

**Губина Т.Н.**

Елецкий государственный университет им. И.А. Бунина, г. Елец, Россия

## **МЕТОДИЧЕСКИЕ ПРИЕМЫ РАЗВИТИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ БУДУЩЕГО УЧИТЕЛЯ ИНФОРМАТИКИ\***

### **АННОТАЦИЯ**

*В статье описываются методические приемы развития алгоритмического мышления студентов, обучающихся по направлению подготовки «Педагогическое образование» (квалификация (степень) «бакалавр», профили «Информатика», «Математика и информатика»). Анализируются результаты педагогического эксперимента по определению влияния этих приёмов на уровень развития алгоритмического мышления обучающихся.*

### **КЛЮЧЕВЫЕ СЛОВА**

*Методика обучения; алгоритмический стиль мышления; высшее профессиональное образование; мыслительные операции; уровень и индикаторы развития.*

**Tatyana Gubina**

Yelets State University of Bunin, Yelets, Russia

## **INSTRUCTIONAL TECHNIQUES OF DEVELOPING THE ALGORITHMIC THINKING OF FUTURE TEACHERS OF INFORMATICS**

### **ABSTRACT**

*This article describes the instructional techniques of developing the algorithmic thinking of students enrolled in the programs "Pedagogical education" (qualification (degree) "bachelor", profiles, "Informatics", "Mathematics and Informatics"). It also gives the analysis of the results of the pedagogical experiment estimating the effects of these techniques on the level of development of algorithmic thinking of students.*

### **KEYWORDS**

*Methods of teaching; algorithmic style of thinking; higher professional education; mental operations; the level and development indicators.*

### **Введение**

Повсеместное внедрение современных информационных и коммуникационных технологий требует от выпускников школ и вузов умений работать с большими объемами разнообразной информации, что предполагает достаточно высокий уровень развития алгоритмического мышления. Первоначально алгоритмическое мышление формируется еще в школе. Особая роль в этом процессе отводится школьным курсам математики и информатики. Значение этих предметов в формировании и развитии алгоритмического мышления определено содержанием ФГОС начального общего, среднего общего, среднего (полного) общего образования [17].

Уровень развития алгоритмического мышления учащихся во многом зависит от способности педагога организовать и управлять процессом обучения, т.е. от его личностных характеристик. Будущий учитель сам должен пройти соответствующее обучение, должен обладать необходимыми компетенциями, соответствующим уровнем алгоритмического мышления. В связи с этим актуальным является поиск методов обучения будущих учителей информатики, содержание которых направлено на формирование алгоритмического стиля мышления.

В настоящей статье рассматриваются методические приемы развития алгоритмического мышления студентов, обучающихся по направлению подготовки 050100 «Педагогическое

---

\* Труды XI Международной научно-практической конференции «Современные информационные технологии и ИТ-образование» (SITITO'2016), Москва, Россия, 25-26 ноября, 2016

образование» (квалификация (степень) «бакалавр», профиль «Информатика»), описывается педагогический эксперимент, анализируются его результаты.

Как показывает многолетняя практика преподавания автором дисциплин "информационно-технологического" цикла в Елецком государственном университете им. И.А. Бунина, студенты первого курса испытывают значительные трудности в составлении алгоритмов решения разного рода задач. Следует отметить, что они вызваны не столько незнанием конструкций, синтаксиса языков программирования, сколько неготовностью к построению компьютерно-математических (формальных) моделей решения задач, к анализу соответствия модели и моделируемого объекта (процесса), к применению различных методов познания: анализа, синтеза, индукции, дедукции и др. при построении модели решения задачи. Хотя владение этими мыслительными операциями, согласно требованиям ФГОС начального общего, среднего общего, среднего (полного) общего образования [17], должно быть уже сформировано до необходимого уровня у выпускников в школе.

Анализ опыта работы учителей информатики в школах г. Ельца позволил выделить несколько основных причин невыполнения требований ФГОС [17]. К ним относятся следующие: во-первых, некоторые учителя информатики не могут методически грамотно выстроить процесс изучения раздела алгоритмизации и программирования, преподнести материал в доступной, понятной школьникам форме, используя разные методы познания; во-вторых, часто информатику в школе преподают учителя математики, физики или выпускники вузов непедагогических направлений подготовки, которые недостаточно глубоко знакомы с методикой преподавания информатики в школе.

На основании этого можно сделать вывод о том, что в одном случае некоторые учителя информатики не владеют в должной степени знаниями основ алгоритмизации, а в другом – не владеют методикой обучения информатике, не готовы к реализации учебного процесса по разделу «Алгоритмизация и программирование» школьного курса "Информатика и информационно-коммуникационные технологии" (ИКТ). В таких случаях учитель информатики не может в должной мере развить у школьников алгоритмическое мышление. Зачастую, учитель информатики считает раздел "Алгоритмизация и программирование" слишком сложным для усвоения учащимися, поэтому не уделяет достаточного времени и усилий на изучение этого раздела в классе. В то же время необходимо отметить, что примерно 60% заданий ЕГЭ по информатике и ИКТ требуют от выпускника школы наличия среднего и высокого уровня алгоритмического мышления. В следствие этой причины появляется нежелание у некоторых выпускников сдавать ЕГЭ по информатике и ИКТ. Многие из них предпочитают сдавать ЕГЭ по физике, поэтому выбирают направления подготовки, связанные с информатикой и ИКТ, на которых в качестве вступительного испытания вузом определена физика.

Такое положение приводит к необходимости дополнительной подготовки студентов первого курса, направленной на повышение уровня их алгоритмического мышления. Однако, часто эти усилия оказываются недостаточно результативными. В вузе не удаётся в полной мере компенсировать недостатки школьного образования. Поэтому основным направлением решения проблемы формирования алгоритмического мышления является повышение качества профессиональной подготовки будущих учителей информатики и, как следствие, уровня подготовки учащихся.

Цель настоящей работы заключается в том, чтобы описать некоторые методические приёмы и проанализировать результаты их влияния на уровень развития алгоритмического мышления будущих учителей информатики.

### **Определение понятия «алгоритмический стиль мышления»**

Прежде, чем приступить к описанию методических приёмов формирования и развития "алгоритмического мышления", необходимо определиться с этим понятием, т.к. в настоящее время нет общепринятого мнения о его характеристиках и объёме.

Термин «алгоритмический стиль мышления» широко используется в научной и методической литературе, посвященной обучению информатике и ИКТ. Это связано с тем, что одной из основных задач изучения этой дисциплины является формирование и развитие алгоритмического мышления учащихся. По мнению А.В. Копаева, ключевой проблемой при выяснении целей обучения и содержания модуля «Алгоритмизация и программирование» является уточнение понятия «алгоритмический стиль мышления» [6]. Развивать эту идею мы будем, опираясь на мнения авторитетных учёных, занимавшихся изучением этой и смежных с ней проблем.

Так С.Л. Рубинштейн считал, что *мышление* есть «движение мысли, раскрывающее связь, которая ведёт от отдельного к общему и от общего к отдельному» [12].

Ю.П. Платонов под *стилем мышления* понимал открытую, постоянно пополняющуюся, «систему интеллектуальных стратегий, приемов, навыков и операций, к которой личность предрасположена в силу своих индивидуальных особенностей» [11]. Там же автор пишет о том, что в понятии «стиль мышления» (или «интеллектуальный стиль») отражается тот факт, что все люди думают, рассуждают по-разному об одном и том же, причем каждый человек считает, что именно привычный для него стиль мышления является наилучшим [11]. В этом контексте можно говорить о том, что современная школа призвана формировать и развивать у учащихся разные стили мышления, учить рассуждать разными способами, решать задачи различными методами, рассматривать проблему с разных сторон.

Содержание и объём понятия "алгоритмическое мышление" рассматривались А.П. Ершовым, А.И. Газейкиной, Г.А. Звенигородским, А.Г. Кушниренко, Т.Н. Лебедевой, Г.В. Лебедевым, Ю.А. Первиным и др. Так, например, А.П. Ершовым, Г.А. Звенигородским, Ю.А. Первиным оно определяется как «умение планировать структуру действий, необходимых для достижения цели, при помощи фиксированного набора средств»; «умение строить информационные модели для описания объектов и систем»; «умение организовывать поиск информации, необходимой для компьютерного решения поставленной задачи» [5]. Следует отметить, что "сущностными характеристиками" (наиболее значимым содержанием) этого определения являются алгоритмическая (конструктивная) и операционная составляющие мышления.

А.Г. Кушниренко и Г.В. Лебедев понимают алгоритмический стиль мышления как «метод и способ, которые необходимы для перехода от непосредственного управления к программному, от умения сделать к умению записать алгоритм» [7].

Я.И. Груденов считает, что процесс формирования алгоритмического стиля мышления способствует развитию умения выбирать наилучшее из возможных решений задачи [3].

В понимании и представлении Т.Н. Лебедевой алгоритмическое мышление определяется в виде «... целесообразной (или рациональной) последовательности совершаемых мыслительных процессов с присущей детализацией и оптимизацией укрупненных блоков, осознанным закреплением процесса получения конечного результата, представленного в формализованном виде на языке исполнителя с принятыми семантическими и синтаксическими правилами» [8].

Анализ информационных источников позволил на этой основе синтезировать содержание понятия "алгоритмическое мышление", выделив в качестве "сущностных" те его характеристики, которые отвечали бы цели данного исследования. Под *алгоритмическим мышлением* мы будем понимать такой стиль мышления человека, который представляет собой систему мыслительных приёмов, конструкций, набора способов действий, необходимых для решения рассматриваемой проблемы, видения проблемы в целом, выделение крупных блоков её решения, построения информационных моделей, организации поиска необходимой информации, получения результата в алгоритмической форме.

### **Методология исследования**

Проблема развития алгоритмического мышления обучающихся изучается психологами и педагогами достаточно давно. В научной и методической литературе описываются разные способы развития алгоритмического мышления. Они базируются на разных теоретико-методологических основаниях.

*Теоретико-методологической основой* данного исследования являются:

- концепция модернизации современного образования (В.А. Болотов, Г.А. Бордовский, В.В. Краевский, Г.П. Щедровицкий и др.);

- проблемно-деятельностный подход в обучении (Л.С. Выгодский, П.Я. Гальперин, А.Н. Леонтьев и др.);

- теория поэтапного формирования умственных действий (П.Я. Гальперин);

- структурный подход в обучении (Т.А. Ильина, Ю.К. Бабанский и др.);

- дифференцированный подход к содержанию и методам обучения (М.К. Акимова, Н.Д. Есипова, В.Т. Козлова и др.).

*Методы исследования:*

*Теоретико-аналитический:* изучение и анализ научной и научно-методической литературы по проблемам методики обучения, направленной на выявление проблем развития алгоритмического стиля мышления и их решение; изучение опыта педагогов; планирование опытно-экспериментальной работы.

*Опытно-экспериментальный:* разработка методических приёмов развития алгоритмического стиля мышления студентов; проведение формирующего эксперимента по внедрению данных разработок в практику обучения будущих учителей информатики; диагностика

полученных результатов.

В качестве одного из основных методических приёмов разрешения проблемы формирования и развития алгоритмического мышления обучающихся многие ученые, например, В.Н. Исаков, В.В. Исакова, Т.Н. Лебедева, И.Н. Слинкина и др., предлагают использовать определённую систему задач с нарастающим уровнем сложности. Мы также считаем: для того чтобы студенты в будущем смогли формировать и развивать алгоритмическое мышление учащихся на уроках информатики, они должны сами овладеть методическими приемами решения задач по теме «Алгоритмизация и программирование», научиться самостоятельно отбирать и формировать определённую систему заданий.

Для оценки эффективности этой методики была организована экспериментальная работа в группах студентов, обучающихся по направлению подготовки «Педагогическое образование» (профиль «Информатика», «Математика и информатика»).

Содержание эксперимента заключается в следующем: была подготовлена система задач, решение которых, по оценке экспертов из числа преподавателей дисциплин "информационно-технологического цикла", должно способствовать развитию алгоритмического мышления. Разработана методика составления алгоритмов для их решения. Эти задачи были включены в содержание таких дисциплин учебного плана специальностей как «Практикум по решению задач на ЭВМ», «Практикум по школьному курсу информатики». На занятиях студенты под руководством преподавателя, корректирующего ход решения задач, строили алгоритмы решения и реализовывали их на языке программирования Паскаль. Следует обратить внимание на то, что реализация, разработанного студентом алгоритма на определённом языке программирования, является важным этапом обучения, т.к. требует выбора структуры данных, необходимых для написания программного кода. Кроме того, необходимо учитывать, что алгоритм, реализующий решение математической задачи, не всегда совпадает с алгоритмом, заложенным в основу программного кода.

Рассмотрим на примере один из методических приемов, способствующих развитию алгоритмического стиля мышления у студентов, направленный на формирование готовности к самостоятельному поиску решения задачи, связанной с построением общего алгоритма решения задач по теме «Умножение многочленов».

*Пример.* Даны действительное число  $a$ , многочлен  $P(x) = p_n x^n + p_{n-1} x^{n-1} + \dots + p_0$ . Требуется получить:

$$\begin{aligned} &(x - a) P(x); \\ &(x^2 + 2ax + 3) P(x); \\ &(x^2 + a^2) P(x). \end{aligned}$$

Если рассматривать данную задачу с позиций математики, то для её решения достаточно вспомнить алгоритм, который изучается еще в школьном курсе математики в 7 классе, умножения одного многочлена на другой многочлен. Однако при разработке алгоритма решения поставленной задачи для дальнейшей его реализации на компьютере – все гораздо сложнее: компьютеру надо «объяснить», как производить такое умножение и как хранить «промежуточные» результаты. Большинство студентов на первом этапе обучения были не в состоянии предложить хоть какой-либо реализуемый способ решения поставленной задачи на языке программирования, несмотря на то, что необходимые знания и инструментарий для её решения к моменту начала обучения в вузе у них уже накоплен.

Какова методика обучения студентов решению подобного типа задач? Можно преподнести ее решение в готовом виде, но такой подход не будет способствовать глубокому пониманию сути решения и не "запускает" мыслительные операции, которые позволяют глубже проникнуть в суть стоящей перед студентом проблемы, выделить составляющие ее элементы. И, как следствие, выпускник не будет способен самостоятельно разрабатывать алгоритмы и, тем более, овладеть методикой преподавания курса алгоритмизации и программирования в школе.

Студентам потребуются "запустить" такие мыслительные операции, как анализ, сравнение, абстрагирование, синтез, обобщение, чтобы суметь разбить задачу на более простые составляющие, найти решения более простых задач, сравнить простые задачи между собой, обобщить построенное решение и т.д.

Для начала следует предложить студентам взять любой (конкретный) многочлен  $P(x)$  и возвести его в квадрат известным из курса математики способом, например, так:

$$\begin{aligned} (x-a) * (2*x^2 + 3*x - 4) &= x*2*x^2 + x*3*x - 4*x - a*2*x^2 - a*3*x - a*(-4) = \\ 2*x^3 + 3*x^2 - 4*x - 2*a*x^2 - 3*a*x + 4*a &= 2*x^3 + (3-2*a)*x^2 - 3*a*x + 4*a. \end{aligned}$$

Как показывает практика, пример с умножением двучлена на трехчлен не наводит студентов на мысль о том, что делать дальше и как это реализовать на компьютере.

Тогда следует предложить взять в качестве  $P(x)$  многочлен нулевой, первой степени, затем

уже многочлен второй степени:

$$(x-a) * (-4*x^0) = x*(-4)*x^0 - a*(-4)*x^0 = -4*x + 4*a$$

$$(x-a) * (3*x^1 - 4) = x*3*x^1 - x*(-4) - a*3*x^1 - a*(-4) = 3*x^2 + (-4-3a)x + 4*a.$$

На данном этапе идет выработка стратегии решения задачи: как получаются коэффициенты результирующего многочлена и как получить суммарный коэффициент при каждом члене многочлена нулевой, первой, второй, третьей степени? Таким образом, переходим от общего случая к частному, расчленив нашу задачу на более простые подзадачи.

Здесь следует отметить, что работу над задачей на данном этапе следует вести с использованием средств наглядности, что в большей степени способствует пониманию сути происходящего: для полноценной мыслительной работы большинству студентов необходимо видеть или представлять ход рассуждений, поскольку у них согласно С.Л. Рубинштейну "в большей степени развито именно наглядное, а не вербальное мышление" [15, С.296].

Подробный разбор примеров нахождения произведения многочлена на многочлен нулевой, первой, второй, третьей степени должен подвести студента к мысли о том, что для хранения коэффициентов потребуется задействовать такую структуру данных как массив и причем не один.

Например, если требуется найти произведение  $(x - a)P(x)$ , то как минимум таких массивов надо будет два: первый массив будет хранить коэффициенты, полученные после умножения  $x$  на  $P(x)$ , а второй – коэффициенты после умножения  $-a$  на  $P(x)$ , причем элементы второго массива должны быть сдвинуты на 1 элемент вправо. Тогда коэффициенты результирующего многочлена могут быть получены простым сложением соответствующих элементов двух массивов. Но для дальнейших расчетов выгоднее и коэффициенты известного многочлена тоже хранить в массиве.

Для того, чтобы студенты пришли к такому выводу, можно предложить следующий методический прием получения коэффициентов результирующего многочлена в виде таблицы (Таблица 1 - Таблица 3).

Таблица 1. Расчет коэффициентов для одночлена  $P(x)$  нулевой степени

	-4			
x	$-4*[x^1]$			
-a		$-4*(-a)[x^0]$		
Результирующий многочлен	$-4x$	$4a$		

Таблица 2. Расчет коэффициентов многочлена для  $P(x)$  первой степени

	$3*x$	-4		
x	$3*[x^2]$	$-4*[x^1]$		
-a		$-a*3*[x^1]$	$-4*(-a)*[x^0]$	
Результирующий многочлен	$3x^2$	$(-4-3a)x$	$4a$	

Таблица 3. Расчет коэффициентов многочлена для  $P(x)$  второй степени

	$2*x^2$	$3*x$	-4	
x	$2*[x^3]$	$3*[x^2]$	$-4*[x]$	
-a		$-2*a*[x^2]$	$-3*a*[x]$	$4*(-a)$
Результирующий многочлен	$2x^3$	$(3-2*a)x^2$	$(-4-3a)x$	$4a$

Такая работа позволяет "запустить" мыслительные процессы (сравнение, синтез, обобщение) студентов, что должно способствовать выдвижению различных предположений о том, как построить алгоритм для решения поставленной задачи.

Так, например, если  $P(x)$  – многочлен нулевой степени, студентами могут быть выдвинуты следующие правильные предположения и идеи:

- степень результирующего многочлена не увеличивается;
- надо создать одномерный массив, в котором сохранить коэффициенты многочлена  $(x - a)$  при каждом его члене размера 2 ( $q$ : array [0..1] of integer);
- индекс элемента массива равен степени одночлена в многочлене, его значение – коэффициенту в этом одночлене;
- запрашивать на ввод коэффициент  $P(x)$  нулевой степени (по сути число);
- умножить каждый элемент массива на введенное с клавиатуры значение;
- вывести значения, хранящиеся в массиве Q в формате  $q_1 x + q_0$ .

После проработки хода решения поставленной подзадачи для многочлена нулевой степени следует предложить студентам реализовать полученный алгоритм на языке программирования Паскаль.

Возможный вариант решения:

```
program palindrom_0;
```

```

const
  a = 10;
var
  i, p: integer;
  q: array[0..1] of integer = (-a, 1);
begin
  writeln('Введите коэффициент при нулевом члене многочлена:');
  read(p);
  for i := 1 downto 0 do
    q[i] := p * q[i];
  write(q[1], 'x+', q[0]);
end.

```

Студенты тестируют реализованный на языке Паскаль алгоритм решения подзадачи и убеждаются в его корректной или некорректной работе.

Затем студенты выдвигают предположение о том, как нужно доработать алгоритм решения подзадачи для случая многочлена первой степени  $P(x) = p_1x + p_0$ . Например,

- необходимо вводить два коэффициента многочлена  $P(x)$ , для этого возьмем две переменные  $p_1$  и  $p_0$ ;

- для хранения коэффициентов многочленов, полученных после умножения  $x$  на многочлен  $P(x)$  и после умножения  $-a$  на  $P(x)$ , можно взять два массива размера 3 элемента: в массиве  $q_1$  последний элемент будет равен нулю, а в массиве  $q_2$  – первый элемент;

- для получения результирующего многочлена сложим соответственные элементы массивов  $q_1$  и  $q_2$ .

Возможный вариант решения:

```

program palindrom_1;
const
  a = 10;
var
  i, j, p1, p0, n: integer;
  q: array[0..1] of integer = (-a, 1);
  q1, q2, q_rez: array[0..2] of integer;
begin
  writeln('Введите коэффициенты многочлена p1 и p0:');
  read(p1, p0);
  for i := 2 downto 0 do
    begin
      q1[i] := 0; q2[i] := 0;
    end;
  q1[2] := q1[2] + p1 * q[1];
  q1[1] := q1[1] + p0 * q[1];
  q2[1] := q2[1] + p1 * q[0];
  q2[0] := q2[0] + p0 * q[0];
  for i := 2 downto 0 do
    begin
      q_rez[i] := q1[i] + q2[i];
    end;
  write(q_rez[2], 'x^2+', q_rez[1], 'x+', q_rez[0]);
end.

```

Затем алгоритм и программный код дорабатываются для решения подзадачи для случая многочлена второй степени  $P(x) = p_2x^2 + p_1x + p_0$  (и в общем  $n$ -ой степени). Могут быть выдвинуты следующие правильные предположения и идеи:

- необходимо вводить три коэффициента многочлена  $P(x)$ , лучше сделать это в цикле (для последующего обобщения задачи на многочлен  $n$ -ой степени). Значит, надо запрашивать степень многочлена (переменная  $n$ );

- в цикле запрашивать все коэффициенты многочлена  $P(x)$  и сохранять их в массив  $p$ ;

- для подсчета и сохранения коэффициентов, полученных после умножения каждого члена многочлена  $P(x)$  на  $x$  и на  $-a$ , в массивах  $q_1$  и  $q_2$  организовать цикл с параметром, начиная с последнего элемента до нулевого;

- вычислять коэффициенты результирующего многочлена  $q\_rez$  как сумму соответственных элементов массивов  $q_1$  и  $q_2$ ;

- вывод результирующего многочлена тоже производить в цикле.

В результате такой работы становится понятно, как нужно модифицировать программный код для перехода к следующей, более сложной задаче.

Возможный вариант решения:

```
program palindrom_1;
const
  a = 10;
var
  i, j, n: integer;
  q: array[0..1] of integer = (-a, 1);
  q1, q2, p, q_rez: array[0..20] of integer;
begin
  writeln('Введите степень многочлена p:');
  read(n);
  writeln('Введите коэффициенты многочлена p:');
  for i := n downto 0 do
    read(p[i]);
  for i := n + 1 downto 0 do //степень результирующего многочлена должна быть на 1 больше чем p
  begin
    q1[i] := 0; q2[i] := 0;
  end;
  for i := n + 1 downto 1 do //по индексам массивов q1, q2
  begin
    q1[i] := q[1] * p[i-1];
    q2[i-1] := q[0] * p[i-1];
  end;
  for i := n + 1 downto 0 do //нахождение результирующего многочлена
  begin
    q_rez[i] := q1[i] + q2[i];
  end;
  for i := n + 1 downto 0 do
  if q_rez[i] < 0 then
    write(q_rez[i], 'x^', i)
  else
    write('+', q_rez[i], 'x^', i);
  end.
```

Далее преподаватель предлагает студентам самостоятельно получить требуемое решение задачи для двух других случаев (домашняя работа).

Развитию алгоритмического стиля мышления в ходе решения задач может способствовать проведение различных тренингов.

#### Тренинг №1. «Насколько телефон испорчен?»

Первый студент описывает проблемную ситуацию второму студенту в ограниченное время (30 секунд), второй передает описание этой проблемы третьему, третий – четвертому, затем оно возвращается к первому студенту. Сравниваются первоначальный и конечный вариант проблемной ситуации. Преподавателем ведется коррекционная работа.

Цель тренинга — научить будущих учителей правильно, четко и конкретно ставить задачи, приводить описание проблемных ситуаций, учить переформулировать проблемную ситуацию.

#### Тренинг №2. «Мозговой штурм»

Преподаватель и студенты, участвующие в обсуждении способов построения алгоритма, предлагают как можно большее количество вариантов решения, в том числе самых фантастических. Затем, анализируя предложенные варианты, из общего их числа совместно выбирают наиболее удачный, который берется в качестве ориентира для построения алгоритма.

Цель тренинга – формирование умения выбирать разные методы анализа и оценки имеющихся вариантов решения поставленной задачи, знания критериев отбора и оценки выдвигаемых гипотез.

#### Тренинг №3. «Выбор одним глазом»

Предлагается несколько возможных вариантов решения проблемной ситуации, нескольких возможных путей построения алгоритмов, некоторые из которых могут быть заведомо неправильными. Студентам необходимо произвести выбор наиболее оптимального, эффективно реализуемого варианта решения задачи.

Цель тренинга — развитие способности интуитивно принимать правильные решения.

Тренинг №4. «Стань на моё место и докажи обратное»

Преподаватель намеренно предлагает обучающимся неправильный, нереализуемый, или неэффективный способ построения алгоритма для решения задачи, тем самым вызывая возможные «возражения» со стороны студентов. Особенно активным студентам предлагается обосновать противоположное мнение, другой способ решения, т.е. доказать, что этого делать не надо.

Цель тренинга – развитие способности критически анализировать предложенные способы решения задачи, приводить контраргументы, доказывать свою точку зрения, в том числе, используя нестандартное мышление.

Тренинг №5. «Предложи свой способ решения»

При возникновении новой проблемной ситуации по ходу решения задачи преподаватель предлагает студентам подумать и записать на листе бумаги тот способ (метод) решения, который он считает правильным и корректным и который поможет выйти из этой ситуации.

Цель тренинга – развитие способности анализировать имеющиеся данные, сопоставлять их с тем результатом, который необходимо получить, учитывать различные сценарии составления алгоритма.

Тренинги и решение подобных задач с использованием рассмотренных выше методических приемов должно создать условия для развития алгоритмического стиля мышления обучающихся, тем самым формируя готовность к анализу данных, формализации задачи, постановке и обнаружению проблем, к анализу получаемых результатов, поиску оптимальных и эффективных способов решения.

С целью экспериментальной проверки педагогической эффективности предлагаемой системы методических приёмов была проведена диагностика сформированности уровня развития алгоритмического мышления студентов. Нами использовались индикаторы, описанные в работах Л.Г. Лучко, И.Н. Слинкиной, Н.Н. Тулькибаевой [9, 14, 16]. Они в обобщённом виде представлены на рис. 1.

**Полученные результаты**

В педагогическом эксперименте участвовало 48 студентов (два потока), обучающихся по направлению подготовки «Педагогическое образование» (профили «Информатика» и «Математика и информатика»). В соответствии с экспериментальной методикой у будущих учителей информатики формировался алгоритмический стиль мышления в рамках дисциплины «Практикум по решению задач на ЭВМ», а также в рамках курса по выбору «Практикум по школьному курсу информатики».

Так как педагогическое воздействие предлагаемой методики обучения осуществлялось только в течение 1 семестра (18 аудиторных часов), ожидать больших сдвигов в значениях уровня алгоритмического мышления студентов не приходилось. Поэтому для получения и обработки экспериментальных данных была использована очень чувствительная к небольшим сдвигам в значениях исследуемых характеристик методика - статистика G-критерия знаков, применение которой описано Е.В. Сидоренко [13].

На первом этапе педагогического эксперимента проводилось два тестирования группы студентов, обучающихся по профилю "Информатика" в количестве 23 человек: в начале и конце эксперимента. Экспериментальные данные для первого потока студентов приведены в Таблице 4.

*Таблица 4. Данные предварительного и повторного тестирования по измерению уровня развития алгоритмического мышления (профиль обучения "Информатика")*

Испытуемые	1	2	3	4	5	6	7	8	9	10	11	12
Значения «до» (1 тест)	32	36	37	38	36	32	37	32	30	30	28	29
Значения «после» (2 тест)	37	39	40	36	36	35	37	39	36	30	35	38
Сдвиг («после»-«до»)	+	+	+	-	0	+	0	+	+	0	+	+

Испытуемые	13	14	15	16	17	18	19	20	21	22	23
Значения «до» (1 тест)	28	29	30	35	38	35	26	27	31	30	34



<b>Значения «после» (2 тест)</b>	28	27	35	38	40	35	32	36	31	28	40
<b>Сдвиг («после»-«до»)</b>	0	-	+	+	+	0	+	+	0	-	+



Рис. 1. Уровни и индикаторы развития алгоритмического мышления

Из таблицы видно, что общий балл, полученный испытуемыми после окончания эксперимента, возрос. Количество нулевых реакций  $n_0=6$ , следовательно, новый объем выборки равен  $n=23-6=17$ . Количество положительных сдвигов – 14, количество отрицательных сдвигов – 3. Значит, «+» сдвиги типичные, а «-» сдвиги – нетипичные.

Были приняты гипотезы:

- нулевая гипотеза -  $H_0$ : сдвиг показателей в типичную сторону является случайным;
- альтернативная гипотеза -  $H_1$ : сдвиг показателей в типичную сторону является не случайным.

По таблице критических значений G-критерия знаков для  $n=17$   $G_{кр}(p \leq 0,05)=4$  и  $G_{кр}(p \leq 0,01)=3$  [13]. Преобладание «типичного» сдвига является достоверным, если  $G_{эмп}=3$  ниже или равен  $G_{0,05}$  и тем более достоверным, если  $G_{эмп}$  ниже или равен  $G_{0,01}$ . Так как  $G_{эмп} < G_{0,05}$  и  $G_{эмп} = G_{0,01}$ , то принимается альтернативная гипотеза -  $H_1$ , т. е. различия не **являются** случайными.

На втором этапе педагогического эксперимента проводилось два аналогичных тестирования студентов второго потока (профиль обучения "Математика и информатика") в количестве 25 человек. Полученные на втором этапе педагогического эксперимента результаты ( $G_{эмп}=2$ ,  $G_{эмп} < G_{0,05}$  и  $G_{эмп} < G_{0,01}$ ) подтверждают выводы, полученные по итогам проведения первого этапа: типичными являются положительные сдвиги уровня развития алгоритмического мышления студентов (количество положительных сдвигов на втором этапе – 15, количество отрицательных сдвигов – 2), поэтому принимается альтернативная гипотеза -  $H_1$ , говорящая о том, что различия в показателях тестирования до обучающего эксперимента и после него не являются случайными. Можно говорить о том, что используемые методические приемы оказывают положительное влияние на развитие студентов обеих групп, т.к. наблюдается "не случайный" сдвиг в сторону повышения уровня их алгоритмического мышления.

Дальнейшее исследование по заявленной проблеме может заключаться в разработке новых методов развития алгоритмического стиля мышления студентов.

### **Заключение**

Будущему учителю информатики еще во время обучения в вузе необходимо овладеть умениями по отбору типовых задач, используемых на уроках для развития алгоритмического мышления, методиками развития алгоритмического мышления, умениями их использования в сочетании с методами формирования предметных, метапредметных знаний и умений обучающихся. Рассмотренные методические приемы могут способствовать развитию алгоритмического стиля мышления обучающихся.

### **Литература**

1. Газейкина, А.И. Стили мышления и обучение программированию студентов педагогического вуза [Электронный ресурс] / А.И. Газейкина // Информационные технологии в образовании. 2006. Режим доступа: <http://ito.edu.ru/2006/Moscow/I/1/I-1-6371.html>.
2. Гальперин, П.Я. Развитие исследований по формированию умственных действий [Текст] / П.Я. Гальперин // Психологическая наука в СССР. - М.: АПН РСФСР, 1959.
3. Груденов, Я.И. Изучение определений, аксиом, теорем. Пособие для учителей [Текст] / Я.И. Груденов. - М. - Просвещение, 1981. - 95 с.
4. Ершов, А.П. О человеческом и эстетическом факторах в программировании [Текст] / А.П. Ершов // Кибернетика. - 1972. - №5. - с. 95-99.
5. Ершов, А.П. Школьная информатика: концепции, состояния, перспективы [Текст] / А.П. Ершов, Г.А. Звенигородский, Ю.А. Первин. - препринт №152 ВЦ СО АН СССР, Новосибирск, 1979. - 51 с.
6. Копаев, А.В. О практическом значении алгоритмического стиля мышления / А.В. Копаев // Информационные технологии в общеобразовательной школе. - 2003. - No 4. - С. 6-11
7. Кушниренко, А.Г. 12 лекций о том, для чего нужен школьный курс информатики и как его преподавать [Текст] / А.Г. Кушниренко, Г.В. Лебедев. - Информатика. - 1999. - №1. - С.2-15.
8. Лебедева, Т.Н. Формирование алгоритмического мышления школьников в процессе обучения рекурсивным алгоритмам в профильных классах средней общеобразовательной школы: автореф. дис. ... канд. пед. наук: 13.00.02 [Текст] / Т.Н. Лебедева. - Екатеринбург, 2005. - 20 с.
9. Лучко, Л.Г. Решение задач школьного курса информатики [Текст]: учебно-методическое пособие [Текст] / Л.Г. Лучко. - Омск: ОмГПУ, 2001. - 80 с.
10. Первин, Ю.А. Дидактическое обоснование школьного курса информатики [Текст] / Ю.А. Первин. - в серии статей «Энциклопедия учителя информатики». «Информатика», 11 (540), 2007.
11. Платонов, Ю.П. Типология стилей мышления руководителей [Электронный ресурс] / Ю.П. Платонов. - URL: <http://www.elitarium.ru/>
12. Рубинштейн, С.Л. Основы общей психологии в 2 т. Т.1. М.: Педагогика [Текст] / С.Л. Рубинштейн, 2000. - 712с.
13. Сидоренко, Е.В. Методы математической обработки в психологии [Текст] / Е.В. Сидоренко. - СПб.: ООО «Речь», 2001. - 350с.
14. Слинкина, И.Н. Использование компьютерной техники в процессе развития алгоритмического мышления у младших школьников [Текст]: автореф. дис. ... канд. пед. наук. - Екатеринбург: УрГПУ, 2000. - 22 с.
15. Слободчиков, В.И. Психология человека: Основы психологической антропологии [Текст] / В.И. Слободчиков, Е.И. Исаев. - М.: Школа—Пресс, 1995. 383с., С.296.
16. Тулькибаева, Н.Н. Теория и практика обучения учащихся решению задач [Текст]: Монография / Н.Н. Тулькибаева. - Челябинск: Изд-во ЧГПУ, 2000. - 239 с.
17. ФГОС среднего(полного) общего образования. Утвержден приказом Министерства образования и науки РФ от 17.05.2012 г. № 413. Режим доступа: <http://минобрнауки.рф/документы/543>.

## References

1. Gazejkina, A.I. Stili myshlenija i obuchenie programirovaniju studentov pedagogicheskogo vuza [Jelektronnyj resurs] / A.I. Gazejkina // Informacionnye tehnologii v obrazovanii. 2006. Rezhim dostupa: <http://ito.edu.ru/2006/Moscow/1/1/I-1-6371.html>.
2. Gal'perin, P.Ja. Razvitie issledovanij po formirovaniju umstvennykh dejstvij [Tekst] / P.Ja. Gal'perin // Psihologicheskaja nauka v SSSR. - M.: APN RSFSR, 1959.
3. Grudenov, Ja.I. Izuchenie opredelenij, aksiom, teorem. Posobie dlja uchitelej [Tekst] / Ja.I. Grudenov. - M. - Prosveshhenie, 1981. - 95s.
4. Ershov, A.P. O chelovecheskom i jesteticheskom faktorah v programirovanii [Tekst] / A.P. Ershov // Kibernetika. - 1972. - №5. - s.95-99.
5. Ershov, A.P. Shkol'naja informatika: koncepcii, sostojanija, perspektivy [Tekst] / A.P. Ershov, G.A. Zvenigorodskij, Ju.A. Pervin. - preprint №152 VC SO AN SSSR, Novosibirsk, 1979. - 51 s.
6. Kopaev, A.V. O prakticheskom znachenii algoritmicheskogo stila myshlenija / A.V. Kopaev // Informacionnye tehnologii v obshheobrazovatel'noj shkole. - 2003. - No 4. - S. 6-11
7. Kushnirenko, A.G. 12 lekcij o tom, dlja chego nuzhen shkol'nyj kurs informatiki i kak ego prepodavat' [Tekst] / A.G. Kushnirenko, G.V. Lebedev. - Informatika. - 1999. - №1. - S.2-15.
8. Lebedeva, T.N. Formirovanie algoritmicheskogo myshlenija shkol'nikov v processe obuchenija rekursivnym algoritmam v profil'nyh klassah srednej obshheobrazovatel'noj shkoly: avtoref. dis. ... kand. ped. nauk: 13.00.02 [Tekst] / T.N. Lebedeva. - Ekaterinburg, 2005. - 20 s.
9. Luchko, L.G. Reshenie zadach shkol'nogo kursa informatiki [Tekst]: uchebno-metodicheskoe posobie [Tekst] / L.G. Luchko. - Omsk: OmGPU, 2001. - 80 s.
10. Pervin, Ju.A. Didakticheskoe obosnovanie shkol'nogo kursa informatiki [Tekst] / Ju.A. Pervin. - v serii statej «Jenciklopedija uchitelja informatiki». «Informatika», 11 (540), 2007.
11. Platonov, Ju.P. Tipologija stilej myshlenija rukovoditelej [Jelektronnyj resurs] / Ju.P. Platonov. - URL: <http://www.elitarium.ru/>
12. Rubinshtejn, S.L. Osnovy obshhej psihologii v 2 t. T.I. M.: Pedagogika [Tekst] / S.L. Rubinshtejn, 2000. - 712s.
13. Sidorenko, E.V. Metody matematicheskoy obrabotki v psihologii [Tekst] / E.V. Sidorenko. - SPb.: OOO «Rech'», 2001. - 350s.
14. Slinkina, I.N. Ispol'zovanie komp'yuternoj tehniki v processe razvitija algoritmicheskogo myshlenija u mladshih shkol'nikov [Tekst]: avtoref. dis. ... kand. ped. nauk. - Ekaterinburg: UrGPU, 2000. - 22 s.
15. Slobodchikov, V.I. Psihologija cheloveka: Osno-vy psihologicheskoy antropologii [Tekst] / V.I. Slobodchikov, E.I. Isaev. - M.: Shkola—Press, 1995. 383s., S.296.
16. Tul'kibaeva, N.N. Teorija i praktika obuchenija uchashhihsja resheniju zadach [Tekst]: Monografija / N.N. Tul'kibaeva. - Cheljabinsk: Izd-vo ChGPU, 2000. - 239 s.
17. FGOS srednego(polnogo) obshhego obrazovanija. Utverzhden prikazom Ministerstva obrazovanija i nauki RF ot 17.05.2012 g. № 413. Rezhim dostupa: <http://minobrnauki.rf/dokumenty/543>.

Поступила: 5.09.2016

### Об авторах:

**Губина Татьяна Николаевна**, доцент кафедры вычислительной математики и информатики Елецкого государственного университета им. И.А. Бунина, кандидат педагогических наук, [gubina-tn@yandex.ru](mailto:gubina-tn@yandex.ru).