

СПЕЦИФІКА ФОРМУВАННЯ ПОЯСНЕНЬ ДО ПРИКЛАДІВ- АПЛІКАЦІЙ У НАВЧАЛЬНОМУ МАТЕРІАЛІ З ПРОГРАМУВАННЯ

*М.Б.Головін, канд. фіз.-мат. наук
Волинський національний університет імені Лесі Українки*

Практичне програмування в навчальній діяльності супроводжується складним, специфічним пізнавальним процесом. Ефективна методична підтримка цієї діяльності з боку викладача, підручника, навчальної комп'ютерної програми може бути реалізована на основі розуміння особливостей психологічних процесів суб'єкта навчання. Метою цієї роботи є концептуальний розгляд розумових дій, що супроводжують комп'ютерне програмування; формулювання на цій основі правил побудови пояснень до аплікаційних прикладів.

Необхідно відмітити, що програмування за своєю специфікою дуже схоже на діяльність стосовно інших складних об'єктів, зокрема, таких як електронні пристрої, комп'ютерні мережі. Інтелектуальні дії спрямовані на складні, добре структуровані об'єкти великого розміру набувають більшого контрасту, яскравості, проявляються закономірності розумових процесів. Об'єкти низької складності, аморфної структури, невеликого розміру завуальовують відповідні особливості. Тому дослідження навчальної діяльності стосовно об'єктів високих технологій є цікавими з точки зору теорії психології та педагогіки.

Пізнавальні проблеми освоєння механізмів роботи складного утворення полягають у тому, що людина не може утримувати в колі уваги та оперувати одночасно всіма багаточисельними функціональними вузлами складного об'єкта. Суб'єкт когнітивних процесів може сприйняти і розрізнити значно більше вузлів, ніж він може зв'язати і запам'ятати. Для зв'язування цих елементарних вузлів об'єкта людині необхідно переводити увагу і при цьому тривалий час формувати структурні уявлення про об'єкт у довготривалій пам'яті. Структура, що формується, має ієрархічну будову і є логічним каркасом уявлень про об'єкт. Таке відображення об'єкта в пам'яті дозволяє

здійснювати абстрактно-логічні переходи структурою об'єкта. Уявлення в процесі роботи з аплікацією поступово диференціюються [1]. Побудова цієї структури приводить до розуміння програми [2]. Протоколи роботи зі складними об'єктами розглянуті в [3, 4].

Ускладнення об'єктів практичної діяльності (програм), що розглядаються в процесі вивчення програмування приводить до того, що діяльність втрачає ознаки спорадичності, дифузності, спонтанності. Дії стають більш системними, узгодженими, послідовними, усвідомленими і тому вони легше піддаються формалізації. Коректність керування процесом навчання з боку викладача може бути підвищена на основі усвідомлення поглибленої деталізації та формалізації дій суб'єкта навчання. Чим більше нюансів у процесі враховуємо, тим точніше керуємо. Розглянемо логіку розумових дій, на прикладі діяльності суб'єкта стосовно типової навчальної програми.

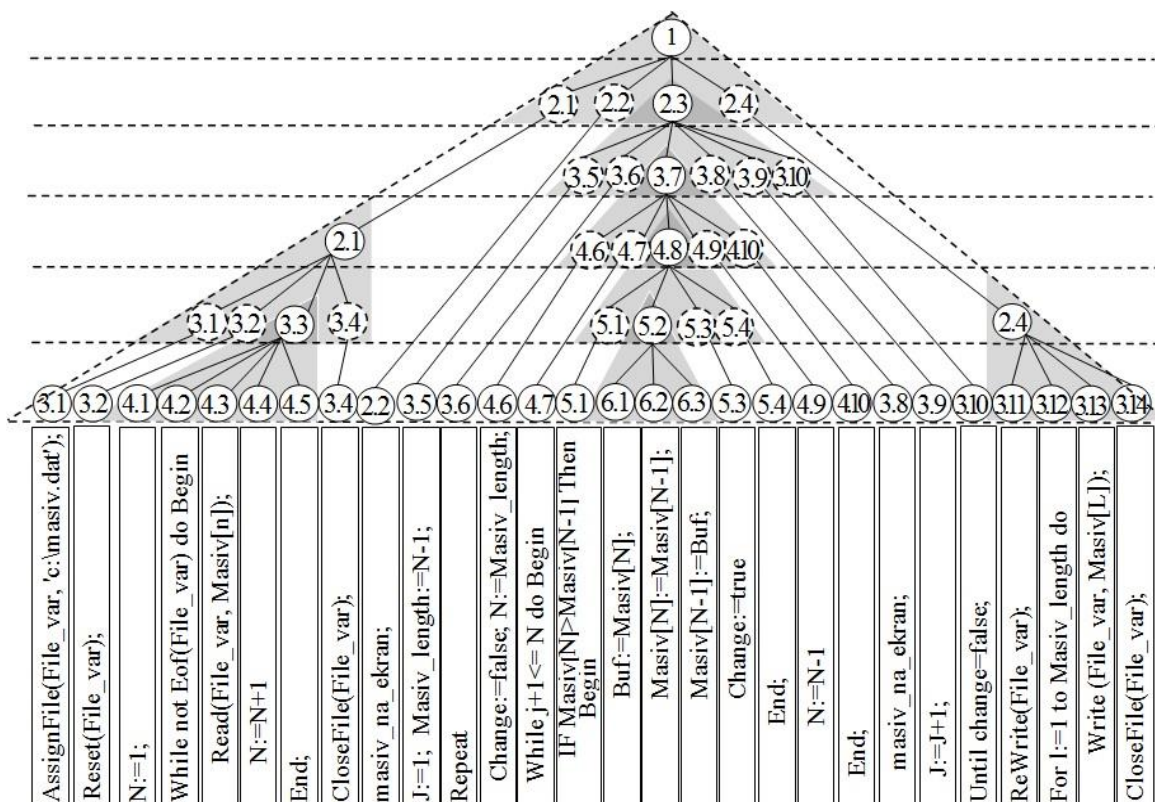
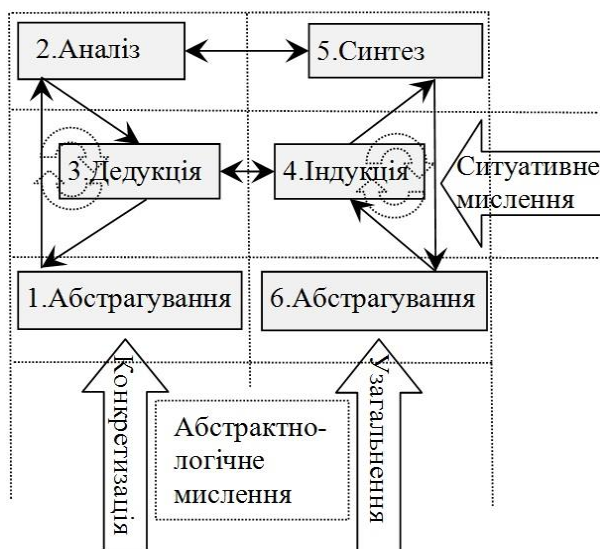


Рис. 1. Ієрархічна структура типового навчального об'єкта та текст програми, відповідний йому

Структура об'єкта, показана на рис.1, є платформою для теоретичного розгляду процесів розумових дій супутніх виконанню завдання. Це структура досить простої навчальної програми. Суцільними кружечками, що складають

основу великого трикутника, позначені реальні неподільні елементи об'єкта - оператори програми. У вершині цього трикутника знаходиться об'єднуючий вузол. Він позначає функції об'єкта як цілісного утворення. У межах великого трикутника є тоновані трикутники меншого розміру. Ці трикутники окреслюють межі зосередження уваги. Вузли основи цих трикутників є складовими частинами вузла у вершині. Накладання трикутників через спільні вузли зв'язує окремі сеанси зосередження уваги. Реалізується логічний зв'язок у діях при переводі уваги.

Розумова діяльність у термінах структури, зображеної на рис.1, може просуватись у вертикальному і горизонтальному напрямках, що відповідає абстрактно-логічному і ситуативному мисленню. Рух ієрархією вниз пов'язаний з конкретизацією, а рух вверх - з узагальненням. Багатокрокові



розумові логічні дії в межах одного рівня ієрархії, в трикутнику зосередження асоціюються в цьому представленні з індукцією або дедукцією. Це ситуативні перетворення які є підготовчими відповідно для абстрактно-логічного руху вверх або вниз.

Специфіка пересувань

Рис.2. Порядок абстрактно-логічних та ситуативних дій стосовно об'єкта навчання структурою, що зображена на рис.1,

може бути пояснена схемою, зображеною на рис.2.

Наприклад, пересування вниз від трикутника зосередження уваги 1, 2.1-2.4 до трикутника 2.3, 3.5 – 3.10 починається з абстрагування від вузлів 2.1, 2.2, 2.4 та концентрації уваги на 2.3. Далі 2.3 диференціюється і може бути представлена діями 3.5, 3.6, 3.7, 3.8, 3.9, 3.10. Таке розчленування є результатом аналізу. Дедукція з'ясовує роль кожного з кроків 3.5-3.10 у контексті призначення кроку 2.3.

Подальший рух структурою вниз аналогічний розглянутому вище. Відбувається перевід уваги з трикутника зосередження 2.3, 3.5-3.10 на трикутник 3.7,4.6-4.10. Він починається з концентрації уваги на вузлі 3.7, абстрагування від всіх інших вузлів поточного трикутника зосередження уваги. Далі аналіз вузла 3.7, дедуктивне формування уявлення про вузли 4.6 – 4.10. Спрацьовує розумовий цикл абстрагування-аналіз-дедукція (1-2-3 рис.2.), що супроводжує транзит структурою вниз (рис.1).

Аналіз і синтез у мислені гармонійно поєднані. Дедуктивний ланцюг розумових дій може перейти в індуктивний. Тому цикл індукція-синтез-абстрагування (4-5-6 рис.2) періодично спрацьовує, як перевірочний для циклу 1-2-3. Домінантними діями при конкретизації є дії 1-2-3 (рис.2).

При узагальненні, тобто при пересуванні структурою вгору (рис.1), кроки циклу 4-5-6 (рис.2) домінують над кроками 1-2-3. Наприклад, індуктивне зв'язування вузлів 6.1, 6.2, 6.3 приводить до формування уявлення про цей фрагмент, як цілісне утворення. Формується синтетичний віртуальний функціональний вузол 5.2. Далі абстрагування від 6.1-6.3, перевід уваги на вузли 5.1-5.4 та індуктивне зв'язування цих вузлів. Утворення про них цілісного синтетичного уявлення 4.8. Абстрагування від вузлів 5.1-5.4. Концентрація уваги на вузлах 4.6-4.10.

Розглянемо в якості базисної аплікації програму сортування. На рис. 1 зображений текст цієї програми (основа рисунку). Структурна надбудова над текстом є когнітивною схемою, яку змушений формувати суб'єкт навчання в процесі освоєння програми. Ця структура логічно зв'язує в єдине ціле численні оператори програми. Схема на рис.1 є зображенням графічно формалізованого відображення програми у пам'яті суб'єкта навчання після її освоєння. Процес освоєння відбувається поступово в реальному часі. Цей процес розглянемо нижче.

Для визначеності опишемо позначення, використані в програмі. `Masiv` - назва масиву для зберігання чисел, завантажених з файлу `c:\masiv.dat`. `Masiv_length` - довжина масиву. Змінна `Vuf` призначена для обміну значень

між комірками масиву. Змінні N, J, L забезпечують роботу рахівників у циклах. У змінній Change логічного типу фіксується інформація про факт обміну значеннями в масиві. Masiv_na_ekran – процедура виводу сукупності чисел з масиву на екран.

Завдання для аплікаційної програми може бути сформульоване коротко і конкретно наступним чином. Відсортувати числа в файлі c:\masiv.dat. Позначимо цю проблему одиницею рис.1.

Проблема сортування 1 диференціюється і представляється сукупністю проблем: 2.1 - прочитати випадкову послідовність чисел з файлу в масив; 2.3 - сортувати числа в масиві в порядку зростання. 2.4 - записати відсортовані дані в файл. Кожна наступна проблема може бути вирішена тільки після вирішення попередньої. Тому загальна стратегія переведення уваги має вигляд, зображений рис.3. Дія 2.2 - вивести масив на екран, є допоміжною, необов'язковою.

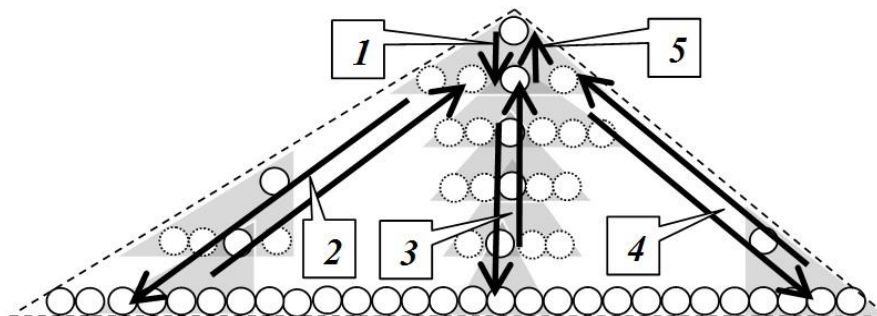


Рис. 3. Стратегічні напрями пересування уваги при освоєнні, реалізації і випробовуванні аплікаційної програми сортування. Номери біля стрілок показують послідовність дій

Розумові дії стосовно читання файлу в масив мають наступний порядок. Тимчасове абстрагування від проблем 2.2, 2.3, 2.4 (рис.1). Концентрація уваги на проблемі 2.1. Диференціація 2.1 формує: 3.1 - співставлення файлу з файловою змінною; 3.2 - відкриття файлу для читання; 3.3 - заповнення масиву; 3.4 - закриття файлу. Мова програмування має засоби для реалізації кроків: 3.1; 3.2; 3.4. Однак, крок 3.3 – заповнення масиву, вимагає деталізації. Тому далі відбувається абстрагування від 3.1; 3.2; 3.4, концентрація уваги на проблемі 3.3. Деталізація 3.3 формує кроки: 4.1 – надання рахівнику індексів масиву початкового значення; 4.2 – початок

циклічного повтору дій 4.3 і 4.4 (тривають до закінчення читання файлу); 4.3 - читання файлу в комірку масиву; 4.4 – зміна рахівника циклів; 4.5 - кінець циклу. Всі дії можуть бути реалізовані засобами мови програмування. Спадна деталізація, що має на меті сформувати когнітивну схему завантаження файлу закінчується. Починається реалізація фрагменту програми, що завантажує файл.

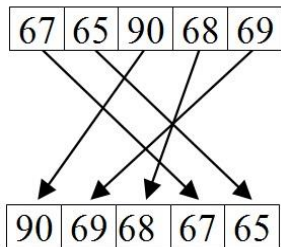


Рис. 4. Проблемний - цільовий стани для кроку 2.3.

Реалізація програми відбувається впродовж висхідного, послідовного укрупнення фрагментів програми. Використовується щойно побудована когнітивна структура. Ця робота закінчується можливістю прочитати файл у масив, тобто реалізацією 2.1. Далі можливою для розв’язання

стає проблема 2.3 - сортування.

Аналіз проблеми сортування 2.3 починається з графічної її формалізації (рис.4). Початковий проблемний стан - це стан несортованого масиву - верхній рядок. Цільовий стан - нижній рядок. Мета полягає в переході від початкового проблемного стану до цільового. З’єднання стрілками відповідних елементів формує уявлення про характер перестановок. Крок 2.3

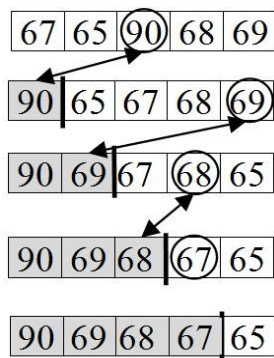


Рис.5. Кожні два суміжні рядки - це проблемний і цільовий стани для кроку 3.7.

реалізований стрілками.

Таке зображення стрілками характеру переміщення елементів масиву є графічною формалізацією діяльності по сортуванню в узагальнених рисах. Однокрокового засобу вирішити проблему у мові програмування нема.

Тому ця проблема диференціюється.

На рис.5 показані проміжні стани вирішення проблеми 2.3. Об’єктами розумових дій стають:

несортована та відсортована область масиву, границя між цими областями, гранична комірка масиву і комірка з максимальним значенням у несортованої області. Кожні два поруч розташовані рядки - це відповідно локальний

проблемний і цільовий стан. Перехід з проблемного в цільовий стан відбувається перенесенням максимального (3.7 рис.1) і зсувом границі (3.9 рис.1). Дії повторюються поки не буде досягнуто стан відсортованості, тоді обміни припиняться (change=false). Мова програмування реалізує такі дії циклом Repeat Until (3.6, 3.10 рис.1).

Диференціювання дії 2.3 в дії 3.5, - 3.10 (рис.1) стосовно вище згаданих об'єктів відбувається впродовж аналізу, дедукції, абстрагування. Дедуктивна робота полягає в формуванні уявлень про роль кожного з кроків 3.5 -3.10 рис.1, як складової частини 2.3. Ключовий крок 3.7 не може бути реалізований одним кроком програми його необхідно диференціювати і конкретизувати.

Графічна деталізація проблеми 3.7, і її розв'язок зображені на рис. 7. Це розв'язок проблеми методом „бульбашки”.

Тепер розглядаються зміни тільки в несортованій області. Від подій у

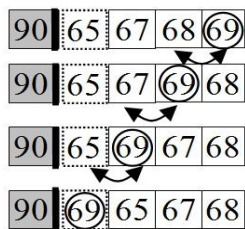


Рис. 6. Кожні два суміжні рядки - це проблемний і цільовий стани для кроку 4.8

сортованій області можна абстрагуватись. Залишається актуальним об'єктом розумових дій гранична комірка. Поняття комірки з максимальним значенням тут актуальне не в сенсі всього масиву, а в сенсі порівняння значень двох розташованих поруч комірок. Поточна пара комірок обміну є новими двома об'єктами

концентрації уваги. Максимальне шукається і транспортується в граничну комірку за рахунок обмінів 4.8 у парах поруч розташованих комірок, як показано на рис.6. Транспортування реалізується зміною координат поточної пари обміну 4.9.

Описані дії повторюються в процесі транспортування. Мова програмування реалізує такі дії циклом (4.7, 4.10 рис.1).

Аналітико-дедуктивна робота формує уявлення про роль кожного з кроків 4.6-4.10, як складових дії 3.7.

Абстрагування від кроків 4.6, 4.7, 4.9, 4.10 концентрація на 4.8 та диференціація цього кроку дозволяє виділити: порівняння значень в

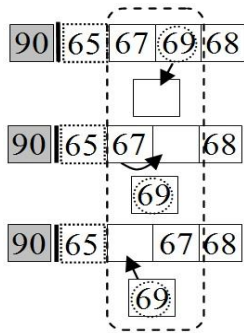


Рис. 7. Кроки 6.1, 6.2, 6.3. позначені стрілками

суміжних комірках масиву та виділення більшого 5.1, 5.4; запуск процесу обміну 5.2, якщо він потрібний. (5.3- фіксація факту обміну.)

Абстрагування від частини кроків і диференціація 5.2 приводить до формування уявлення про реалізацію обміну значеннями між

комірками пам'яті 6.1, 6.2, 6.3 через новий об'єкт розумових дій - буферну комірку. Ці кроки реалізуються засобами мови програмування. Рух ієрархією вниз закінчується. На цьому етапі когнітивна схема сортування набуває стану часткової сформованості (рис.8). Розглянемо цей стан.

Суцільними кружечками позначені транзитні вузли роль яких

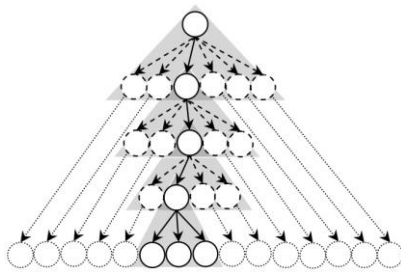


Рис.8. Пересування ієрархією вниз шляхом 3 (рис.3.)

усвідомлюється, суцільною стрілкою позначається перехід уваги від загального до конкретного через проміжні транзитні вузли. Переривчастою лінією позначені вузли від яких у процесі розумових дій відбувалось абстрагування, але які брались до уваги в процесі

дедуктивних дій у відповідних трикутниках зосередження уваги. Перед абстрагуванням вузли включались у структуру (схему) відображення об'єкта у пам'яті. Уявлення про ці вузли є орієнтовним, дифузним, їх набір може

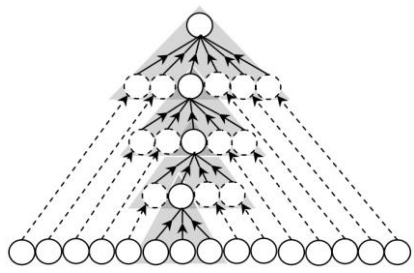


Рис. 9. Пересування ієрархією вверх по шляху 3 (рис.3.)

бути не повним. Їх проекція на рівень реалізації позначена кружечками, що окреслюються крапками.

Процес реального написання програми супроводжує пересування ієрархією вверх.

Пересування структурою вверх є поступовим рис.9. Увага концентрується на нижньому трикутнику і через

проміжні трикутники переходить на верхній. При поступовому пересуванні ієрархією відбуваються наступні важливі зміни. Вузли реалізації об'єкта, уявлення про які було дифузне, слабо визначене, або зовсім відсутнє, поступово проявляються та конкретизуються у вигляді тексту програми.

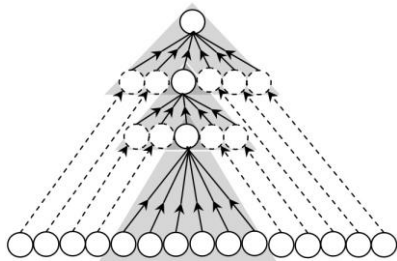


Рис. 10. Пересування ієрархією вверх альтернативний варіант.

варіант показаний на рис. 10.

При порівнянні рис. 9 і рис. 10 видно, що два нижніх трикутники зосередження уваги об'єдналися в один. Спостереження за коментарями студентів впродовж роботи з програмами свідчить, що трикутники зосередження уваги в процесі роботи з програмою можуть укрупнюватись. Так опускання ієрархією вниз і є можливим по набору трикутників, зображених на рис.8, а підйом - схемою, зображеною на рис.10. Необхідно відмітити, що досвідчені в програмуванні студенти схильні пересуватись структурою, як вверх так і вниз укрупненими трикутниками зосередження уваги. Вони бачать стандартні часто вживані програмні фрагменти цілісно укрупненими логічно закінченими блоками – паттернами „візерунками” (pattern англ.) операторів. Звичайні студенти сприймають програму окремими операторами. (Застосування паттернів у шахових задачах розглянуто в роботі [5].)

При пересуванні структурою вниз первинною є деталізація проблеми, що вирішується. Проблема представляється сукупністю все більш детальних і конкретних зв'язаних локальних проблем. Окремі кроки з вирішення локальних проблем довгий час залишаються узагальненими, дифузними, погано конкретизованими. Так, наприклад, при пересуванні структурою вниз відмічалась необхідність використання циклів. Однак параметри цих циклів

легше написати при сходженні ієрархією вгору. Деякі кроки на цьому етапі взагалі не проглядаються.

Мета цієї поступової деталізації в тому, щоб представити проблему в вигляді сукупності окремих локальних проблем, які можуть бути вирішені засобами мови програмування. Крім деталізації, важливим моментом у цьому процесі є спосіб формалізації цих окремих локальних проблем. На початкових фазах процесу - це може бути графічний або вербальний спосіб, на кінцевих - математичний. Необхідно ще раз нагадати, що увага сконцентрована завжди не на всьому об'єкті, а лише на його частині. Чим нижче суб'єкт мислення опускається структурою вниз, тим «вужчу» частину загальної проблеми він «бачить». У колі його уваги знаходиться завжди обмежена кількість вузлів об'єкта та їх зв'язків. Пересування уваги після сеансу розумової дії супроводжується ротацією об'єктів зосередження. Це стосується як руху структурою вниз, так і руху вгору.

При досягненні рівня реалізації, коли локальні проблеми і кроки, що ці проблеми вирішують максимально конкретизуються настає фаза пересування структурою вгору. При пересуванні вниз домінуючими об'єктами зосередження уваги були об'єкти проблеми. Тепер домінуючими об'єктами розумових дій стають кроки, що вирішують конкретизовані локальні проблеми. При пересуванні вгору кількість об'єктів у колі уваги збільшується. Так рух вгору у сортуванні починається з трьох кроків та трьох локальних проблемних ситуацій. Закінчується - узгодженням вже п'ятнадцяти кроків і великої кількості можливих проблемних ситуацій.

Це приводить до необхідності зменшення кількості об'єктів зосередження уваги при сходженні структурою вгору. Це зменшення відбувається поетапно в режимі узагальнення і абстрагування у кожній парі суміжних трикутників.

Висновки.

Повна орієнтуюча основа для дій, стосовно аплікаційної програми, формується у відповідності до когнітивних процесів, що супроводжують ці

дії. Орієнтуюча основа повинна відображати всі стратегічні шляхи, якими відбуваються транзити уваги вниз і вгору структурою програми.

Розумова структура, що зв'язує окремі оператори програми, повинна формуватись у правильному порядку. Спочатку зверху вниз визначеним стратегічним напрямком, а потім знизу вгору. Рух вниз у термінах деталізації проблеми, диференціації і конкретизації кроків, якими ці проблеми розв'язуються. При русі вниз ставляться проблеми і формуються шляхи їх вирішення. Рух вгору повинен відбутись у термінах мови програмування, в режимі еволюційного пояснення шляхів, якими проблеми (поставлені при русі вниз) будуть вирішені. При русі уваги структурою вгору відбувається поступове укрупнення програмних модулів. Це укрупнення повинно супроводжуватись періодичним випробовуванням дієздатності програми. Пояснення до аплікації повинні вказати ключові моменти, коли такі випробовування доречні.

Пояснення до програмних аплікацій не повинно бути дифузним. Пояснення повинно мати модульний характер. Локальна проблема відповідає окремому модулю. Проблема повинна бути правильно (лаконічно і повно) сформульована. Для кожного проблемного стану повинна бути сформульована відповідна цільова ситуація. Ланцюги проблем та їх вирішень повинні бути логічно, без розривів, зв'язані.

Розмір окремих модулів, розтлумачення механізму роботи програми, не повинен бути перевантаженим об'єктами і зв'язками, на яких треба зосереджувати увагу. Це стосується, як кількості об'єктів проблемних і цільових ситуацій, так і кількості об'єктів - засобів вирішення проблем. Зрозуміло, якщо пояснення орієнтоване на експертів, а не на ординарних студентів, блоки зосередження уваги можуть бути більшими, а відповідно їх кількість меншою. Механізми роботи стандартних фрагментів програми в таких випадках не розтлумачуються, вони розглядаються цілісно паттернами (фрагмент програми – його функція).

Логіка викладання не повинна бути плутаною. Вона повинна бути формалізована у відповідності з логікою циклів абстрагування-аналіз-дедукція і відповідно індукція-синтез-абстрагування з можливими переходами між цими циклами.

Література

1. Чуприкова Н.И. Психология умственного развития: Принцип дифференциации. –М.: Столетие, 1997. – 478 с.
2. Головін М.Б. Структура аплікаційної комп'ютерної програми та проблеми її розуміння // Педагогічний пошук. Вип. 4. – Луцьк, 2006. – С. 18.
3. Головін М.Б. Модельний аналіз психолого-педагогічних особливостей діяльності з освоєння та створення об'єктів високої складності // Педагогічний пошук. Вип. 2(54). – Луцьк, 2007. – С. 60-63.
4. Головін М.Б. Психолого - педагогічна модель практичної діяльності студента по освоєнню та створенню об'єктів високої складності // Проблеми педагогічних технологій, Випуск 2-4 (31-33). Луцьк, 2006. –С. 314-325.
5. Джон Р. Андерсен Когнитивная психология. СПб.: Питер, 2002. 492 с.