

## АКТИВІЗАЦІЯ НАВЧАЛЬНО-ПІЗНАВАЛЬНОЇ ДІЯЛЬНОСТІ ПРИ ВИВЧЕННІ ПРОГРАМУВАННЯ З ВИКОРИСТАННЯМ МОДЕЛЬНО-СИМВОЛІЧНОЇ ТЕХНОЛОГІЇ

В.С. Єремєєв, С.М. Прийма  
м. Мелітополь, Мелітопольський державний педагогічний університет

Реалізація основних положень Національної доктрини розвитку освіти України у XXI столітті вимагає оновлення освітньої парадигми, впровадження нових концепцій навчання, пошуку шляхів вдосконалення педагогічного процесу та посилення його технологічного аспекту. Особливої актуальності набувають вказані заходи у процесі вивчення такої фундаментальної науки як інформатика.

Дана публікація присвячена розробці основних принципів активізації самостійної навчально-пізнавальної діяльності студентів при вивченні інформатики, зокрема такого її розділу як основи алгоритмізації та програмування, з використанням модельно-символічної технології організації розвивального навчання.

В роботах автора даної педагогічної технології П.О. Барабохи показано ефективність та доцільність її використання у процесі вивчення дисциплін природничого циклу [1, 2].

Предметом нашого дослідження виступають шляхи активізації навчально-пізнавальної діяльності студентів вищого навчального закладу у процесі вивчення програмування на основі евристично-пошукової моделі, що реалізована використанням модельно-символічної технології.

Метою дослідження є обґрунтування ефективності та доцільності використання модельно-символічної технології у процесі вивчення курсу програмування.

Виходячи з мети дослідження, було сформульовано наступні завдання:

- розробити логіко-теоретичну основу модельно-символічної технології;

- виконати класифікацію проблемної символіки;
- розробити методичні рекомендації до організації лабораторних занять при вивченні курсу “Візуальне програмування”.

Система понять курсу програмування, що формується на фундаментальних основах інформатики, дозволяє краще відобразити глибинні причинно-наслідкові зв'язки, фундаментальні категорії дисципліни. Адже, будь-яке знання виражається в поняттях, категоріях, принципах, законах, закономірностях, символах, концепціях, теоріях. Відповідно, вільне володіння основними змістовними одиницями, які виступають орієнтиром для всього процесу вивчення курсу програмування, сприяє формуванню не тільки конкретних предметних знань, а й розвитку теоретичного, діалектичного мислення, що дозволить майбутнім фахівцям самостійно адаптуватися до швидкої зміни навколишнього інформаційного середовища.

Формуючи певне поняття предметного матеріалу, слід починати з такої логічної операції як поділ. Саме операція логічного поділу дозволяє розкрити обсяг поняття, розподіляючи його на види за певною ознакою. Чітке розуміння структури операції поділу, що включає ділене поняття, членів поділу та підставу, дозволяє уникнути помилок при виконанні даної операції [3, с. 154-155]. Поділ поняття на ділене поняття (рід) та види на підставі специфічного вияву ознаки в різних видах діленого називається поділом за видозмінюваною ознакою. Більш простішим діленням є дихотомічний поділ, за допомогою якого ділене поняття розподіляють на два суперечливі поняття. На практиці найчастіше використовують поділ за видозмінюваною ознакою, а дихотомічний поділ, за звичай, застосовують на початковій стадії вивчення поняття, коли є ясність відносно частини предметів, що обмежені діленим поняттям [4, с. 100].

Слід зауважити, що використовуючи будь-який з видів логічного поділу, необхідно дотримуватися певних правил, а саме:

- поділ поняття слід здійснювати за однією підставою;

- поділ повинен бути відповідним;
- члени поділу повинні виключати один одного;
- поділ повинен бути послідовним.

Отже, визначивши обсяг певного поняття, більш детально розглянемо членів його поділу, що надалі дозволять зрозуміти внутрішні взаємозв'язки та сутність поняття. Для цього скористаємося логічними відношеннями між поняттями.

Поняття, які мають спільну родову ознаку або спільне родово поняття називають порівнюваними [3, с. 142]. Наприклад, процедурне програмування – об'єктно-орієнтоване програмування, процедура – функція, цикл з передумовою – цикл з післяумовою, константа – змінна, інкапсуляція – наслідування, підпрограма – функція, сортування методом обміну – сортування методом вибору мінімального елемента. Якщо ж поняття не мають спільного родового поняття, то їх називають непорівнюваними. Наприклад, наукова дисципліна “інформатика” – мова програмування, комп'ютер – програма, об'єкт – об'єктно-орієнтоване програмування. Слід зауважити, що коли ми говоримо про непорівнюваність понять, то в цьому є доля умовності. Адже, визначаючи, що непорівнювані поняття не мають спільних родових ознак, ми вже порівнюємо їх за змістом. Окрім цього, при порівнянні можна завжди використати такий широкий рід як “предмет” чи “явище” чи “дещо”, який буде включати в себе об'єми будь-якого поняття, а вони по відношенню до нього будуть порівнюваними. Ось чому, говорячи про порівнювані та непорівнювані поняття, слід мати на увазі не потенційну можливість відшукування загального роду, а фактичну наявність в змісті понять спільних ознак, що безпосередньо визначають характеристики даного роду та дають підставу для ділення [4, с. 82].

Порівнювані поняття поділяють в свою чергу на сумісні і несумісні.

Сумісними називають поняття, видові ознаки яких забезпечують повне або часткове збігання їх обсягів.

Наприклад, програмування – об'єктно-орієнтоване програмування, підпрограма – процедура, цикл – цикл з умовою, цикл з умовою – цикл з післяумовою, програмне забезпечення – прикладне програмне забезпечення, функція – підпрограма, що

повертає значення результату через змінну, ім'я якої співпадає з її іменем.

Слід зазначити, що незважаючи на три види відношень (відношення тотожності, підпорядкування та часткового збігання), практичну цінність в умовах впровадження модельно-символічної технології, на нашу думку, має лише відношення підпорядкування. Дане відношення фіксує зіставлення родового і видового поняття. Наприклад, програмування – об'єктно-орієнтоване програмування, підпрограма – процедура, цикл з умовою – цикл з післяумовою. В даному випадку поняття, яке входить до обсягу іншого поняття, називають “підпорядкованим”, а поняття, яке вміщує у свій обсяг інше поняття, називається “підпорядковуючим” [3, с. 144].

Більший потенціал в межах модельно-символічної технології мають несумісні поняття, видові ознаки яких обумовлюють повне незбігання їх обсягів.

Несумісні поняття можуть знаходитися у трьох відношеннях:

- суперечності;
- протилежності;
- супідрядності [3, с. 145].

У відношенні суперечності знаходяться поняття, зміст яких заперечує один одного, а сума їх обсягів вичерпує обсяг родового поняття [3, с. 145]. Наприклад, значення логічної змінної true – значення логічної змінної false, процедура – функція, цикл з передумовою – цикл з післяумовою.

Протилежними називають поняття, зміст яких різниться у вищому ступені. Це означає, що вони виражають деякі протилежні характеристики у певному упорядкованому переліку властивостей, які постійно змінюються, і у свої сумі не вичерпують обсяг родового поняття. Наприклад, початкове значення лічильника – кінцеве значення лічильника.

Коли видові поняття одного роду не знаходяться ні у відношенні суперечності, ні у відношенні протилежності, а сума їх об'ємів становить лише частину об'єму родового поняття, то їм притаманне відношення супідрядності [3, с. 146]. Наприклад, цикл з передумовою – цикл з параметром, інкапсуляція – наслідування. Слід зауважити, що для відношення супідрядності

необхідна наявність більш загального, родового, поняття [4, с. 184].

Таким чином, знання логічних відношень між поняттями дозволяє краще збагнути зміст понять та допомагає визначити перелік операцій мислення, які можна виконати над поняттями, і, відповідно, уникнути помилок у побудові проблемно-символічних сигналів, що лежать в основі модельно-символічної технології.

Так, наприклад, до непорівнюваних понять можна застосувати команди впливу, взаємовпливу, взаємодії та встановлення причинно-наслідкового зв'язку. До порівнюваних понять, що знаходяться у відношенні підпорядкування, можна застосувати тільки команду встановлення причинно-наслідкового зв'язку. Побудова проблемно-диференційованого завдання, в основі якого знаходиться пара понять у відношенні підпорядкування, буде некоректною, якщо до нього застосувати команду порівняння (незважаючи на те, що ці поняття є порівнюваними).

Всі несумісні поняття, що знаходяться у відношенні суперечності, протилежності та супідрядності можуть комбінуватися з командами порівняння, впливу, взаємовпливу, взаємодії.

Для розуміння вищесказаного матеріалу скористаємося поняттями “цикл з передумовою” та “цикл з післяумовою”. Дані поняття утворилися в результаті поділу поняття “цикл з умовою” за такою ознакою як “місце умови”. В свою чергу поняття “цикл з умовою” є членом ділення поняття “цикл” за такою ознакою як “наявність умови”.

Враховуючи те, що поняття “цикл з передумовою” та “цикл з післяумовою” утворилися в результаті поділу родового поняття, вони будуть порівнюваними. Видові ознаки цих понять не забезпечують повного або часткового збігання їх обсягів, отже, поняття будуть несумісними. Можна стверджувати, що зміст одного з понять заперечує інше, а сума їх обсягів вичерпує обсяг родового поняття, а, отже, вони знаходяться у відношенні суперечності.

Встановлення відношення між даними поняттями дозволяє визначити одну з команд, наприклад, порівняння, та скласти

відповідне проблемно-диференційоване завдання, яке полягало б у порівнянні двох циклів з умовою, визначенні узагальнюючого слова (родового поняття) та 3 рис подібності і відмінності між ними, виділенні найголовніших з них (див. рис.1.).

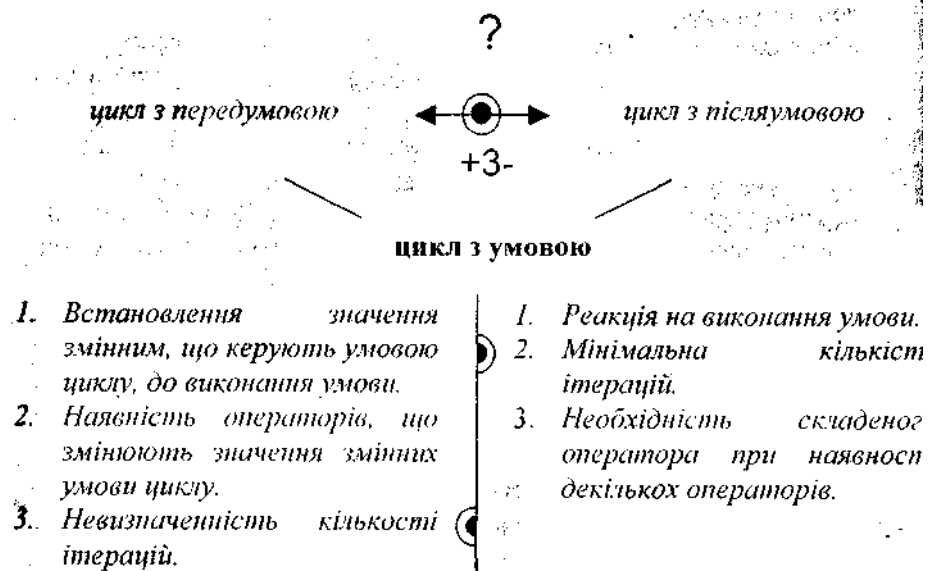
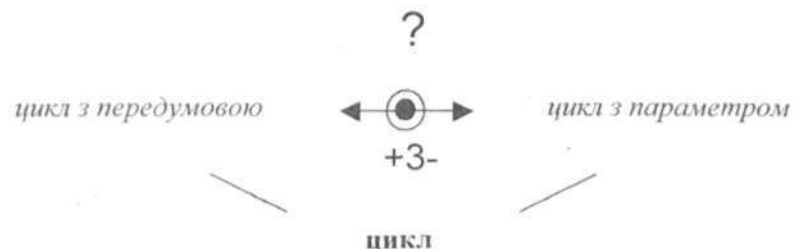


Рис. 1. Приклад проблемно-символічного сигналу

Подальший розгляд та формування поняття "цикл" можна здійснити на прикладі пари понять "цикл з передумовою" ("цикл з умовою") та "цикл з параметром" ("цикл без умови") (див. рис.2.).

Таким чином, вказані проблемно-символічні сигнали реалізують план формування поняття "цикл".

Слід зазначити, що масштабне впровадження технології вказало на необхідність класифікації символіки з метою її подальшого коректного використання. Для цього було проаналізовано блок символіки, визначено перелік основних символів (команд) та перелік додаткової символіки (параметри).



1. Призначення.
2. Можливість невиконання тіла циклу жодного разу.
3. Наявність циклічних процедур.

1. Встановлення значення змінним, що керують умовою циклу, до виконання умови.
2. Наявність операторів, що змінюють значення змінних умови циклу.
3. Невизначеність кількості ітерацій

Рис. 2. Приклад проблемно-символічного сигналу

До блоку команд ми віднесли символи, що відповідають загальним інтелектуальним операціям (порівняння, узагальнення, визначення причинно-наслідкових зв'язків та найголовнішого тощо):



Блок символів-параметрів, що доповнює та уточнює завдання, був складений з наступних символів:



Символ К (в попередніх версіях конструктора проблемно-символічних сигналів символ 3 (три)), що визначає кількісну

характеристику завдання, ми назвали ключем (опцією).

Наступний крок полягав у складанні можливих коректних комбінаціях, на базі яких було визначено загальну модель проблемної символіки:



– де квадратні дужки, за аналогією з форматом команд операційної системи MS DOS, вказують на не обов'язковий елемент.

Наведена модель не тільки підтверджує правильність складеної моделі, але й дозволяють уникнути некоректності у поєднанні символів, що

позитивно відображається на такому критерії технологічності як відтвореність.

Запропонований нами підхід до організації практичних робіт в умовах впровадження модельно-символічної технології базується на реалізації наступних етапів самостійності: від повного керівництва вчителя через дозовану допомогу до самокерування пізнавальною діяльністю, під час якої самостійність реалізується повністю. Звісно, процес проведення занять у відповідності до такого підходу вимагає від викладача високої кваліфікації, уваги до процесу організації самостійної діяльності, до логіки побудови та організації кожного заняття та всієї спільної діяльності [5, с. 456]. Реалізація основних принципів такого підходу передбачає кардинальні зміни в організації практичних робіт, зокрема, у вирішенні питань диференціювання. Відомо, що провідним принципом диференціювання освіти, а особливо в умовах впровадження розвивального навчання, повинно виступати не диференціація змісту завдань (одним-простіше, іншим-складніше), що домінує при традиційному навчанні, а диференціація допомоги студентам з боку викладача без істотного зниження складності змісту (одні студенти потребують більшої допомоги, інші - меншої). Слід зазначити, що завжди можна виділити групу студентів, котрим можна надати повну самостійність [6, с. 247-248]. Такий підхід дасть змогу кожному студенту досягти максимального прояву своїх можливостей та здібностей, творчого самоствердження та самовизначення.

Залишаючи традиційні три типи диференційованих рівнів А, В, та С, що передбачають визначення рівня оволодіння студентами знань, вмінь та навичок, слід більше уваги звертати на рівень самостійності.

Так, використання рівня А підносить студентів на рівень усвідомленого, творчого та подальшого застосування знань. Цей рівень передбачає вільне володіння фактичним матеріалом, прийомами навчальної роботи й розумових дій, можливість кожному студенту повністю виявити себе через самостійну пізнавальну діяльність, поміркувати над проблемою.

Робота над рівнем В передбачає осмислення й усвідомлення матеріалу. Але для оволодіння такими прийомами навчальних та розумових дій, які необхідні для вирішення питань програми рівня А, в програмі рівня В містяться загальні методичні рекомендації виконання пізнавальних завдань.

Рівень С передбачає засвоєння навчального матеріалу на рівні відтворення і включає багаторазове повторення, членування матеріалу на смислові групи, визначення головного, застосування прийомів запам'ятовування. В зміст цього рівня вводиться детальний інструктаж про те як навчатися, на що звертати увагу, який з вивченого впливає наслідок. При роботі на рівні С студентам дозволяється користуватися підручником, лекційним матеріалом та додатковою літературою, оскільки прямої відповіді на запитання там немає. Виконання рівня С передбачає ознайомлення із комп'ютерною реалізацією завдання – кінцевою метою кожного заняття (ніякої творчості, головна мета - досягти конкретного результату).

Запропонована нами методика організації практичних занять в умовах впровадження модельно-символічної технології полягає в наступному.

На початку вивчення курсу студентам пропонується базовий комплект до практичних занять, що містить *завдання до всього курсу, перелік практичних робіт із зазначенням теми та мети даного заняття, вимоги до оформлення практичних робіт, контрольні запитання для всіх трьох рівнів після кожної практичної роботи.*

Завдання до всього курсу. По-перше, завдання повинні бути практично значущими для студентів. Необхідно встановити

зв'язок навчання із життям та практичною професійною діяльністю. По-друге, завдання повинні бути творчими, тобто сприяти творчому розвитку кожної особистості.

Перелік практичних робіт із зазначенням теми та мети даного заняття. В цьому розділі рекомендовано більш детально вказати на завдання та кінцевий результат окремого заняття, звертати увагу на обов'язковий рівень реалізації (опис загальної структури, перелік функціональних можливостей та наявних компонентів). Слід звернути увагу на те, що цього матеріалу досить для роботи на рівні А.

Контрольні запитання для всіх трьох рівнів після кожної практичної роботи. Метою контрольного завдання рівня С є перевірка осмислення алгоритму виконуваних дій. Враховуючи те, що детальна інструкція складається з певної послідовності кроків, то завдання можна побудувати на визначенні негативних наслідків зміни даної послідовності. Пояснимо це на прикладі п.п. 7 (I<sub>7</sub>), 8 (I<sub>8</sub>) та 9 (I<sub>9</sub>) практичної роботи "Середовище швидкої розробки додатків Delphi. Створення головного та контекстно-залежного меню" курсу "Візуальне програмування", що формулюється як:

"... 7. Додайте до Вашої форми компонент класу *TRorirMenu* та, скориставшись Конструктором меню (*Menu Designer*), організуйте контекстно-залежне меню за зразком, що наведений у п.4.

8. За допомогою Інспектора об'єктів, встановіть значення властивості *RorirMenu* компоненту *Main\_button* в "*RorirMenu1*".

9. Завантажте проект на виконання та, скориставшись натисненням правої кнопки миші на Головній кнопці, перевірте функціональність створеного Вами контекстно-залежного меню".

Зміст завдання відповідно буде наступним:

"До яких негативних наслідків, що вплинуть на результат виконання завдання, призведе видалення з інструкції практичної роботи п.8 (I<sub>8</sub>)?"

Наведене завдання у формі проблемно-символічного сигналу матиме вигляд, зазначений на рис. 3.

Таким чином, виконання даного завдання передбачає розуміння виконуваних дій, їх взаємозв'язок, визначення необхідності дотримання саме такої послідовності. Все це сприяє не простому бездумному виконанню інструкції практичного завдання даного рівня, а осмисленню кожного етапу завдання.

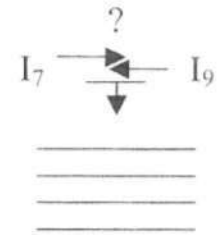


Рис. 3

Як було вказано раніше, програма рівня В містить перелік методичних рекомендацій до виконання завдання. Контрольне завдання даного рівня полягає у ретельному опрацюванні тексту цих рекомендацій, визначенні в ньому пари найголовніших понять, положень чи дій, адекватному підборі проблемно-диференційованого символу, узагальнюючого слова та самостійному складанні на їх основі проблемно-символічного сигналу з подальшим його вирішенням.

Для прикладу скористаємося наступним фрагментом рекомендацій щодо роботи з компонентами-меню:

"Компонент класу *TMainMenu* визначає головне меню форми. На формі можна розташувати скільки завгодно примірників даного компоненту, але відобразяться в смузї меню у верхній частині форми буде тільки той із них, що зазначений у властивості *Menu* форми. Натомість, компонент класу *TRorirMenu* використовується для створення контекстно-залежних меню. На відміну від головного меню, даний компонент може бути створений для будь-якого віконного компонента. Для пов'язування даного меню з будь-яким компонентом, необхідно у властивості *RorirMenu* компонента вказати ім'я компонента-меню. Робота щодо формування та редагування змісту меню аналогічна роботі з головним меню форми."

Завдання повинно бути сформульоване наступним чином:

"На основі п.2 рекомендацій до виконання практичної роботи визначте пару термінів, підберіть до неї узагальнююче слово та виконайте порівняння термінів між собою з метою встановлення 3 рис подібностей та відмінностей."

Виконане студентами завдання у площині проблемної графіки матиме вигляд, зазначений на рис.4.



Рис. 4

Завдання даного рівня передбачає вміння опрацювати текст, вільне володіння основними положеннями використання проблемної символіки, самостійність при виконанні завдання.

Виконання програми рівня А вимагає від студента самостійного опрацювання теоретичного матеріалу, визначення алгоритму власних дій, застосування творчого підходу до вирішення проблеми. У зв'язку з цим контрольне завдання цього рівня полягає у перенесенні основних етапів власної роботи у площину проблемної символіки. Студенти самостійно аналізують суттєві етапи (на одне заняття – від 2 до 5 етапів) власних дій, встановлюють в них найголовніше та, аналогічно до завдання рівня В, складають проблемно-символічний сигнал. Під час виконання даного завдання слід рекомендувати студентам використовувати 2-5 проблемні символи, а від так і складати завдання на порівняння, взаємодію та встановлення причинно-наслідкових зв'язків.

Далі вказані методичні рекомендації до організації роботи студентів під час лабораторних занять з курсу програмування.

Безпосередньо практичному заняттю передують самостійна підготовка студентів. Маючи загальну направленість (завдання до всього курсу) та мету конкретного заняття студенти обирають стратегію його реалізації. Опрацьовується теоретичний матеріал підручника та лекцій, додаткова література; складається план роботи безпосередньо в аудиторії; опрацьовуються контрольні запитання та формується перелік запитань до викладача.

На початку лабораторного заняття студентам надається можливість отримати відповіді на запитання, які вони підготували заздалегідь. На даному етапі слід звернути увагу на те, що не викладач ставить запитання, відповіді на які в нього вже є, а самі студенти проявляють пізнавальну активність, шукаючи відповіді на питання, що виникли у них під час вирішення завдання.

Якщо опрацювавши матеріал та отримавши відповіді на свої запитання, студент все ж таки відчуває невпевненість в можливості самостійного виконання завдання (рівень А), то на занятті він отримує методичні рекомендації рівня В з його відповідними критеріями оцінювання.

Після отримання програми рівня В у студента з'являється вибір: самостійно виконувати завдання цього рівня або ж зробити запит детальної інструкції виконання завдання, і тим самим обрати рівень С.

Слід зазначити, що надання можливості вільного вибору рівня складності доцільне при свідомому ставленні студентів до рівня своїх знань. На початковому етапі бажано запровадити контрольні завдання, успішне виконання яких дозволяло б працювати на певному диференційованому рівні.

Завершальним етапом роботи студента на лабораторному занятті є опрацювання контрольних завдань, зміст яких був поданий нами раніше.

Переваги такої форми організації лабораторних робіт полягають в можливості особистості проявити такі індивідуальні риси мислення як самостійність та критичність; активізації самостійної навчально-пізнавальної діяльності, що проявляється в самостійній підготовці до кожного практичного заняття; можливості переходу студентів від одного диференційованого рівня на інший в залежності від підготовленості до певного заняття; економії навчального часу та раціонального його використання на заняттях за рахунок самостійної підготовки вдома; можливості вільного вибору, що сприяє розвитку критичного ставлення до своїх знань та відповідальності за зроблений вибір; обов'язковому контролю в кінці кожного заняття, що виключає можливість списування алгоритму розв'язку або результатів роботи.

Отже, в ході проведеного нами дослідження було отримано наступні результати:

- розроблено логіко-теоретичну основу модельно-символічної технології, на основі якої сформульовано нові принципи підбору опорних пар понять програмного матеріалу;
- виконано класифікацію проблемної символіки з визначенням переліку основних символів та сукупності додаткової символіки;
- складено коректні комбінації символів-команд з блоком символів-параметрів;
- визначено загальну модуль проблемної символіки;
- розроблено методичні рекомендації до організації лабораторних занять при вивченні курсу “Візуальне програмування” у вищому навчальному закладі.

Результати педагогічного експерименту підтверджують доцільність та ефективність застосування модельно-символічної технології при вивченні курсу програмування для активізації навчально-пізнавальної діяльності студентів, підвищення рівня знань та розвитку логічного мислення.

#### Література.

1. Барабоха П.А. Методика применения системы проблемно-графических сигналов (ПГС) в преподавании географии: Дис... канд. пед. наук: 13.00.02. – С.-Пб., 1993. – 158 с.
2. Барабоха П.А. Программа системного применения проблемно-символических сигналов (ПСС) в преподавании географии. – Уч.-метод. пособие. – К.: Реформа, 1998. – 48 с.
3. Конверський А.Є. Логіка: Підручник для студентів вищих навчальних закладів. – К.: Український Центр духовної культури, 1999. – 400 с.
4. Логика: Учебное пособие / В.Ф. Берков, Я.С. Яскевич, В.И. Павлюкевич. – Мн.: ТетраСистемс, 1998. – 480с.
5. Кушніренко А.Г., Лебедев Г.В. 12 лекцій о том, для чего нужен школьный курс информатики и как его преподавать. Методическое пособие. – М.: ЛБЗ, 2000. – 464 с.
6. Фурман А.В. Психолого-педагогічна теорія навчальних проблемних ситуацій: Дис... д.-ра психол. наук: 19.00.07. – К., 1993. – 449 л.