що вимагає модернізації (тоновані кружечки під трикутником). Однак, ментальна структура – орієнтовна основа матеріалізованих дій відсутня в обох випадках. Діяльність на модернізацію вимагає проміжних розумових дій, подібних до приведених вище. На рис.3.2 ментальна схема програми для модернізації в завершеному стані. Можна сказати, що і в першому, і в другому випадку шляхи розумових дій схожі. Матеріалізованим діям написання окремих операторів при створенні програми відповідають дії по передбаченню операторів в тексті при модернізації. Адже суть пізнавального процесу в передбаченні результатів [2]. Мета створення і модернізації однакова – працююча програма. Вона досягається при повністю сформованій ментальній схемі об'єкта рис.2.14.

Рис.3 Стани ментальних схем об'єктів дій

Спорідненим до модернізації завданням є завдання на розпізнавання програмних блоків готової програми. Ментальне відображення такого завдання представлено на рис.3.3. Суть завдання полягає в тому, що необхідно знайти в тексті програми фрагмент з визначеними функціями, виділити його і зберегти для подальшої автоматизованої перевірки. Навчальний сенс застосування таких завдань в стимулюванні процесу освоєння готових програм. Когнітивна структура, що зв'язує оператори, відсутня. При ознайомленні передбачається, що в процесі виконання завдань така структура буде створена.

Завдання на дописування відрізняється від завдання на модернізацію тим, що початковий текст програми, що дається із завданням, не є працюючим.

Цікавими є завдання на відлагодження програми. Навіть невеликі виправлення у незнайомій програмі потребують формування повної ментальної структури цього об'єкта. Час виконання таких завдань часто одного порядку з часом виконання завдань на створення програми.

Необхідно зазначити, що для практичних завдань всіх типів важливе значення має наявність у суб'єкта конкретизованої орієнтовної основи дій. В усіх випадках відсутність конкретизованої основи дій вимагає додаткової розумової роботи в термінах проблемних

ситуацій. Інформаційна база проблеми може знаходитись в області далекій від програмування.

Представлена формалізація ментальних структур дає можливість враховувати трудомісткість розумових дій стосовно різних завдань та створювати однорідні за трудомісткістю пакети завдань.

#### Література

1. Холодная М.А. Психология интеллекта: парадоксы исследования. — СПб.: Питер, 2002. - 272с.
2. Найссер У. Познание и реальность. Смысл и принципы когнитивной психологии. М.: Прогрес, 1981. - 225с.
3. Головін М.Б. Специфіка формування пояснень до прикладів – аплікацій у навчальному матеріалі з програмування //Проблеми сучасного підручника, Збірник наукових праць. Випуск 7. Київ-Луцьк, 2007. – С.339-350.
4. Гальперин П.Я. Типы ориентировки и типы формирования действий и понятий // Доклады АПН РСФСР. – 1958. – № 2. – С.75-78.