

ИСПОЛЬЗОВАНИЕ ПРИНЦИПА АНАЛОГИИ ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ

Н. В. Иванова, Лю Минь

*Витебский государственный университет
имени П. М. Машерова
Витебск, Беларусь
E-mail: ivanata2@tut.by*

Один из методов научного познания – аналогия – широко применяется при обучении программированию. В данной работе рассматриваются возможности использования аналогии при обучении языку программирования Си при условии, что язык программирования Паскаль был изучен ранее.

Ключевые слова: аналогия свойств, аналогия отношений, строгая аналогия, не-строгая аналогия, ложная аналогия.

Под аналогией понимается сходство двух предметов (или двух групп предметов) в каких-либо свойствах или отношениях. Часто аналогия определяется как умозаключение о принадлежности предмету определенного признака (т. е. свойства или отношения) на основе сходства в признаках с другим предметом.

Использование аналогии в познании – основа для выдвижения предположений, догадок, гипотез. Рассуждения по аналогии часто приводили к научным открытиям.

В обучении аналогия понимается как педагогический прием, который заключается в установлении сходства в каком-нибудь отношении между изучаемыми предметами и явлениями, различными в остальных отношениях. Ценность аналогии как приема обучения состоит в том, что она облегчает усвоение учебного материала, активизирует мысль обучаемого, побуждает его к поискам, наводит на предположения и догадки, правильность которых требует специальных исследований.

Аналогия свойств или отношений предметов в традиционной логике отражается следующей схемой.

Предмет *A* обладает свойствами *a, b, c, d, e, f*.

Предмет *B* обладает свойствами *a, b, c, d*.

Вероятно, предмет *B* обладает свойствами *e, f*.

Посредством аналогии осуществляется перенос информации с одного предмета (*оригинал*) на другой (*модель*). Посылки относятся к оригиналу, заключение – к модели. Рассматривая характер информации, переносимой с оригинала на модель, выделяют два вида аналогий: *аналогия свойств* и *аналогия отношений*.

В *аналогии свойств* рассматриваются два единичных предмета или два множества однородных предметов, а переносимыми признаками являются свойства этих предметов. Так, например, в языках программирования Паскаль и Си аналогичными свойствами обладают простые типы данных: переменным логического типа могут быть присвоены только значения true или false; переменная типа char (символ) занимает ячейку памяти, достаточ-

ную для размещения в ней ASCII-кода символа; целочисленная переменная длиной 16 бит может быть использована для представления чисел в диапазоне от $-32\,678$ до $32\,767$ и т. д.

В *аналогии отношений* информация, переносимая с оригинала на модель, характеризует отношения между двумя предметами. Имеем отношение $(a R b)$ и отношение $(c Q d)$. Аналогичными являются отношения R и Q , но при этом a не аналогично c , а $b - d$.

Если рассматривать степень достоверности заключения (характер выводного знания), то умозаключение по аналогии можно разделить на три вида:

- 1) строгая аналогия (дает достоверное заключение);
- 2) нестрогая аналогия (дает вероятное заключение);
- 3) ложная аналогия (дает ложное заключение).

Строгая аналогия. Отличительной особенностью строгой аналогии является наличие необходимой связи между сходными признаками и переносимым признаком. Схема строгой аналогии такая.

Предмет **A** обладает признаками a, b, c, d, e .

Предмет **B** обладает признаками a, b, c, d .

Из совокупности признаков a, b, c, d необходимо следует e .

Предмет **B** обязательно обладает признаком e .

Чаще всего строгая аналогия применяется в научных исследованиях, в математических доказательствах. На строгой аналогии основан метод моделирования.

Строгая аналогия дает достоверный вывод, т. е. истину, обозначаемую в классической логике и в теории вероятностей через 1. Вероятность вывода по строгой аналогии равна 1.

Строгую аналогию мы используем в лабораторных работах для студентов. Например, при обучении работе с массивами в начале работы мы предлагаем студентам подробно разобрать решение одной задачи и текст программы сначала на языке Паскаль, затем – на Си.

Условие задания. Дан одномерный целочисленный массив. Составить программу вычисления суммы четных элементов этого массива.

Программа на языке Паскаль.

```
uses crt; {Подключение библиотеки ввода-вывода}
const n = 6; {Размерность массива}
var i, s : integer; {i - счетчик цикла, s - переменная накопления
суммы четных элементов массива}
    a : array [1..n] of integer; {Объявление одномерного массива}
begin
  ClrScr; {Очистка экрана}
  for i := 1 to n do
  begin
    Write('Введите ', i, '-й элемент массива ');
    ReadLn(a[i]); {Ввод элемента массива}
  end;
  s:=0; i:=1; {Инициализация переменных}
  for i:=1 to n do
  begin
    if a[i] mod 2 =0 then s:=s+a[i]; {Накопление суммы четных эле-
ментов}
    i:=i+1; {Увеличение счетчика цикла}
  end;
  WriteLn('Сумма четных элементов массива равна ', s);
  ReadKey; {Задержка экрана}
end.
```

Программа на языке Си.

```
#include<stdio.h> // Библиотека ввода-вывода
#include<conio.h> // Библиотека работы с окнами
#define N 6 // Макроопределение (размерность массива)
void main(void) // Главная функция
{
// Объявление локальных переменных
int A[N]; // Одномерный целочисленный массив из N элементов
int i, s; // i - счетчик цикла, s - сумма четных элементов
clrscr(); // Очистка экрана
for (i=0; i<N; i++)
{
printf("Введите %i элемент массива: ", i);
scanf("%i", &A[i]); // Ввод элементов массива
}
s=0; // Присвоение начального значения переменной s
for (i=0; i<N; i++)
{
if (A[i]%2==0) // Проверка на четность
s+=A[i]; // Накопление суммы
}
// Вывод полученного значения суммы на экран
printf("\nСумма четных элементов равна s=%i", s);
getch(); // Задержка экрана
}
```

Тексты программ подробно откомментированы. Деятельность студентов по выполнению этого задания включает в себя следующие шаги.

1. Выполнить программу на языке Паскаль, зафиксировать исходные данные и результаты.
2. Выполнить программу на языке Си и зафиксировать результаты для тех же наборов исходных данных, что в п. 1).
3. Сравнить тексты программ и отметить особенности синтаксиса языка программирования Си, использованные в примере.

Поскольку в примере использована строгая аналогия, то при одних и тех же наборах данных результаты выполнения программ совпадают. Программы отличаются только синтаксисом языка программирования. Так как язык Паскаль уже знаком студентам, основное внимание они вынуждены сконцентрировать на особенностях синтаксиса языка Си.

Нестрогая аналогия. Позволяет получить лишь вероятное заключение. Если ложное суждение обозначить через 0, а истинное – через 1, то степень вероятности выводов по нестрогой аналогии лежит в интервале от 0 до 1, т. е. $0 < P(a) < 1$, где $P(a)$ – вероятность заключения по нестрогой аналогии.

Для повышения степени вероятности выводов по нестрогой аналогии следует выполнить ряд условий:

- 1) число общих признаков должно быть возможно большим;
- 2) необходимо учитывать степень существенности сходных признаков, т. е. сходные признаки должны быть существенными;
- 3) общие признаки должны быть по возможности более разнородными;

- 4) необходимо учитывать количество и существенность пунктов различия. Если предметы различаются в существенных признаках, то заключение по аналогии может оказаться ложным;
- 5) переносимый признак должен быть того же типа, что и сходные признаки.

В обучении программированию при решении задач применяется нестрогая аналогия с уже решенными ранее одностипными задачами. Также аналогия используется тогда, когда, пытаясь решить предложенную задачу, мы начинаем с другой, более простой. Например, при решении задачи, связанной с обработкой двумерных массивов, удобным бывает решить эту задачу относительно линейного массива – строки или столбца.

Ложная аналогия. При нарушении указанных выше условий аналогия может дать ложное заключение, т. е. стать ложной. Вероятность заключения по ложной аналогии равна 0. Ложные аналогии часто делаются неумышленно, в результате незнания правил построения аналогий или отсутствия фактических знаний относительно предметов *A* и *B* и их свойств, на основании которых осуществляется аналогия.

Для использования аналогии в обучении языкам программирования необходимо, чтобы учащиеся владели начальным уровнем знаний какого-то одного из указанных языков. Начальный уровень знаний языка программирования включает в себя: знание алфавита языка, знание типов данных и их описание средствами языка, знание основных структурных операторов. Кроме того, язык Паскаль чаще используется как учебный, а язык Си – как язык промышленного программирования [1]. Поэтому в нашей методике язык Паскаль предпочтительно изучать первым.

При обучении программированию различают обучение технологии написания программы и обучение алгоритмизации. Использование аналогии в нашей методике предполагается в двух аспектах:

- 1) опираясь на аналогию языков программирования Паскаль и Си, совершенствуем технологию написания программ;
- 2) выявляя аналогию между алгоритмами «базовых» задач и алгоритмом новой, творческой задачи, развиваем навыки алгоритмизации.

Нельзя написать программу, не имея алгоритма решения задачи. Верно также и то, что алгоритм зависит от того, на каком языке он будет реализован. Использование аналогии как педагогического приема эффективно и при обучении технологии написания программ, и при обучении алгоритмизации.

ЛИТЕРАТУРА

1. *Потопахин, В. В.* Почему Паскаль? / В. В. Потопахин // Информатика. – Издательский дом «Первое сентября», 2008. – № 6. – С. 18–20.