

## РОЗУМОВІ ТА МАТЕРІАЛІЗОВАНІ ДІЇ, ЇХ ВЕРБАЛЬНЕ СУПРОВОДЖЕННЯ В КОНТЕКСТІ ПРАКТИЧНОГО НАВЧАЛЬНОГО ПРОГРАМУВАННЯ

Оптимізація процесів практичної навчальної діяльності неможлива без детального уявлення про психологічні пізнавальні процеси. У цій роботі зроблена спроба розглянути взаємозв'язки розумових, матеріалізованих та вербальних дій у процесі навчального програмування.

Робота спирається на концепцію когнітивної психології про те, що інтелектуальна діяльність детермінується структурною організацією пізнавальної сфери [1, 2]. Тут розвивається модельний підхід до процесів навчання, представлений, зокрема, в роботі [3, 4, 5].

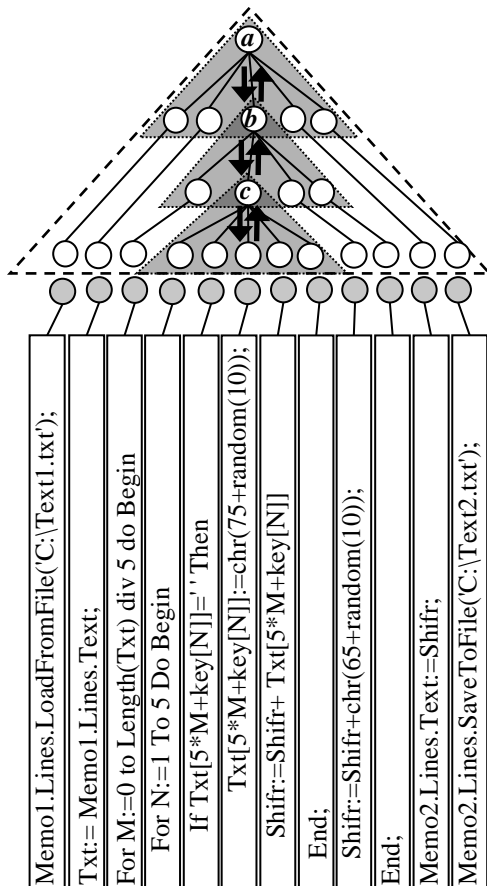


Рис.1 Когнітивна схема навчального об'єкта та відповідний йому текст програми. Стрілками позначені транзити уваги схемою.

трикутники меншого розміру. Ці трикутники окреслюють вузли, на яких концентрується увага у процесі розумових дій. Сукупності вузлів всередині тонованих трикутників будемо називати конструктами. Ці структури є логічно завершеними. Вузли основи конструкта є складовими частинами вузла у вершині. Конструкти частково накладаються. Через спільні вузли суміжних конструктів відбувається логічний зв'язок у розумових діях при переводі уваги, реалізуються ланцюги абстрактно-логічних розумових дій.

З рис.1 видно, що поруч розташовані оператори можуть розглядатись у різних сеансах зосередження уваги. Перенесення відображення оператора схемою з рівня реалізації у відповідний трикутник зосередження уваги позначено лінією.

На рис.1 подано оригінальну графічну формалізацію когнітивної схеми програмного об'єкта. Це структура в трикутнику, що позначений пунктиром. Знизу під структурою сама програма. Ментальна структура має ієрархічну будову і є логічним каркасом уявлень про об'єкт. Схема, зображена на рисунку, є завершеною. Процес її формування відбувається поступово у реальному часі і буде розглянутий нижче. Програма шифрує текст перестановкою знаків. Рядки `Shifr:=Shifr+chr(65+random(10));` та `If Txt[5*M+key[N]]=' ' Then Txt[5*M+key[N]]:=chr(75+random(10));` підсилюють захист шифрованого тексту. Перший рядок вставляє через кожні 5 знаків випадковий шостий, а другий рядок замінює пробіли на випадкові знаки.

У програмі такі позначення: `Txt` 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.

Реальні оператори зображені як рядок тонованих кружечків зовні трикутника, позначеного пунктиром. У вершині цього 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. Всередині великого трикутника є тоновані

Ієрархічність схеми реалізується через віртуальні функціональні вузли, позначені на рис.1 буквами **a**, **b**, **c**. Ці вузли дозволяють зв'язати в єдине ціле відображення реальних операторів - вузлів у основі трикутника. Так проблема **a** шифрування тексту, що знаходиться у файлі, представляється ієрархічною структурою у яку в якості складових входять локальні проблеми: шифрування тексту по всій його довжині - **b** та завантаження і збереження файлу. Проблема **c** є частиною проблеми **b**. Вузлом **b** позначена проблема шифрування тексту в межах довжини ключа. Ієрархічна організація пізнавальної схеми надає суб'єкту діяльності важливу спроможність - масштабування уявленнями про об'єкт. Масштабування у процесі діяльності дозволяє суб'єкту створювати об'єкти як завгодно високої складності.

На рис.1 стрілками показаний порядок формування ментальної схеми об'єкта в процесі діяльності по створенню програми. Проблема **a** є початковою. При продумуванні проекту проблема **a** породжує проблему **b**, яка в свою чергу породжує - **c**. Проблема **c** можна практично розв'язати наявними операторами. При цьому увага концентрується одночасно на всіх операторах конструкта **c**. Реалізація конструкта **c** у вигляді програмного фрагмента та успішне його випробовування дає можливість надалі сприймати відповідний програмний фрагмент монолітно. Це дозволяє перейти до конструкта **b** і включити в поле уваги для логічної обробки додаткові оператори. Програмна реалізація конструкта **b** інтегрує уявлення про цей фрагмент. Як і в попередньому випадку, поле уваги частково звільняється. Відбувається перехід до конструкта **a**. Крім вузла **b**, увага концентрується також на операторах, що забезпечують завантаження з файлу та запис у файл. Після інтегрування уявлень про конструкт **a** та успішного випробовування відповідного програмного фрагмента робота над програмою завершується.

Необхідно відмітити, що поняття про призначення програмних блоків **a**, **b**, **c** на концептуальному лаконічному рівні є більш абстрактним, ніж поняття про призначення окремих операторів. Найбільший ступінь абстрактності у поняття **a**, найменший - у **c**. Це пояснюється тим, що при кожному переході ієрархією вгору приходиться абстрагуватись від деталей реалізації чергового конструкту.

Процес реалізації програми відбувається усвідомлено, якщо суб'єкту дій приходиться реалізовувати в ній нові, креативні, в контексті своєї попередньої діяльності, алгоритмічні рішення. Слово „креативні” в сенсі програмування має контрастне забарвлення, адже це слово “create” саме і означає створювати. Усвідомленість дій під час створення нового програмного фрагмента, а також підготовчих роздумів є дуже важливою обставиною. Адже усвідомлені розумові дії реалізуються через вербальну складову мислення.

У сенсі педагогіки і методики проведення лабораторних занять цікавими є три аспекти. По-перше, формування оптимізованих інтелектуальних умінь і навичок, що підтримують перенесення уваги і логічні дії в процесі створення програми. По-друге, формування системи вербалізованого понятійного апарату, що включає взаємозв'язані професійні поняття різного ступеня узагальненості і абстрактності. Саме цей понятійний апарат підтримує логічні дії і дозволяє реалізовувати перенос уваги, як змінюючи масштаб „бачення” об'єкта, так і без його зміни. По-третє, цікавим є і оптимізація траєкторії формування ментальної структури складної програми, а також сутність дій при утрудненнях інтелектуальної роботи.

Ментальне відображення програми формується у довготривалій пам'яті як орієнтовна основа майбутніх матеріалізованих дій. Ця структура тривалий час може відтворюватись відокремлено від інших знань. Формування цієї конкретизованої ментальної структури, яка підтримує процес програмування, є актом навчання. Актом навчання є також, безумовно, і формування узагальненої вербально організованої понятійної системи знань. Це знання декларативного типу. До навчальних процесів відноситься вдосконалення системи інтелектуальних умінь і навичок, що узгоджують та гармонізують логічні дії та перенос уваги. Зрозуміло, що покращення системи умінь та

навичок є формуванням неусвідомленої сфери мислення. Сам процес формування може відбуватись як усвідомлено – експліцитно, так і неусвідомлено - імпліцитно. Усвідомлене формування умінь і навичок під керівництвом викладача є більш привабливим, адже вироблені таким чином уміння та навички є, з однієї сторони, оптимізованими, а з другої - здатними до подальшої диференціації та вдосконалення у процесі навчання.

Модельний розгляд навчальних процесів у контексті розумових, матеріалізованих і вербальних дій може бути цікавим для викладачів, що проводять лабораторні заняття з програмування. Цікавими є також модельний аналіз дій при утрудненнях у контексті моделі. Розглянемо ці аспекти діяльності нижче.

На рис.2 розумові та матеріалізовані дії представлені окремо. Розумова діяльність розглядається в контексті конкретизації (1) уявлень про об'єкт, та дій по узагальненню (4) таких уявлень. Результатом розумових дій є ментальне відображення об'єкта. Це ж відображення є ментальною основою матеріалізованих дій.

Виконання матеріалізованих дій у сенсі програмування має два аспекти (рис.2) це: формування матеріалізованої орієнтовної основи дій (2) і самі матеріалізовані дії з програмування (3).

Перший аспект матеріалізованих дій представляє собою формалізацію

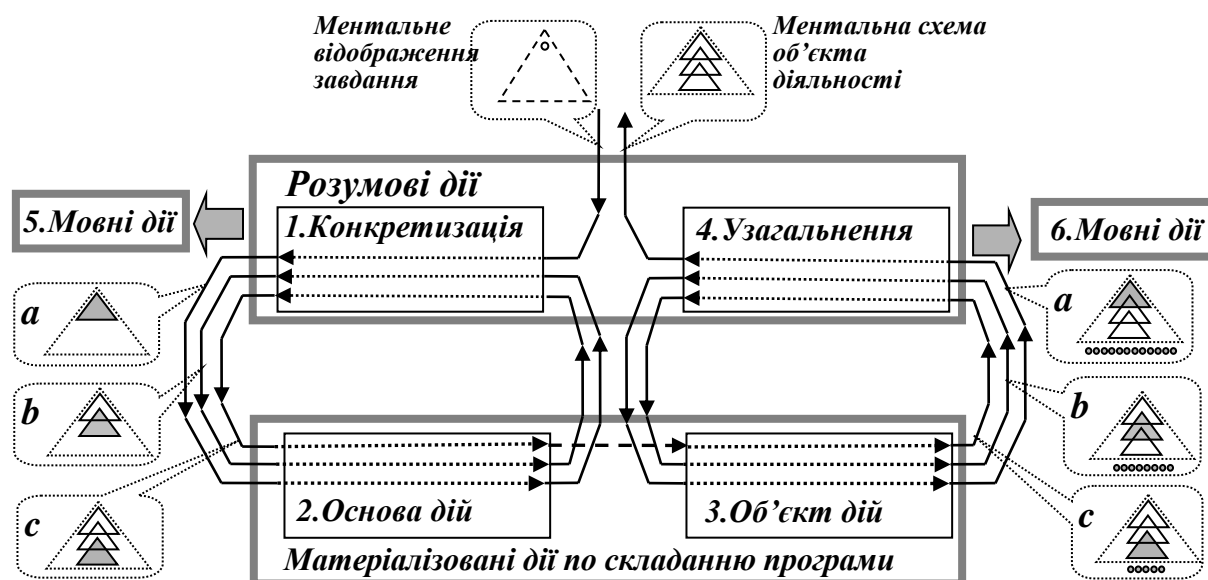


Рис.2 Розумові та матеріалізовані дії в контексті практичного програмування.

ментальних уявлень про майбутню програму у вигляді алгоритмів. Найбільш популярний формат формалізації - алгоритмічні блок-схеми. Програмування, як діяльності, має прикладний характер. Сфера застосування програми (фізика, математика, тощо) та проблема, яку ця програма вирішує, диктує необхідність матеріалізованої фіксації локальних проблемних ситуацій, які виникають при програмуванні. Фіксація цих проблем так само, як і алгоритмічних кроків їх розв'язання, може відбуватись у вербальному вигляді або у вигляді графічних схем. Можливі різноманітні формати.

Другий аспект матеріалізованих дій полягає у безпосередньому написанні програми на конкретній мові програмування та її налаштуванні.

На виносках рис.2 показана еволюція формування ментальної структури об'єкта та порядок написання тексту програми.

Ментальна структура об'єкта виконує роль ментальної основи матеріалізованих дій. Видно, що ментальне відображення об'єкта дій еволюціонує в процесі діяльності. Розумові та матеріалізовані дії об'єднуються спіралеподібними траєкторіями. Різні витки спіралі на рис.2 показують циклічні зв'язки розумових та матеріалізованих дій стосовно різних конструктів ментального відображення об'єкта та, відповідно, змістовних

фрагментів програми. Конструкт, на якому сконцентрована увага, позначений відповідним тонованим трикутником, що знаходиться у складі більш загальної ментальної структури. Конструкти, на яких увага не зосереджена, нетоновані. Кожний виток траєкторії відповідає за свій конструкт. Так, зовнішній виток відповідає за „верхній” конструкт *a* рис.1, а внутрішній - за „нижній” *c*. Виток в середині відповідає за „середній” конструкт *b*.

Конкретизація, за своєю внутрішньою суттю, є діяльністю по формуванню ментальної та матеріалізованої орієнтовної основи дій. У цьому сенсі розумові (1) і матеріалізовані (2) дії гармонійно поєднані. Конструкти формуються в порядку зменшення абстрактності *a, b, c*. Важливим методичним моментом, що впливає з моделі є те, що графічна формалізація кожного з конструктів, зокрема у вигляді алгоритмічної блок-схеми повинна включати обмежену кількість вузлів. Суміщення на блок-схемі двох, або трьох конструктів у повній деталізації може привести до перевантаження поля уваги і зменшення ефективності алгоритмічної схеми, як орієнтовної основи для матеріалізованих дій написання програми. Тому, конструкт *b* у конструкті *a* виглядає як недеталізований вузол. Конструкт *c* у конструкті *a* взагалі невидимий. Відповідно, при деталізації *b* конструкт *c* виглядає як вузол. Конструкт *a* - невидимий.

Дії, що розглядаються на рис.2, є взаємостимулюючими. Так розумова дія 1 з деталізації та конкретизації конструкта *a* дозволяє формалізувати вузли основи конструкта *a* (матеріалізувати) його (2) та побачити вузол *b*, як абстрактне монолітне утворення, яке на поточному етапі конкретизації ще не може бути реалізоване. Напроти, інші вузли основи конструкта *a* вже на поточному етапі можуть реалізуватись окремими операторами.

Деталізація конструкта *b* (1) і його формалізація (2) дозволяє побачити конструкт *c*, як окремий неподільний одиночний абстрактний вузол. Інші вузли основи конструкта *b* вже на поточному етапі можуть реалізуватись окремими операторами.

Деталізація вузла *c* (1) та представлення його у вигляді конструкта породжує відповідну матеріалізовану основу дій (2). Окремі елементи цієї основи можуть бути реалізовані надалі операторами мови програмування. Утворюється ланцюг зв'язаних дій формування орієнтовної основи. Цей ланцюг розумових і матеріалізованих дій не розривається, він продовжується спіраллю, що з'єднує матеріалізовані дії написання тексту програм (3) і розумові дії з узагальнення (4).

Треба відмітити, що хоч графічна формалізація орієнтовної основи матеріалізованих дій є зручною, однак цільовою формалізацією, є все ж таки мовна формалізація, адже програма пишеться мовою програмування.

Первинними при конкретизації (рух ментальною структурою вниз) є розумові аналітико-дедуктивні дії, що базуються на узагальненому інформаційному базисі стосовно згаданих вище локальних проблем і способів їх вирішення. Вторинними є матеріалізовані дії по формалізації кроків, що ведуть до вирішення проблеми.

На нижньому конструкті *c* відбувається перехід до зміни напрямку пересування ментальною схемою об'єкта. Цей перехід позначений жирною пунктирною стрілкою, що з'єднує пункт 2 і 3. З пункту 3 починається пересування ієрархічною ментальною схемою об'єкта вгору. Ситуація стосовно розумових дій радикально міняється.

Набір нового програмного фрагмента відбувається на основі уявлень, сформованих при конкретизації. Окремі оператори, достатньо жорстко обумовлені та формалізовані. Кожний оператор має свій зовнішній вигляд – формат та дію. Цілісний логічно завершений фрагмент програми є джерелом для подальших індуктивно-синтетичних дій з узагальнення. У цьому сенсі матеріалізовані дії при узагальненні первинні, а розумові індуктивно-синтетичні - вторинні. Ці розумові дії відбуваються стосовно образів щойно записаних операторів та локальних проблемних ситуацій, які ці оператори долають.

Узагальнення за своєю внутрішньою суттю є розумовою діяльністю, що має на меті створити основу для подальших ефективних практичних матеріалізованих дій. У

цьому сенсі матеріалізовані (3) та розумові (4) дії є гармонійно поєднані. Поточний ментальний образ об'єкта детермінує матеріалізовані дії. Як і в попередньому випадку, відбувається чергування матеріалізованих і розумових дій. На рис.2 ці дії показані стрілками, з'єднаними у спіралеподібну траєкторію, що багатократно проходить через пункти 3 та 4. Кожний виток траєкторії відповідає за свій конструкт. Увага пересувається від нижнього конструкта до верхнього, утворюючи послідовність *c, b, a*.

Дії, що розглядаються, є взаємостимулюючими. Суттєвим є те, що після успішної програмної реалізації *c*, внутрішній механізм дії *c* перестає бути актуальним у подальших розумових діях. У контексті реалізації *b* вузол *c* використовується як абстрактне, одиничне, монолітне утворення з притаманними йому функціями. Аналогічно, у контексті реалізації *a*, механізм реалізації *b* втрачає актуальність. Така технологія розумових дій дозволяє не перевантажувати поле уваги при узагальненні, та поступово укрупнювати масштаб бачення об'єкта.

Сеанси перенесення уваги та меж її концентрації у процесі практичного програмування можуть об'єктивно фіксуватись при спостереженні за навчальним процесом на екрані комп'ютера. Базується ця фіксація на відслідковуванні еволюції тексту програми у процесі її створення. Реперними подіями такої фіксації є сеанси перевірки логічно завершених фрагментів програми на дієздатність. Особливо зручними для дослідження меж уваги є такі теми програмування, як робота з файлами, текстами, масивами, кодування та шифрування. Тут джерелом додаткової інформації про сенс навчально-практичних дій у контексті переносу уваги є файл з результатами роботи фрагмента програми.

Надзвичайно важливою привабливою обставиною, що полегшує спостереження за процесами навчального програмування є те, що це спостереження може відбуватись і автоматизованим чином. Навчання відбувається у файловій системі комп'ютера, стан якої можна контролювати відповідними діагностичними програмами. Для цього на кожній обчислювальній машині у багатозадачному операційному середовищі Windows додатково до засобів програмування запускається відповідна сервісна діагностична програма.

Мовні коментарі стосовно поточної діяльності теж легко фіксуються. Однак, розшифровка відповідних реплік утруднена. Мовні конструкції контекстозалежні. Сенс повідомлень можна зрозуміти тільки здійснюючи спостереження за відповідною матеріалізованою діяльністю на екрані комп'ютера.

Змістова складова мовних висловів у процесі формування основи для матеріалізованих дій має характер поступової деталізації і конкретизації майбутнього механізму роботи програмного об'єкта.

У процесі матеріалізованих дій, пов'язаних з безпосереднім програмуванням, сутність мовних висловів теж структурується концентрацією уваги на відповідних конструктах. Однак, змістова складова сенсу висловів має зворотній характер. Вони направлені на концентрацію, абстрагування та узагальнення уявлень про об'єкт, що формується.

Після матеріалізованих дій з написання кожного з логічно закінчених фрагментів відбувається випробування. Вдалі результати випробування переводять написаний фрагмент у статус завершеного. Розміри цих фрагментів поступово збільшуються. Останнє випробування - це випробування найбільшого програмного модуля – всієї програми. Ці тенденції написання програми завжди спостерігаються при практичному програмуванні. У модельному процесі збільшення тексту програми можна бачити за видовженням рядка тонованих кружків, що позначають оператори. Цікавим є і характер формування тексту програми. Текст програми формується, як правило, не в тій послідовності, в якій записані оператори у тексті завершеної програми. Розглянемо цей момент на прикладі програми, приведеної на рис.1. З попереднього викладення зрозуміло, що спочатку формується фрагмент, відповідний „нижньому” в ієрархії конструкта *c*. Це середня частина тексту, що складається з п'яти рядків. На наступному етапі до написаних

рядків додаються ще три: один перед сформованим текстом, два -після. Новий текст відповідає конструкту *b*. На останньому етапі, що відповідає розумовим діям з конструктом *a*, в текст впроваджуються ще чотири рядки, два до і два після вже сформованого тексту.

Представлений вище модельний розгляд в цілому узгоджується з загальновизнаними базовими технологіями програмування, такими, як метод покрокової низхідної деталізації і метод модульного програмування [6, 7].

Окремі конструкти є тими ментальними логічно завершеними утвореннями, що розташовані на всіх рівнях ієрархії і на яких в порядку черговості локалізується увага. Вони є образами фрагментів програми, що піддаються перевірці на правильність роботи. Будемо вважати, що результат перевірки відомий після проходження пункту 4 (рис.2). Результат перевірки має синтетичну узагальнюючу природу і два варіанти значення: „Так” - результат перевірки позитивний, „Ні” – негативний.

Зрозуміло, що стосовно кожного поточного конструкта можливий, як позитивний рис.3*a* так і негативний рис.3*b*, 3*c*, 3*d*, 3*e*, 3*f*. результат перевірки дієздатності готового логічно закінченого фрагмента програмного об'єкта. При негативному результаті перевірки програмне утворення може або взагалі не працювати, або мати функції, відмінні від бажаних.

Якщо результат перевірки позитивний, то відбувається абстрагування від вузлів основи поточного конструкта, концентрація уваги на вузлах суміжного „верхнього” конструкта. Здійснюється перехід до діяльності з поточного конструкта до „верхнього” суміжного. Реалізується розімкнутий виток спіралі. У ситуації, коли поточний конструкт є найвищим у ієрархії, тобто він охоплює роботу всієї програми, відбувається вихід з

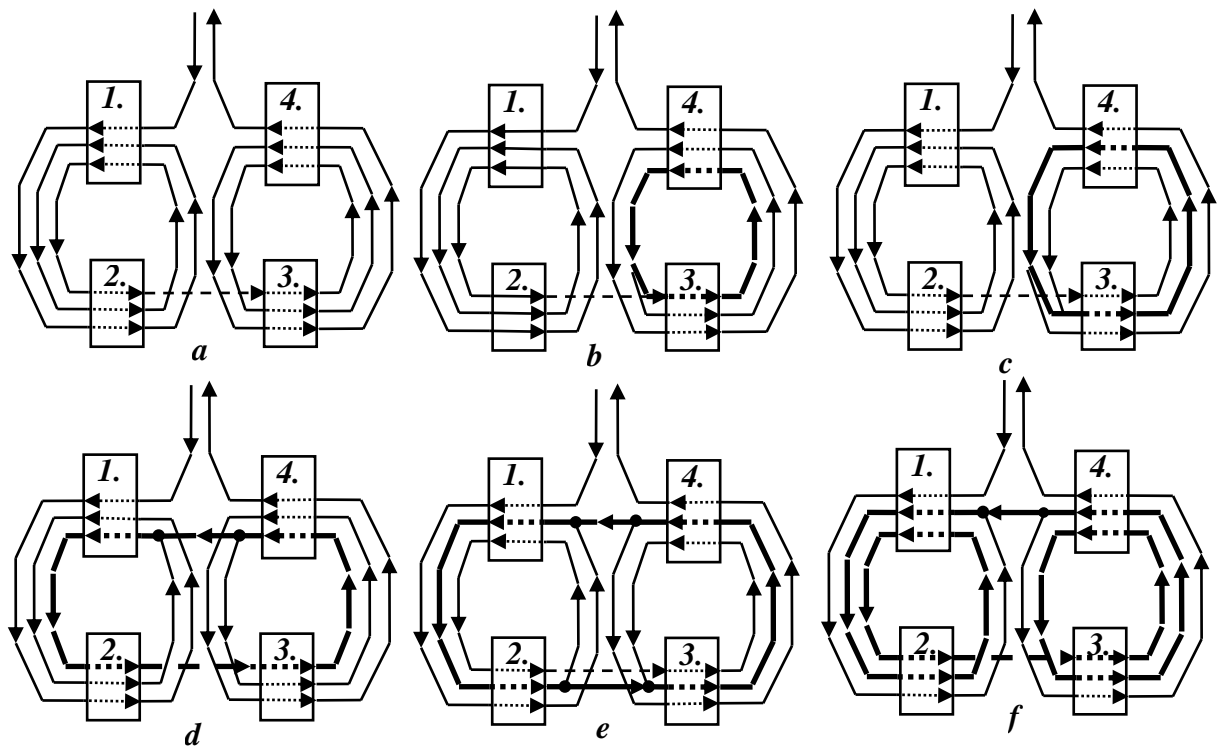


Рис.3 Розумові та матеріалізовані дії при формуванні програми, ментальне відображення якої включає три конструкти. *a* - дії без утруднень; *b*, *c* - утруднення долаються повторним підбором операторів; *d*, *e*, - утруднення долаються зміною орієнтовної основи дій та повторним підбором операторів; *f* – утруднення долаються змінами в постановці локальної проблеми, в орієнтовній основі дій та в підборі операторів.

діяльності. Робота над програмою успішно завершується.

Якщо результат перевірки негативний, то відбуваються повторні розумові та матеріалізовані дії стосовно поточного конструкта. При негативному результаті виток траєкторії розумових і матеріалізованих дій, відповідний поточному конструкту, замикається. Специфіка замикання може бути різною. У одних випадках (рис.3b, 3c) утруднення долаються тільки повторним підбором операторів без зміни орієнтовної основи дій. У інших випадках (рис.3d, 3e) відбувається зміна орієнтовної основи дій та повторний підбір операторів. У деяких випадках (рис.3f) приходиться вдаватися до змін у постановці локальної проблеми. Останнє тягне за собою, як зміни в орієнтовній основі дій, так і в підборі операторів.

На думку автора, модельні уявлення про психологічні процеси, представлені в роботі можуть стати в нагоді при розробці нових методичних підходів до викладання програмування та при автоматизації цього процесу за допомогою навчаючих тренажерних програм.

### *Література*

1. Холодная М.А. Психология интеллекта: парадоксы исследования. — СПб.: Питер, 2002. - 272с.
2. Величковский Б.М. Когнитивная наука: Основы психологии познания: в 2 т. — Т.1 — М.: Смысл: Издательский центр «Академия», 2006. — 448 с.
3. Головін М.Б. Специфіка формування пояснень до прикладів-аплікацій у навчальному матеріалі з програмування // Проблеми сучасного підручника. Збірник наукових праць. (присвячується 15-річчю АПН України.) Випуск 7. Київ-Луцьк, 2007 . –С. 339-350. <http://esnuir.eenu.edu.ua/bitstream/123456789/11575/1/formuvannja%20pojasnen.pdf>
4. Головін М.Б. Кількість і складність розумових дій у контексті діагностики когнітивних процесів, що детермінують практику навчального програмування // Вісник Черкаського університету. Серія педагогічні науки. Випуск 125. Черкаси, 2008. – С. 34 - 41. [http://www.nbu.gov.ua/old\\_jrn/Soc\\_Gum/Vchu/N125/N125p034-040.pdf](http://www.nbu.gov.ua/old_jrn/Soc_Gum/Vchu/N125/N125p034-040.pdf)
5. Головін М.Б. Зміст підготовки висококваліфікованого фахівця з інформаційних комп'ютерних технологій у контексті когнітивних процесів (на прикладі програмування) // Інформаційні технології в освіті. Випуск 2. Херсон, 2008. – С. 66-73. <http://ite.kspu.edu/issue-2/p-66-73>
6. Лингер Р., Миллс Х., Уитт Б. Теория и практика структурного программирования. Москва.: «Мир», 1982 – 402 с.
7. Соммервилл Иан. Инженерия программного обеспечения, 6-е издание. : Пер. с англ. — М.: Издательский дом "Вильямс", 2002. — 624 с.

### *Анотація на роботу:*

*Розглянуто навчальне програмування в контексті структурної організації ментального відображення об'єктів цієї діяльності.*

*Educational programming in the context of structural organization of this activity objects' mental reflection are considered.*

*Ключові слова : понятійні структури, когнітивні процеси, модель навчання, conceptual structures, cognitive processes, model of education.*