

МЕТОДИКА ВЫКЛАДАННЯ ІНФАРМАТЫКІ

УДК 004.4(07)

А. И. Лапо,
соискатель кафедры информатики
и основ электроники БГПУ, учитель информатики Лицея БГУ

МЕТОДИКА ПОСТРОЕНИЯ СИСТЕМ ЗАДАЧ ПО ПРОГРАММИРОВАНИЮ В ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ

Введение. Программирование как раздел школьной информатики достаточно долго изучался только на повышенном и углубленном уровнях изучения информатики. Это наложило свой отпечаток как на подбор задач, решаемых школьниками, так и на методику обучения основам программирования в школьном курсе информатики, поскольку чаще всего школьников обучали так же, как и студентов младших курсов.

После изменения школьной программы в 2008 г. программирование стало изучаться в рамках базового курса информатики с седьмого класса. Данная тема вызывает трудности у учителей при ее преподавании и у учащихся при ее изучении. Проблемы подготовки школьных учителей в области преподавания программирования рассмотрены в работе [1].

Основными причинами и проблемами, требующими решения, являются:

- отсутствие методик обучения программированию в базовой школе для учащихся 7–9 классов;
- отсутствие опыта преподавания программирования у многих учителей;
- отсутствие системы задач, удовлетворяющей образовательным запросам и возрастным особенностям учащихся, особенно в 7–9 классах.

Изучение программирования занимает в курсе информатики порядка 30 % учебного времени, и основная часть этого времени отводится решению задач.

Задачный подход обладает значительным потенциалом для формирования личностных образовательных результатов учащихся, но данный потенциал не реализуется в полной мере в связи с отсутствием научно обоснованной методики построения системы задач по программированию и ее реализации в рамках школьного курса информатики.

Основная часть. Построенная система задач должна удовлетворять принципам **открытости, сложности и нелинейности**. Эти принципы определены в работах С. М. Окулова [2–3]. Построенная система задач должна содержать **наглядные и понятные** примеры, «и чем младше школьники, тем больше примеров необходимо приводить при описании языка» [4]. Согласно теории Балла [5], система задач должна содержать **индивидуальные задачи и родовые**, каждой из которых соответствует некоторый класс индивидуальных задач.

При построении системы задач по определенной теме можно выделить следующие этапы:

- определение фундаментальных составляющих системы (С. М. Окулов называет их аттракторами [2–3]). Они определяются целями, задачами и содержанием учебной программы;
- выбор родовых задач, которые покрывают фундаментальные составляющие системы. Обучение решению таким задачам осуществляется «при помощи подражания и опыта». Это этап линейного развития системы, который готовит скачек – переход к возможности решения индивидуальных задач;
- разработка индивидуальных задач, соответствующих родовым. Решение индивидуальных задач должно основываться на решении родовых. Но вместе с тем индивидуальные задачи должны иметь различный уровень сложности, для того чтобы обеспечить личностное развитие учащегося. Только в этом случае принцип нелинейности обеспечит переход на новый уровень;
- разработка индивидуальных задач, требующих применения опыта решения родовых задач в измененных или новых условиях.

Типы задач

При построении системы используются задачи разных типов (примеры задач в приложении, для индивидуальных задач приведены 3 варианта, хотя на практике их количество можно увеличить до 10–15).

А. Задача с разобранным алгоритмом и готовой программой.

В. Задача с частично описанным алгоритмом и программой с пропусками

С. Задача, аналогичная задаче с разобранным алгоритмом и готовой программой.

Д. Задача, требующая применения стандартных подходов, но без разбора аналогичной.

Е. Нестандартная задача, задача повышенного уровня сложности.

Для родовых задач обязательна разработка задач типа А (задача с разобранным алгоритмом и готовой программой).

Задачи типа С (задача, аналогичная задаче с разобранным алгоритмом и готовой программой) и D (задача, требующая применения стандартных подходов, но без разбора аналогичной) разрабатываются как индивидуальные. Количество индивидуальных вариантов может соответствовать количеству учащихся в группе.

Задачи типа В (задача с частично описанным алгоритмом и программой с пропусками) разрабатываются в том случае, когда уровень подготовки учащихся низкий или тема достаточно сложная и плохо усваивается учащимися.

Задачи типа Е разрабатываются для учащихся с высоким уровнем подготовки.

Уровни сложности задач. Кроме классификации задач по типам, предлагается классификация задач по уровням сложности. Выделим следующие уровни усвоения учебного материала:

- 1) репродуктивный, низкий, узнавание, припоминание;
- 2) репродуктивный, удовлетворительный, воспроизведение материала на уровне памяти;
- 3) репродуктивный, средний, понимание учебного материала;
- 4) продуктивный, достаточный, действия в знакомой ситуации, по образцу;
- 5) продуктивный, высокий, действия в измененной ситуации;
- 6) продуктивный, творческий, действия в незнакомой ситуации.

В таблице представлено соответствие задач различных типов и деятельности учащихся при их решении.

| Задание | Уровень усвоения знаний учащихся | Деятельность ученика |
|--|---|---|
| задача с разобранным алгоритмом и готовой программой | 1-й уровень | запустить программу, ввести предложенные тесты |
| | 2-й уровень | подобрать свои тесты, рассматривающие определенные ситуации, описанные в задании |
| | 3-й уровень | объяснение отдельных частей программы, внесение небольших изменений в программу, подбор тестов, не учтенных в описании задания |
| | 4-й уровень | внесение в программу изменений для получения других (похожих) результатов, исследование и объяснение ситуаций «А что будет делать программа, если...» |
| | 5-й уровень | нетривиальное изменение программы, объяснение действия программы при нестандартных исходных данных |
| задача с частично описанным алгоритмом и программой с пропусками | 2–3-й уровни | заполнение пропусков в описании алгоритма и программе, подбор тестов |
| | 4–5-й уровни | деятельность учащихся аналогична работе над задачей с разобранным алгоритмом и готовой программой |
| задача, аналогичная задаче с разобранным алгоритмом и готовой программой | 3-й уровень (варианты 1–4 в заданиях для индивидуализированной работы) | незначительное изменение разобранный задачи для получения решения новой (применяются в основном те же команды, но с другими параметрами) |
| | 4-й уровень (варианты 5–7 в заданиях для индивидуализированной работы) | изменения, которые нужно произвести в программе, требуют от учащихся применения команд, незначительно отличающихся от тех, которые были разобраны |
| | 5-й уровень (варианты 8–10 в заданиях для индивидуализированной работы) | программа требует значительных изменений: ввода дополнительных исходных данных, рассмотрения отдельных ситуаций и др. |

| Задание | Уровень усвоения знаний учащихся | Деятельность ученика |
|---|--|---|
| | 6-й уровень (варианты 11–12 в заданиях для индивидуализированной работы) | задача требует нестандартных действий или достаточно объемна по своей реализации |
| задача, требующая применения нестандартных подходов, но без разбора аналогичной | 4–6-й уровни | деятельность учащихся во многом аналогична деятельности по соответствующим пунктам для предыдущего типа задач, но решение полностью самостоятельно, не имеет образца. В разработанной системе задач предлагается 12 вариантов, однако на практике не все учащиеся способны к решению таких задач, поэтому все 12 вариантов используются только в классах с высоким уровнем подготовки |
| нестандартная задача, задача повышенного уровня сложности | 5–6-й уровни | от учащихся требуется использование не только материала изучаемой темы, но и предыдущего материала, а часто и материала, выходящего за рамки школьной программы. Такие задачи обычно уникальны и тяжело поддаются типизации, поэтому могут предлагаться без вариантов |

Построение системы задач. Рассмотрим построение системы задач на примере изучения линейных алгоритмов в разделе «Основы алгоритмизации и программирования» 7 класс.

Согласно учебной программе в этом разделе изучаются:

- понятие типа данных, числовые типы данных, понятие переменной;
- организация ввода-вывода данных;
- команда присваивания;
- арифметические операции и выражения;
- стандартные функции.

В результате учащиеся должны уметь записывать арифметические операции и выражения на языке программирования, составлять и реализовывать линейные алгоритмы [3].

Фундаментальной составляющей (аттрактором) данного раздела будет задача: «Найти значения арифметического выражения».

Родовые задачи (им соответствуют задачи типа А и В):

- нахождение значения арифметического выражения (задание 1);
- простейшее моделирование на основе задач из курсов геометрии, физики и др. (задание 3);
- использование целочисленных операций на примере выделения цифр из числа (задание 5).

Индивидуальные задачи разрабатываются для каждого типа родовой задачи (тип С – это задания 2, 4, 6).

В качестве задач типа D можно предложить:

- задачи на преобразование величин (часы в минуты, километры в метры, мегабайты в байты и др.) – родовой является задача на использование целочисленных операций (задание 7);

- задачи на взаимосвязь времени на часах со стрелками и углами поворота стрелок – родовыми являются задачи на использование целочисленных операций и простейшее моделирование.

В качестве задачи типа E предлагается задача о получении степени числа за определенное количество умножений (задание 8).

Принципы фундаментальности и открытости позволят легко перенести построенную систему задач в новые условия, например, при изучении второго языка программирования в старших классах. Оставив неизменной фундаментальную составляющую – нахождение значения арифметического выражения, можно изменить состав родовых и индивидуальных задач. Принцип сложности позволит подобрать их в соответствии с возрастными и индивидуальными особенностями развития учащихся.

Важной составляющей задач типа А является тестирование программы. Задания, собранные в этом пункте, позволяют не только проверить правильность написания программы, но и уровень понимания учащимся того, что и как делает данная программа.

Для выполнения всех пунктов, а они подбираются в соответствии с уровнями усвоения учебного материала, учащийся должен уметь «читать» программу. Это умение является не менее важным, чем умение писать программу (недаром при обучении грамоте сначала учат читать, а потом писать). Подбор собственных тестов, проверка работы программы в ситуации «а что будет, если...», изменение программы для нестандартных ситуаций той же самой задачи могут заставить учащегося взглянуть на задачу по-другому, перейти от решения родовой задачи к индивидуальным. Все это дает право говорить

о «неисчерпаемости даже одной задачи, не говоря уже о серии или пучке. И эта неисчерпаемость приводит к возвратам на новом витке развития, к нелинейности в разработке каждой конкретной программы» [2].

Заключение. Описанная методика дает практический инструментарий для построения систем задач по программированию в школьном курсе информатики. Построенная система задач не является универсальной. Это один из возможных способов помочь учащимся освоить азы программирования. Использование построенной системы задач позволяет осуществить реализацию деятельностного и дифференцированного подходов в обучении, учета индивидуальных особенностей учащихся.

ПРИЛОЖЕНИЕ

Задание 1а. Даны x, y, z . Написать программу для вычисления $a = \frac{2x + 3y - z}{3 + 2x}$.

Этапы выполнения задания

I. Определение исходных данных: переменные x, y, z .

II. Определение результатов: переменная a .

III. Алгоритм решения задачи.

1. Ввод исходных данных.
2. Вычисление значения выражения.
3. Вывод результата.

IV. Описание переменных.

Все переменные, определенные для решения задачи, имеют тип *real*. (Переменные x, y, z могут быть и целого типа, а переменная a только вещественного типа, поскольку в выражении используется операция деления.)

V. Программа:

```
Var x,y,z,a: real;
Begin
  Write('введите значение x = ');
  Readln(x);
  Write('введите значение y = ');
  Readln(y);
  Write('введите значение z = ');
  Readln(z);
  a := (2*x + 3*y - z)/(3 + 2*x);
  Writeln('a = ',a);
```

End.

VI. Тестирование программы.

1. Запустите программу и введите значения $x = 3, y = 2, z = 0$

ЛИТЕРАТУРА

1. Лапо, А. И. Проблемы подготовки школьных учителей информатики в области программирования / А. И. Лапо // Сборник науковых прац академії паслядыпломной адукацыі. – 2013. – Вып. 11. – С. 284–298.
2. Окулов, С. М. Информатика: развитие интеллекта школьников / С. М. Окулов. – М.: БИНОМ. Лаборатория знаний, 2008. – 212 с.
3. Окулов, С. М. Развитие интеллекта школьника как принцип организации синергетической среды обучения информатике. Автореферат диссертации на соискание ученой степени доктора педагогических наук / С. М. Окулов. – Киров, 2004. – 56 с.
4. Андреева, Е. В. Методика обучения основам программирования на уроках информатики. Лекции 1–4 / Е. В. Андреева – М.: Педагогический университет «Первое сентября», 2006. – 60 с.
5. Балл, Г. А. Теория учебных задач: Психолого-педагогический аспект / Г. А. Балл. – М.: Педагогика, 1990. – 184 с.
6. Информатика. VI–XI классы : учебная программа. – Минск : НИО, 2009. – 24 с.

Проверьте, результат должен быть следующим: $a = 1.333$.

Проверить правильность вычислений можно на калькуляторе.

2. Заполните таблицу

| | x | y | z | a |
|---|---------|--------|--------|-----|
| 1 | 2 | -5 | 1 | |
| 2 | 2,362 | 4,258 | -7,458 | |
| 3 | -1,2358 | -4,554 | 5,562 | |

3. Подберите такие значения x, y, z , чтобы значение $a = 2,000$.

4. Что изменится и почему, если в программе заменить команду `Writeln('a = ',a)`; на команду `Writeln('a = ', a:7:3)`;

5. Как распечатать только целую часть результата?

Задание 1б. Даны x, y . Написать программу для вычисления $b = \frac{\sqrt{x+4}}{|y|-2}$.

Этапы выполнения задания

I. Определение исходных данных: переменные _____.

II. Определение результатов: переменная _____.

III. Алгоритм решения задачи.

1. Ввод исходных данных.
2. Вычисление значения выражения.
3. Вывод результата.

IV. Описание переменных:

Все переменные, определенные для решения задачи, имеют тип _____

V. Программа:

```
Var x,y,a: real;
Begin
  Write('введите значение ____');
  Readln(____);
  Write('введите значение ____');
  Readln(____);
  b: = _____;
  writeln(_____);
End.
```

VI. Тестирование программы.

1. Запустите программу и введите значения $x = 9, y = -7$.

Проверьте результат должен быть следующим: $a = 1,000$.

2. Заполните таблицу:

| | X | Y | B |
|---|---------|--------|---|
| 1 | 16 | -5 | |
| 2 | 2,362 | 4,258 | |
| 3 | -1,2358 | -4,554 | |

3. Подберите такие значения x, y , чтобы значение $b = 2.000$.

4. Можно ли объединить в одну программу вычисление двух выражений (задания 1а и 1б)? Как это сделать?

Задание 2. Даны x, y, z . Написать программу для вычисления a и b .

$$1. a = y + \frac{x}{y^2 + \frac{x^2}{y^3}}; b = \frac{\sqrt{x+y}}{\sqrt{x+y} + \sqrt{y}}$$

$$2. a = y^2 + \frac{x^2}{y - \frac{x^2}{y^3}}; b = \sqrt{x+y} \cdot \frac{x-y}{2+x+y^2}$$

$$3. a = \frac{3+x^2-y^3}{x-y^2} (x+3)^2;$$

$$b = |x+y| \frac{\sqrt{x^2+y^2+z^2}}{\sqrt{|x|} + \sqrt{|y|} + \sqrt{|z|}}$$

Задание 3. Напишите программу для решения геометрической задачи.

Образец выполнения задания.

Задан квадрат с длиной стороны a . Найти его площадь и длину диагонали.

Этапы выполнения задания

I. Определение исходных данных: переменная a (длина стороны).

II. Определение результатов: переменные S (площадь) и d (длина диагонали).

III. Алгоритм решения задачи.

1. Ввод исходных данных.
2. Вычисление значений площади по формуле $S = a^2$ и длины диагонали по формуле $d = a\sqrt{2}$.
3. Вывод результата.

IV. Описание переменных.

Все переменные, определенные для решения задачи, имеют тип *real*.

V. Программа:

```
Var a,S,d: real;
Begin
  Write('введите значение a = ');
  Readln(a);
  S: = SQR(a);
  D: = a*SQR(2);
  Write('площадь = ',S:7:3);
  Write('длина диагонали = ',d:7:3);
End.
```

VI. Тестирование.

1. Запустите программу и введите значение $a = 2$.

Проверьте: результат должен быть следующим: $S = 4,000; d = 2,828$.

2. Заполните таблицу:

| | a | S | d |
|---|-------|---|---|
| 1 | 5 | | |
| 2 | 2,35 | | |
| 3 | 0,129 | | |

3. Можно ли подобрать такое значение a , чтобы значение d содержало нули в дробной части?

4. Что нужно изменить в программе, чтобы значения a и d выводились в разных строчках?

5. Какие еще возможны варианты записи формулы для вычисления площади?

Задание 4.

1. Даны гипотенуза и катет прямоугольного треугольника. Найти второй катет и радиус вписанной окружности.

2. Равнобедренный треугольник задан длинами основания и высоты. Найти его площадь и длину боковой стороны

3. Треугольник задан длинами сторон a, b, c . Найти длины медиан треугольника, опущенных на стороны треугольника.

Задание 5. Напишите программу для решения задачи. При решении воспользоваться операциями *div* и *mod*.

Образец выполнения задания.

Задано двузначное число. Поменять местами первую и вторую цифры чис-

ла. Например, при вводе числа 25, получаем число 52.

Этапы выполнения задания

I. Определение исходных данных: переменная a (исходное число).

II. Определение результатов: переменная b (преобразованное число).

III. Алгоритм решения задачи.

1. Ввод исходных данных

2. Для преобразования числа необходимо выполнить следующие действия:

а) выделить из исходного числа вторую цифру и сохранить ее в переменной $a1$. Для выделения цифры нужно найти остаток от деления исходного числа на 10.

б) выделить из исходного числа первую цифру и сохранить ее в переменной $a2$. Для выделения цифры нужно найти целую часть от деления исходного числа на 10.

в) получить искомое число b . Для этого нужно умножить на десять значение переменной $a1$ и к полученному произведению прибавить значение переменной $a2$.

3. Вывод результата.

IV. Описание переменных.

Все переменные, определенные для решения задачи, имеют тип *integer*.

V. Программа:

```
Var a,b,a1,a2: integer;
```

```
Begin
```

```
  Write('введите значение a = ');
```

```
  Readln(a);
```

```
  a1:= a mod 10; {Выделение последней
цифры в двузначном числе}
```

```
  a2:= a div 10; {Выделение первой
цифры в двузначном числе}
```

```
  b:= a1*10 + a2;
```

```
  Write('искомое число = ',b);
```

```
End.
```

VI. Тестирование.

1. Запустите программу и введите значение $a = 25$.

Проверьте: результат должен быть следующим: искомое число = 52.

2. Заполните таблицу:

| | a | B |
|---|----|---|
| 1 | 15 | |
| 2 | 46 | |
| 3 | 63 | |
| 4 | | |
| 5 | | |

3. Для каких значений переменной a значение переменной b будет таким же? Добавьте в таблицу такое значение.

4. Всегда ли в результате выполнения программы получим двузначное число? Почему? Приведите пример и добавьте его в таблицу.

5. Каким будет результат, если ввести однозначное число? Почему?

6. Попробуйте ввести трехзначное число (например, 125). Объясните получившийся результат.

Задание 6.

1. Задано двузначное число. Найдите сумму цифр числа.

2. Задано двузначное число. Найдите разность между количеством десятков и количеством единиц числа.

3. Задано двузначное число. Получите новое число, в котором количество десятков совпадает с количеством единиц в исходном числе, а количество единиц в два раза меньше количества десятков в исходном числе.

Задание 7. Напишите программу для решения задачи. При решении воспользоваться операциями *div* и *mod*.

1. С начала месяца прошло m часов. Определить, какое сейчас число.

2. С начала суток прошло k минут. Определить, который сейчас час (в часах и минутах).

3. С начала месяца прошло t секунд. Определить, какое сейчас число и который сейчас час (в часах, минутах, секундах).

Задание 8. Напишите программу для решения задачи.

1. Определить полное количество часов, прошедших от начала суток до того момента (в первой половине суток), когда часовая стрелка повернулась на f градусов ($0 \leq f < 360$, f – целое число).

2. Определить полное количество минут, прошедших от начала часа до того момента (в первой половине суток), когда минутная стрелка повернулась на f градусов ($0 \leq f < 360$, f – целое число).

3. Определить f – угол (в градусах) между положением часовой стрелки в начале суток и ее положением в h часов m минут ($0 \leq h < 12$, $0 \leq m < 60$). На какой угол повернута минутная стрелка?

Задание 9. Написать программу для решения задачи.

1. Дано действительное число a . Не пользуясь никакими другими арифметическими действиями кроме умножения, получить a^9 за четыре операции.

2. Дано действительное число a . Не пользуясь никакими другими арифметическими действиями кроме умножения, получить a^{15} за пять операций.

3. Дано действительное число a . Не пользуясь никакими другими арифметическими действиями кроме умножения, получить a^4 , a^{12} и a^{28} за шесть операций.

SUMMARY

This article presents the method of tasks construction. The types of tasks and relevant levels for the students are defined in this article. Using this technique, the system of problems tasks is built for the topic «Basics of algorithms and programming. Linear algorithms» appropriate the training program at grade 7.

Поступила в редакцию 25.04.2014 г.

Репозіторій БДПУ