

УДК 372.800.4

**ОБЗОР МЕТОДИЧЕСКИХ ПОДХОДОВ К ОБУЧЕНИЮ
ПРОГРАММИРОВАНИЮ В ШКОЛЕ****Гребнева Д.М.***ФГАОУ ВПО «Российский государственный профессионально-педагогический университет,
филиал, Нижний Тагил, e-mail: grebdash@gmail.com*

В статье рассматриваются пять основных методических подходов к обучению программированию в школе: системный, деятельностный, когнитивный, проблемный и семиотический. Выделяются основные понятия и идеи подходов, анализируются их достоинства и недостатки. Системный подход рассматривается как основополагающий и делается вывод о том, что другие методические подходы, которые могут быть применены к обучению программированию, должны дополнять и расширять его основные идеи и принципы, а не противоречить ему. Деятельностный подход представлен с позиций формирования универсальных учебных действий обучающихся, при этом основное внимание уделяется знаково-символическим действиям обучающихся. Когнитивный подход основывается на таксономии Б. Блума и предполагает овладение учащимися операциями программирования, начиная с элементарных, постепенно продвигаясь к сложным. Проблемный подход подразумевает построение курса программирования на основе системы специально разработанных задач или ситуаций, содержание которых интересно для учащихся и расширяет их опыт программирования. Семиотический подход направлен на целенаправленное развитие знаково-символических действий обучающихся. В заключение статьи сделан вывод о дидактическом потенциале идей и принципов семиотического подхода для повышения эффективности обучения программированию.

Ключевые слова: обучение программированию, методический подход, системный подход, деятельностный подход, когнитивный подход, проблемный подход, семиотический подход

**THE REVIEW OF METHODOLOGICAL APPROACHES TO LEARNING PROGRAMMING
AT SCHOOLS****Grebneva D.M.***The branch of the Russian State Vocational Pedagogical University in Nizhny Tagil city, Nizhny Tagil,
e-mail: grebdash@gmail.com*

This article discusses five main methodical approaches to teaching programming at schools. They are system, activity, cognitive, and semiotic approaches. The key concepts and ideas of the approaches are outlined. The advantages and disadvantages of these approaches are analyzed. A system approach is seen as fundamental, and it is concluded that the other methodological approaches that can be applied to learning programming should complement and enhance its basic ideas and principles, but not to contradict them. The activity approach is presented from the viewpoint of formation of student's universal learning activities, with the focus on the sign-symbolic actions of students. The cognitive approach is based on the Bloom taxonomy and requires students to master the programming, starting with elementary, gradually moving to the complex activities. Problem approach involves the construction of a programming course based on specially developed tasks or situations, the content of which is interesting for the students and expanding their programming experience. The semiotic approach is aimed at focused development of signs and symbolic actions of students. In conclusion it is concluded that there is didactic potential of the ideas and principles of the semiotic approach to improve the effectiveness of teaching programming.

Keywords: teaching programming, methodical approach, system approach, activity approach, cognitive approach, problematic approach, semiotic approach

В дидактике термин «подход» определяется как совокупность принципов, определяющих стратегию обучения, при этом каждый принцип регулирует разрешение конкретных противоречий, возникающих в процессе обучения, а их взаимодействие – разрешение основных его противоречий [12]. Таким образом, исходным содержанием понятия «подход» является определенная идея, концепция, совокупность принципов, обуславливающих организацию того или иного явления, процесса, например, процесса обучения программированию.

В настоящее время существует ряд российских и западных исследований, посвященных разработке методических подходов

к обучению программированию, среди которых можно выделить: системный подход (И.О. Одинцов и др.), деятельностный подход (Е.А. Ракитина и др.), когнитивный подход (J. Reinfelds и др.), проблемный подход (Е.В. Касьянова, К.Ю. Поляков и др.), семиотический подход (P. Andersen, К. И. Баумане, Н.И. Рыжова и др.).

Проведем сравнительный анализ обозначенных методических подходов, уделяя особое внимание системному и деятельностному подходу в связи с тем, что согласно ФГОС, они являются основополагающими в общеобразовательной школе.

В основе системного подхода лежит рассмотрение объектов как систем. Он

ориентирует исследование на раскрытие целостности объекта, на выявление многообразных типов связей в нем и сведение их в единую теоретическую картину [9]. Как показал анализ научной литературы, основными категориями системного подхода является система, структура и среда.

Система – множество элементов, находящихся в отношениях и связях друг с другом, образующих определенную целостность, единство [9].

Также исследователи системного подхода (В.С. Леднев, В.Г. Афанасьев и др.), подчеркивают, что система – это совокупность объектов, взаимодействие которых вызывает появление новых интегративных качеств, не свойственных отдельно взятым образующим систему компонентам [3].

Структура – совокупность устойчивых связей между элементами объекта, которые обеспечивают его целостность и тождественность самому себе, то есть сохранение основных свойств при различных внешних и внутренних изменениях [9].

Структурность является главным свойством реальности – «всё как-то устроено: все объекты реальности из чего-то состоят, имеют составляющие их части и в тоже время, сколь бы сложными они ни были, сами являются составными частями чего-то более сложного. И так до бесконечности как в сторону увеличения масс и масштабов, так и в сторону их уменьшения. Притом всё находится в движении – всё течёт, всё изменяется» [6].

Поэтому любые объекты реальности требуют двоякого их рассмотрения: в статике и динамике.

Среда – совокупность всех объектов/субъектов, не входящих в систему, изменение свойств и/или поведение которых влияет на изучаемую систему, а также тех объектов/субъектов, чьи свойства и/или поведение которых меняются в зависимости от поведения системы [13].

Следует отметить, что системный подход к познанию и преобразованию любого объекта является ведущим общенаучным подходом. Применение же данного подхода в обучении программированию позволяет выявить такой компонент как система обучения программированию со всеми ее характеристиками: целостность, связь, структура и организация, уровни системы и их иерархия, управление, самоорганизация системы, ее функционирование и развитие.

Идеям и принципам применения системного подхода в обучении информатике и программированию посвящены работы И.О. Одинцова, Н.В. Макаровой и др.

Основная идея системного подхода в контексте обучения программированию

состоит в рассмотрении учебных задач в тесной взаимосвязи со средствами, методами программирования и в целом с технологическим процессом. Изучение каждого из этих направлений должно происходить не изолировано, а в тесных связях и зависимостях между ними.

Учитывая межпредметные связи программирования с другими школьными предметами и областями научного знания, можно утверждать, что системный подход является основополагающим в его преподавании, и другие методические подходы, которые могут быть применены к обучению программированию, должны дополнять и расширять его основные идеи и принципы, а не противоречить ему.

Деятельностный подход в обучении основан на «принципиальном положении о том, что психика человека неразрывно связана с его деятельностью и деятельностью обусловлена». Категория деятельности является основной в этом подходе. При этом деятельность понимается как преднамеренная активность человека, проявляемая в процессе его взаимодействия с окружающим миром, и это взаимодействие заключается в решении жизненно важных задач, определяющих существование и развитие человека [15].

В обучении информатике и программированию концепция деятельностного подхода предполагает:

- формирование готовности школьника на активное, адекватное и эффективное применение информационных и телекоммуникационных технологий в процессе обучения информатике,
- выявление и формирование творческой индивидуальности школьника,
- развитие его будущих профессиональных взглядов [16].

В настоящее время системный и деятельностный подходы интегрированы и положены в основу ФГОС второго поколения. Системно-деятельностный подход направлен на обеспечение реализации идеи непрерывного образования на уровне школы при условии сформированности у обучающихся универсальных учебных действий (УУД): регулятивных, познавательных, коммуникативных и личностных.

Под универсальными учебными действиями, в широком значении понимают «умение учиться, то есть способность субъекта к саморазвитию и самосовершенствованию путем сознательного и активного присвоения нового социального опыта» [17]. В более узком (собственно психологическом) значении термин «универсальные учебные действия» определяется как сово-

купность способов действия учащегося (а также связанных с ними навыков учебной работы), обеспечивающих его способность к самостоятельному усвоению новых знаний и умений, включая организацию этого процесса [17].

В повышении эффективности изучения программирования приоритетное место занимают познавательные УУД [18,19]. Познавательные универсальные учебные действия формируют систему способов познания окружающего мира, построения самостоятельного процесса поиска, исследования и совокупность операций по обработке, систематизации, обобщению и использованию полученной информации.

В блоке познавательных УУД выделяют общеучебные действия, знаково-символические, логические действия и действия постановки и решения проблем.

Сформированность общеучебных универсальных действий позволяет учащимся:

- самостоятельно выделять и формулировать проблему, ставить цели, задачи, находить необходимую информацию, осуществлять поиск решения поставленной задачи;
- структурировать знания;
- осознанно использовать термины программирования в речевых высказываниях в различных предметных областях;
- выбирать наиболее эффективный способ решения задачи по программированию в зависимости от конкретных условий.

Особую роль для повышения эффективности обучения программированию играют знаково-символические действия, которые обеспечивают конкретные способы преобразования учебного материала, представляют действия моделирования, выполняющие функцию отображения учебного материала; выделение существенного, отрыва от конкретных ситуативных значений, формирования обобщенных знаний. К ним относят:

- замещение – перенос существенных свойств объекта на модель;
- кодирование – написание алгоритма на понятном для исполнителя языке;
- схематизацию – представление результатов двух предыдущих этапов в наглядной, удобной для восприятия форме.
- моделирование – преобразование объекта из чувственной формы в модель (пространственно-графическую или знаково-символическую), где выделены существенные характеристики объекта

При выборе методов и способов решения задач, определении оптимальных алгоритмов, существенную роль играют логические универсальные действия. Уровень их развития влияет на выполнение учащимися действий по переводу записи алгоритма ре-

шения задачи из одной знаковой системы в другую. К этим действиям относятся:

- анализ объектов с целью выделения признаков (существенных, несущественных);
- синтез – составление целого из частей, в том числе самостоятельное достраивание с восполнением недостающих компонентов;
- выбор оснований и критериев для сравнения, сериации, классификации объектов;
- подведение под понятие, выведение следствий.

Многообразие и емкость содержания каждого из перечисленных учебных действий создают необходимые предпосылки успешного решения задач по программированию. Учитель должен быть уверен, что учащиеся владеют всей необходимой системой действий, составляющих умение программировать. Для этого целесообразно соотнести этапы решения задач по программированию со структурой действия, предложенной Н. Ф. Талызиной [20]:

1) предмет действия. Предметом может быть как материальный объект, так и слова, представления понятия;

2) цель действия. Для успешного усвоения действия школьников необходимо учить осознавать цель. Цель неразрывно связана с мотивом действия.

3) мотив. Побуждает человека ставить и достигать различные цели, выполнять соответствующие действия. Если ученик не видит необходимости выполнения тех или иных действий, то это ведет к тому, что учебная деятельность становится ему неинтересной, он не видит в ней никакого смысла.

4) система операций. Действие включает в себя несколько операций, которые должны выполняться в определенной последовательности: алгоритм действий. В одних случаях данный алгоритм неизменный, в других допускает изменения.

5) ориентировочная основа представляет собой систему условий, на которую реально опирается человек при выполнении действия. Она может быть полной или неполной, правильной или неправильной; от того, насколько ориентировочная основа будет полной и правильной, зависит эффективность решения задач.

6) результат. Результатом действия всегда является какой-либо продукт (материальный или мыслимый). Он может совпадать с целью, но может и не совпадать. Соотношение результата и цели определяет успешность усвоения действия.

Соотношение традиционных этапов решения задач по программированию со структурой действия представлено в табл. 1

Обобщая вышесказанное, можно сделать вывод, что для повышения эффективности обучения программированию необходимо сформировать у учащихся все виды познавательных универсальных учебных действий. При этом главную роль играют знаково-символические действия, что связано с необходимостью работы учащихся с разными знаково-символическими системами.

В ходе решения задач по программированию представленные компоненты действия обеспечивают три основных функции

действий: ориентировочную, исполнительную, контрольно-корректировочную.

Согласно Н.Ф. Талызиной, центральной является ориентировочная часть действия [20]. Именно данная часть обеспечивает успех деятельности. Ее можно раскрыть как процесс исполнения ориентировочной основы действия. Ученики часто недооценивают ориентировочную часть, спешат к исполнительной, то есть к преобразованию предмета, к получению результата.

Таблица 1

Соотношение этапов решения задач по программированию с действиями учащихся

Этап	Предмет действия	Цель действия	Система операций	Ориентировочная основа	Результат
Постановка задачи.	Текст задачи	Определение конечной цели программы.	Анализ текста задачи. Определение исходных данных, их ограничений. Определение выходных данных.	Анализ возможных решений, условий функционирования программы.	Четкая формулировка решаемой задачи.
Математическая формализация.	Математическая модель задачи (необходимые формулы, расчеты, алгоритмы и др.)	Формализация условий задачи	Выделить существенные свойства объекта Оптимизировать требования задачи Подобрать необходимые формулы	Контроль адекватности применения выбранных методов и средств обучения	Создание математической модели
Построение алгоритма, дающего решение задачи.	Блок-схема/ Псевдокод/ План решения	Разработка алгоритма	Определить последовательность шагов решения задачи, выполнить трассировку, частично формализовать алгоритм	Общие правила и рекомендации составления наглядной схемы алгоритма	Построение алгоритма в удобной для восприятия, наглядной форме
Составление программы.	Язык программирования	Написание программы с соблюдением синтаксических и семантических правил	Записать общую структуру программы, перевести созданный на предыдущем этапе алгоритм на язык программирования	Синтаксические и семантические правила программирования	Программа на выбранном языке
Отладка и тестирование программы.	Среда программирования, интерпретатор	Проверка корректности программы	Проверить синтаксические ошибки, выполнить контрольный пример	Средства проверки корректности программы	Интерпретация программы исполнителем
Анализ полученных результатов.	Программа, исполнитель	Верификация полученных результатов	Проанализировать работу программы при разных исходных данных	Соотнесение кода и действий исполнителя	Получение учащимся новых знаний

Однако на примере программирования можно показать, что ориентировочная часть занимает гораздо большее время и требует больших усилий по сравнению с исполнительской частью, которая сводится лишь к набору текста программы в выбранной среде.

Контрольная часть направлена на проверку правильности как результатов ориентировочной части, так и исполнительной, на слежение за ходом исполнения, на проверку соответствия его намеченному плану. В случае обнаружения ошибки, отклонения от правильного пути необходима коррекция, исправление. Следует отметить, что коррекционная часть действия при программировании также занимает достаточно большое время в связи с необходимостью отладки программы, ее верификации.

Если ориентировочным, контрольным и корректировочным действиям важно обучать учащихся постоянно, то чисто исполнительские функции должны быть по возможности исключены из учебного процесса, потому что они формируют механические навыки, не обеспечивают понимания.

Однако в процессе обучения программированию необходимо использование не только общего системно-деятельностного подхода, так как он не учитывает выделенных нами особенностей программирования: высокую степень абстрактности базовых понятий программирования, метапредмет-

ность и мультимодальность учебного материала. Следовательно, обозначенный во ФГОС системно-деятельностный подход должен быть дополнен идеями и принципами других методических подходов.

Как показал анализ научной и методической литературы, учебников по информатике в школьной практике в настоящее время распространены когнитивный и проблемный подходы к обучению программированию.

Когнитивный подход основывается на таксономии Б. Блума и предполагает овладение учащимися операциями программирования, начиная с элементарных, постепенно продвигаясь к сложным (рис. 1).

Основная цель когнитивного подхода – развитие когнитивных качеств обучающихся: быстроты и экономичности мышления, способности к решению нестандартных проблем, готовности воспринимать и анализировать противоречивую информацию. Обобщенным показателем уровня развития перечисленных качеств является интеллект – относительно устойчивая структура умственных способностей индивида [21].

В основные задачи обучения программированию на основе когнитивного подхода, помимо развития интеллекта, входит передача знаний учащимся. Однако это должны быть не формальные, а действенные знания, дающие возможность широкого и разнопланового оперирования ими.



Рис. 1. Схема модели когнитивной таксономии целей при обучении программированию

Анализ учебной литературы по информатике и ИКТ для 7-9 классов показывает, что когнитивный подход к обучению программированию является основным. В рекомендованных Министерством образования и науки РФ учебниках по информатике [22] в качестве самостоятельной работы учащихся предлагаются, в основном, задачи трех первых уровней: на знание синтаксиса языка программирования и умение работать в выбранной среде, на объяснение программы и ее модификацию в соответствии с изменившейся задачей. Наиболее полно все типы учебных задач рассмотрены в учебнике И.Г. Семакина [23]. Так, учащимся предлагается объяснить действие набранной программы, проанализировать ее действие при разных входных параметрах. Следует также отметить, что автор накладывает некоторые ограничения на решение задач, предлагает учащимся решить одну задачу несколькими способами. Данные приемы способствуют развитию алгоритмического мышления учащихся.

Реализация когнитивного подхода возможна с помощью метода демонстрационных примеров, метода ключевых задач, метода раскрутки задач и др. Метод демонстрационных примеров, предложенный в исследовании М.В. Швецкого [24], основан на идее Н.О. Вирта «...программирование представляет собой обширную и разнообразную деятельность,

часто требующую сложной умственной работы. Ошибочно считать, что ее можно свести лишь к использованию готовых рецептов. В качестве метода обучения остается тщательный выбор и рассмотрение характерных примеров» [25]. Метод демонстрационных примеров имеет определенные преимущества: позволяет научить учеников чтению программ, использовать ключевые задачи при программировании других задач и др.

Однако когнитивный подход, по мнению И.А. Бабушкиной, В.Ю. Нефедовой, А.Ю. Петухова и др., требует значительных временных затрат и предполагает решение большого числа однотипных вычислительных задач, которое приводит к снижению мотивации [26, 27, 28]. Следует отметить также, что при обучении программированию на основе когнитивного подхода, несмотря на работу учащихся с разными знаковыми системами, нет целенаправленного развития их знаково-символических действий.

Проблемный подход предполагает, что весь курс программирования строится на основе системы специально разработанных «проблемных» задач или ситуаций, содержание которых интересно для учащихся и, по возможности, связано с их будущей профессиональной деятельностью. Решая такого вида задачи, ученики расширяют свой опыт программирования (рис. 2).

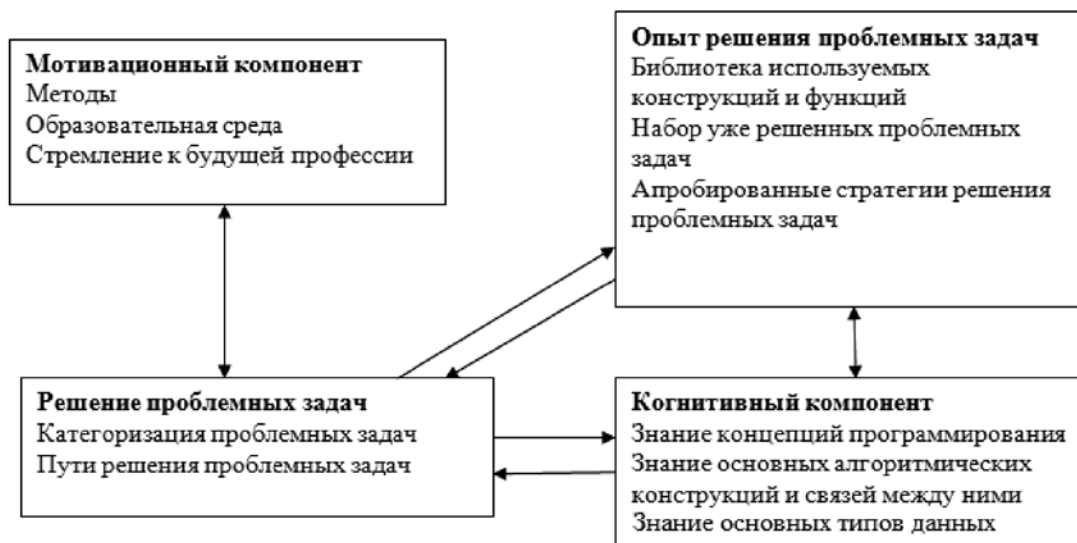


Рис. 2. Схема проблемной модели обучения программированию

Наиболее четко и последовательно понятие проблемной ситуации представлено психологом А.М. Матюшкиным [29], который выделяет следующие компоненты:

- неизвестное достигаемое знание или способ действия;
- познавательная потребность, побуждающая человека к интеллектуальной деятельности;
- интеллектуальные возможности человека, включающие его творческие способности и прошлый опыт.

Проблемному подходу соответствуют такие методы обучения, как проблемное изложение, исследовательский метод, эвристический метод и др.

По сравнению с когнитивным подходом в проблемном обучении программированию подчеркивается понимание процесса изучения учебного материала с точки зрения знаний и личного опыта учащихся. Как отмечает И. Н. Скопин, при изучении программирования на основе проблемного подхода «... изучение основ курса становится мотивированным, а не догматически преподнесенным ученикам как непререкаемая истина» [30].

С нашей точки зрения, в полном объеме проблемный подход не может быть использован на первом этапе обучения программированию, поскольку опирается на опыт деятельности в данном направлении у учащихся. Однако элементы проблемных ситуаций при обучении программированию успешно применяются и в начале обучения.

Обобщая вышесказанное, можно сделать вывод, что проблема изучения специфичной работы с новыми, не используемыми в естественном языке знаками, характерной для программирования, остается недостаточно исследованной. В связи с этим, как показывает анализ, при обучении программированию до сих пор возникают значительные трудности. Таким образом, в обучении программированию возникает проблема учета закономерностей взаимодействия человека со знаками и знаковыми системами, рассматриваемых семиотикой.

Семиотический подход к обучению программированию. В настоящее время все большую популярность в обучении информатике приобретает семиотический подход. Его основа разрабатывалась в философии и лингвистике (Ч.С. Пирс, Ч. Морис, Ф. де Соссюр и др.). Однако в связи с тем, что любая предметная область может быть описана как система знаков, а «...все информационные процессы представляют собой процесс прохождения информации по цепочке – иерархии кодов внутри некоторой системы..., т.е. процесс перекодирования (перевода) информации» [34], семиотический подход постепенно становится общеметодо-

логическим и развивается как в рамках специальной науки о знаках – семиотике, так и междисциплинарных областях знаний (теория коммуникаций, кибернетика и др.).

Целесообразность применения семиотического подхода к обучению программированию может быть обоснована тем, что язык программирования изначально имеет знаковую природу. Как всякий искусственный язык он имеет «повторный перевод», другими словами, знания о мире формализуются, «переводятся» на обычный язык, а затем находят свое отражение в алгоритмических конструкциях, встроенных функциях и др., что влечет за собой постепенное развитие языка программирования.

Применение семиотического подхода в информатике рассматривается О.Ф. Брыксиной, Н.А. Кургановой, К.Р. Пиотровской, В.И. Фоминым и др. Так, О.Ф. Брыксина рассматривает семиотический подход к формированию ИКТ-компетентности выпускника ВУЗа [35]. В.И. Фомин на основе данного подхода рассматривает профессиональную компетентность специалиста через умения оперировать различными знаковыми системами в контексте задач предметной области [36]. В диссертационном исследовании Н. А. Кургановой рассматривается семиотический подход к изучению содержательной линии «Формализация и моделирование» школьного курса информатики и развитию знаково-символической деятельности учащихся [37].

К основным категориям семиотического подхода относят знак, символ и образ. В семиотике существуют различные толкования данных понятий. Для четкого определения понятий в рамках работы был проведен их контент-анализ (табл. 2, 3, 4).

На основе представленной таблицы, в данном исследовании будем понимать знак как материальный предмет, являющийся представителем другого объекта или процесса и обладающий смысловым значением, принятым в определенной сфере деятельности.

Смысловым значением знаков языка программирования, как правило, являются какие-либо действия, например, «считать переменную», «сложить два числа», «повторить» и др., а также объекты окружающего мира (текстовая строка, массив чисел, файл и др.).

Исходя из проведенного анализа, будем понимать образ как внутреннее представление ученика об объекте действительности на основе чувственных восприятий и логических умозаключений. При этом следует отметить, что наглядный образ не мыслимое представление одного конкретного предмета – это прежде всего представление главных и важных сторон изучаемого предмета, составляющих его собственную сущность [57].

Таблица 2

Исследование содержания понятия «знак»

Автор	Признак понятия										
	Материальный пред-мет	Представитель другого объекта или процесса	Психическое образование	Ориентир деятельности	Смысловое значение	Субъективное образование	Интерпретирующая мысль	Средство общения	Аналог другого объекта	Элемент формализованной системы	Социально признанное единство содержания и выражения
Ч.С. Пирс	+	+			+	+	+				+
Ч.У. Моррис	+			+	+						
Ф. Соссюр	+		+		+						+
Р. Карнап	+				+					+	
Г. Фреге	+	+			+	+					
Л. Витгенштейн	+				+						
С.Ю. Головин	+				+						
А.А. Грицанов	+	+			+						
С.Я. Левит	+				+			+	+		
К. Фрумкин	+	+		+	+						
Л.С. Выготский	+				+			+			
А.В. Славин	+				+			+			+
Н.Г. Салмина	+			+	+						
А.В. Соколов	+				+						+
И.Г. Семакин	+				+					+	
	15	4	1	3	15	2	1	3	1	2	4

Таблица 3

Исследование содержания понятия «образ»

Автор	Признак понятия									
	Результат отражения реальности	Сознательное восприятие	Система отношений между характеристиками объекта	Целостная структура	Приобретенный опыт	Чувственные восприятия	Логические умозаключения	Форма освоения реальности	Внутреннее представление	
А.А. Реан	+		+	+						+
В. Вундт	+				+	+	+			+
Н. Бердяев	+									+
Л.С. Выготский	+									+
А. В. Славин	+									+
Ж. Пиаже	+	+								+
О.М. Фрейдберг	+			+						+
В. Кемеров								+		+
В. Зорин	+	+								+
Д.А. Леонтьев	+									+
	9	2	1	2	1	1	1	1		10

Контент-анализ понятия «символ» приведен в табл. 4. В научной литературе по программированию понятия «знак» и «символ» употребляются как синонимы (знаки языка программирования, символы языка программирования). Так, алфавит языка программирования понимается как словарь исходных символов. Однако, согласно проведенному контент-анализу, более правильно использовать понятие «знаки программирования».

В кодировании знаково-символические средства выполняют коммуникативную функцию, в случае решения задач средствами программирования: между учеником и компьютером. Ученику следует выделить алгоритмические конструкции: командные слова, встроенные функции, а также подпрограммы, которые необходимы для решения задачи.

В схематизации знаково-символические средства выполняют ориентировочную

Таблица 4

Исследование содержания понятия «символ»

Автор	Признак понятия						
	Отличительный знак	Внутреннее единство представления и значения	Аллегория	Выражение неизвестного	Устойчивый образ	Многозначность	Контекстуальное понимание
Е.Ф. Губский	+	+					
Г. Шмидт	+	+					
В. Кемеров			+				
В.М. Лейбин			+				
Э. Фромм				+			
В.В. Зеленский				+			
К. Юнг				+		+	+
С. В.Ильинский					+		
С.Ю. Головин						+	+
	2	2	1	2	1	2	2

Сущность семиотического подхода к обучению программированию заключается в целенаправленном развитии у учащихся знаково-символических действий (замещения, кодирования, схематизации, моделирования). При этом учитывается, что обучающийся, уже знаком с некоторыми знаковыми системами (русским языком, дорожными знаками, позиционными системами счисления) и имеющиеся у него знания, опыт деятельности можно эффективно использовать при изучении особенностей языков программирования как знаковых систем [65].

Под замещением понимают самый простой уровень знаково-символической деятельности, когда функции (признаки, свойства) замещаемого объекта переносятся на знаково-символическое средство. Так, при решении учебных задач учащимся нужно определить входные и выходные данные, на основе которых определить переменные, их тип и свойства.

роль, заключающуюся в структурировании реальности, выявлении связей между явлениями. В контексте обучения программированию схемы используются как средства активной наглядности (выступают в функции материализации), например, в качестве наглядной опоры плана деятельности по созданию программного кода. Схема может быть представлена учащимся в удобной для него форме (в виде блок-схемы, псевдокода, плана решения задачи и др.) и вмещать в себя результаты предыдущих этапов.

После создания схемы учащийся переходит к следующему этапу составления программы: ее записи на языке программирования с соблюдением синтаксических и семантических правил.

Потребность в применении семиотического подхода к обучению программированию и детализации процесса перевода учащимися алгоритма решения с естественного языка на язык программирования на основе

выделения подэтапов (замещение, кодирование, схематизация, моделирование) особенно ощутима при решении учебных задач, где используются «новые» непривычные рассуждения, из которых сразу же выделить алгоритмические структуры достаточно сложно.

Так как подход понимается как совокупность принципов [12], раскроем принципы семиотического подхода в контексте обучения программированию.

1. Принцип контекстуального понимания знака. Модель знака универсальна и может быть использована для анализа знаков, с которыми учащиеся встречаются в учебной деятельности и повседневной жизни. Отметим, что в определенном языке программирования каждому знаку всегда соответствует один смысл, а смыслу, в свою очередь, – один знак. Это необходимо для того, чтобы при машинной обработке текста программы каждый знак должен быть интерпретирован однозначно. Однако в разных языках программирования ряд знаков трактуется по-разному. Так знак « \Leftarrow » на языке Си интерпретируется как «присвоить», на языке Паскаль как «сравнить значение», «найти значение выражения» в среде MathCAD. То же самое мы наблюдаем в повседневной жизни: часто один и тот же знак может использоваться для обозначения разных объектов или действий. В этом проявляется явление полисемии – наличие у языкового знака более одного значения [66]. Наглядно полисемия знака « \Leftarrow » представлена на рис. 3.

Принцип контекстуального понимания знаковых систем выделяет умение правильно читать программу в качестве одного из основных умений учащихся в области программирования. Следует отметить, что исследования, проведенные в области программирования L.E. Winslow, показали, что между умением учащихся писать программу и читать программный код существует достаточно слабая корреляция [67]. Другими словами, умение «переводить» программу, записанную на каком-либо языке программирования, на естественный язык, не формируется само по себе в процессе обучения программированию, а требует от учителя применения специальных методов, упражнений.

Большинство терминов программирования также являются межпредметными, и могут использоваться в речи в различных контекстах. Из принципа контекстуального понимания знаковых систем следует, что при обучении программированию целесообразно представлять алгоритмические конструкции в нескольких формах (текстовое, графическое, аналитическое на разных языках программирования) и предлагать специальные задания на осмысление знаковых систем. Например, привести примеры синтаксического и семантического правил из различных знаковых систем: языка программирования Си, дорожных знаков, обозначения химических элементов и др. По мнению Н.А. Кургановой, такого рода упражнения, направленные на более глубокое осмысление природы знаков, на связь



Рис. 3. Полисемия знака

естественного и искусственных языков, «... приводит к полноценному формированию различных видов знаково-символической деятельности» [68].

2. *Принцип учета ведущего канала восприятия.* В целях концентрации внимания обучаемого его деятельность должна представлять собой логически обоснованную строгую последовательность предъявления учебной информации, а темп реализации обучающих воздействий на обучаемого (темп чтения лекции, развития исходной ситуации учебного задания) должен соответствовать уровню познавательных возможностей обучаемого. Из принципа дискретности осмысления семантической информации вытекает важное следствие: в том случае, если на один семантический объект-приемник одновременно поступают различные семантические сообщения, имеет место потеря семантической информации. Следовательно, учебный материал должен подаваться отдельными, связанными между собой блоками и при предъявлении очередного блока изначально должен повторяться предыдущий. Также должны учитываться особен-

ности восприятия информации учащимися с разными ведущими каналами восприятия (визуалами, аудиалами и кинестетиками). При работе с визуалами учителю необходимо использовать слова, описывающие цвет, размер, форму, местоположение; выделяя цветом различные пункты или аспекты содержания; записывая действия; используя схемы, таблицы, наглядные пособия и др. Аудиалы хорошо воспринимают вариации голоса (громкость, паузы, высоту) и нуждаются в озвучивании поставленных перед ними задач. При работе с кинестетиками следует помнить, что они воспринимают большую часть информации через осязание и с помощью движений, таким образом, ученикам просто необходимо работать с реальными моделями, исполнителями и др.

3. *Принцип возрастания степени абстрактности.* С начальной школы ученики изучают естественные и искусственные знаковые системы в иерархически обоснованной последовательности (рис. 4) от знаков и знаковых систем наименьшей абстрактности к знакам и системам все большей степени абстрактности [69].



Рис. 4. Иерархия знаковых систем по степени абстрактности

Это связано с тем, что процесс познания всегда начинается с рассмотрения конкретных, чувственно-воспринимаемых предметов и явлений, их внешних признаков, свойств, связей. Только в результате изучения чувственно-конкретного человек приходит к каким-то обобщенным представлениям, понятиям, к тем или иным абстракциям. Рассматриваемый принцип подчеркивает тот факт, что четкое понимание проблемы, хорошее знание ее предметной области (то есть работа на уровне естественной знаковой системы) должно предшествовать попыткам ее формализации.

Следовательно, с учетом рассмотренного принципа обучение учащихся решению задач по программированию должно состоять из нескольких этапов: постановка задачи, знакомство с исполнителем (естественная знаковая система), составление схемы решения задачи (образная система), написание программы (языковая система), компиляция программы (система записи, графема), эмпирическое наблюдение за исполнителем (формализованная система первого порядка) и верификация (формализованная система второго порядка).

4. Принцип полифункциональности знака. Принцип определяет необходимость рассмотрения языка программирования не только как средства решения учебных задач, но и как знаковой системы, что предполагает знакомство учащихся с системой специальных знаков и их конструкций, выделение

репрезентативной, экспрессивной и прагматической функций знаков.

5. Принцип триединства представлений. Данный принцип означает, что ученик должен соотносить знак (например, операторы, командные слова языка программирования) или получаемую из совокупности знаков систему (программу) и обозначаемый ими объект действительности с учетом своего опыта, сформированных ценностей. Как отмечает С.А. Бешенков, в процессе обучения важно «...не оставить ученика внутри знаковой системы» [18], он должен уметь грамотно интерпретировать полученные им результаты применительно к реальным объектам или процессам, понимать необходимость верификации полученных результатов.

Следует подчеркнуть, что значение знака зависит от объективно существующей действительности (денотата) и субъективных представлений об этой действительности (концепта). Причем любой знак может мыслиться не только в связи с обозначаемым им объектом, но и в связи с тем, какой смысл может быть приписан этому знаку (рис. 5, 6).

По мере накопления информации, обогащения и уточнения знаний об объекте исследования создаются такие наглядные модели, которые наиболее адекватно отражают структуру исследуемого объекта.

Накопление знания о том или ином объекте или явлении происходит постоянно, соответственно обогащается и смысловое содержание знака (рис. 7).

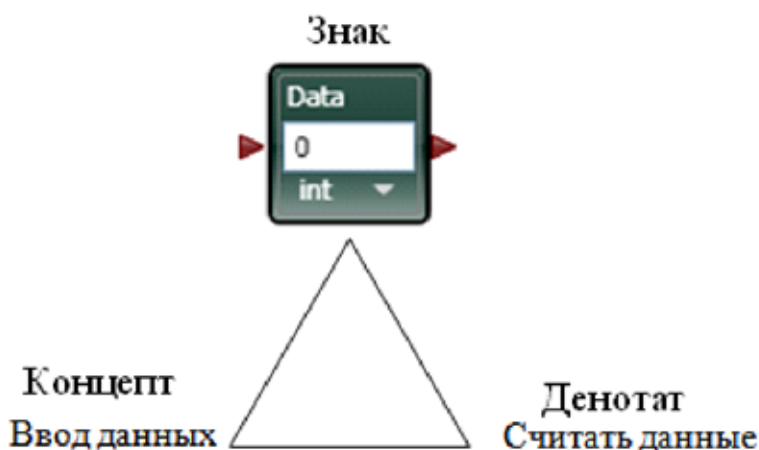


Рис. 5. Триадачная модель знака на примере блока «Data»



Рис. 6. Триадичная модель знака на примере изображения стрелки

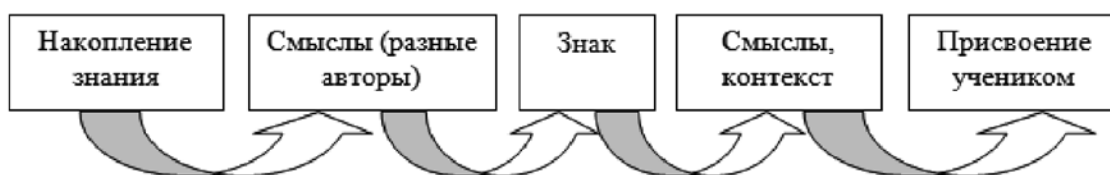


Рис. 7. Процесс создания знака

В итоге ученик должен «присвоить» знак, то есть уметь использовать его в умственной деятельности и речи. К сожалению, умение устного объяснения понятий программирования, сравнение употребления их в разных контекстах часто не фиксируется как одна из важнейших целей обучения и формируется явно на недостаточном уровне. С этим связаны трудности учащихся при объяснении сути созданных программ, защите проектов.

Также для осознанного решения задач программирования нужна не только когнитивная, но и эмоциональная база, которая, согласно исследованиям в семиотике, является неотъемлемой частью образа, который складывается у учащегося. В доказательство данного тезиса можно привести высказывание Г. Фреге, который объясняет процесс интерпретации знака следующим образом: «...представление о вещи есть внутренний образ, возникший у меня на основе моих впечатлений об этой вещи, а также в результате моей деятельности, физической и мыслительной, связанной с этой вещью» [39].

Нарушение принципа триединства представления ведет к тому, что школьник может осуществлять разнообразную и успешную деятельность по решению задач, в том числе практического содержания, но в рамках типовых подходов, методом аналогии. При решении же учебных задач творческого ха-

рактера у таких учащихся возникают значительные трудности.

Анализ методических подходов к обучению программированию позволил сделать вывод о дидактическом потенциале идей и принципов семиотического подхода для повышения эффективности обучения программированию. Определены методические основания использования семиотического подхода в обучении программированию учащихся 7–9 классов: согласованность с основополагающим системно-деятельностным подходом, целенаправленное развитие знаково-символических действий учащихся, повышение уровня учебной мотивации и уровня сформированности предметных знаний.

Список литературы

1. Andersen P. A Theory of Computer Semiotics: Semiotic Approaches to Construction and Assessment of Computer Systems. – URL: <http://acl.ldc.upenn.edu/J/192/192-4011.pdf>.
2. Reinfelds J. A Three Paradigm Course for CS Majors, Proceedings, 26th ACM SIGCSE Technical Symposium on Computer Science Education, 1995, p. 223-227.
3. Афанасьев В.Г. О системном подходе в социальном познании // Вопросы философии. – 1973. – №6. – С. 23-34.
4. Баумане К.И. Методика обучения семиотике языков программирования будущих учителей информатики : дис. ... канд. пед. наук: СПб., 2004. – 289 с.
5. Касьянова Е.В. Адаптивные методы и средства поддержки дистанционного обучения программированию: дис. ... канд. физ.-мат. наук. – Новосибирск, 2006. – 181 с.
6. Леднев В.С. Научное образование. – М.: МГАУ, 2002. – 120 с.

7. Одинцов И.О. Профессиональное программирование. Системный подход. / И.О. Одинцов СПб.: БХВ-Петербург, 2004. – 624 с.
8. Поляков К.Ю., Еремин Е.А. Информатика: учебник для 10 класса. В 2-х частях. – М., 2012. – 688 с.
9. Прохоров А.М. Большой энциклопедический словарь / А.М. Прохоров. – СПб.: Норинт, 2004 – 1456 с.
10. Ракитина Е.А. Построение методической системы обучения информатике на деятельностной основе: дис.... д-ра. пед. наук. – М., 2002. – 485 с.
11. Рыжова Н.И. Развитие методической системы фундаментальной подготовки будущих учителей информатики в предметной области: дис.... док. пед. наук. – СПб., 2000. – 429 с.
12. Слостенин В.А. Педагогика. / В.А. Слостенин, И.Ф. Исаев, Е.Н. Шиянов. – М.: Академия, 2007. – 170 с.
13. Новиков А.М. Педагогика: словарь системы основных понятий/ А.М. Новиков. – М.: Издательский центр ИЭТ, 2013. – 268 с.
14. Макарова Н.В. Системно-деятельностный подход при обучении информатике в средней школе // Педагогическое образование в России. – 2002. – №5. – С. 88-95.
15. Фридман Л.М. Наглядность и моделирование в обучении. – М.: Знание, 1984. – 80 с.
16. Григорьева М.А. Деятельностный подход в обучении школьников информатике с использованием мобильных компьютерных систем: дис. ... канд. пед. наук. – М., 2011 – 182 с.
17. Козлов В.В. Фундаментальное ядро содержания образования / В.В. Козлов, А.М. Кондаков. – М.: Просвещение, 2011. – 79 с.
18. Бешенков С. А. Курс информатики в современной школе: доклад на 7-м заседании семинара «Методологические проблемы наук об информации». – М.: ИНИОН РАН, 2012. – 9 с.
19. Миндзаева Э.В. Развитие универсальных учебных действий в курсе информатики 5-6 классов: дис. канд. пед. наук. – М., 2009. – 180 с.
20. Талызина Н.Ф. Педагогическая психология. – М.: Академия, 2006. – 288 с.
21. Локалова Н.П. Школьная неуспеваемость: причины, психокоррекция, психопрофилактика. – СПб.: Питер, 2009. – 368 с.
22. Приказ об утверждении федеральных перечней учебников, рекомендованных (допущенных) к использованию в образовательном процессе в образовательных учреждениях, реализующих образовательные программы общего образования и имеющих государственную аккредитацию, на 2015/2016. – URL: минобрнауки.рф/документы/5812 (дата обращения 5.12.2015).
23. Семакин И.Г. Информатика. Базовый курс. 9 класс. – М.: Бином, 2007. – 158 с.
24. Швецкий М.В. Методическая система фундаментальной подготовки будущих учителей информатики в педагогическом вузе в условиях двухступенчатого образования: авторефер. дис. ... док. пед. наук. – СПб, 1994. – 36 с.
25. Вирт Н. Алгоритмы и структуры данных. – М.: Мир, 1989. – 360 с.
26. Бабушкина И.А. Практикум по объектно-ориентированному программированию. – М.: Бином, 2004. – 366 с.
27. Нефедова В.Ю. Объектно-ориентированное программирование для школьников. Теоретические и методические рекомендации к обучению в системе дополнительного образования: LAP LAMBERT Academic Publishing, 2012. – 129 с.
28. Петухов А.Ю. Формирование информационной компетентности школьников в системе дополнительного образования на примере учебного модуля курса «Программирование»: дис. ... канд. пед. наук. Бийск, 2006. – 271 с.
29. Матюшкин А.М. Проблемные ситуации в мышлении и обучении. – М.: Директмедиа Паблишинг, 2008. – 392 с.
30. Скопин И.Н. Проблемные задачи при изучении общих методов информатики и программирования // Вестник Российского университета дружбы народов – 2011. – №4. – С. 21–33.
31. Пирс Ч.С. Избранные философские произведения. – М.: Логос, 2000. – 412 с.
32. Соссюр Ф. Курс общей лингвистики. – М.: Эдиториал УРСС, 2004. – 278 с.
33. Моррис Ч.У. Основания теории знаков // Семиотика. – 1983 – №2. – С. 3789.
34. Мечковская Н. Б. Семиотика. Язык. Природа. Культура. – М.: Академия, 2007. – 432 с.
35. Брыксина О. Ф. Формирование профессиональной готовности студентов педагогического колледжа к реализации содержания образования на уровне учебного предмета «Информатика»: дис. ... канд. пед. наук. Самара, 1999. – 226 с.
36. Фомин В. И. Развитие содержания подготовки к информационно-аналитической деятельности на основе семиотического подхода: дис. ... док. пед. наук: 13.00.02 / В. И. Фомин. – Самара, 2009. – 385 с.
37. Курганова Н. А. Развитие знаково-символической деятельности учащихся в процессе обучения информатике на основе семиотического подхода: автореферат дис. ... канд. пед. наук. Омск, 2006. – 25 с.
38. Карнап Р. Значение и необходимость. Исследование по семантике и модальной логике. – М.: ЛКИ, 2007. – 384 с.
39. Фреге Г. Смысл и денотат // Семиотика и информатика. – 1997. – №35. – С. 352379.
40. Витгенштейн Л. Избранные работы. – М.: Издательский дом «Территория будущего», 2005. – 440 с.
41. Головин С.Ю. Словарь практического психолога. – Минск: Харвест, 1998. – 301 с.
42. Грицанов А.А. Новейший философский словарь. – Минск: Интерпрессервис; Книжный Дом, 2001. – 1280 с.
43. Левит С.Я. Культурология. XX век. Энциклопедия. Том 1. – СПб.: Университетская книга, 1998 – 907 с.
44. Фрумкин К.Г. Термины Хайдеггера. – URL: http://pounivers.narod.ru/ofirs/h_voc.htm (дата обращения 8.12.2015)
45. Выготский Л.С. Психология развития человека. – М.: Изд-во Эксмо, 2005. – 1136 с.
46. Славин А. В. Наглядный образ в структуре познания. – Л.: Политиздат, 1971. – 267 с.
47. Салмина Н.Г. Знак и символ в обучении / Н.Г. Салмина. – М.: Изд-во Моск. ун-та, 1988. – 288 с.
48. Соколов А.В. Общая теория социальной коммуникации. – СПб.: Изд-во В.А. Михайлова, 2002. – 461 с.
49. Реан А.А., Бордовская Н.В., Розум С.И. Психология и педагогика. – СПб.: Питер, 2002. – 432 с.
50. Вундт В. Введение в психологию. – М.: КомКнига, 2001. – 167 с.
51. Бердяев Н.А. Избранные произведения. – Ростов-на-Дону: Феникс, 1997. – 544 с.
52. Пиаже Ж. Избранные психологические труды. М.: Международная педагогическая академия, 1994 г. – 680 с.
53. Фрейденберг О.М. Образ и понятие // Историческая поэтика: хрестоматия-практикум под ред. С.Н. Бройтман. – М.: Академия, 2004. – С. 15–29; 91–113.
54. Леонтьев Д.А. Психология смысла: природа, строение и динамика смысловой реальности. – М.: Смысл, 2007. – 511 с.
55. Зорин В. И. Евразийская мудрость от А до Я: философский толковый словарь. – Алматы: Создiк-Словарь, 2002. – 407 с.

-
56. Кемеров В. Е. *Философская энциклопедия*. – М.: Панпринт, 1998. – 1064 с.
57. Усольцев А. П. *Управление процессами саморазвития учащихся при обучении физике*. – Екатеринбург: Изд-во УрГПУ, 2006. – 213 с.
58. Губский Е.Ф. *Философский энциклопедический словарь* – М.: ИнфраМ, 2009. – 569 с.
59. Шмидт Г. *Философский словарь*. – М.: Республика, 2003. – 576 с.
60. Лейбин В.М. *Словарь-справочник по психоанализу*. – М.: АСТ, 2010. – 956 с.
61. Фромм Э. *Забытый язык. Введение в науку понимания снов, сказок и мифов*. – М.: АСТ, 2010. – 320 с.
62. Зеленский В.В. *Толковый словарь по аналитической психологии*. – М.: Когнито-Центр, 2008. – 336 с.
63. Юнг К. *Архетип и символ*. – М.: Ренессанс, 1991. – 304 с.
64. Ильинский С. *Нейролингвистическое программирование: оперативный словарь-справочник*. – М.: АСТ, 2006. – 478 с.
65. Кауфман В.Ш. *Языки программирования: концепции и принципы*. – М.: ДМК-пресс, 2010. – 464 с.
66. Ярцева В. Н. *Лингвистический энциклопедический словарь*. – М.: Сов. Энциклопедия, 1990. – 685 с.
67. Winslow L.E. *Programming Pedagogy – A Psychological Overview*, SIGCSE Bulletin 28(3), 1996, p. 17-22.
68. Курганова Н.А. *Развитие знаково-символической деятельности учащихся в процессе обучения информатике на основе семиотического подхода: автореф. дис... канд. пед. наук*. – Омск, 2006. – 25 с.
69. Соломоник А.Б. *Семиотический подход к отбору материала для обучения в средней школе // Медиа. Информация. Коммуникация* – 2012. – № 3.– URL: <http://mic.org.ru>.