

**(ОБРАЩАЕМ ВНИМАНИЕ, ЧТО НЕКОТОРЫЕ ФРАГМЕНТЫ РАБОТЫ
ИЗВЛЕЧЕНЫ ИЗ ДАННОГО ДОКУМЕНТА)**

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРЕПОДАВАНИЯ ОСНОВ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ В СРЕДНИХ УЧЕБНЫХ ЗАВЕДЕНИЯХ	5
1.1 Теоретические основы развития алгоритмического мышления учащихся .	5
1.2 Формирование алгоритмического мышления в средних учебных заведениях	7
1.3 Особенности методики обучения учащихся основам программирования	15
1.4 Выводы по главе 1	17
ГЛАВА 2 СТРУКТУРА ОБУЧЕНИЯ ИНФОРМАТИКЕ В СРЕДНЕЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ШКОЛЕ	18
2.1 Особенности изучения информатики в младших, средних и старших классах	18
2.2 Факультативные курсы по информатике	22
2.3 Реализация политехнизма	23
2.4 Межпредметные и внутрипредметные связи при преподавании информатики	24
ГЛАВА 3 МЕТОДИКА ОБУЧЕНИЯ ОСНОВАМ АЛГОРИТМИЗАЦИИ	27
3.1 Основные требования к учащимся	27
3.2 Понятие алгоритма	27
3.3 Методические рекомендации по решению задач	27
ГЛАВА 4 МЕТОДИКА ИЗУЧЕНИЯ ЯЗЫКА ПРОГРАММИРОВАНИЯ	28
4.1 Основная цель изучения раздела. Общие вопросы методики	28
4.2 Понятие языка программирования, данные, величины, команды	28
4.3 Изучение языков программирования	28
4.4 Системы программирования	Ошибка! Закладка не определена.
4.5 Методические рекомендации по решению задач по курсу алгоритмизации и программирования	28
ЗАКЛЮЧЕНИЕ	29
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	31
ПРИЛОЖЕНИЕ 1	32
ПРИЛОЖЕНИЕ 2	33

ВВЕДЕНИЕ

Современный уровень развития науки и техники, компьютеризация современного общества, инновационные технологии предъявляют новые требования к умениям и навыкам учащихся. От нового поколения требуются умения планировать свои действия, самостоятельно добывать нужные знания, решать разнообразные практические и теоретические задачи, моделировать будущий процесс, адекватно и умело действовать, сталкиваясь с возникающими проблемами, словом составлять алгоритмы.

Под алгоритмическим стилем мышления понимается система мыслительных действий, приемов, которые направлены на решение как теоретических, так и практических задач, результатом чего есть алгоритмы как специфические продукты человеческой деятельности.

В условиях современного информационного общества и глобальных коммуникаций становятся актуальными вопросы педагогической эффективности методов, средств и организационных форм обучения информатике, благодаря которым у учащихся будет развиваться алгоритмический стиль мышления, возможный в школьном возрасте.

Навыки планирования, привычка к точному и полному описанию своих действий помогают учащимся разрабатывать алгоритмы решения задач самого разного происхождения. Проблема развития алгоритмического стиля мышления учащихся особенно актуальна в современном образовательном процессе. Алгоритмическое мышление является необходимой частью научного взгляда на мир. В то же время оно включает и общие учебные навыки и умения, такие как: умение сравнивать, анализировать, обобщать, абстрагировать, видеть структурные, иерархические и причинно-следственные связи, разбивать задачи на подзадачи.

Целью работы является создание системы работы по развитию алгоритмического мышления учащихся на второй ступени обучения.

В процессе работы необходимо решить следующие задачи:

1. Провести анализ теоретических основ преподавания алгоритмизации и программирования на уроках информатики в школе.
2. Рассмотреть структуру обучения информатике в средней общеобразовательной школе.
3. Разработать методические рекомендации для обучения основам алгоритмизации.
4. Разработать методические рекомендации по решению задач по курсу алгоритмизации и программирования.

Объект исследования – процесс обучения ОАиП в средних учебных заведениях.

Предмет исследования – методика преподавания основ алгоритмизации и программирования в средних учебных заведениях.

Методы исследования: наблюдение, анализ, методы педагогического проектирования для достижения цели исследования.

Практическая значимость проведенного исследования заключается в разработке учебно-методических материалов для организации обучения ОАиП на уроках информатики в средних учебных заведениях.

ГЛАВА 1

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРЕПОДАВАНИЯ ОСНОВ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ В СРЕДНИХ УЧЕБНЫХ ЗАВЕДЕНИЯХ

1.1 Теоретические основы развития алгоритмического мышления учащихся

В настоящее время теория и педагогика преподавания информатики находятся в стадии активного развития. Несмотря на то, что этот предмет существует в школах почти два десятилетия, многочисленные задачи в рамках этой новой педагогической области появились лишь недавно, им не хватает обширного теоретического обоснования или экспериментальной проверки.

В соответствии с образовательным стандартом [7] по «Информатике» как учебному предмету учебная программа на уровне общего среднего образования направлена на развитие компьютерной грамотности, развитие логического и алгоритмического мышления и формирование информационной культуры у учащихся.

Преподавание информатики основывается на двух фундаментальных подходах: системной методологии и методологии, основанной на деятельности. Системный подход предполагает рассмотрение предмета как взаимосвязанной системы, интеграцию его целей, задач и функций для целостного развития учащихся во взаимосвязи и взаимозависимости. Между тем, деятельностный подход направлен на создание оптимальных условий для вовлечения учащихся в разнообразную деятельность, направленную на овладение современными информационными технологиями, с акцентом на продуктивные и творческие аспекты их вовлечения.

На раннем этапе своего обучения учащиеся должны усвоить концепцию алгоритмов, получив четкое представление о том, что влечет за собой алгоритм.

Владение алгоритмами, включающее последовательность действий, переплетенных с образным и логическим мышлением, является критерием интеллектуального мастерства и творческих способностей человека.

Изучение алгоритмизации в рамках школьной серии по информатике преследует две основные цели: во-первых, способствовать развитию алгоритмического (или операционного) мышления учащихся; во-вторых, углубляться в аспекты программирования. Программирование начинается с разработки алгоритма, выделяя продвинутое алгоритмическое мышление как важнейшую подготовку для опытного программиста. В настоящее время

учебники по информатике подчеркивают ключевую роль этой темы в развитии учащихся.

Начальный аспект связан с укреплением основополагающего элемента учебной программы по информатике: разъяснением сути управления программным обеспечением. Учащиеся углубляются в понимание языков программирования, постигают природу программ на этих языках и разрабатывают программы в современных средах программирования.

Последующий аспект ориентирован на карьеру. В современных условиях профессия программиста занимает видное место и престижна. Включение программирования в школьную программу позволяет учащимся выявить свои способности в этой области. Однако более специализированный курс информатики в средней школе лучше соответствовал бы этой цели.

Решение задач на компьютере требует создания алгоритма. Эта способность включает в себя разработку стратегий, эмпирическую проверку гипотез, предвидение результатов, анализ, оптимизацию, уточнение алгоритмов и формальное изложение их на языке исполнителя. Оценка алгоритмических навыков школьников зависит от этих способностей, отражающих уровень их развития.

Следовательно, особого внимания заслуживает определение приоритетов в совершенствовании алгоритмических навыков подрастающего поколения.

Начинать обучение алгоритмизации и программированию с помощью визуализированной среды, такой как Scratch, представляется разумным подходом. Это предположение вытекает из педагогического эксперимента, проведенного в период с 2011 по 2016 год в образовательных учреждениях Минска, Минской, Витебской областей и «Образовательном центре Парка высоких технологий» [6].

Целью эксперимента была оценка эффективности методики, разработанной для обучения учащихся 4-9 классов основам алгоритмизации и программирования с использованием языка программирования Scratch в визуализированной среде. Полученные результаты подчеркнули дидактические достоинства визуальных языков в рамках образовательного процесса, охватив несколько аспектов:

Они лучше соответствуют психологическим особенностям школьников, используя знакомую лексику и интерфейсы, интегрируя игровые элементы и облегчая практико-ориентированные задачи, такие как анимация персонажей.

Эти визуальные языки способствуют согласованности между деятельностью и познанием, визуализируя результаты, согласуясь с эффективными моделями обучения.

Важно отметить, что они синхронизируются с основополагающим содержанием учебной программы «Основы алгоритмизации и программирования».

1.2 Формирование алгоритмического мышления в средних учебных заведениях

Огромные возможности для развития алгоритмического стиля открываются при изучении основ алгоритмизации и программирования. Программирование помогает учащимся пройти все основные этапы формализованного решения некоторой творческой точно сформулированной задачи.

Актуальность изучения основ алгоритмизации и программирования претерпела значительные изменения. Главенствующее место в образовании заняли компьютерные информационные технологии, позволяющие упростить процесс обработки информации. Курс информатики стал иметь ярко выраженное практическое назначение, программирование отошло на второй план. Направление в сторону подготовки пользователя компьютером явно понизил уровень формирования алгоритмического мышления.

Технология решения задач на компьютере – это не только составление программы и получение загрузочного модуля, а формирование модели, составление алгоритма, отладка программы и ее тестирование. Без программирования развитие алгоритмического стиля мышления практически невозможно, так как отсутствует возможность компьютерного эксперимента проверки работоспособности алгоритма. Поэтому изучение основ алгоритмизации и программирования на второй ступени обучения основано на работе в системе программирования.

Решение задачи на компьютере невозможно без создания алгоритма. Умения решать задачи, разрабатывать стратегию ее решения, выдвигать и доказывать гипотезы опытным путем, прогнозировать результаты своей деятельности, анализировать и находить рациональные способы решения задачи путем оптимизации, детализации созданного алгоритма позволяют судить об уровне развития алгоритмического мышления учащихся.

Поскольку алгоритмическое мышление в течение жизни развивается под воздействием внешних факторов, то в процессе дополнительного воздействия возможно повышение уровня его развития. В методической литературе по информатике отмечены различные способы формирования алгоритмического мышления учащихся: проведение систематического и целенаправленного применения идей структурного подхода (А.Г. Гейн, В.Н. Исаков, В.В. Исакова,

В.Ф. Шолохович), повышение уровня мотивированности задач (В.Н. Исаков, В.В. Исакова), постоянная умственная работа (Я.Н. Зайдельман, Г.В. Лебедев, Л.Е. Самовольнова). Эффективным способом формирования алгоритмического мышления учащихся на второй ступени обучения в курсе «Основы алгоритмизации и программирования» является обучение построению алгоритмов и их использованию при решении большого класса задач.

Формирование алгоритмической культуры учащихся способствует осознанному восприятию учебного материала, что предполагает обязательное наличие общих представлений: об алгоритме и его свойствах; о языковых средствах записи алгоритмов (развернутая форма, табличная форма, блок-схема); об алгоритмических процессах (линейном, разветвляющемся, циклическом). Алгоритмическая культура учащегося должна содержать следующие компоненты: понимание сущности алгоритма и его свойств; понимание сущности языка как средства для записи алгоритма; владение приёмами и средствами для записи алгоритмов; владение алгоритмами школьного курса информатики; понимание элементарных основ программирования на компьютере. Обучение алгоритмам должно строиться с учётом следующих принципов:

- создание у учащихся полной ориентировочной основы его применения;
- осуществление алгоритмизации на основе приёмов, раскрывающих их происхождение;
- алгоритмическая линия должна пронизывать весь процесс обучения информатики в школе;
- обеспечение взаимосвязи алгоритмов.

Работа по алгоритмам развивает интерес учащихся к процессу обучения, они стремятся заменить предложенный алгоритм более простым и обосновать целесообразность такой замены, что развивает их творческое и конструктивное мышление. Алгоритмизация обучения предполагает единство между анализом и синтезом и активно влияет на развитие творческого мышления учащихся. Свободное творчество возможно только на базе осознанных алгоритмов.

Алгоритмическое мышление помогает формировать навыки:

- уметь планировать структуру действий, необходимых для достижения заданной цели при помощи фиксированного набора средств;
- строить информационные структуры для описания объектов и средств;
- организовывать поиск информации, необходимой для решения поставленной задачи;
- правильно, четко и однозначно формулировать мысль в понятной собеседнику форме и правильно принимать текстовое сообщение;
- своевременно обращаться к компьютеру при решении задач из любой области;

- формировать навыки анализа информации, умение структурировать ее;
- критически оценивать результаты.

Обучение школьника основам алгоритмического мышления базируется на понятии исполнителя. Основой для введения исполнителей служат задачи. Чтобы описать исполнителя, нужно задать среду, в которой он действует, и действия, которые он может выполнять при нажатии кнопок управления или по введенным командам. При этом используются такие среды как «ЛогоМиры», «Robot», «Чертежник», «Алгоритмика», пакет «Роботландия». Знакомство с ними происходит на занятиях факультативного курса «Занимательная информатика», предназначенного для учащихся 6-7 классов. В рамках действующей учебной программы на уроках информатики изучается компьютерный исполнитель «Чертежник», реализованный в среде программирования Pascal ABC.

Составление программ – весьма сложный процесс, включающий в себя значительное число качественно разнообразных этапов. Наиболее сложные из них – постановка задачи и ее алгоритмизация. Именно этим этапам необходимо уделять основное внимание при разработке уроков. Как правило, в учебных пособиях при объяснении данных разделов приводятся конкретные задачи различных типов и соответствующие им алгоритмы с некоторыми частными пояснениями.

К сожалению, большая часть учащихся может решить задачи только основываясь на разработанных ранее аналогичных примерах, т.е. не способна к самостоятельному продуктивному творчеству. В качестве основного средства описания алгоритмов методически правильно выбирать блок-схемы алгоритмов – наиболее наглядный, понятный и наиболее естественный для человека способ, т.к. человек мыслит образами и ему легче воспринимать образы, нежели текст. Развитие умения просчитывать результаты работы предложенных блок-схем являются так же достаточно эффективным средством формирования алгоритмического стиля мышления.

Очевидно, что разработка алгоритмов должна проводиться по нарастанию уровня сложности: реализация линейного следования, ветвления, цикла с параметром. Только после усвоения учащимися основных алгоритмических структур можно приступать к задачам на обработку структурированных типов данных. При этом необходимо приучать учащихся самостоятельно оценивать правильность разработанных алгоритмов на основе анализа полученных при расчетах результатов. Необходимо ориентировать учащихся на самостоятельность, обучать их правилам разработки текстовых примеров и анализу результатов.

Очень важным моментом является усвоение учащимися типовых алгоритмов решения стандартных классов задач. Например, одна из трудно

усваиваемых на уроках тем – обработка массивов данных. При изучении данной темы желательно четко сформулировать основные классы задач на обработку массивов и дать учащимся примеры алгоритмов их решения.

Для иллюстрации рассмотрим задачи по теме «Обработка одномерных массивов данных» средствами языка программирования Pascal ABC. Самый распространяемый класс задач на обработку одномерного массива – нахождение суммы элементов массива. Примерами таких задач могут быть задачи на нахождение разнообразных суммарных итоговых показателей.

Пусть требуется найти сумму элементов заданного массива из 10 целых чисел.

Пример решения данной задачи может иметь вид:

Program summa;

Var a: array[1..10] of integer; i, s: integer;

Begin

s:=0;

for i:=1 to 10 do

begin

 read(a[i]);

 s:=s+a[i];

end;

writeln(s);

End.

В программе используется стандартный алгоритм накопления суммы по шагам: сначала предполагается, что искомое значение суммы равно нулю; далее на каждом шаге цикла к имеющему значению суммы прибавляется очередное слагаемое, и полученный результат сохраняется как значение переменной s.

Для изучения данного алгоритма можно предложить учащимся для обсуждения следующие вопросы:

Что измениться в данном алгоритме, если:

- вместо суммы потребуется найти произведение элементов массива;
- потребуется найти сумму не всех элементов, а только тех, которые удовлетворяют определенным условиям (нахождение положительных элементов массива);
- потребуется найти среднее арифметическое значение элементов массива?

Таким образом, учащиеся могут самостоятельно сформулировать следующие типовые алгоритмы обработки массивов:

- нахождение суммы элементов массива, состоящих из n чисел;
- нахождение произведение элементов массива, состоящих из n чисел;

– нахождение суммы элементов массива, удовлетворяющей определенным условиям;

– нахождение среднего арифметического значения элементов массива.

Для закрепления навыков работы со стандартными алгоритмами можно предложить учащимся задачи, на комбинировании изученных ранее алгоритмов.

Данная методика доступна любой категории учащихся на начальной стадии обучения информатике и программированию. Она базируется на известных подходах к построению программ – структурном программировании и проектировании «сверху вниз» и является их развитием. Методика достаточно универсальна и применима при решении задач обработки не только числовой, но и текстовой и графической информации.

И в дополнение, как обязательный элемент обучения алгоритмизации должна входить отладка алгоритма каждого вида. Она позволяет учащимся «прочувствовать» суть каждого вида алгоритмов, предлагает им правила самоконтроля – правила нахождения ответов к задачам алгоритмизации, позволяет понять и закрепить принципы отладки программ. Как показывает практика, использование в обучении основ алгоритмизации и программирования методики формализованного перехода от схемы алгоритма к программе на языке программирования позволяет научить основную массу учащихся составлять алгоритмы и программы достаточно сложных задач.

Самая высокая степень развития алгоритмического мышления – умение решать нестандартные (олимпиадные) задачи, здесь, естественно, уже не обойтись без навыков творческого мышления, без наличия непродуктивного, т.е. не основанного на шаблонах подхода к решению задач.

Алгоритмический стиль мышления характеризуется точностью, определенностью, формальностью и, как правило, связывается с теоретической деятельностью, при этом мышление должно быть фоновым, незаметным и полезным, как умение читать.

В практике обучения информатике учитель пользуется различными методами и способами донесения как теоретической, так и практической информации до учащегося. Отметим лишь некоторые из методов: словесные; наглядные; демонстрационные; алгоритмические.

Поскольку на современном этапе наиболее актуален деятельностный подход, то и в качестве метода обучения целесообразно выбрать демонстрационный метод. Обычно на каждом уроке применяются все вышеназванные методы, однако основной упор сделан на демонстрацию и самостоятельную работу учащихся. В итоге, учащиеся в полной мере, а главное за короткий срок, овладевают значительным объемом знаний, умений и навыков в работе.

Основным средством, позволяющим достичь высоких успехов в обучении, является комплекс дидактических материалов.

Применение средств ИКТ и ЭСО вносит определенную специфику в известные общедидактические методы обучения. Так, объяснительно-иллюстративные методы при использовании мультимедийного проектора заметно повышают познавательную активность учащихся за счет увеличения наглядности и эмоциональной насыщенности.

В свою очередь, практические методы делятся на: метод целесообразно подобранных задач; выполнение практикумов; выполнение лабораторных работ; программирование.

С точки зрения деятельностного подхода к процессу обучения, а также в целях закрепления или познания теоретического материала и формирования умений, необходимо использовать метод целесообразно подобранных задач. Учителю нужно построить такую систему упражнений, которая удовлетворяла бы таким условиям:

- выполнение каждого упражнения базируется на выполнении предыдущего и направлено на решение сформулированной проблемной ситуации;

- задачи должны быть разных уровней сложности, и ученик может выбрать – решать ли все задачи подряд или только некоторые из них, соответствующие его уровню подготовки;

- задачи должны нести практическое применение;

- формулировка задач интересна, опирается на учебный опыт детей.

Можно рассмотреть программирование как частно-дидактический метод обучения информатике, характеризующийся, в том числе тем, что умственная активность и самостоятельность учащихся может быть и репродуктивной, частично-поисковой, исследовательской.

Также часто используется такой частно-дидактический метод обучения, как метод демонстрационных примеров.

При объяснении нового материала и для закрепления теоретических знаний удобно использовать демонстрационные примеры. Они в состоянии частично заменить собой учителя и взять на себя управление познавательной деятельностью учащихся. Использование данного метода позволяет научить ребят чтению программ, увеличить их общение друг с другом и с преподавателем, а также заимствовать демонстрационные примеры для дальнейшего модифицирования.

На протяжении урока учащиеся успевают выслушать объяснение учителя по предлагаемой теме, посмотреть, как озвученный материал реализуется в конкретной изучаемой программе, попробовать самостоятельно поработать над изучаемой темой.

Такие приёмы ведут к лучшему усвоению нового материала, поскольку каждый учащийся вовлечён в работу.

Самостоятельная работа учащихся проходит как составная часть урока. Учитель объясняет и демонстрирует системы команд и последовательности их выполнения для достижения результата, а затем основное время уделяем самостоятельной работе с последующим комментарием и подведением итога.

Такой вид самостоятельной работы самый эффективный. Развивается память, способность воспроизводить, алгоритмически мыслить, но по определённым стандартам, по определенным логическим схемам.

Более фундаментальным видом самостоятельных занятий является специальные уроки, или часть уроков, полностью посвященных самостоятельному труду учеников: лабораторные работы; практические работы; работа с рабочими тетрадями или учебником и т. д.

Уроки, полностью посвященные самостоятельной работе, требуют тщательной подготовки. Именно на таких уроках учащиеся лучше усваивают новый материал, учатся работать с разными источниками, могут без помощи учителя в нужный момент сориентироваться и найти ответ на поставленный вопрос.

Этапы работы в обучающей программе:

1. Визуальная адаптация к обучающей программе

Сообщение названия обучающей программы. Формирование у школьников эмоционального отношения к программе. Выделение смысловых зон на экране компьютера, элементов управления, объектов программы. Постановка перед учащимися учебной проблемы.

2. Объяснение алгоритма работы

Организация внимания учащихся, процесса восприятия, осознания, осмысления, первичного обобщения и систематизации нового материала. Объяснение и демонстрация учителем системы команд и последовательности их выполнения для достижения результата. Указание на местонахождение программы и способа завершения работы с ней. Обучение учащихся обобщённым способам деятельности, а не только использованию конкретных клавиш управления программой. Уяснение вначале того, что необходимо сделать, а затем уже как это надо сделать.

3. Закрепление алгоритма работы

Закрепление знания учащимися способов и средств достижения результата. Учитель, задавая вопросы учащимся, добивается от них правильных и полных ответов, демонстрирует пошаговое выполнения задания и анализирует ошибочные варианты выполнения действий.

4. Разъяснение задания для самостоятельной работы

Формулировка и комментирование задания для самостоятельной работы учащихся, постановка перед ними проблемы. Мотивирование учащихся на быструю и качественную работу. Сообщение критериев, по которым будет оцениваться самостоятельная работа учащихся. Указание на дополнительные возможности работы с данной программой.

5. Работа учащихся на компьютерах

Закрепление знания учащимися способов и средств достижения результата. Выполнение серии заданий. Контроль и корректировка выполнения действий различными способами.

Для того, чтобы данная методика принесла свои результаты должна вестись постоянная тренировка с учениками, они должны быть приучены такой работе на уроке.

С использованием данного метода повышается уровень самостоятельной работы учащихся, реализуется принцип личностно-ориентированного подхода. Нужно признать тот факт, что большинство учеников не способны самостоятельно работать, они стремятся увильнуть, избежать самостоятельной работы. Чаще ученики желают получить всё в готовом виде без труда и собственных усилий. С использованием данной методики ученик обязательно будет работать самостоятельно. В связи с акцентом на личностную составляющую повышается качество усвоения знаний и приобретенных навыков. У учителя высвобождается время для персональной помощи слабым ученикам.

Эффективность использования данного метода подтверждается мониторингами результатов учебной деятельности. У многих учащихся появилась уверенность в собственных силах (так как работа в любом случае будет выполнена), интерес к процессу обучения (у многих он отсутствует из-за хронического непонимания материала). Взаимодействие учителя и ученика становится более продуктивным, что позволяет сократить учебное время, требуемое на отработку основных необходимых навыков, позволяет учащимся более глубоко и творчески отрабатывать умения и тем самым формировать алгоритмический стиль мышления. Ребята с большим желанием принимают участие в олимпиадном движении и конкурсе «Инфомышка». Реализованные возможности действуют на ученика развивающе, стимулируют интерес.

Развитие алгоритмического мышления является в целом комплексной проблемой. Система заданий, предлагаемая на самостоятельной работе на уроке информатики с целью развития алгоритмического мышления, является оптимальной формой работы с учащимися. Задания повышают развитие алгоритмического мышления учащихся, вооружают их навыками решения нестандартных задач, значительно расширяют и углубляют знания о

нестандартных задачах, формируют эмоциональную восприимчивость, формируют практические умения по решению жизненных задач.

1.3 Особенности методики обучения учащихся основам программирования

Образовательные учреждения несут ответственность за развитие способностей учащихся и повышение их интеллекта. Среди различных дисциплин информатика обладает наибольшими перспективами для развития алгоритмического мышления школьников — неотъемлемого аспекта интеллектуального роста.

Тщательный анализ эволюции образовательных стандартов в области информатики выявляет важнейшую цель: развитие алгоритмических способностей школьников на разных этапах изучения этой дисциплины. Это подчеркивает важность интеграции алгоритмического мышления в школьное образование.

Алгоритмическое мышление обладает отличительными характеристиками, отличающими его от других стилей мышления. Его общие атрибуты, такие как связность и эффективность, позволяют целостно взглянуть на проблемы и сформулировать предварительный образ решения. Между тем, специфические свойства включают дискретность, абстракцию и намеренное выражение с помощью языка. Эти специфические черты облегчают систематическое выполнение алгоритма, позволяя абстрагироваться от конкретных данных и перейти к обобщенному подходу к решению проблем, сформулированному с помощью формализованного языка.

Компоненты, неотъемлемые от алгоритмического мышления, включают способность формализовать задачу и разложить ее на отдельные, взаимосвязанные логические блоки.

«Алгоритмическое мышление определяется следующими компонентами:

- анализ требуемого результата и выбор на этой основе исходных данных для решения проблемы;
- выделение операций, необходимых для решения;
- выбор исполнителя, способного осуществлять эти операции;
- упорядочение операций и построение модели процесса решения;
- реализация процесса решения и соотнесение результатов с тем, что следовало получить;
- коррекция исходных данных или системы операций в случае несовпадения полученного результата с предполагаемым» [4].

В работах Л.Г. Лучко и И.Н. Слинкиной были определены три основных уровня развития алгоритмического мышления:

- операционный владеет некоторыми разрозненными операциями, но не может сочетать их, не владеет структурой их вложенности;
- системный знает некоторые способы сочетания операций конструкций создания этих сочетаний, умеет решать стандартные задачи на применение алгоритмического мышления;
- методологический умеет использовать уже имеющиеся мыслительные схемы решения некоторых алгоритмических проблем, может преобразовать их в изменяющихся условиях или трансформировать имеющиеся» [8].

Соответственно данным уровням были выделены умения, характеризующие каждый этап развития алгоритмического мышления:

- решать задачи алгоритмического характера;
- производить анализ задачи;
- составлять алгоритм;
- записывать алгоритм;
- производить синтаксический анализ составленного или предложенного алгоритма;
- выполнять алгоритмы;
- проводить оптимизацию алгоритма;
- производить мыслительные операции.

На основе этих уровней выделяются требования к развитию алгоритмического мышления:

Операционный уровень характеризуется тем, что ученик имеет представление об алгоритме.

Системный уровень характеризуется тем, что ученик имеет представления об алгоритме, его свойствах, составляет небольшие линейные алгоритмы или с простейшими ветвлениями и циклом; владеет конкретными операциями классификации; знает способы решения некоторого класса алгоритмических задач; имеет представление об исполнителе и системе команд исполнителя.

Методологический уровень характеризуется тем, что ученик имеет представления об алгоритме, знает его свойства, умеет составлять и записывать формальные и неформальные алгоритмы линейной структуры, с простейшими ветвлениями и циклами; владеет операциями классификации, взаимно однозначного соответствия; легко справляется с задачами алгоритмического характера; имеет представление об исполнителе, системе команд исполнителя [4].

1.4 Выводы по главе 1

Информационные технологии претерпевают стремительную эволюцию, требуя продвинутого программного обеспечения для решения разнообразных задач обработки информации в связи с растущим потенциалом компьютерных технологий. Программирование представляет собой глубоко интеллектуальное занятие в разработке программного обеспечения. План будущей программы формулируется с помощью алгоритма, который в конечном итоге переводится в машиночитаемый код. Инструментами для создания этих программ служат различные языки программирования.

Будущий программист должен преуспевать в разработке алгоритмов решения задач, владеть знаниями по крайней мере на одном языке программирования и обладать всесторонним пониманием смежных областей информатики, включая математику, физику и дизайн.

Алгоритмическое мышление составляет важную часть интеллектуальной деятельности человека с применением современных информационных технологий.

ГЛАВА 2

СТРУКТУРА ОБУЧЕНИЯ ИНФОРМАТИКЕ В СРЕДНЕЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ШКОЛЕ

2.1 Особенности изучения информатики в младших, средних и старших классах

Информатика на сегодняшний день является динамическим предметом, учитывающим современные тенденции развития и усовершенствования использования ИТ в учебном процессе и предполагает вариативность при составлении программ.

В младшей школе рекомендуется изучение так называемого «пропедевтического» курса информатики. При изучении пропедевтики курса учащиеся получают элементы знаний и умений работы с компьютерными программами, овладевают навыками конструирования, управления разными объектами (данный вид работы идет в игровой форме).

Практические навыки и умения необходимо закреплять и углублять во время работы на уроках математики, языка, рисования, музыки. Если нет такой возможности, учителю следует знакомить детей с этими программами на уроках информатики.

Во время проведения уроков в компьютерном классе, особенно в младшей школе, следует тщательно соблюдать санитарно-гигиенические нормы, проводить на уроке физкультминутки. Дети в этом возрасте могут на уроке работать с компьютерными программами 10-15 минут.

Дети от 7 до 10 лет быстро овладевают навыками и умениями работы с компьютером, у них развивается образное мышление, во время обучения создается комфортный эмоциональный фон, развивается их самостоятельность в познавательной работе, усиливается процесс индивидуализации обучения. Обучение в этом возрасте должно идти в игровой форме. Хотя использование игр во время обучения не вызывает значительного повышения эффективности овладения информационной культурой, но создает положительный эмоциональный фон, который необходимо сохранить и перенести для дальнейшего обучения.

В первом классе оправдывает следующая форма работы:

Сначала задание выполняет учитель, затем задание выполняется вместе всеми учениками (коллективная фаза) под наблюдением учителя, и только потом задача выполняется учениками самостоятельно. Во втором, третьем классах коллективная фаза выполнения задачи существует только тогда, когда программа или задачи сложны в информативном или учебном плане. Но

постепенно эту фазу следует исключать из работы. У детей уже есть определенные навыки работы с компьютером, есть некоторая база знаний, поэтому даже «средний» ученик может выполнить самостоятельно задание после первичного ознакомления с ним. В 4-5 классах можно постепенно переходить к индивидуальной работе учащихся по компьютерным программам. В связи с развитием психики ребенка, в этот период он начинает чувствовать себя личностью, ребенку в этом возрасте не безразлично мнение о нем других людей. Индивидуальная работа с компьютерными программами снимает этот дискомфорт. Диалог ведут двое: ученик и компьютер, поэтому ошибку, если ее допустил ученик, никто кроме компьютера не увидит и ошибку легко исправить. Ученик также имеет возможность работать в собственном темпе, связанном с уровнем имеющихся знаний.

Этапы работы с компьютером:

- Знакомство с компьютером начинается со знакомства с правилами поведения в компьютерном классе, с организацией рабочего места, гигиены труда.

- Знакомство на начальном уровне с принципами работы компьютера, ролью вычислительной техники в повседневной жизни и производстве.

- Дети переходят к знакомству с правилами работы с клавиатурой (используются разные клавиатурные тренажеры), мышью (используются простейшие программы, работу которых можно управлять мышью).

- Знакомство с графическим редактором (закрепление работы с мышью, работа с инструментами).

- Знакомство с простым детским музыкальным редактором.

- Работа с программными комплексами, поддерживающими изучение материала по другим школьным предметам.

- Параллельно идет знакомство с понятием объекта (имя объекта, его свойства, функции). Но это знакомство идёт на интуитивном уровне без сложных теоретических выкладок.

- Решение логических задач (без использования компьютерных программ).

К программным средствам, которые используются особенно в младшей школе, выдвигается ряд требований – это высокое качество и соответствие психолого-дидактическим принципам. Техника, используемая для детей (особенно мониторы) тоже должна соответствовать высокому техническому уровню.

Сегодня существуют учебники, рабочие тетради, диск.

Эти программы рассчитаны на активизацию учебно-познавательной деятельности учащихся, разработанные с учетом психологических особенностей детей. Каждый ребенок преследует конкретную цель, может

работать со скоростью, которая соответствует уровню его личного развития. При работе с программами дети не только получают начальные знания по информатике, но и развивают память, фантазию, логическое мышление и т.д.

Главная цель пропедевтического курса – преодоление психологического барьера между компьютером и ребенком и овладение более простыми навыками работы с компьютером.

Наряду с работой детей с прикладными программами, школьная практика имеет примеры использования интеллектуальной обучающей системы ЛОГО для развития алгоритмического мышления.

Швейцарским психологом Ж.Пиаже была выдвинута идея обучения через открытие. Особую популярность идеи Пиаже получили в компьютерном обучении благодаря работам Пейперта. Причиной создания С.Пейпертом своей теории явилось противоречие между старыми педагогическими средствами и новыми технологиями, а также отсутствие подлинного интереса к обучению и формированию опасений у детей перед новыми для них открытиями. С.Пейперт видит причину этого в принятом распределении людей на способных и не способных, на склонных к математическим наукам и предпочитающих гуманитарные знания. С.Пейперт придерживается более гуманной точки зрения, согласно которой все люди все равно успешно могут овладеть знаниями из любых отраслей наук. Дело не в умении, а в организации процесса обучения. Им предложена концепция школы будущего, в основе которой лежат естественная любознательность детей и способы удовлетворения этой любознательности. С.Пейперт выдвигает идею среды обучения. Она базируется не на подходах, при которых на компьютер пытаются перевести функции учителя или учебника, а основана на концепции микросред, представляющих собой некоторые модели реального мира, которые с той или иной степенью детализации творит сам ребенок. Это приводит к необходимости разработки компьютерных версий школьных курсов.

Главное предназначение предметных сред – возбуждать ребенка активно участвовать в процессе конструирования так называемых объектов, с помощью которых можно думать. Это совершенно особенные объекты, в которых возведены в единое целое признаки культурной среды, интегрированная возможность персональной идентификации в процессе обучения. Причем методическое обеспечение не удручает, а скорее стимулирует самостоятельность учащегося. При правильном применении оно становится частью среды, обеспечивая возможность достижения цели без помощи преподавателя. Простейшими аналогами этих сред могут выступать имитационно-моделирующие программы и программно-методические комплексы с объектно-ориентированным подходом.

Свои теоретические положения С.Пейперт воплотил в обучении ЛОГО, что является примером новых технологий открытого типа, использование которого изменяет методику обучения математике, физике и другим предметам. Особенностью среды ЛОГО является установка на то, что ребенок должен сам "программировать и обучать" компьютер в процессе природного "диалога с машиной", а по мере овладения средством переходить к непосредственному использованию научных идей, математических и лингвистических понятий, приобщаться к искусству интеллектуального моделирование.

Благодаря работе в среде ЛОГО дети приобретают уверенность в своих силах и могут выражать свои знания предмета в форме, отличной от традиционно принятой при изучении естественных наук.

Экспериментальная проверка идей С.Пейперта показала, что их реализация не всегда приводит к автоматическому развитию мышления. Основное внимание в таких обучающих системах обращается не на целенаправленный поиск эффективных способов решения задач, а на обучение путем проб и ошибок, что снижает дидактические возможности моделей данного типа.

Поэтому многие исследователи отошли от теории учения без обучения, и были вынуждены регламентировать учебный процесс. Понимается не возврат к традиционной схеме обучения, а поиск путей более полного использования компьютера как особого средства учебной деятельности.

В средних классах изучается базовый курс. В школьной практике существуют две ситуации – дети впервые знакомятся с компьютером или продолжают обучение. Цель базового курса – дать необходимые знания, умения и навыки будущему пользователю персонального компьютера. Учащиеся получают представление о возможностях компьютера, овладевают первичными навыками практического использования компьютера для обработки текстов и графических изображений, хранения и поиска информации, обработки больших числовых массивов информации.

В этот период уделяется внимание развитию алгоритмического мышления путем знакомства с понятием алгоритма, изучением его основных структур, овладением приемами построения алгоритмов.

Структура учебного курса:

- Общие сведения о работе с компьютером (знакомство с аппаратной частью информационных технологий, с работой операционных систем).
- Программные средства общего назначения (системы обработки графической информации, текстовой информации, табличные процессоры, системы управления базами данных).
- Основы алгоритмизации и программирования.

При тематическом планировании изучения базового курса информатики в средних классах следует помнить, что для лучшего усвоения некоторых тем у учащихся нет достаточных знаний по другим предметам (математика, физика), поэтому некоторые темы следует оставлять для изучения в старших классах.

Старшие классы. На этом этапе происходит формирование необходимых навыков использования компьютера как инструментального средства профессиональной деятельности, большое внимание уделяется работе с прикладными программами, особенно программами компьютерного моделирования.

Для желающих углубить свои знания по информатике – факультативы и кружки.

2.2 Факультативные курсы по информатике

Характер построения и содержания факультативных курсов может быть слишком разнообразен. В тоже время общеобразовательный курс должен содержать ядро:

1. Устройство и принципы функционирования компьютера;
2. Алгоритмы, средства их разработки и описания;
3. Знание основ одной из парадигм программирования.

Таким образом, движение по курсу информатики на факультативах и кружках должно идти как по горизонтали (включаются темы, которые не вошли в базовый курс), так и по вертикали (учащиеся углубляют свои знания при изучении программирования).

Цель факультативов – проведение профориентационной работы, то есть более подробное знакомство с профессиями, при работе которых инструментально используется компьютер, и углубление знаний по данному предмету.

Работа кружков по информатике может идти по разным направлениям в зависимости от контингента учащихся и профильной специализации школы. Можно организовать кружки, где учащиеся решают олимпиадные задачи по информатике, разрабатывают проекты и реализуют их на одном из современных языков программирования, а можно спланировать работу кружка по приобретению навыков использования современных информационных технологий для делопроизводства, изучать основы компьютерного дизайна, создания web-сред и т.д.

Внеурочная работа по информатике имеет целью повышение интереса к изучению данного предмета, проведение профориентационной работы.

Виды работы:

- экскурсии (знакомство с современной техникой и направлениями профессионального использования ИТ);
- диспуты (например, "Эрудит - информатик"; "Проблема выбора. Разветвление");
- викторины (например, "Путешествие по гипертексту"; "Информация вокруг нас");
- устные журналы (например, "Виртуальный мир"; "Страна алгоритмов");
- интеллектуальные игры (например, "Циклы в нашей жизни").

2.3 Реализация политехнизма

Политехнизм – система обучения, при которой учащиеся теоретически и практически знакомятся с основными отраслями производства.

Курс информатики призван внести значительный вклад в трудовую и политехническую подготовку учащихся. В этом отношении предполагается систематическое раскрытие взаимосвязи теоретических и прикладных аспектов курса, раскрытие роли и значения новых информационных технологий и работы программистов в современном общественном производстве.

В процессе изучения информатики последовательно и систематически формируется культура умственного труда на качественно новом уровне и такие важнейшие общетрудовые умения как умение планировать свою работу, рационально ее выполнять, критически соотносить начальный план работы с реальным процессом ее выполнения.

Необходимо специально подчеркивать роль задач и отводить значительное время для их рассмотрения практических в курсе информатики. В рамках данного предмета рассматриваются темы алгоритмизации, программирования, решения задач с помощью компьютеров. Решение задач пронизывает все содержание учебного предмета и является основным средством формирования компьютерной грамотности и информационной культуры учащихся.

Благодаря значительному количеству задач преподавания курса информатика имеет ярко выраженную практическую направленность: большинство задач, предлагаемых в школьных учебниках, имеют широкое применение на практике. Усвоение учащимися средств решения этих задач значительно распространяет математический инструментальный школьника, способствуя усилению связи обучения с жизнью.

Не следует воспринимать курс информатики как некоторый прикладной раздел курсов математики и физики. Математические и физические задачи

являются лишь материалом, средством для формирования у учащихся основных положений курса информатики – методов и средств обработки информации. Следует правильно понимать роль прикладных задач и не сводить дело только к разбору средств решения этих задач, потому что это не главное в предмете информатики. Главное – показать современный подход к применению ИТ для автоматизации процесса решения задач и получения решения более оптимальным и рациональным способом.

Практическая часть курса информатики впервые явно и систематически формирует у учащихся полное представление о современных подходах к решению задач и исследованию различных явлений и процессов.

При раскрытии сущности компьютерного подхода к решению задачи необходимо четко показать 2 аспекта самого понятия «решения задачи»:

1) важно показать учащемуся как решать задачу (этапы решения, технологию, алгоритм решения);

2) не менее важно учить тому, как найти решение задачи, то есть как найти способ, метод, идею решения.

Важным моментом в формировании у учащихся понимания особенностей компьютерного решения задач является разбор этапов решения задачи с помощью компьютера от ее постановки до анализа полученных результатов.

2.4 Межпредметные и внутрипредметные связи при преподавании информатики

Одним из важных направлений повышения практической значимости результатов обучения является организация межпредметных связей между школьными предметами на основе общих знаний и методов научного познания. Проблема межпредметных связей вытекает из дидактического принципа систематичности, который отражает общефилософское понятие о связи явлений и согласуется с физиологическими и психологическими понятиями о работе мозга. Последовательное осуществление межпредметных связей в обучении естественным наукам способствует приобретению обобщенных знаний, умений и навыков.

Стремление сделать преподавание математики, физики и других предметов более эффективным, а результаты обучения более практически значимыми приводят к необходимости использования информационных технологий. Предметное использование ИТ формирует у учащихся навыки сознательного пользователя компьютеров, показывает возможности использования ИТ в дальнейшей практической и научной деятельности.

1. Исследование физических основ строения частей компьютера может естественно войти в содержание курса физики. Важно отметить, что это не требует прибавления к курсу физики принципиально новых вопросов. Знакомство с физическими основами работы полупроводниковых устройств предусмотрено имеющейся программой по школьному курсу физики, а с возникновением микропроцессорной техники поменялась только разработка производства транзисторов, физические принципы их работы остались по существу прежними.

2. Межпредметные связи математики с другими школьными дисциплинами могут осуществляться по двум направлениям:

– различные естественные науки – источник задач для математики с практическим содержанием;

– математические теории – инструмент исследований в естественных науках.

Связь математики с другими науками осуществляется в основном через математическое моделирование явлений и процессов и статистический анализ экспериментальных данных. Использование средств новых информационных технологий позволяет облегчить этап решения задачи внутри модели за счет выполнения громоздких вычислений. На основе графических возможностей компьютер наглядно представляет функциональные или статистические зависимости между рассматриваемыми величинами, что помогает осознать связи между соответствующими величинами.

Этапы построения, характеристики математической модели органически учить в курсе математики. Часто учащиеся при описании каких-либо процессов определяют функции и делают это на интуитивном уровне. Учителям математики следует четко выделить и описать свойства и характеристики математических моделей и показать, как эти знания можно использовать при решении математических задач.

Такая тема, как этапы решения задач с помощью компьютера, объединяет в себе знания в области математики и информатики. Начальные этапы этого процесса (анализ условия задачи, выделение и описание входных и исходных данных, построение модели, построение метода) можно учить в курсе математики.

3. ИТ должны использоваться учащимися для вычислительной работы не только в курсе информатики, но и в курсах математики, химии, физики; для анализа данных обучающего эксперимента и поиска закономерностей при проведении лабораторных работ; для исследования функций; при построении и анализе математических и информационных моделей физических, химических и других процессов и явлений В курсе биологии, географии, истории и ряда других гуманитарных предметов ИТ должны использоваться в качестве

предметного средства, как информационная система, банк данных, автоматизированный справочник.

ГЛАВА 3

МЕТОДИКА ОБУЧЕНИЯ ОСНОВАМ АЛГОРИТМИЗАЦИИ

3.1 Основные требования к учащимся

3.2 Понятие алгоритма

3.3 Методические рекомендации по решению задач

ГЛАВА 4

МЕТОДИКА ИЗУЧЕНИЯ ЯЗЫКА ПРОГРАММИРОВАНИЯ

4.1 Основная цель изучения раздела. Общие вопросы методики

4.2 Понятие языка программирования, данные, величины, команды

4.3 Изучение языков программирования

4.5 Методические рекомендации по решению задач по курсу алгоритмизации и программирования

ЗАКЛЮЧЕНИЕ

Обучение информатике в школе способствует развитию личности учащихся, подготовке их к активной жизни и труду в информационном обществе.

Реализация указанной составляющей цели обучения предполагает формирование у учащихся научного мировоззрения, информационной культуры, алгоритмического и критического стилей мышления, информатических и ключевых компетентностей, развития творческих способностей, умений и навыков работать с современными средствами информационных и коммуникационных технологий.

Содержание учебного предмета направлено на овладение учащимися научных основ информатики, соответствующих фундаментальных понятий, принципов построения и функционирования средств информационных и коммуникационных технологий, овладение ими.

Задачи обучения предмету информатика в школе:

- сформировать у учащихся базовые знания об информации и информационных процессах, значении информации и знаний на современном этапе развития информационного общества;
- предоставить учащимся основные исторические сведения о развитии информатики и средств вычислительной техники, вклад зарубежных и отечественных ученых в них развитие;
- сформировать у учащихся умение использовать программные и информационно-коммуникационные средства в учебной и повседневной деятельности;
- обеспечить овладение учащимися основами информационной культуры;
- сформировать умение творчески выполнять учебные задания, разрабатывать рациональные алгоритмы выполнения и анализ их выполнения;
- сформировать у учащихся начальные навыки программирования, умения налаживать программы и анализировать полученные результаты.

Выполнение указанных задач обеспечивает формирование у учащихся научного мировоззрения, информационной культуры, алгоритмического и критического стилей мышления, развития творческих способностей, умений и навыков работать с современными средствами информационных и коммуникационных технологий.

Приобретенные учащимися в процессе обучения информатике знания, умения и навыки обеспечат основу для формирования у учащихся предметной информационной компетентности и ключевых компетентностей, предусмотренных Государственным стандартом базового и полного общего среднего образования.

Приобретенные учащимися в результате обучения компетентности позволяют им:

- объяснять свойства информации и закономерности информационных процессов;
- овладеть основными методами научного познания;
- понимать научные основы обработки сведений, применять основные понятия, связанные с алгоритмизацией обработки данных, управление объектами и процессами;
- быть готовым к активной жизнедеятельности в условиях информационного общества, стать в будущем не только полноценным членом такого общества, но и его создателем;
- использовать приобретенные знания в дальнейшей учебной и практической деятельности в условиях информационного общества.

В школе необходимо изучать программирование по следующим причинам:

1. В целях фундаментализации курса информатики. Одним из основных принципов информатики является принцип программного управления работой компьютера. Понять его невозможно, не зная, что такое программа для ЭВМ. Таким образом, знакомство с программированием есть элемент общеобразовательного содержания информатики.

2. С точки зрения профориентационной функции предмета. Любой школьный предмет должен наряду с образовательной, воспитательной и развивающей функциями выполнять еще и профориентационную. Программирование является профессиональной сферой деятельности, очень важной, современной и престижной.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЕ 1

ПРИЛОЖЕНИЕ 2