

Н. Ю. Добровольская,
Кубанский государственный университет, г. Краснодар

ПРИМЕНЕНИЕ ОБОБЩЕННЫХ СХЕМ ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ

Аннотация

В статье рассматривается применение обобщенных схем как языково-независимых фрагментов алгоритмического решения задач при конструировании учебных материалов, генерации наборов разноуровневых тестовых заданий. Применение схем в обучении программированию развивает у школьников умение формализовать задачу, соотносить ее решение с базовым алгоритмом, закладывает навыки блочно-модульного программирования.

Ключевые слова: методика обучения информатике, компьютерные дидактические материалы, техника алгоритмизации.

DOI: 10.32517/2221-1993-2019-18-6-37-40

Современные требования к профессиональной подготовке учителей информатики предполагают внедрение информационных технологий во все элементы учебного процесса. Одной из составляющих профессиональной деятельности учителя является конструктивная деятельность, которая включает отбор учебного материала и формирование наборов заданий оценивания знаний. Сочетание педагогических и компьютерных технологий на этой стадии деятельности позволяет решать указанные педагогические задачи более качественно, с меньшими временными затратами. В связи с этим профессиональная компетенция учителей информатики — способность к преподаванию физико-математических дисциплин и информатики в общеобразовательных организациях, профессиональных образовательных организациях и организациях дополнительного образования — должна быть расширена умением конструировать компьютерные дидактические материалы. Компьютерная форма дидактического обеспечения учебного курса не только позволяет использовать различные варианты тестирования и визуализации учебной информации, но и автоматизирует процесс построения больших наборов разноуровневых заданий, обеспечивая тем самым индивидуализацию обучения. Такие задания отвечают формату заданий компьютерной дидактики, большой вклад в развитие которой внесла А. И. Архипова [1]. Технологии компьютерной дидактики позволяют разрабатывать как компьютерные учебно-методические комплексы, так и локальные задания.

Конструирование компьютерных дидактических материалов по программированию в курсе школьной информатики предполагает формализацию заданий и соответствующих им алгоритмических решений. Такая формализация не только позволяет учителю разрабатывать большое число однотипных заданий для различных видов контроля знаний, но и способствует развитию навыков программирования у школьников. Учащиеся должны научиться конструировать алгоритмическое решение задачи, не зависящее от конкретных конструкций языка программирования. Дидактические материалы могут быть как теоретического, так и тестового содержания. Основой конструирования дидактических материалов являются обобщенные схемы алгоритмов [2].

Приобретение начальных навыков алгоритмизации в процессе изучения информатики закладывает основу будущего умения разрабатывать программы практически на любом языке программирования. Однако у многих обучаемых довольно слабо развито умение формализовать задачу, соотносить с ней тот или иной алгоритм решения. Для развития этого умения необходимо применение строгой методической последовательности изложения учебного материала, от простого к сложному, использование дидактических средств, позволяющих проверить уровень сформированности навыков алгоритмизации. В качестве такого дидактического средства нами предлагаются обобщенные схемы решения задач по программированию.

Контактная информация

Добровольская Наталья Юрьевна, канд. пед. наук, доцент, доцент кафедры информационных технологий, Кубанский государственный университет, г. Краснодар; адрес: 350040, г. Краснодар, ул. Ставропольская, д. 149; e-mail: dnu10@mail.ru

N. Yu. Dobrovolskaia,
Kuban State University, Krasnodar

USE OF GENERALIZED SCHEMES IN TEACHING PROGRAMMING

Abstract

The article discusses the use of generalized schemes as language independent fragments of algorithmic problem solving in the design of training materials, the generation of sets of multilevel test tasks. The use of schemes in teaching programming develops in schoolchildren the ability to formalize a problem, to correlate its solution with the basic algorithm, and to establish block-modular programming skills.

Keywords: methods of teaching informatics, computer didactic materials, algorithmization technique.

Вне зависимости от конкретного языка программирования соответствующий раздел курса школьной информатики содержит набор типовых тем: вычисление арифметических выражений, обработка последовательности чисел, определение свойства числа, работа с массивом, с матрицей и т. д. Для каждой темы нами сконструировано несколько обобщенных схем, которые легко модифицируются при использовании конкретного языка программирования.

Обобщенная схема представляет собой фрагмент алгоритма на некотором языке программирования, включающий постоянные элементы и варьируемые. Наполнение варьируемых элементов схемы конкретными значениями позволяет решить определенную задачу [4]. Обобщенные схемы применяются для формирования умения у обучаемых решать задачи по программированию данного раздела и для тестирования приобретенных навыков.

Приведем пример обобщенной схемы по теме «Обработка одномерных массивов».

Схема 1.

Однотипная обработка элементов массива (язык программирования Pascal).

```
var
  a: array[1..100] of integer;
  i, s: integer;
begin
  readln (n);
  for i:=1 to n do readln(a[i]);
  Z(s);
  for i:=1 to n do
    if (Q(a[i])) then R(s, a[i]);
  writeln(s);
end.
```

В схеме присутствуют три варьируемых поля, заданные функциями:

- $Z(s)$ представляет собой инициализацию переменной s ;
- $Q(a[i])$ проверяет элемент массива $a[i]$ на заданное свойство;
- $R(s, a[i])$ определяет действие над элементом массива $a[i]$ с использованием переменной s .

Для поля $Z(s)$ возможны следующие конкретные значения: $s=0$; $s=1$; для инициализации вычисления суммы, количества или произведения элементов.

Поле $Q(a[i])$ определяет свойство элемента массива, например:

- является ли число положительным ($a[i] > 0$);
- оканчивается ли число цифрой 3 ($a[i] \bmod 10 = 3$) и т. д.

Причем при дальнейшем изучении учебного материала в качестве $Q(a[i])$ можно использовать отдельно реализованную функцию.

Поле $R(s, a[i])$ задает действие над элементами массива, например:

- поиск количества элементов массива ($s := s + 1$);
- вычисление суммы чисел ($s := s + a[i]$) и т. д.

Учащимся необходимо показать, как соотносить обобщенную схему с конкретными формулировками заданий. Для каждой сконструированной схемы строится таблица соответствия условий заданий и значений варьируемых полей (см. табл.).

Таблица

Значения варьируемых полей

№ п/п	Задание	$Z(s)$	$Q(a[i])$	$R(s, a[i])$
1	В массиве целых чисел найти количество отрицательных элементов	$s := 0$;	$a[i] < 0$	$s := s + 1$;
2	В массиве целых чисел найти сумму нечетных элементов	$s := 0$;	$a[i] \bmod 2 < 0$	$s := s + a[i]$;
3	В массиве целых чисел найти произведение элементов, оканчивающихся на 7	$s := 1$;	$a[i] \bmod 10 = 7$	$s := s * a[i]$;

Обобщенные схемы могут соответствовать более сложным заданиям, включающим составные или сложные условия. В последнем случае одна схема будет входить в состав другой. При этом используется базовая парадигма программирования — конструирование решения задачи снизу вверх.

Использование обобщенных схем достаточно эффективно при конструировании авторских дидактических материалов. Тестовые задания на основе обобщенной схемы предлагаются нами в двух формах: открытой и закрытой. В первом случае значение одного из варьируемых полей задается неверно и обучаемый должен предложить в качестве ответа верную строку-замену. Во втором случае варьируемое поле не заполняется, но предлагается ряд различных значений поля и обучаемый должен указать номер верного ответа. Все задания легко автоматизируются с помощью макросов MS Excel [3, 5].

Приведем пример конструирования тестовых заданий. Рассматривается тема «Обработка последовательностей чисел». Обобщенная схема приведена на языке программирования C++.

Схема 2.

Последовательность чисел содержит N вводимых значений.

```
cin >> n;
Z(s);
for (i=0; i<n; i++)
{
  cin>>x;
  if (Q(x))
    R(s, x);
}
cout<<s;
```

Здесь введены три варьируемых поля, заданные функциями:

- $Z(s)$ представляет собой инициализацию переменной s ;
- $Q(x)$ соответствует условному выражению для числа x ;
- $R(s, x)$ определяет действие над числом x .

Для поля $Z(s)$ возможны следующие конкретные значения: $s=0$; $s=1$; для инициализации суммы, количества или произведения чисел.

Поле $Q(x)$ определяет свойство числа, например проверку:

- четности числа ($x \% 2 == 0$);
- делимости на 3 ($x \% 3 == 0$);
- является ли число положительным ($x > 0$);
- оканчивается ли число цифрой 3 ($x \% 10 != 3$) и т. д.

Поле $R(s, x)$ задает действие над числами:

- вычисление количества чисел ($s++$);
- вычисление суммы чисел ($s+=x$);
- вычисление произведения чисел ($s*=x$).

Применение схемы рассмотрим на двух примерах. Первый пример демонстрирует конструирование тестового задания в закрытой форме. Второй пример является вариантом открытого тестового задания, построенного по обобщенной схеме.

Пример 1.

Дана последовательность, состоящая из N целых чисел. Необходимо найти сумму четных чисел последовательности.

```
cin >> n;
int s=0;
for (i=0; i<n; i++)
{
    cin>>x;
    if (Q(x))
        s+=x;
}
cout <<s;
```

Необходимо указать номер варианта ответа, содержащий строку, соответствующую значению функции $Q(x)$:

- 1) ($x \% 10 == 2$)
- 2) ($x \% 2 != 0$)
- 3) ($x \% 2 == 0$)
- 4) ($x > 0$)

Пример 2.

Программа написана с ошибкой. На вход подается последовательность, состоящая из N целых чисел. Необходимо найти количество положительных чисел последовательности. Найдите и исправьте ошибку в программе.

```
cin >> n;
int s=0;
for (i=0; i<n; i++)
{
    cin>>x;
    if (x>0)
        s+=x;
}
cout <<s;
```

Такая форма задания предполагает открытую форму ответа. Обучаемый должен указать неверную строку и предложить строку-замену.

Далее конструируются обобщенные схемы более высокого уровня абстракции, не зависящие от языка программирования. Рассмотрим подобную конструкцию, соответствующую схеме 1.

Схема 3.

Обработка последовательности чисел, содержащей N вводимых значений.

```
Ввод количества чисел n;
Z(s);
Цикл с параметром
{
    Ввод числа x;
    Если (Q(x)) то R(s, x);
}
Вывод результата s;
```

Такие схемы позволяют абстрагироваться от конкретного языка программирования, и при наличии готовых обобщенных схем одного языка легко сгенерировать аналогичные схемы для другого языка программирования. Подобные схемы можно предлагать на факультативных занятиях со школьниками, при подготовке к олимпиадам. В этом случае необходимо знание хотя бы двух языков программирования, например Pascal и C++.

Следующий пример демонстрирует построение обобщенной схемы для заданного языка программирования на основе абстрактной схемы 3.

Схема 4.

Последовательность чисел содержит N вводимых значений. Схема для языка программирования Pascal:

```
readln(n);
Z(s);
for i:=1 to n do
begin
    readln(x);
    if (Q(x)) then
        R(s, x);
end;
writeln(s);
```

Выводы.

Навык конструирования собственных учебных материалов позволяет учителю расширять дидактическое обеспечение курса, выстраивать индивидуальные траектории обучения, реализовывать творческие и исследовательские способности. Использование обобщенных схем при формировании навыков алгоритмизации и программирования предоставляет возможность, с одной стороны, доступно объяснить алгоритмы решения базовых задач, научить блочному программированию, с другой стороны, за счет автоматизации структуры обобщенных схем конструировать однотипные тестовые задания. Соотнесение обобщенных схем с решением конкретной задачи раскрывает структуру алгоритма, отмечает базовые операторы и структуры данных, выделяет варьируемые фрагменты, зависящие от текущей задачи. Объединение схем в иерархические конструкции позволяет легко повышать уровень сложности решаемых задач.

Список использованных источников

1. Грушевский С. С., Архипова А. И. Электронные образовательные ресурсы инновационной компьютерной дидактики как средство информатизации педагогического образования // Историческая и социально-образовательная мысль. 2014. № 1. С. 75–81.
2. Добровольская Н. Ю. Формирование умения формального исполнения алгоритма как основы алгоритмических навыков учащихся // Преподавание математики и информатики

в школе и вузе: Материалы межвузовской научно-практической конференции. Краснодар: Изд-во Куб. гос. ун-та, 2017. С. 56–58.

3. Добровольская Н. Ю., Харченко А. В. Применение информационных технологий в обучении // Актуальные проблемы информационно-правового пространства: Сборник статей по материалам ежегодных Всероссийских научно-практических конференций. Краснодар: Юг, 2017. С. 28–31.

4. Добровольская Н. Ю., Харченко А. В. Применение технологии фасетов при изучении основ программирова-

ния // Математическое образование в школе и вузе: теория и практика (MATHEDU-2014): Материалы IV Международной научно-практической конференции, посвященной 210-летию Казанского университета и Дню математики (г. Казань, 28–29 ноября 2014 года). Казань: Изд-во Казан. ун-та, 2014. С. 231–233.

5. Харченко А. В. Опыт творческой педагогической деятельности при конструировании учебных задач на основе фасетной технологии // Проблемы современного педагогического образования. 2017. № 57(2). С. 265–272.

НОВОСТИ

Четыре золотые медали завоевали российские школьники на Международной олимпиаде по информатике

В 31-й Международной олимпиаде по информатике, которая состоялась в Баку (Азербайджан), приняли участие 327 школьников из 87 стран мира.

Российская сборная завоевала четыре золотые медали и заняла первое место в медальном зачете.

Медалей удостоены:

- Ильдар Гайнуллин (лицей имени Н. И. Лобачевского Казанского (Приволжского) федерального университета, Казань) — золото;
- Егор Лифарь (школа «Интеллектуал», Москва) — золото;
- Александр Морозов (Президентский физико-математический лицей № 239, Санкт-Петербург) — золото;
- Владимир Романов (школа-интернат имени А. Н. Колмогорова Московского государственного университета имени М. В. Ломоносова, Москва) — золото.

Членов сборной и их наставников поздравила министр просвещения Российской Федерации О. Ю. Васильева.

«Очередная победа российской сборной на Международной олимпиаде по информатике — это не просто успех ребят и их тренеров, заслуги которых в этом году вновь трудно переоценить. Это наглядная демонстрация сложившейся в России системы обучения и подготовки в области программирования, в сфере информационных технологий. Уже с первых классов у нас по всей стране проходят профильные занятия и уроки, которые позволяют ребятам раскрыть свой талант с самых юных лет, вовлекая их в эту сферу. Поздравляю нашу сборную и всех причастных с блистательным выступлением», — сказала О. Ю. Васильева.

Руководителем сборной выступил доцент факультета информационных технологий и программирования Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики А. С. Станкевич, его заместитель — проректор по международным программам и цифровым инновациям Московского физико-технического института А. В. Малеев.

Отбор и подготовка национальной сборной проходили на базе Московского физико-технического института.



(По материалам, предоставленным пресс-службой Минпросвещения России)