

## РОЗВИТОК АЛГОРИТМІЧНОГО Й ОПЕРАЦІЙНОГО МИСЛЕННЯ У ПРОЦЕСІ ВИВЧЕННЯ ПРИКЛАДНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Барболіна Т.М.

Поняття алгоритмічного стилю мислення широко використовується в методичній літературі. Зокрема, у [1] воно визначається як система мисленневих способів дій, прийомів, методів та відповідних їм мисленневих стратегій, що спрямовані на розв'язування як теоретичних, так і практичних задач, і результатом яких є алгоритми як специфічні продукти людської діяльності. Поза сумнівом, алгоритмічне мислення складає важливу частину інтелектуальної діяльності людини із застосуванням сучасних інформаційних технологій. Як компоненти алгоритмічного стилю мислення виділяють [2]:

- аналіз необхідного результату і вибір на цій основі початкових даних для розв'язання проблеми;
- виділення операцій, необхідних для розв'язання;
- вибір виконавця, здатного здійснювати ці операції;
- впорядкування операцій і побудова моделі процесу розв'язування;
- реалізація процесу розв'язування і співвідношення результатів з тим, що слід було отримати;
- корекція початкових даних або системи операцій у разі неспівпадіння отриманого результату з передбачуваним.

Особливості операційного стилю мислення спочатку пов'язувалися з необхідними для професійного програміста вміннями й навичками, які дозволяли ефективно використовувати комп'ютер як інструмент діяльності. Проте, як зазначає Ю.А. Первін [3], навички операційного стилю мислення, хоча і виглядають безпосереднім продовженням і розширенням «технологічних» умінь спілкування з комп'ютером, мають загальнокультурну, загальноосвітню, загальнолюдську цінність і потрібні в сучасному інформаційному суспільстві кожній людині, незалежно від прикладних задач її професійної діяльності. До умінь і навичок, що формують операційний стиль мислення, відносяться [3]:

- уміння планувати структуру дій, необхідних і достатніх для досягнення поставленої мети за допомогою певного набору засобів;
- уміння будувати інформаційні моделі для опису об'єктів і систем;
- уміння організувати пошук інформації, необхідної для розв'язування поставленої задачі;
- дисципліна і структурованість мовних засобів комунікації;
- навички своєчасного звернення до комп'ютера під час розв'язування задач із різних предметних галузей;
- технічні навички взаємодії з комп'ютером.

У даній статті розглядаються практичні аспекти формування елементів алгоритмічного й операційного мислення. Формування алгоритмічного мислен-

ня пов'язується, як правило, з вивченням розділу «Основи алгоритмізації і програмування», проте алгоритмічний підхід важливий не тільки в навчанні алгоритміки чи програмування, а є одним з головних у навчанні інформатики взагалі, алгоритмічний стиль мислення є необхідною базою для засвоєння учнями як елементів програмування, так і технологічних компонентів інформатики.

Вивчення прикладних програмних засобів невідменно пов'язано зі свідомим виконанням послідовності дій для досягнення певного результату. Формування навичок формального виконання й побудови таких послідовностей є не лише пропедевтикою вивчення основ алгоритмізації та програмування, а й створює передумови для розвитку алгоритмічного мислення учнів.

Одним із компонентів алгоритмічного мислення є вичленування окремих операцій. Тому під час вивчення прикладних програм слід приділяти увагу формуванню навичок виокремлення елементарних (для даного програмного засобу) дій, подавання складних дій у вигляді організованої послідовності простих.

Найбільш типовим завданням є вимога описати послідовність дій для виконання тієї чи іншої стандартної операції (наприклад, копіювання файлу засобами операційної системи, виділення несуміжних діапазонів у середовищі табличного процесора, зміни міжрядкового інтервалу у текстовому процесорі тощо). Водночас слід вимагати від учнів точного формулювання алгоритмів, що сприяє не лише усвідомленню послідовності дій і оволодінню термінологією, але й формуванню іншого важливого елементу операційного мислення — структурованості мовних засобів комунікації, вміння правильно, чітко й однозначно формулювати думку.

Для демонстрації важливості точного формулювання можна запропонувати іншому учневі формально виконати запропонований алгоритм. У такому випадку більш зрозумілими й помітними стають неоднозначності у висловленнях, пропуски окремих кроків, які сприймаються учнями як очевидні дії. Щоправда, розглянуті переваги можуть проявитися не одразу, оскільки на початкових етапах учні підмінюють неправильну послідовність дій тією, яка забезпечує досягнення результату. У цьому випадку учитель повинен запобігати такій підміні дій, слідкуючи за правильністю виконання.



Доцільно не лише розглядати з учнями алгоритми виконання стандартних операцій, але й пропонувати завдання на побудову послідовності дій у більш складних ситуаціях.

**Приклад 1.** Описати дії для створення в текстовому процесорі таблиці такого вигляду (рис. 1):


Рис. 1

Виконуючи це завдання, учень повинен не лише зазначити вказівку, яку необхідно виконати для додавання таблиці, але й вказати необхідні значення параметрів (кількість рядків і стовпців), пояснити спосіб об'єднання комірок (враховуючи виділення необхідних комірок).

Децю складнішим і через те цікавішим варіантом таких завдань є побудова учнями якомога більшої кількості можливих алгоритмів розв'язання одного й того самого завдання, що далі передбачає визначення найефективнішого плану дій.

**Приклад 2.** Використовуючи засоби створення автофігур у текстовому процесорі, створити зображення шахівниці таким чином, щоб під час роботи було виконано якнайменше дій, а результат був високої якості. Пояснити свій варіант алгоритму виконання цього завдання.

Раціональний підхід полягає не у великій кількості повторів вставки чорної та білої клітин, а у формуванні «наддеталі», з кількох копій якої формується ціле. Такий алгоритм близький до принципу дихотомії: спочатку формується деталь, що складається з чорної та білої клітин, з цих двох клітин — фрагмент із чотирьох клітин і т. д. Розглянутий підхід дозволяє виконати 5 копіювань замість 62.

Діяльність учнів у процесі розв'язування таких завдань можна організувати по-різному. Так, учитель може оголосити завдання, продемонструвати зразок, а далі бажуючі пояснюють кроки можливого виконання задачі. Інший варіант полягає у тому, що після оголошення завдання учитель пропонує учням виконати завдання практично. Після виконання завдання відбувається обговорення учнями різних алгоритмів з наступним їх аналізом. Цей спосіб організації виконання завдань дозволяє кожному з учнів взяти активну участь у практичному вирішенні проблеми, причому практичні предметні маніпуляції полегшують процес складання алгоритму.

У будь-якому випадку необхідним є етап формування й обговорення запропонованих алгоритмів, аналіз їх ефективності, з'ясування найбільш раціонального способу розв'язування задачі. Так створюються умови для розуміння ефективності алгоритму як співвідношення між витратами на розв'язання задачі (час, кількість виконаних дій) та отриманим результатом (його якістю, точністю).

Крім завдань на формулювання алгоритмів, доцільно пропонувати вправи на встановлення правильного порядку виконання дій, встановлення пропущених кроків, визначення результату виконання алгоритму без застосування відповідного програмного засобу. Можна також поєднати завдання визначення результату виконання алгоритму із дослідженням ефективності, запропонувавши учням створити алгоритм, виконання якого призводить до такого самого результату, що й даний, і порівняти їх ефективність.

**Приклад 3.** Розгляньте наведений нижче алгоритм роботи з текстовим редактором **Блокнот** і визначте результат його виконання. Як можна досягти такого самого результату раціональніше?

1. Натиснути клавішу **Home**.
2. Відпустити клавішу **Home**.
3. Натиснути клавішу **Shift**.
4. Якщо справа немає символів, то перейти до п. 8.
5. Натиснути клавішу **→**.
6. Відпустити клавішу **→**.
7. Перейти до п. 4.
8. Відпустити клавішу **Shift**.
9. Натиснути клавішу **Del**.
10. Відпустити клавішу **Del**.
11. Кінець.

Наведений алгоритм призначений для знищення поточного рядка тексту. Підвищити ефективність можна за рахунок більш раціонального виділення рядка: наприклад, замінивши кількаразове натиснення клавіші **→** натисненням клавіші **End**.

Слід відзначити, що розгляд алгоритмів розв'язування тих чи інших завдань у середовищах прикладних програмних засобів не лише сприяє розвитку алгоритмічного мислення, але й виступає пропедевтикою вивчення основних понять алгоритмізації і програмування: алгоритм, вказівка алгоритму, виконавець алгоритму. Хоча переважна більшість алгоритмів, які можуть бути сформульовані учнями під час розв'язування задач з використанням прикладного програмного забезпечення, є лінійними, можна навести приклади, які дозволять продемонструвати й інші базові алгоритмічні конструкції.

**Приклад 4.** Створюючи орнамент (рис. 2), неважко знайти фрагменти, що повторюються. Це дозволяє спростити процес побудови за рахунок копіювання таких фрагментів. Кількаразова дія вставки підводить до поняття циклу. Крім того, вишня за формою така сама, як і яблуко, що дозволяє одержати її зменшенням та зміною заливки. У термінах алгоритмізації побудови вишні та яблука можна розглядати як використання допоміжного алгоритму з різними параметрами.



Рис. 2

Правильна побудова алгоритму неможлива без коректного визначення основних властивостей ре-

зультату, який необхідно одержати. Нерідко діти припускаються помилок не через незнання правил виконання тих чи інших операцій, а внаслідок неправильного визначення характеристик зразка (параметрів форматування тексту чи комірок електронної таблиці, параметрів ліній у графічному зображенні). Для формування навичок визначення характеристик зразка, умінь формулювання функціональних можливостей використовуваних засобів можна застосовувати такі завдання.

**Приклад 5.** Для кожної з наступних пар фрагментів (рис. 3) тексту вказати, за допомогою яких операцій можна другий фрагмент одержати з першого (приклади операцій: застосувати курсив, відмінити жирне накреслення, збільшити розмір шрифту, вставити вирівнювання по центру тощо).

Приклад тексту 1	Приклад тексту 1
Приклад тексту 2	Приклад тексту 2
Приклад тексту 3	Приклад тексту 3
Приклад тексту 4	Приклад тексту 4
Приклад тексту 5	Приклад тексту 5

Рис. 3

Пари фрагментів у завданні подаються за зростанням складності: у перших двох парах передбачається зміна одного параметра форматування шрифту (курсивне накреслення і гарнітура відповідно), у третій — двох параметрів форматування шрифту (розмір шрифту і підкреслення), у четвертому і п'ятому прикладі змінюються параметри форматування не лише шрифту, але й абзацу.

Навички визначення істотних характеристик об'єкта є необхідними і в інформаційному моделюванні. Уміння будувати інформаційні моделі важливе не лише під час розробки програм, його значення зростає у прикладних інформаційних системах (електронних таблицях, базах даних тощо), основу яких складають інформаційні моделі.

Під час добору «життєвих» задач для розв'язування на уроках слід враховувати привабливість, практичну значущість задач, близькість предметної галузі. Цікавими, і часто несподіваними за кінцевим результатом, є задачі про числа-велетні, наведені у [4]. Більшість цих задач пов'язані з геометричною прогресією і демонструють швидкість експоненційного зростання.

Розглянемо одну з таких задач: розповсюдження чуток у невеликому місті. Аналіз умови дозволяє визначити такі припущення, які вводяться для спрощення моделі.

1. Час, який необхідний для повідомлення новини, є однаковим.
2. Кількість людей, яким одночасно повідомляється новина, є однаковою.
3. Кожний, хто почув новину, повідомляє її іншим протягом найближчого періоду часу.
4. Кожний бере участь у передачі новини один раз.
5. Кожного разу новина повідомляється людям, які її до цього моменту не чули.

Зрозуміло, що в реальній передачі чуток у місті ці умови не виконуються: передача новини може займати різний час, кожен «носій» новини може повідомляти її різній кількості людей (і навіть не повідомляти взагалі) тощо. Проте навіть у такому спрощеному варіанті задача дозволяє побачити лавиноподібний розвиток подій. У класі, де учні мають сильну математичну підготовку, можна побудувати також математичну модель задачі, спираючись на формулу суми перших членів геометричної прогресії.

На основі висловлених вище припущень може бути побудована така математична модель розповсюдження чуток:

$$S = p_0 + p_1 + \dots + p_n, \quad p_i = k \cdot p_{i-1} \quad (i = 1, \dots, n),$$

де  $S$  — загальна кількість людей, яким відома новина;  $p_0$  — кількість людей, які знали новину в початковий момент процесу;  $p_i$  ( $i = 1, \dots, n$ ) — кількість людей, яким повідомляється новина на  $i$ -му «етапі» передачі;  $k$  — кількість людей, яким одночасно повідомляється новина.

Для реалізації моделі в табличному процесорі доцільно в послідовних комітках обчислювати значення  $p_i$ , а потім просумувати одержані значення. Для поетапного аналізу результатів доцільно також знаходити суми  $S_k = p_0 + p_1 + \dots + p_k$  ( $k < n$ ). Можлива структура відповідної таблиці:

- «етап»;
- кількість людей, яким повідомили новину протягом останнього проміжку часу;
- загальна кількість людей, які знають новину на даний момент часу.

Залежно від рівня підготовки учнів величина  $k$  може використовуватися у формулах як константа або її значення розміщується в окремій комітці (тоді у формулі використовується абсолютне посилання). Для більшої наочності замість номера «етапу» передачі новини можна в таблицю вносити відповідні моменти часу.

Переваги побудови комп'ютерної моделі полягають не лише в можливості швидко побачити результат, але й прослідкувати динаміку кількості мешканців, які знають новину. Особливо наочним такий результат буде, якщо побудувати графік за даними електронної таблиці. Крім того, комп'ютерна модель дозволяє учням поекспериментувати і визначити, як зміниться результат у разі зміни окремих параметрів.

Прикладні задачі, що вимагають побудови математичної моделі, можуть використовуватися як елементи домашнього завдання (записати необхідні співвідношення у вигляді математичної або табличної моделі), практичної роботи, причому одна й та сама задача може використовуватися протягом кількох уроків, під час яких на матеріалі цієї задачі відбувається вивчення різних можливостей табличного процесора.

Для формулювання завдань, що вимагають побудови інформаційної моделі, можна використовувати матеріали не лише з математичної, але й з художньої літератури. Такі завдання не лише забезпечу-

ють міжпредметні зв'язки, але й демонструють важливість розглядуваних понять у різних сферах людської діяльності, неочікувані застосування підвищують інтерес до вивчення предмету, дозволяють формувати розуміння того, які задачі можуть бути розв'язані за допомогою комп'ютера, добирати відповідні програмні засоби.

**Приклад 6.** Прочитайте уривок з оповідання А. Коуан-Дойла «Блакитний карбункул» і створіть базу даних, яка замінювала б книги торговця Брекенриджа.

— Бачите цю книжку?

— Ну то й що?

— Це список тих, у кого я купую товар. Бачите? Ось тут, на цій сторінці, імена сільських постачальників, а цифра після кожного прізвища означає сторінку в гробухові, де ведуться їх рахунки. А цю сторінку, списану червоним чорнилом, бачите? Це список моїх міських постачальників. Погляньте-но на третє прізвище. Прочитайте його вголос.

— Місіс Окшот, Брикстон-роуд, 117, сторінка 249.

— Цілком вірно. Тепер відкрийте 249-у сторінку в гробухові.

— Ось: «Місіс Окшот, Брикстон-роуд, 117 — постачальниця дичини та яєць».

— Про що йдеться в останньому записі?

— «Грудень, 22. 24 гусака по 7 шилінгів 6 пенсів».

— Правильно. Запам'ятайте це. А внизу?

— «Продано містеру Віндігету, «Альфа», по 12 шилінгів».

Враховуючи складність процесу проектування баз даних, початковий аналіз предметної галузі доцільно проводити у формі евристичної бесіди. У ході такої бесіди слід визначити основні таблиці, що входять до бази даних (доцільно розробити окремі таблиці для зберігання інформації про постачальників, факт купівлі та дані продажу).

Структуру окремих таблиць можна також з'ясувати у ході бесіди або в процесі групової роботи (кожна група працює над однією таблицею). Під час визначення необхідних полів слід скористатися прикладами записів, які наведені у фрагменті. Зокрема, аналіз запису «Місіс Окшот, Брикстон-роуд, 117 — постачальниця дичини та яєць» дозволяє зробити висновок про необхідність таких полів таблиці «Постачальники»: **Прізвище, Адреса, Об'єкт\_поставок**. Враховуючи, що містер Брекенридж розділяв міських і сільських постачальників, доцільно додати відповідне поле (наприклад, **місце\_проживання**). Для однозначного визначення записів у таблиці, що необхідно в майбутньому для встановлення зв'язків, доречно також додати поле **Код\_постачальника**.

Аналогічно на основні записів у книгах формуються структури таблиць «Закупівлі» (**Код\_закупівлі, Постачальник, Дата, Назва\_товару, Ціна**) та «Продажі» (**Код\_Продажу, Код\_Закупівлі, Ім'я\_покупця, Адреса\_покупця, Ціна**).

На наступному етапі роботи з базою даних слід встановити зв'язки між таблицями: поле **Код\_Постачальника** таблиці «Постачальники» зв'язується з полем **Постачальник** таблиці «Закупівлі» та поле **Код\_закупівлі** таблиці «Закупівлі» з однойменним полем таблиці «Продажі».

Формуванню навичок свідомого застосування тих чи інших засобів інформаційних технологій сприяють також завдання, формулювання яких не містить вказівки щодо програмного засобу, який повинен використовуватися для їх розв'язування.

**Приклад 7.** Розробити такі документи для забезпечення документнообігу в бібліотеці:

1) шаблон акта на одержання літератури, який повинен містити текст про те, хто й кому передає літературу, а також оформлену у вигляді таблиці інформацію по кожному найменуванню (автор, назва, ціна та кількість примірників);

2) шаблон для зберігання інформації про книги, наявні в бібліотеці (шифр книги, автор, назва, рік видання, місце видання, короткий опис), з можливістю підрахунку кількості книг за роками видання.

Не зважаючи на схожість завдань, вони вимагають різного інструментарію. Перший документ передбачає поєднання тексту й інформації, поданої у вигляді таблиці, причому без необхідності її подальшого опрацювання. Тому для підготовки такого документа краще використати текстовий процесор. Другий документ, очевидно, необхідно готувати в середовищі систем управління базами даних (хоча можливою є також реалізація бази даних у середовищі табличного процесора). Таким чином, завдання реалізує прикладну спрямованість курсу інформатики, примушує дитину зважити переваги й недоліки використання різних програмних засобів для конкретних задач.

Розглянуті у статті прийоми сприяють формуванню в учнів навичок планування своїх дій, визначення істотних характеристик об'єктів та встановлення зв'язків між ними, свідомого застосування засобів інформаційно-комунікаційних технологій для розв'язування практичних задач, дозволяють підвищити інтерес до вивчення інформатики.

### Література

1. *Копаяев О.В.* Алгоритм як модель алгоритмічного процесу / О.В. Копаяев // Комп'ютерно-орієнтовані системи навчання: Зб. наук. праць — К.: НПУ ім. М.П. Драгоманова. — Випуск 6. — 2003. — С. 206–213.
2. *Газейкина А.И.* Стили мышления и обучение программированию студентов педагогического вуза / А.И. Газейкина // Материалы XVI Международной конференции-выставки «ИТО-2006». — Режим доступу: <http://ito.edu.ru/2006/Moscow/index.html>.
3. *Первин Ю.А.* Методика раннего обучения информатике / Ю.А. Первин // Вопросы Интернет-образования. — № 5. — Режим доступу: [http://vio.fio.ru/vio\\_05/](http://vio.fio.ru/vio_05/).
4. *Перельман Я.И.* Живая математика / Я.И. Перельман. — М.: ОГИЗ, 1947. — 184 с.

★ ★ ★