Н.В.Морзе
Академія праці і
соціальних відносин ФПУ

Метод демонстраційних прикладів при навчанні інформатики

У традиційній методиці навчання предметів природничо-математичного циклу в школі розв'язування задач розглядається як метод навчання і як засіб закріплення теоретичного матеріалу, розвитку мислення і творчих здібностей учнів. Ці функції задач лишаються і при навчанні інформатики в школі. Однак для сучасної методики навчання інформатики всі більш значущим стає подальше розширення дидактичних функцій задач, які враховували б використання основного об'єкта і засобу навчання інформатики – комп'ютер.

Останнім часом велике значення набуває діяльнісний підхід до процесу навчання, який в основному проявляється в концепції "навчання через задачі". Джерела такого підходу до навчання лежить в роботах Д. Пойа.

У конкретних методиках "*навчання через задачі*" розглядається в різних формах. Наприклад, використання задач для мотивації деякої діяльності учнів, для закріплення теоретичного матеріалу, а також для навчання учнів нового теоретичного матеріалу.

"*Навчання через задачі*" – це основний метод навчання, що відноситься до проблемного навчання, яке відрізняється організацією навчання шляхом самостійного одержання знань у процесі розв'язування навчальних проблем, орієнтацією на творче мислення і пізнавальну активність учнів [6]. Існує таке означення методу "*навчання через задачі*" – проблемне навчання, що здійснюється за допомогою системи задач, об'єднаних між собою однією загальною ідеєю дослідження (проблемою), яка орієнтується на одержання нових теоретичних знань [5]. Цей метод пов'язаний з *методом доцільно дібраних задач*, сутність якого полягає в наступному:

з боку вчителя – в побудові системи вправ (або системи доцільно дібраних задач), причому виконання кожної з вправ системи базується на виконанні попередньої і спрямовано на розв'язування сформульованої проблемної ситуації;

з боку учнів – у розв'язуванні деякої проблемної ситуації, яка сформульована вчителем;

вчитель "втручається" в діяльність учнів (якщо це необхідно) при формулюванні кожної наступної задачі або в ході її розв'язування.

Основна ідея цих двох методів полягає в навчанні за допомогою задач, тобто у використанні розв'язування задач як методу навчання. Доведено, що метод доцільно дібраних задач більш загальний, ніж "навчання через задачі".

При вивченні кожного розділу шкільного курсу інформатики можна використовувати метод доцільно дібраних задач. Ця ідея була використана вже в шкільному підручнику з інформатики (1995), де вивчення кожного розділу будується на основі розв'язування задач за допомогою комп'ютера (побудова математичної моделі, її реалізація на комп'ютері, аналіз результатів).

Реалізація цього методу передбачає розробку системи задач (вправ), яка відповідає концепції навчання інформатики та пристосована до навчання діяльності, що відображає специфіку цього предмету. Це означає, що задачі (вправи) повинні слугувати і мотивом для подальшого розвитку теорії (введення нових понять, нових властивостей об'єктів, що вивчаються) і можливістю для її ефективного застосування.

Основну ідею використання при навчанні методу доцільно дібраних задач, залежно від мети використання задач, можна умовно подати так:

1) задачі – засіб для закріплення теоретичного матеріалу: теорія – задача – теорія;
2) задачі – засіб для пізнання теоретичного матеріалу: задача – теорія – задача.

При навчанні інформатики з використанням цього методу важливо пам'ятати:

1) що основним методом інформатики (як науки) вважається обчислювальним експеримент;
2) необхідно використовувати навчальні прикладні задачі.

Одним із частково-дидактичних методів, що базується на ідеях методу доцільно дібраних задач, є моделювання. Метод комп'ютерних моделей був розширений з виділенням "навчальних інформаційних моделей", які одержали назву "демонстраційні приклади", а новий метод навчання з їх використанням – *метод демонстраційних прикладів* [4]. Розглянемо більш детально цей метод.

У педагогічній практиці поняття "*модель*" трактується як деякий об'єкт, дещо подібний (аналогічний) до оригіналу. Моделювання є найбільш адекватним сучасним вимогам до системи освіти методом включення комп'ютера в навчання, яке забезпечує активний вид навчально-пізнавальної діяльності. Переваги навчального комп'ютерного моделювання пов'язані з подоланням формальності засвоєння знань, розвитком дослідницьких і конструкторських навичок, розвитком інтелектуальних здібностей учнів. Використання комп'ютерного моделювання в навчальному процесі (дослідження явищ на

основі готових моделей, побудова моделей самими учнями) дозволяє підвищити інтенсивність навчання і активність пізнавальної діяльності учнів.

Комп'ютерна модель – це програмне середовище для обчислювального експерименту, яке об'єднує в собі на основі математичної моделі явища чи процесу засоби аналізу об'єкта експерименту та відображення інформації [3]. Виділення даного поняття дозволило визначити перший класифікаційний рівень поділу моделей на *традиційні і комп'ютерні*. Однак існують моделі, які мають властивості і традиційних, і комп'ютерних. Такі моделі будемо називати *комп'ютерними інформаційними моделями*.

Зазначимо, що:

1) комп'ютерна інформаційна модель являє собою сукупність символів деякого алфавіту, тому є традиційною моделлю;
2) комп'ютерна інформаційна модель шляхом виконання процесу трансляції може бути перетворена на програмний засіб, тому є комп'ютерною моделлю;
3) принциповою основою можливості застосування комп'ютера для аналізу математичних моделей у обчислювальному експерименті є інформаційний характер математичних моделей.

Як вже відмічалось, *навчальні комп'ютерні моделі й навчальні інформаційні моделі* одержали назву *демонстраційних прикладів.* Всі навчальні моделі залежно від їх призначення умовно можна поділити на моделі-замінники, моделі-уявлення, моделі-інтерпретатори, дослідницькі моделі та комп'ютерні моделі.

Навчальні комп'ютерні моделі мають дві основні відмінності від традиційних навчальних моделей:

1) універсальність навчальних комп'ютерних моделей;
2) навчальні комп'ютерні моделі є не лише засобом опанування знаннями, а й роблять доступними способи діяльності: робота з навчальними комп'ютерними моделями дозволяє учням опанувати вміння і навички користувача комп'ютера.

Ці особливості дозволяють виділити навчальні комп'ютерні моделі в окремий клас. Поняття *"навчальна інформаційна модель*" визначається як програмне середовище, яке об'єднує в собі на основі комп'ютерної інформаційної моделі засоби експериментування з об'єктом дослідження і розвинуті засоби відображення інформації [4].

Навчальні інформаційні моделі можна поділити на такі групи:

а) моделі алгоритмів зберігання, передавання і опрацювання інформації;
б) моделі структур даних;

в) моделі віртуальних машин;
г) моделі, які демонструють реалізацію об'єктно-орієнтованого підходу до комп'ютерного інформаційного моделювання.

Вчитель може здійснювати добір навчальних інформаційних моделей, які найбільш адекватні тому чи іншому етапу дидактичного циклу навчання. Наприклад, при поясненні нового матеріалу і на етапі закріплення доцільно віддавати перевагу роботі учнів з навчальними інформаційними моделями типів а),б), г), а навчальні інформаційні моделі типу в) більше слід використовувати при самостійній роботі школярів.

У практичній діяльності вчителя часто виникає необхідність використання системи навчальних інформаційних моделей, які описують процес конструювання об'єктів або віртуальних машин. Методичне призначення подібних демонстрацій при навчанні програмуванню описано Н. Віртом:" Читачеві демонструється, як поступово створюється програма, йому надаються різні "моментальні знімки" її розвитку, причому ці розробки демонструють метод поетапного уточнення деталей. Я вважаю важливим, розглядаючи програми в їх закінченому вигляді, приділяти достатньо уваги деталям, оскільки саме в них приховуються основні труднощі в програмуванні" [1].

Демонстраційні приклади подаються у вигляді вихідних текстів програм мовами програмування. Тому часто комп'ютерні засоби навчання для підтримки методу демонстраційних прикладів мінімальні: потрібен лише текстовий процесор (бажано з підтримкою гіпертексту) і система програмування обраною мовою.

Особливого практичного застосування *метод демонстраційних прикладів* набуває *при навчанні програмуванню*. Розглянемо особливості. Щоб навчання програмуванню мало ефект необхідно, щоб воно супроводжувалося лабораторними роботами (лабораторним практикумом), які передбачають практичну роботу учнів за комп'ютерами. Метою лабораторних робіт є закріплення та поглиблення теоретичних знань, напрацювання школярами практичних навичок реалізації алгоритмів певних класів за допомогою мови програмування, яка вивчається. Важливо зазначити, що метод демонстраційних прикладів найчастіше використовується в формі лабораторних робіт (лабораторного практикуму).

Використання методу демонстраційних прикладів при навчанні програмуванню базується на концепції відомого методиста в навчанні програмуванню Н. Вірта: "Програмування – це мистецтво конструювання. Як можна навчити конструкторській, винахідницькій діяльності? Існує такий

метод: виділити найпростіші будівельні блоки із багатьох вже існуючих програм і дати їх систематичний опис… Але програмування являє собою велику і різнопланову діяльність, яка часто потребує складної розумової праці. Помилково вважати, що її можна звести лише до використання готових рецептів. За метод навчання нам лишається обрати ретельний добір і розгляд характерних прикладів. Зрозуміло, не слід вважати, що вивчення прикладів всім однаково корисно. При цьому підході багато залежить від кмітливості й інтуїції учня…" [1].

Для реалізації цього методу навчання вчитель створює для кожної лабораторної роботи методичні вказівки, які повинні містити:

1) текст, до якого входить:
2) програмне формулювання теми, основну мету, вимоги до підготовки учнів, заплановані результати навчання;
3) стислий опис теорії (понять і алгоритмів), яка необхідна для виконання завдання;
4) демонстраційні приклади. При складанні і доборі демонстраційних прикладів необхідно дотримуватися принципу Н. Вірта: "подавати програми в їх остаточному вигляді тією мовою, щоб вони могли реально виконуватися в обчислювальній машині"[1];
5) завдання для самостійного виконання.

Можна поділити демонстраційні приклади для навчання програмування на таки типи:

а) Приклади для ілюстрації певних аспектів синтаксису і семантики обраної для вивчення мови програмування. Таким чином, відомості про синтаксис і семантику мови програмування учень обирає безпосередньо із демонстраційних прикладів, які містять детальні коментарі. Зрозуміло, що у разі потреби школярі повинні скористатися консультацією вчителя або навчальними посібниками з мови програмування, яка вивчається.

б) Приклади для демонстрації реалізації класичних алгоритмів (пошук, впорядкування та ін.). Казиев В. вводить поняття дидактичних алгоритмічних одиниць – "функціонально завершені реалізації тих чи інших алгоритмічних процедур, які найчастіше використовуються і на базі яких будується розв'язування більшості задач і які складають операційне середовище користувача"[2]. Використання таких програмних одиниць дозволяє перенести акцент навчання програмування з вивчення елементів мови, синтаксису конкретної мови програмування на алгоритмічну складність, технологію і методи програмування.

в) Приклади для демонстрації реалізації структурних даних за допомогою засобів мови програмування, яка вивчається.

г) Приклади для демонстрації реалізації об’єктів у мові програмування, яка вивчається.

Продемонструємо технологію викладання (дії вчителя) при використанні методу демонстраційних прикладів при навчанні програмуванню. На першому лабораторному занятті вчитель демонструє учням, як виконуються такі основні операції:

1) Виклик системи програмування за допомогою засобів операційної системи.
2) Здійснення процесу компілювання заздалегідь написаної простої програми.
3) Застосування текстового редактора системи програмування для створення простої програми.
4) Вихід із системи програмування.

Перше заняття, присвячене новій темі, відбувається так: 1) на учнівські комп’ютери разом з системою програмування обраною мовою розміщуються заздалегідь підготовлені вчителем методичні вказівки; 2) вчитель за допомогою демонстраційних прикладів (із методичних вказівок) пояснює основні питання теми, що вивчається; 3) вчитель розподіляє завдання (із методичних вказівок) для самостійного виконання відповідно до рівня підготовки учнів.

На наступних заняттях з даної теми вчитель консультує учнів і приймає виконані завдання.

При цьому учень може мати доступ до програми розв’язування конкретної задачі, яку можна вважати прикладом відкритої програми, яка подається в закінченому вигляді, цілком, відразу на екрані комп'ютера і досліджується учнем інтерактивно з самого початку. Перше, що необхідно йому зробити з відкритою програмою, – запустити її, щоб дізнатися, як вона працює. Це етап орієнтації і знайомства з програмою та мовою програмування і відповідними алгоритмічними структурами, які в ній використовуються.

При роботі з такими програмами учням пропонується виконати ряд експериментальних вправ та потім зробити відповідні висновки або дати відповіді на спеціальні запитання.

Другий етап вивчення відкритої програми продовжується по коментарях (методичних вказівках). Коментарі повинні відноситися не до опису операторів програми, а до змісту, значенню дій в контексті постановки початкової задачі.

Працюючи з такою програмою учень спочатку з одного боку бачить текст на нечіткій, живій мові, а з іншого – послідовність формальних операторів мови програмування, що не має поки для нього ніякого значення.

Практика свідчить, що текст коментарів не відразу цілком зрозумілий всім учням. Прихована в них алгоритмічна складова виявляється тільки в зіставленні з формальним способом запису. Тобто відбувається переклад учнями задачі з мови коментарів на мову програмування і навпаки, крім того використовується запис алгоритму навчальною алгоритмічною мовою.

Звідси очевидні найвищі вимоги до точності обох способів опису у відкритій програмі і відповідності між ними. Оригінальність ситуації в тому, що результат такого перекладу – не алгоритм на мові програмування. Він вже присутній зліва в тексті і сам по собі може бути незрозумілий, але він виконує коректуючу роль при переході від коментарів до команд. Відбувається зустрічний перехід від незрозумілого до зрозумілого в їх способах опису. Основний логічний прийом учня тут – зіставлення текстів і пошук аналогій між досить далекими поняттями.

Учнів слід навчати не лише працювати з готовими коментарями, і записувати спочатку коментарі, а потім ідею її розв'язування.

Наприклад, при оформленні завдань з лабораторних робіт чи при виконанні комплексних практичних робіт від учнів вимагається записати ідею розв'язування задачі, наприклад, у вигляді:

Для задачі: *Скласти алгоритм і для знаходження в лінійній таблиці літерного типу номера заданого елемента Якщо таких елементів в таблиці кілька, то вказати елемент з найменшим номером. Якщо ж елемента в таблиці немає – вивести відповідне повідомлення.*

Схема розв'язування задачі

1. *Будемо перебирати всі елементи таблиці, починаючи з першого, до тих пір, поки не переберемо всі елементи або не знайдемо потрібний елемент.*
2. *Кожний елемент таблиці порівнюємо з даним, якщо елементи співпадуть, виводимо відповідне повідомлення і номер шуканого елемента таблиці.*

Третій етап роботи з відкритою програмою – внесення до неї зміни. При внесенні змін до такої програми учень може приступити до її радикальної модифікації або до власної розробки.

Такий підхід до засвоєння відкритої програми (демонстраційного прикладу) передбачає високу мотивацію і рівень самостійності учнів.

Конструювання алгоритму – це інший, більш традиційний підхід. Алгоритми в цьому випадку відкривають знову разом з учнем без комп'ютера, повторюючи процес його природного створення, від простого до складного, закінчуючи створенням програми. Зрозуміло, що коментарі спочатку тільки намічаються

і не розгортаються детально, а лише проговорюються. Саме коментування та опис ідеї алгоритму представляє значну проблему для учнів. Справа тут в тому, що створення коментарів, крім відомих здібностей до вміння виражати свої думки стисло в письмовому вигляді, в цьому випадку вимагає і точного перекладу задачі з формальної мови на змістовну, вербалізації знань.

При розв'язуванні задачі на конструювання алгоритму важливо висунути та обґрунтувати ідею її розв'язування, вміти структурувати цю ідею, а потім вже визначити, як елементи цієї структури можна формалізувати засобами вибраної мови програмування.

При відборі задач (демонстраційних прикладів) доцільно керуватися такими правилами:

- задача, на прикладі розв'язування якої проводиться пояснення нового матеріалу, повинна бути змістовною (не формалізованою), практично значимою і бажано цікавою; для розв'язування задачі потрібно введення одного-двох (не більше) нових операторів мови або двох невідомих для учнів стандартних функцій;
- задача, на прикладі розв'язування якої проводиться закріплення вивченого матеріалу, повинна передбачати використання вивченого оператора у "модифікованому" вигляді, який відображає інші аспекти його застосування;
- задачі, які використовуються при вивченні типових методів розв'язку, бажано спочатку формулювати нечітко і довизначати умови в процесі діалогу з учнями;
- задачі для закріплення вивченої теми повинні бути як репродуктивного характеру, так і такими, що вимагають внесення елементів творчого розв'язку.

Слід відмітити, що розробка власної нової програми - трудомісткий процес, тим більше, якщо програма повинна виконувати досить складні функції. Тому, по-перше, це практично неможливо зробити в рамках одного уроку; по-друге, налагодження програми вимагає часу і відволікає часто на неістотні дрібниці; по-третє, робить недосяжними поставлені цілі і, по-четверте, виконання описаних завдань неефективно без попередньої (домашньої) підготовки учня до заняття.

Перераховані міркування обумовили ту обставину, що на уроці слід проводити роботи по готових програмах, але, передбачати самостійне виправлення і доповнення їх учнями у вузлових місцях. Це вносить елемент творчості, сприяє засвоєнню теми. Учень заздалегідь знайомиться з теоретичним матеріалом, завданнями, і приступає до роботи, маючи самостійно розроблені фрагменти алгоритмів і програм. Тільки така організація навчання приводить до досягнення всіх цілей лабораторної роботи.

Якщо учні постійно виконуватимуть завдання з програмування, то основу їх діяльності (навчання) становитиме самостійна практична діяльність стосовно виконання завдань, у ході якої і відбувається формування знань, вмінь та навичок.

Використання методу демонстраційних прикладів дозволяє навчити учнів читанню програм, які достатньо часто зустрічаються в демонстраційних прикладах. Н. Вірт з цього приводу пише: ”Довгі програми… можуть розглядатися як вправи в мистецтві читати програми, яким часто нехтують порівняно з мистецтвом писати програми” [1]. Навички читання програм є основою для встановлення правильності програм, які написані іншими, для їх модифікації, а також для верифікації власних програм. Такі навички полегшують добір і використання вже описаних у літературі програмних проектів. Учням слід продемонструвати, що процес читання добре документованої програми, як правило, відбувається згори донизу, починаючи з проекту в цілому і просуваючись далі до більш низького рівня деталізації. Погано документовану програму легше читати знизу догори, щоб спочатку знайти проміжні абстракції, а далі, використовуючи вже знайдені абстракції, перейти до більш високого рівня. Тому процес читання погано документованої програми знизу догори часто називають *поетапним абстрагуванням*.

Вчитель може використовувати демонстраційні приклади не лише для формування вмінь та навичок учнів, а й при вивченні нового теоретичного матеріалу, узагальненні й систематизації знань і для підсумкового контролю.

Метод демонстраційних прикладів доцільно використовувати при вивченні прикладних програм.

Використання методу демонстраційних прикладів дозволяє інтенсифікувати спілкування учнів між собою і з вчителем, який проводить лабораторні роботи, передавати один одному демонстраційні приклади, що були написані іншими учнями, аналізувати їх, модифікувати тощо.

Наявність локальної комп’ютерної мережі, демонстраційного екрана дозволяє в багатьох випадках ефективно використати ідею копіювання в навчанні, причому вчитель має можливість одночасно працювати з усіма учнями при збереженні принципу індивідуальності. Специфічні особливості програм для комп’ютера в курсі інформатики дозволяють ефективно використати готовий програмний модуль, виготовлений кваліфікованим програмістом, для різних навчальних експериментів. Наприклад, а) модуль запускається учнями з неоднаковими початковими даними, а результати, що одержуються при цьому, аналізуються; б) вчитель вводить у модуль штучні помилки, пропонуючи учневі відшукати їх і виправити; в) у модулі «урізаються» деякі з можливостей, які

учень повинен відновити і порівняти потім результат своєї роботи із зразком.

Можна навести чимало інших конкретних прикладів навчального застосування зразків готових програм. Головне тут у тому, що учень має можливість скопіювати кращі сторони готового програмного продукту, який пред'являє йому вчитель. Вчителеві ж не становитиме великих труднощів перетворити одну "навчальну допомогу" на іншу, для цього лише потрібно необхідним чином відредагувати ту, що подається учнем та порівняти з програмою-зразком. Подібний матеріал, що концентрує в собі методичні знахідки вчителя, може поступово нагромаджуватися в ході роботи.

Даний метод дозволяє використовувати метод "*учнівського підручника*". Метод можна використовувати на лабораторних роботах, присвячених систематизації і узагальненню знань. Суть його полягає в тому, що учні одержують на свої комп'ютери демонстраційні приклади з теми, що вивчається, і за допомогою текстового процесора доповнюють його таким чином, щоб запропонований матеріал був цікавим і пізнавальним для товаришів.

Для підготовки "*учнівського підручника*" від учнів вимагається:

1) самостійне вивчення додаткової наукової і методичної літератури;
2) активне застосування одержаних знань на практиці;
3) вміння розв'язувати основні типи задач з теми, що вивчається, тому що, як правило, вказівки до лабораторної роботи закінчуються підрозділом "приклади розв'язування задач", в яких наводяться умови задач, розв'язки яких потрібно було включити в демонстраційні приклади;
4) вміння чітко і ясно викласти свої знання.

Наприкінці такої лабораторної роботи кожен учень, користуючись локальною мережею, може передати на комп'ютер вчителя свій варіант демонстраційних прикладів. Кращі варіанти обговорюються. Розвитком подібного методу є методи "*бригадного*" і "*класного*" підручника.

Зазначимо, що відомий у методиці навчання математики *метод ключових завдань* [7] є до деякої міри безмашинним варіантом методу демонстраційних прикладів. *Ключовою задачею* називається задача, оволодіння способам розв'язування якої дозволяє учневі розв'язувати будь-яку задачу з даної теми на рівні шкільних вимог. Метод ключових задач заснований на тому, що розв'язування більшості шкільних завдань зводиться до розв'язування певної послідовності заздалегідь дібраних вчителем ключових задач.

Однак, при застосуванні методу демонстраційних прикладів вчителеві необхідно враховувати, що деякі процедури і функції, які використовуються в даному методі, – лише приклади, за зразком і подібністю яких можна створювати реальні програми опрацювання інформації. Іноді спроби безпосереднього застосування тієї чи іншої процедури чи функції можуть не відразу привести до мети, оскільки сама постановка задачі в деяких демонстраційних прикладах умовна. Вчитель повинен пояснити учням, що практична реалізація того чи іншого алгоритму може вимагати суттєвої модифікації демонстраційних прикладів.

ЛІТЕРАТУРА

1. Вирт Н. Алгоритмы + структуры данных= программы. – М.:Мир, 1985.– 406 с.
2. Казиев В. Дидактические алгоритмические единицы // Информатика и образование, 1991, 6,с.93-95.
3. Лаптев В.В., Немцев А. Учебные компьютерные модели // Информатика и образование, 1991,4, с.70-73.
4. Лаптев В.В., Швецкий М.В. Метод демонстрационных примеров в обучении информатике студентов педагогического вуза // Педагогическая информатика, 1994, 2, С.7-16.
5. Методика преподавания математики в средней школе: Общая методика. Учеб. Пособие для студентов пед. институтов. / А.Я.Блох, Е.С.Канин, Н.Г. Килина и др.; Сост. Р.С. Черкасов, А.А Столяр. – М.: Просвещение, 1985, – 336 с.
6. Подласый И.П. Педагогика. Учеб. Для студентов высших пед. учеб. Заведений. – М.: Просвещение, 1996, – 432 с.
7. Хазанкин Р.Г., Зильбергер Н.И. Ключевые задачи в обучении математике // Учитель Башкирии, 1984, 9, С.58-61.