

DOI: 10.32517/2221-1993-2025-24-5-44-51

**Н. И. Заводчикова***Ярославский государственный педагогический университет им. К. Д. Ушинского, г. Ярославль, Россия;
Средняя школа № 76, г. Ярославль, Россия***И. А. Быкова***Ярославский государственный педагогический университет им. К. Д. Ушинского, г. Ярославль, Россия;
Средняя школа № 26, г. Ярославль, Россия*

ОСОБЕННОСТИ МЕТОДИКИ РАБОТЫ С ЗАДАЧЕЙ ПО ПРОГРАММИРОВАНИЮ ПРИ РЕАЛИЗАЦИИ СЕМИОТИЧЕСКОГО ПОДХОДА К ОБУЧЕНИЮ*

Аннотация

На первых этапах обучения программированию методика работы с задачей, в которой учтены особенности организации семиотической деятельности школьников, позволяет предупредить формализм в усвоении знаний. Работа над задачей включает несколько этапов, на которых учащиеся оперируют с различными знаково-символическими системами. На этапе моделирования действий с конкретными входными данными используются знаково-символические системы, связанные с условием задачи; при построении схемы алгоритма — естественный язык и язык блок-схем; при кодировании — знаково-символическая система, определяемая языком программирования.

Целесообразно при управлении деятельностью учащихся по поиску решения задачи использовать набор упражнений, включающий задания на моделирование действий с конкретными входными данными; на определение правил изменения величин и формулировку алгоритма; на сигнификацию выделенных величин, определение их типа и назначения в программе; на визуализацию схемы алгоритма; на замещение элементов схемы командами языка программирования; на разработку программы для реализации алгоритма.

В статье приводится пример набора упражнений указанной структуры для организации работы учащихся над задачей по теме «Обработка строковых величин» в VII классе.

Ключевые слова: семиотический подход к обучению программированию, знаково-символическая деятельность, цифровой образовательный контент, обработка строковых величин, обучение программированию.

Информация об авторах

Заводчикова Надежда Ивановна, канд. пед. наук, доцент, доцент кафедры теории и методики обучения информатике, физико-математический факультет, Ярославский государственный педагогический университет им. К. Д. Ушинского, г. Ярославль, Россия; учитель информатики, средняя школа № 76, г. Ярославль, Россия; *e-mail*: zaw-nadejda@yandex.ru

Быкова Ирина Альбертовна, старший преподаватель кафедры теории и методики обучения информатике, физико-математический факультет, Ярославский государственный педагогический университет им. К. Д. Ушинского, г. Ярославль, Россия; учитель информатики, средняя школа № 26, г. Ярославль, Россия; *e-mail*: i.bukova@yandex.ru

* Материалы к статье можно скачать на сайте ИНФО: https://infojournal.ru/journals/school/school_05-2025/

N. I. Zavodchikova

Yaroslavl State Pedagogical University named after K. D. Ushinsky, Yaroslavl, Russia;
School 76, Yaroslavl, Russia

I. A. Bikova

Yaroslavl State Pedagogical University named after K. D. Ushinsky, Yaroslavl, Russia;
School 26, Yaroslavl, Russia

SPECIFICS OF THE METHODOLOGY FOR WORKING WITH PROGRAMMING PROBLEMS WHEN IMPLEMENTING A SEMIOTIC APPROACH TO LEARNING

Abstract

In the initial stages of teaching programming, a methodology for working with problems that considers the specifics of organizing school students' semiotic activity helps to prevent formalism in knowledge acquisition. Problem-solving involves several stages where students operate with various sign-symbolic systems. At the stage of modeling actions with specific input data, sign-symbolic systems related to the problem condition are used; when building an algorithm flowchart — natural language and the flowchart language; during coding — the sign-symbolic system defined by the programming language. It is advisable to use a set of exercises to guide students' problem-solving activities.

This set should include tasks on modeling actions with specific input data; determining rules for changing values and formulating the algorithm; signifying identified variables, defining their type and purpose in the program; visualizing the algorithm flowchart; substituting flowchart elements with programming language commands; and developing a program to implement the algorithm.

The article provides an example of an exercise set with the specified structure for organizing student work on a problem related to "String processing" in the 8th grade.

Keywords: semiotic approach to teaching programming, sign-symbolic activity, digital educational content, string processing, programming education.

Information about the authors

Nadejda I. Zavodchikova, Candidate of Sciences (Education), Docent, Associate Professor at the Department of Theory and Methods of Teaching Informatics, Faculty of Physics and Mathematics, Yaroslavl State Pedagogical University named after K. D. Ushinsky, Yaroslavl, Russia; an informatics teacher, School 76, Yaroslavl, Russia; e-mail: zaw-nadejda@yandex.ru

Irina A. Bikova, Senior Lecturer at the Department of Theory and Methods of Teaching Informatics, Faculty of Physics and Mathematics, Yaroslavl State Pedagogical University named after K. D. Ushinsky, Yaroslavl, Russia; an informatics teacher, School 26, Yaroslavl, Russia; e-mail: i.bukova@yandex.ru

1. Введение

Раздел «Алгоритмы и программирование» является одним из основных в школьном курсе информатики [3, 17]. Л. Л. Боссова отмечала, что изучение программирования развивает мышление человека, способствует формированию у людей новых ценностей цифрового общества, пониманию правил поведения в цифровой среде [2]. В основной школе согласно федеральной рабочей программе изучение программирования предусмотрено в VIII—IX классах на базовом уровне и в VII—IX классах — на углубленном [17]. В качестве первого языка для изучения программирования в последнее время выбирают язык Python [4, 5, 12]. Как правило, используется задачный подход в организации учебной деятельности, поэтому вопросы, связанные с различными аспектами методики работы с задачей, являются актуальными.

Традиционно выделяют следующие этапы работы с задачей по программированию:

- анализ условия задачи и поиск ответа на вопрос задачи для некоторых входных данных;
- формализация условия;
- разработка алгоритма;
- программирование алгоритма;
- отладка и тестирование программы.

С психологической точки зрения при прохождении этих этапов у учащихся происходит совмещение различных планов: действия с конкретными данными, знаково-символическое описание этих данных и структуры алгоритма, написание кода программы [9, 10, 13]. При этом учащиеся оперируют с несколькими знаково-символическими системами, что заставляет обратить внимание на особенности организации семиотической деятельности школьников.

2. Структура набора упражнений для решения задачи по программированию

Н. Г. Салмина выделяет следующие виды знаково-символической деятельности: моделирование, схематизация, кодирование и замещение [14]. При решении задач по программированию указанные виды деятельности осуществляются условно последовательно и соответствуют этапам перехода от действий с частными случаями входных данных через модель изменения значений величин к алгоритму решения и от алгоритма к программе (рис. 1). В данную схему не включен этап отладки и тестирования.

На каждом из этапов используются различные знаково-символические системы. Условно можно считать, что система знаков, соответствующая определенному языку программирования, используется для создания собственно кода программы, т. е. на этапе кодирования. На этапе схематизации алгоритма решения задачи школьники оперируют элементами естественного языка и языка блок-схем. На этапе моделирования действий с частными случаями входных данных используются знаково-символические системы, связанные с условием задачи.

Исследования Г. А. Глотовой, Н. Г. Салминой, Е. Е. Сапоговой показали, что моделирование является наиболее сложным видом знаково-символической деятельности [8, 14, 15]. Вероятно, поэтому основные затруднения возникают у школьников на этапе перехода от решения задачи для частных случаев входных данных к построению модели изменения значений величин, участвующих в решении задачи.

Как отмечено нами в [10], на первых этапах обучения программированию целесообразно использовать набо-

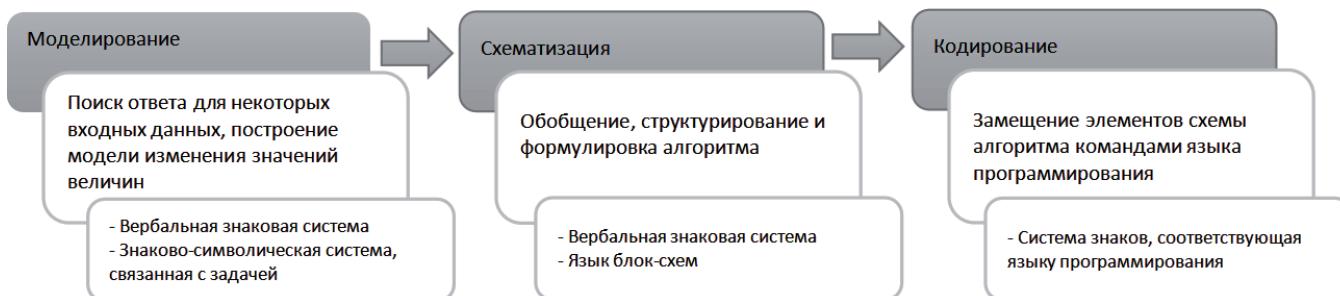


Рис. 1. Обобщенная схема решения задачи по программированию

ры взаимосвязанных упражнений, в которых выделены отдельные этапы решения задачи. В описанных нами ранее упражнениях [6, 10, 11] решение задачи начинается с моделирования изменения значения величин, участвующих в решении задачи. Управление этим процессом осуществляется с помощью таблицы, где столбцы соответствуют величинам, использующимся при решении, а в ячейках отображается процесс изменения значений этих величин. При этом заголовки столбцов таблицы определяются учителем, также подразумевается цикличность выполняемых действий.

В указанных упражнениях не обеспечивается активность школьника в определении характера последовательности действий, выполняемых при поиске ответа на вопрос задачи для частных случаев входных данных, и выделении величин. Для стимулирования познавательной активности учащихся при анализе существенных свойств последовательности выполняемых действий можно предложить им выбрать визуальный аналог описания этой

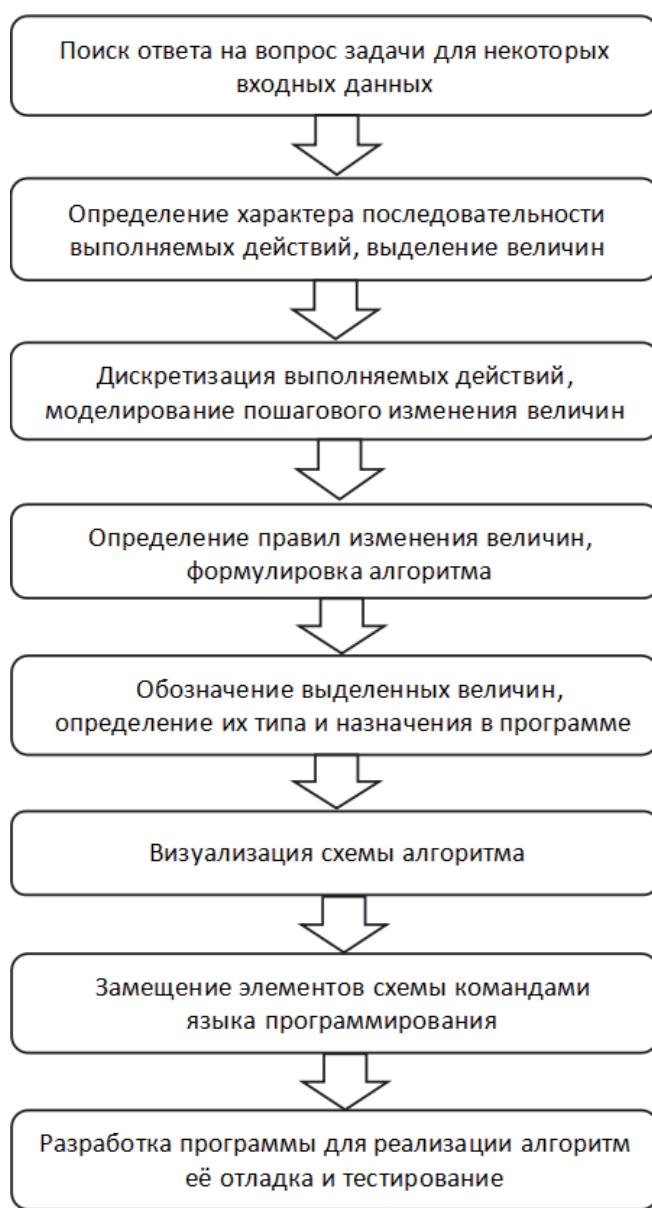


Рис. 2. Последовательность действий учащихся при решении задачи по программированию

последовательности. Практика обучения программированию показывает, что для этого удобно использовать блок-схемы. Выделение величин должно происходить во внешней речи, которая служит средством обобщения и фиксирования выполняемых действий. Добавление в набор упражнений заданий на поиск ответа на вопрос задачи для конкретных входных данных и анализ этого процесса будет способствовать сопоставлению учащихся различных планов: «действия», «слова» и «образа».

На рисунке 2 приведена последовательность действий учащихся при решении задачи по программированию.

Таким образом, наборы упражнений для решения задачи по программированию должны содержать:

- задания на нахождение ответа на вопрос задачи для конкретных входных данных;
- задания на определение характера последовательности действий и выделение величин, участвующих в решении задачи;
- задания на моделирование пошагового изменения значений выделенных величин;
- задания на определение правил изменения величин и формулировку алгоритма;
- задания на сигнификацию выделенных величин, определение их типа и назначения в программе;
- задание на визуализацию схемы алгоритма;
- задания на замещение элементов схемы командами языка программирования;
- задание на разработку программы для реализации алгоритма, ее отладку и тестирование.

3. Пример набора упражнений для управления процессом решения задачи

Проиллюстрируем уточненную схему методики работы с задачей по программированию на примере следующей задачи:

«На вход программе подается строка *s*. Выведите строку, в которой каждый символ строки *s* удваивается. Например, для строки 'bit' программа должна вывести 'bbiitt'».

Задание 1.

Какую строку должна вывести программа для исходной строки:

- «лето»;
- «алгоритм»?

Вопросы для обсуждения:

- Какая величина является исходной (аргументом)? Назовите ее тип.
- Какая величина является искомой (результатом)? Назовите ее тип.
- Необходимы ли для работы программы какие-либо промежуточные величины?

Задание 2 (рис. 3).

Отметьте, какие алгоритмические конструкции вы использовали, чтобы найти ответ. Какие величины участвуют в решении задачи?

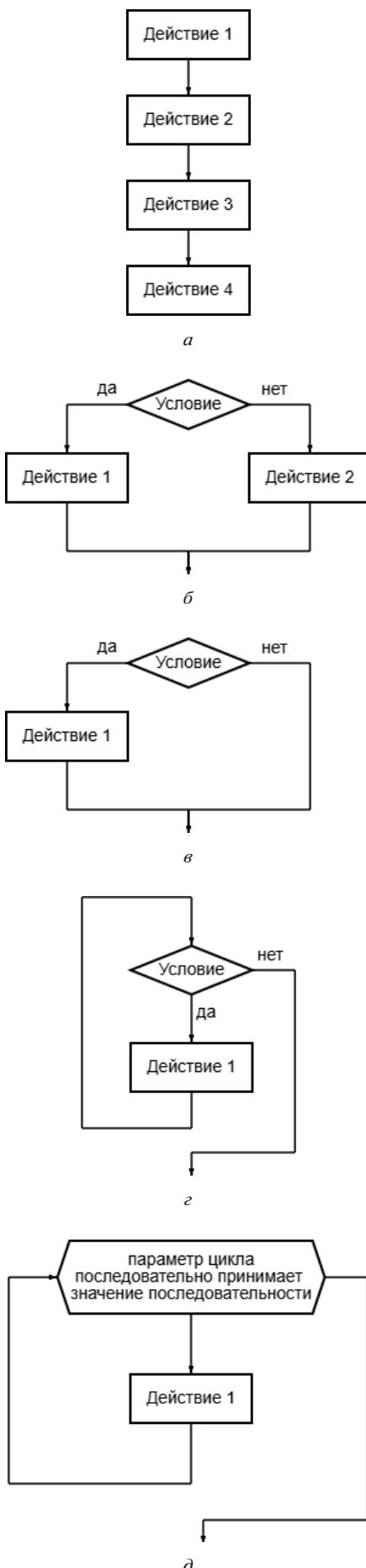


Рис. 3. Рисунок к заданию 2

Вопросы для обсуждения:

- Будет ли меняться блок-схема *a* при изменении длины входной строки?
- Осуществляли ли вы проверку какого-либо условия при выполнении задания 1?
- Можно ли в ваших действиях при выполнении задания 1 выделить повторяющиеся шаги? Какие шаги повторялись? Что в этих шагах было общего, а что менялось?
- Какая величина будет являться параметром цикла? Как она будет изменяться?
- Какие действия совершаются в теле цикла?
- Какие величины участвуют в решении задачи?

Задание 3.

Заполните пропуски в таблице изменения значений величин (табл. 1).

Таблица 1

Таблица к заданию 1

Исходная строка		
лето		

Вопросы для обсуждения:

- Как необходимо озаглавить второй столбец таблицы? Третий?
- Какие значения примут величины на первом шаге цикла? На втором?

Ответ представлен в таблице 2.

Таблица 2

Ответ к заданию 3

Исходная строка	Текущий символ	Искомая строка
лето	л	лл
	е	лле
	т	ллеет
	о	ллеетто

Задание 4.

Сформулируйте правила изменения величин (табл. 3).

Таблица 3

Таблица к заданию 4

Величина	Правило
Исходная строка	

Примерный вариант ответа представлен в таблице 4.

Таблица 4

Примерный вариант ответа к заданию 4

Величина	Правило
Исходная строка	Вводится и далее не меняется
Текущий символ	Последовательно принимает значение очередного символа исходной строки
Искомая строка	На каждом шаге дописываются справа два текущих символа

Задание 5 (вариант 1).

Сформулируйте алгоритм для решения задачи.

Задание 5 (вариант 2).

Заполните пропуски в алгоритме решения:

Ввод исходной строки

_____ все символы строки s
добавить _____
в исходную строку.

Вывести исходную строку.

Примерный вариант ответа.

Ввод исходной строки

Инициализация искомой строки

Последовательно перебрать все символы строки s
добавить **справа два текущих символа** в ис-
ходную строку.

Вывести исходную строку.

Задание 6.

Выберите имена для выделенных величин, укажите их тип и назначение в программе (табл. 5).

Таблица 5

Таблица к заданию 6

Величина	Имя	Тип	Назначение
Исходная строка	s		аргумент

Примерный вариант ответа представлен в таблице 6.

Таблица 6

Примерный вариант ответа к заданию 6

Величина	Имя	Тип	Назначение
Исходная строка	s	строковый	аргумент
Текущий символ	symb	строковый	промежуточная
Искомая строка	s1	строковый	результат

Задание 7 (вариант 1).

Нарисуйте блок-схему для решения задачи.

Задание 7 (вариант 2).

Заполните пропуски в блок-схеме (рис. 4).

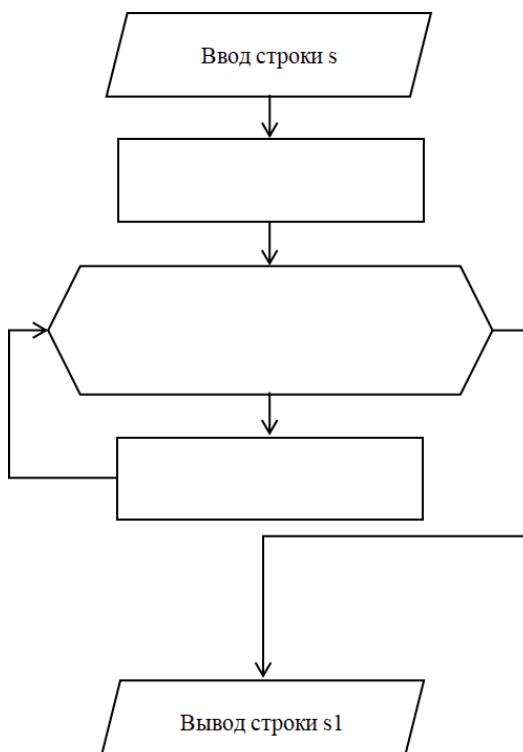


Рис. 4. Рисунок к заданию 7

Примерный вариант ответа представлен на рисунке 5.

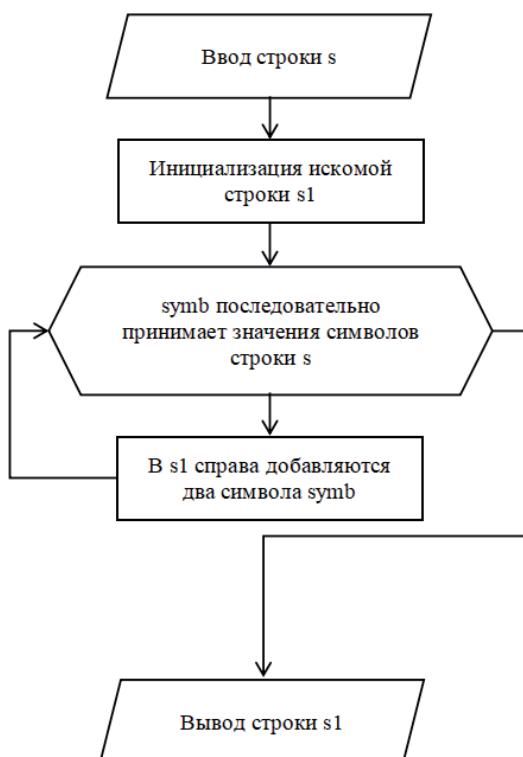


Рис. 5. Примерный вариант ответа на задание 7

Задание 8 (вариант 1).

Отметьте команды, которые описывает правило, по которому происходит изменение величин (табл. 7).

Таблица 7

Таблица к заданию 8

Правило	Команда
Ввод строки <i>s</i>	а) <code>s = print()</code> б) <code>input(s)</code> в) <code>s1 = input()</code> г) <code>s = input()</code>
Инициализация строки <i>s1</i>	а) <code>s1 = 0</code> б) <code>s1 = ''</code> в) <code>s1 == ''</code> г) <code>s1 = s</code>
<i>symb</i> последовательно принимает значение очередного символа исходной строки	а) <code>for symb in s:</code> б) <code>if symb in s:</code> в) <code>for symb in range(len(s)):</code> г) <code>for s in symb:</code>
Добавить в <i>s1</i> справа два текущих символа <i>symb</i>	а) <code>s1 = s1 + symb + symb</code> б) <code>s1 += symb * 2</code> в) <code>s1 = symb + s1</code> г) <code>symb += s1</code>
Вывод строки <i>s1</i>	а) <code>s1 = print()</code> б) <code>input(s1)</code> в) <code>print(s)</code> г) <code>print(s1)</code>

Задание 8. (вариант 2).

Напишите программу для решения задачи.

Задание 9.

Напишите программы для решения следующих задач:

- На вход программе подается строка *s*. Выведите строку, в которой каждый символ строки *s* утраивается. Например, для строки 'bit' программа должна вывести 'bbbiiittt'.
- На вход программе подается строка *s* и число *k*. Выведите строку, в которой каждый символ строки *s* повторяется *k* раз. Например, для строки 'bit' и *k* = 5 программа должна вывести 'bbbbbbiiiiittttt'.
- На вход программе подается строка *s*. Выведите строку, в которой каждый символ строки *s*, кроме первого и последнего, удваивается. Например, для строки 'bit' программа должна вывести 'bii'.
- На вход программе подается строка *s*. Выведите строку, в которой каждый символ строки *s*, являющийся гласной буквой, удваивается. Например, для строки 'bit' программа должна вывести 'biit'.
- На вход программе подается строка *s*. Выведите строку, которая получится при удалении из строки *s* всех гласных букв. Например, для строки 'bit' программа должна вывести 'bt'.

Задание 10.

Что будет выведено на экран в результате работы алгоритма?

а)

```
s='abc'
s1 = ''
for x in s:
    s1 = s1 + x
print(s1)
```

б)

```
s='abc'
s1 = ''
for x in s:
    s1 = s1 + x + x + x
print(s1)
```

в)

```
s='abc'
s1 = ''
for x in s:
    s1 = x + s1
print(s1)
```

г)

```
s='abc'
s1 = ''
for x in s:
    s1 = x + s1 + x
print(s1)
```

Ответ. а) abc; б) aaabbbcc; в) cba; г) cbaabc.

Рассмотрим назначение каждого из заданий.

Задание 1 содержит два варианта входной строки, что позволяет организовать обсуждение целесообразности выбора схемы линейного алгоритма при выполнении задания 2.

Блок-схемы, предложенные школьникам в задании 2, дают возможность совместить конкретные действия с обобщенным образом этих действий.

Задание 3 способствует переносу действий, совершаемых при выполнении задания 1, во внешний план, в результате чего учащиеся могут сформулировать правила изменения величин (задание 4). Языковые средства на данном этапе, включаясь в состав действий с конкретными данными, выступают как средство обобщения.

Два варианта задания 5 предназначены для учащихся с различным уровнем алгоритмического мышления и общего развития. Необходимо учитывать, что самостоятельная формулировка алгоритма занимает достаточно много времени.

При выполнении задания 6 происходит сигнификация выделенных величин.

В задании 7 с использованием введенных обозначений строится схема алгоритма решения, происходит ее визуальная структуризация, вербальные формулировки сокращаются, отдельные элементы схемы приобретают большую смысловую нагрузку.

Замещение элементов схемы командами языка программирования приводит к написанию кода программы для решения задачи (задание 8).

Дальнейшее закрепление изученного алгоритма происходит при выполнении лабораторной работы (задание 9), которая содержит:

Задача

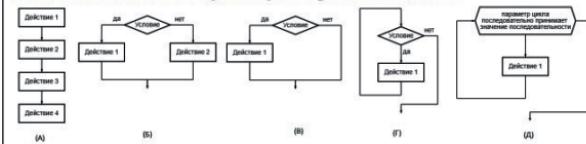
На вход программе подаётся строка S. Выведите строку, в которой каждый символ строки S удваивается.

Например, для строки 'sport', программа должна вывести 'sspoortt'.

Решение

- 1) Для строки 'лето', ответом будет
Для строки 'задача', ответом будет

- 2) Какие алгоритмические конструкции вы использовали чтобы найти ответ? Какие величины участвуют в решении задачи?



- 3) Заполните пропуски в таблице изменения значений величин:

Исходная строка		
лето		

- 4) Сформулируйте правила изменения величин

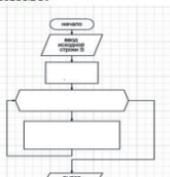
Величина	Правило
Исходная строка	

- 5) Сформулируйте алгоритм для решения задачи

- 6) Выберите имена для выделенных величин, укажите их тип и назначение в программе

Название величины	Имя	Тип	Назначение
Исходная строка	S		аргумент

- 7) Заполните пропуски в блок-схеме:



- 8) Напишите программу для решения задачи

Программа	Экран	Задача
<pre>s='abc' s1 = '' for x in s: s1 = s1 + x + x + x print(s1)</pre>		
<pre>s='abc' s1 = '' for x in s: s1 = x + s1 print(s1)</pre>		
<pre>s='abc' s1 = '' for x in s: s1 = x + s1 + x print(s1)</pre>		

Рис. 6. Фрагмент рабочего листа для управления процессом решения задачи

- задачи, аналогичные разобранной;
- задачи, обобщающие некоторые элементы условия задачи;
- задачи, в которых изменен некоторый элемент условия;
- задачи, в которых требуется выполнить обратные действия.

Задание 10 позволяет проверить обратимость освоенных знаково-символических действий.

4. Рабочий лист как средство управления процессом решения задачи

Большинство из описанных заданий можно оформить в виде интерактивных упражнений в цифровой образовательной среде и использовать для управления индивидуальной познавательной деятельностью учащихся. Пример таких интерактивных упражнений, разработанных на платформе LearningApps, можно посмотреть по ссылке: <https://learningapps.org/view36657741>. Однако, как показывают исследования [1, 7, 16] и наш практический опыт, влияние совместной речевой деятельности учащихся на интериоризацию осваиваемых действий очень велико. Поэтому целесообразно контролировать процесс перехода от моделирования действий с некоторыми входными данными к схематизации алгоритма в рамках «живого» общения с учащимися.

Организовать фронтальную работу по решению указанных выше упражнений (задания 1–8, 10) можно с помощью рабочего листа (рис. 6). Отдельные элементы рабочего листа учащиеся могут заполнять в малых группах с последующим фронтальным обсуждением.

5. Заключение

Как правило, на первых этапах обучения программированию школьники сразу приступают к этапу кодирования и отладки программы, не уделяя достаточно внимания предварительному анализу задачи. Систематическое использование описанной в статье методики способствует формированию некоторого устойчивого поведенческого паттерна: начинать решение задачи по программированию с этапа моделирования действий для конкретных входных данных и схематизации этих действий. Организация речевой деятельности на указанном этапе позволяет повысить уровень осознанности ориентировочной, исполнительной и контролирующей частей действий при написании программы.

Список источников

- Ананьев Б. Г. Психология чувственного познания. М.: Изд-во АПН РСФСР, 1960. 486 с.
- Босова Л. Л. О новых подходах к изучению школьной информатики в условиях цифровой трансформации общества //

Информатика в школе. 2022. № 4. С. 5–14. EDN: DKRLZV. DOI: 10.32517/2221-1993-2022-21-4-5-14.

3. Босова Л. Л. Программирование как инструмент формирования вычислительного мышления обучающихся // Информатика в школе. 2020. № 10 (163). С. 4–10. EDN: GURIPH. DOI: 10.32517/2221-1993-2020-19-10-4-10.

4. Босова Л. Л., Аквиланов Н. А., Кочергин И. О., Штепна Ю. Л., Бурцева Т. А. Информатика. 8–9 классы. Начала программирования на языке Python. Дополнительные главы к учебникам. М.: БИНОМ. Лаборатория знаний, 2022. 96 с.

5. Босова Л. Л., Босова А. Ю. Информатика. 8 класс. М.: БИНОМ. Лаборатория знаний, 2021. 176 с. EDN: EVEMHI.

6. Быкова И. А. Поэтапное формирование умственных действий при обучении школьников программированию // Информатика в школе. 2023. № 2 (181). С. 79–86. EDN: WWFCQF. DOI: 10.32517/2221-1993-2023-22-2-79-86.

7. Выготский Л. С. Мышление и речь. М.: Эксмо, 2022. 544 с.

8. Глотова Г. А. Человек и знак: семиотико-психологические аспекты онтогенеза человека. Свердловск: Изд-во Урал. ун-та, 1990. 256 с. EDN: TZPEEH.

9. Гребнева Д. М., Егорова Л. Е. Обучение школьников программированию на основе семиотического подхода. Нижний Тагил: НТГСПИ (ф) РГППУ, 2018. 112 с. EDN: YOCWNN.

10. Заводчикова Н. И., Быкова И. А. Организация различных видов знаково-символической деятельности при обучении программированию // Информатика в школе. 2023.

№ 5 (184). С. 33–38. EDN: UZXXOD. DOI: 10.32517/2221-1993-2023-22-5-33-38.

11. Заводчикова Н. И., Быкова И. А. Управление познавательной деятельностью учащихся при изучении программирования на уроках информатики // Математика и проблемы образования. Материалы 41-го Международного научного семинара преподавателей математики и информатики университетов и педагогических вузов. Киров, 2022. С. 213–215. EDN: OQIDQI.

12. Каракозов С. Д., Маняхина В. Г. Python как базовый язык обучения программированию в школе // Информатика в школе. 2020. № 1 (154). С. 26–30. EDN: GPEZPH. DOI: 10.32517/2221-1993-2020-19-1-26-30.

13. Курганова Н. А. Развитие знаково-символической деятельности учащихся в процессе обучения информатике на основе семиотического подхода: автореф. дис. ... канд. пед. наук. Омск, 2006. 22 с. EDN: NKAICB.

14. Салмина Н. Г. Знак и символ в обучении. М.: Изд-во Московского университета, 1988. 288 с.

15. Сапогова Е. Е. Моделирование как этап развития знаково-символической деятельности дошкольника // Вопросы психологии. 1992. № 5–6. С. 26–30.

16. Соколов А. Н. Внутренняя речь и мышление. М.: Прогресс-Книга, 1967. 248 с.

17. Федеральная рабочая программа среднего общего образования. Информатика (базовый уровень) (для 7–9 классов образовательных организаций). https://edssoo.ru/wp-content/uploads/2023/08/15_ФРП-Информатика-7-9-классы_база.pdf