

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ
СЕВЕРО-ВОСТОЧНОЕ ОКРУЖНОЕ УПРАВЛЕНИЕ
ОБРАЗОВАНИЯ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ЦЕНТР ОБРАЗОВАНИЯ № 1430

Научно-практическая разработка

*Изучение темы «Алгоритмы и алгоритмические
структуры в профильных классах»*

Автор: Алябьева Н.М. – учитель информатики и ИКТ

Москва 2012

ВВЕДЕНИЕ.....	3
Общая характеристика работы.....	3
Глава 1. Теоретические основы профильного обучения информатике.....	7
1.1. Дидактические аспекты профильного обучения информатике	7
1.2. Психолого-педагогические аспекты изучения информационных технологий	11
Глава 2. Разработка методического материала.....	14
2.1. Методические рекомендации по обучению	14
2.1.1 Средство обучения	14
2.1.2 Visual Basic for Application	18
2.1.3 Задачи для самостоятельного решения	25
2.2 Содержание обучения.....	26
2.2.1 Содержательные аспекты темы «Алгоритмы и алгоритмические структуры».....	29
2.2.2 Свойства алгоритмов	30
2.2.3 Виды алгоритмов и их реализация	31
2.2.4 Методы изображения алгоритмов	33
2.2.5 Словесное описание алгоритма	33
2.2.6 Блок-схема алгоритма	33
2.2.7 Псевдокод	35
2.2.8 Программное представление алгоритма	36
2.2.9 Порядок разработки иерархической схемы алгоритма	36
2.3 Организационные аспекты занятий в компьютерном классе.....	39
2.4 Разработка уроков.....	42
Выводы по главе 2.....	48
Глава 3. Организация и результаты эксперимента.....	50
3.1 Цель и гипотеза эксперимента.....	50
3.2 Доступность теста и сравнение знаний групп.....	51
3.3 Критерии оценки итоговых программ на VBA.....	57
3.4 Отношение к ИКТ.....	59
Выводы по главе 3.....	64
Заключение.....	65
Список источников.....	66
Литература.....	66
Электронные ссылки.....	67
ПРИЛОЖЕНИЯ.....	68
Приложение 1. Термины и сокращения.....	69

ВВЕДЕНИЕ

Общая характеристика работы

Актуальность

Данная работа является актуальной в свете перехода на Федеральные государственные образовательные стандарты (ФГОС) третьего поколения. Введение ФГОС привело к существенному изменению всей системы образования, и ключевым моментом преобразований стал реальный, а не декларируемый переход от знаниевого подхода к компетентностному.

В этом отношении переход к профильному обучению представляется авторам исследования наиболее перспективным. Реализация процессов профилизации старшей школы требует построения интеграционной системы взаимодействия общего и профессионального образования для обеспечения профильной и предпрофессиональной подготовки обучающихся 10-11 классов в рамках модернизации Российского образования.

Таким образом, процесс интеграции общего и профессионального образования является средством формирования компетенций у обучающихся профильных классов, учреждений начального и среднего профессионального образования как способа достижения нового качества образования, что позволит разрешить следующие **противоречия:**

- между потребностью в реализации компетентностного подхода и отсутствием апробированной интегративной системы реализации данного процесса в системе профильных классов общеобразовательных школ, ОУ НПО и СПО;

- между потребностью обучающихся в получении качественного общего образования и сокращением часов на изучение общеобразовательных предметов в учебном плане учреждений начального и среднего профессионального образования по сравнению с общеобразовательной школой;

- между потребностью в реализации процессов профилизации старшей школы и отсутствием системы профильной и предпрофильной подготовки старшеклассников, обеспечивающей интеграцию общего и профессионального образования.

Информатика является одним из тех предметов, в которых дифференциация обучения реализуется наиболее естественным образом. Этому способствует сам характер информатики как науки и совокупности множества информационных технологий, история ее появления в школе в те годы, когда многообразию в школьном образовании способствовали внешние условия. Заметим, что даже базовый курс информатики является в некотором смысле дифференцированным, так как по-разному излагается в

различных учебниках. Однако истинная дифференциация курса информатики связана не с методическими различиями в изложении одного и того же материала, как в базовом курсе, а с реальными различиями в содержании дифференцированных курсов. Подобное возможно лишь на старшей ступени школы, после изучения базового курса информатики. Тема алгоритмы и алгоритмические структуры не дается в полном объеме в базовом курсе. Но для поступления в высшие учебные заведения знания предмета данной темы является необходимым. Поэтому введение в профильном курсе информатики темы «Алгоритмы и алгоритмические структуры» не только поможет выпускнику подготовиться к единому государственному экзамену при поступлении в высшие учебные заведения, а также разовьет алгоритмическое мышление.

Объектом исследования

является процесс изучения информатики в профильном курсе.

Предмет

Методика изучения темы «Алгоритмы и алгоритмические структуры»

Цель

Разработка темы «Алгоритмы и алгоритмические структуры» профильного общеобразовательного курса «Основы алгоритмизации и программирование» для повышения качества обучения учащихся в профильных классах.

Гипотеза исследования:

Оптимальное решение может быть обеспечено в условиях: 1) отбора методов повышения эффективности аспекта посредством ИКТ и 2) отбора содержания темы «Алгоритмы и алгоритмические структуры» курса «Основы алгоритмизации и программирование» на основе анализа развития мышления в процессе информационной деятельности по обработке процесса изучения информатики в профильном курсе, решения алгоритмических задач; 3) разработки электронной поддержки и 4) соответствующей методики обучения, 5) обучения самостоятельному освоению новых ИКТ.

Задачи исследования:

1) на основе анализа состояния и перспектив применения методов информатики и средств ИКТ в дидактике определить основные виды информационной деятельности и основные компоненты содержания обучения и развития алгоритмического мышления в профильных классах;

- 2) выявить принципы отбора содержания и методики обучения учащихся профильных классов в области «Теоретической информатики» в условиях применения ИКТ в образовательном процессе;
- 3) сформулировать уровневые характеристики выпускников и требования к знаниям, умениям и навыкам пользования средствами ИКТ;
- 4) разработать блочно-модульную программу теоретического курса «Основы алгоритмизации и программирование» на основе тезауруса и сетевого моделирования;
- 5) адаптировать программно-методическую поддержку для преподавания основных разделов темы «Алгоритмы и алгоритмические структуры» профильного профессионально ориентированного курса «Основы алгоритмизации и программирование»;
- 6) разработать методику обучения указанной теме;
- 7) экспериментально апробировать содержание и методику обучения и подтвердить гипотезу.

Методологической основой исследования являются фундаментальные работы по философии образования и методологии психолого-педагогической науки Бабанского Ю.К., Давыдова В.В., Краевского В.В., Лернера И.Я., Скаткина М.Н., Талызиной Н.Ф. и др.; по теории профессионального обучения и обучения в высшей школе - работы Сластенина В.А.; Абдуллиной О.А., Архангельского С.И., Попова В.В., Филатова О.К. и др.; по теории деятельностного подхода - работы Вербицкого А.А., Гальперина П.Я. и др.; по теории опережающего обучения Пасхина Е.С., Урсула А.Д. и др.; по методологии и практике информатизации образования - работы Ваграменко Я.А., Каракозова С.Д., Кузнецова А.А., Лапчика М.П., Могилева А.В., Пака Н.И., Роберт И.В., Уварова А.Ю., Шолоховича В.Ф. и др.; исследования в области содержания : профессиональной (педагогической) информатики Брановского Ю.С., Жданова С.А., Колина К.К., Кузнецова Э.И., Лавиной Т.А., Лапчика М.П., Лучко О.Н., Соколовой И.В. Швецкого М.В.; общеобразовательной информатики Бешенкова С.А., Гейна А.Г., Григорьева С.Г., Добудько Т.В., Ершова А.П., Захаровой Т.Б., Кувалдиной Т.А., Кузнецова А.А., Кушниренко А.Г., Лесневского А.С., Пугача В.И., Румянцева И.А., Смольниковой И.А., Хеннера Е.К.; пропедевтики информатики Горячева А.В., Дубинину В.В., Зарецкого А.В., Паутову Л.И., Первина Ю.А., Семёнова А.Л. и др.

Методы исследования: изучение и анализ философской, педагогической, психологической, технической и математической литературы по проблематике исследования и по информатизации образования; анализ отечественного и зарубежного

опыта использования средств ИКТ в учебной деятельности; методы отбора содержания учебной дисциплины и составления сценария темы учебной дисциплины; анализ средств и методов преподавания информатики на всех уровнях и в различных типах учебных заведений в России и за рубежом; беседы, тестирование, анкетирование и телемониторинг учащихся профильных классов; поисковый, констатирующий и формирующий педагогический эксперимент и дальнейшая обработка результатов эксперимента, методы доказательств.

Научная новизна и теоретическая значимость исследования определяются выявлением основных видов информационной деятельности педагогов; выделением и систематизацией принципов, методических требований обучения в условиях применения средств ИКТ; формулировкой уровневых характеристик профессионального обучения по ИКТ учащихся профильных классов; отбором понятий для построения инварианта курса «Основы алгоритмизации и программирование» с применением тезаурусного метода и сетевого моделирования.

Практическая значимость исследования заключается в том, что на основе полученных результатов предложены модифицируемые в дальнейшем ПО УН для педагогов в любой специализации; определена блочно-модульная структура и содержание программы теоретического курса «Основы алгоритмизации и программирование». Разработанная методика курса «Применение ИКТ в деятельности учащихся профильных классов» и три ПМК могут быть использованы для проведения лекционных и практических занятий по вопросам применения ИКТ в учебной деятельности.

Обоснованность и достоверность научных положений, выводов и рекомендаций обеспечивалась опорой на достижения психолого-педагогической науки, согласованностью полученных выводов с основными положениями концепции новой науки «Теоретическая информатика» и современной концепции информатизации образования, а также опытом и перспективами использования средств ИКТ в учебной деятельности, комплексной методикой исследования, соответствием использованных методов задачам исследования, апробацией полученных результатов в учебном процессе, использованием математических методов обработки и статистической значимостью результатов авторского педагогического эксперимента, логичностью выводов.

Апробация результатов исследования.

Осуществлялась через публикацию статей в сети Internet; в процессе обсуждения отдельных вопросов на заседаниях в рамках выступления на окружных

семинарах (СВАО, Москва, 2010), отдельные аспекты научно-практической разработки результаты и выводы доложены на заседаниях кафедры, методических советах в школе (ГБОУ ЦО №1430, Москва, 2011); в процессе разработки и реализации проектных работ учащихся, опубликованных в сборнике «Доклады пленарных заседаний. Всероссийская научная конференция молодых исследователей «Шаг в будущее»».

По разработанным программно-методическим комплексам для преподавания основных разделов темы «Алгоритмы и алгоритмические структуры» профильного профессионально ориентированного курса «Основы алгоритмизации и программирование» проводились практические занятия в 2009-2010 уч. годах - в 10 «а» классе 2011 года выпуска, в 2010-2011 уч. годах – в 10 «а» классе 2012 года выпуска ГБОУ ЦО №1430. Эффективность методического комплекса подтверждена не только автором разработки, но и коллегами, работающими в параллели.

Глава 1. Теоретические основы профильного обучения информатике

1.1. Дидактические аспекты профильного обучения информатике

Дифференциация обучения — важная задача современной школы. Она позволяет реализовать многообразие образовательных траекторий, способствует индивидуализации обучения, развитию познавательной активности учащихся, выбору профессионального пути, помогает определиться с продолжением образования в вузе. Информатика является одним из тех предметов, в которых дифференциация обучения реализуется наиболее естественным образом. Этому способствует сам характер информатики как науки и совокупности множества информационных технологий, история ее появления в школе в те годы, когда многообразие в школьном образовании способствовали внешние условия. Заметим, что даже базовый курс информатики является в некотором смысле дифференцированным, так как по-разному излагается в различных учебниках. Однако истинная дифференциация курса информатики связана не с методическими различиями в изложении одного и того же материала, как в базовом курсе, а с реальными различиями в содержании дифференцированных курсов. Подобное возможно лишь на старшей ступени школы, после изучения базового курса информатики.

Профильное обучение — средство дифференциации и индивидуализации обучения, позволяющего за счет изменений в структуре, содержании и организации образовательного процесса более полно учитывать интересы, склонности и способности

учащихся, создавать условия для обучения старшеклассников *в соответствии с их профессиональными интересами и намерениями в отношении продолжения образования.*

Профильное обучение направлено на реализацию личностно-ориентированного учебного процесса. При этом существенно расширяются возможности *выстраивания учеником индивидуальной образовательной траектории.*

Переход к профильному обучению преследует следующие *основные цели:*

- обеспечить *углубленное изучение отдельных предметов* программы полного общего образования;
- создать условия для существенной дифференциации содержания обучения старшеклассников с широкими и гибкими возможностями *построения школьниками индивидуальных образовательных программ;*
- способствовать установлению равного доступа к полноценному образованию разным категориям обучающихся *в соответствии с их способностями, индивидуальными склонностями и потребностями;*
- расширить возможности социализации учащихся, обеспечить преемственность между общим и профессиональным образованием. *Более эффективно подготовить выпускников школы к освоению программ высшего профессионального образования.*

Профилизация обучения в старших классах соответствует структуре образовательных и жизненных установок большинства старшеклассников. К 15-16 годам у большинства учащихся складывается *ориентация на сферу будущей профессиональной деятельности.* Социологические исследования показывают, что большинство старшеклассников (более 70%) отдадут предпочтение тому, чтобы «знать основы главных предметов, а углубленно изучать только те, которые выбираются, **чтобы в них специализироваться**».

Реализация профильности старшей ступени ставит выпускника основной ступени (9 класс) перед **необходимостью совершения ответственного выбора – предварительного самоопределения в отношении профилирующего направления собственной деятельности.**

Модель общеобразовательного учреждения с профильным обучением на старшей ступени предусматривает возможность разнообразных комбинаций учебных предметов,

что и будет обеспечивать гибкую систему профильного обучения. Эта система включает в себя следующие *типы учебных предметов*:

Базовые общеобразовательные предметы – являются **обязательными** для всех **учащихся**. В набор обязательных общеобразовательных предметов входят: математика, история, русский и иностранный языки, физическая культура, а также интегрированные курсы обществоведения и естествознания.

Профильные общеобразовательные предметы – предметы **повышенного уровня**, **определяющие направленность каждого конкретного профиля обучения**. Эти предметы являются **обязательными** для учащихся, выбравших данный профиль обучения.

Согласно рекомендациям Департамента образования **не менее 5% материала по темам профильного предмета изучается в рамках проектно-исследовательской деятельности**. Представление выполненного проекта осуществляется в рамках выпускного экзамена в 11 классе, также представляются на школьной научно-практической конференции.

Элективные курсы – **обязательные для посещения курсы по выбору учащихся, входящие в состав профиля** обучения на старшей ступени школы; одни из курсов могут **поддерживать** изучение основных профильных предметов на заданном профильным стандартом уровне, другие - служат для **внутрипрофильной специализации** обучения для построения индивидуальных образовательных траекторий; по элективным курсам итоговая аттестация не проводится.

В нашей школе профильное обучение реализуется на основе **индивидуальных учебных планов** учащихся. При такой организации каждый ученик может выбрать профильные предметы из различных образовательных областей. Остальные предметы изучаются на базовом уровне в соответствии со стандартами образования.

При составлении индивидуального учебного плана необходимо выбрать базовые общеобразовательные предметы, профильные предметы и элективные курсы, которые в совокупности и составят индивидуальную образовательную траекторию учащегося.

По закону образовательную программу разрабатывает сама школа, и эта программа **может превышать** требования Госстандарта. Для реализации такой программы **могут**

потребоваться дополнительные пособия. Выбор учебной программы и учебника для работы предоставляется непосредственно учителю.

Организация поступления в старшую профильную школу

В существующей практике число желающих продолжить образование в старших классах определенного общеобразовательного учреждения больше, чем реальные возможности приема в эти классы. Возникает ***ситуация конкурсного приема***, которая становится особенно актуальной в условиях перехода на профильное обучение.

Конкурсный набор в старшие классы отдельных общеобразовательных учреждений ***не входит в противоречие*** с законодательно закрепленным правом получения каждым ребенком общего (полного) среднего образования (ст.16, п.1, абз. 2 Закона Российской Федерации «Об образовании»). Закон гарантирует гражданам право получения образования этого уровня, что, однако, не есть синоним права получения его в конкретном образовательном учреждении.

Важную роль в организации конкурсного набора играет ***введение накопительной оценки (портфолио – «портфель достижений»)***, которая учитывает различные достижения учащихся по исполнению тех или иных ***проектов, написанию рефератов, творческих работ, участие в конференциях, конкурсах, олимпиадах различного уровня, реальные результаты на курсах по выбору и т.п.***

Информатика и информационные технологии

Изучение информатики в профильной школе (10-11 классы) реализуется на двух уровнях:

1. базовый (общеобразовательный) уровень — ***уровень А***, из расчёта 34 учебных часов (1 час в неделю) в старшей школе
2. профильный уровень — ***уровень Б***, 136 учебных часов (4 часа в неделю) на профильном уровне.

Изучение информатики и информационных технологий в старшей школе на профильном уровне направлено на достижение следующих целей:

- освоение и систематизация знаний, относящихся к математическим объектам информатики; построению и описанию объектов и процессов, позволяющих осуществлять их компьютерное моделирование; средствам моделирования;

- развития алгоритмического мышления, способностей к формализации овладение умениями строить математические объекты информатики, в том числе логические формулы и программы на формальном языке; использовать инструментарий и настраивать его;
- приобретение опыта проектной деятельности, создания, редактирования, оформления, сохранения, передачи информационных объектов различных типов с помощью современных программных средств; построение компьютерных моделей, коллективной реализации информационных проектов, формальной деятельности в различных сферах, востребованных на рынке труда.

1.2. Психолого-педагогические аспекты изучения информационных технологий

Проанализировав работы специалистов, которые занимались изучением мышления С. Л. Рубинштейна [12], П. Я. Гальперина [8], Н. Ф. Талызиной [8], М. Б. Воловича [4], Н. Н. Поспелова [11], В. В. Давыдова [5], Ж. Пиаже [10], рассмотрим теорию поэтапного формирования мыслительных действий.

Она была разработана в 50-е годы нашего столетия. Ее создателем является советский психолог и педагог П. Я. Гальперин.

По П. Я. Гальперину, любое новое умственное действие, например воображение, понимание, мышление, наступает после соответствующей внешней деятельности. Этот процесс проходит несколько этапов, обуславливающих переход от внешней деятельности к психологической. Эффективным, т.е. действительно ускоряющим развитие учащегося, является только такое обучение, при котором учитываются эти этапы. Такое обучение основывается не на выявлении и развитии врожденных способностей личности, а на овладении ею такими видами и способами познавательной деятельности, которые являются приобретением общества и которые позволяют личности стать членом этого общества.

П. Я. Гальперин широко понимает обучение. Он пишет, что обучением условно можно назвать любую деятельность, поскольку тот, кто ее выполняет, получает новую информацию и умения, и одновременно получаемая им информация получает новое качество.

В деятельности, выполняемой учащимися, П. Я. Гальперин выделяет три стороны: ориентировочную, исполнительную и контрольную. Ориентировочная сторона основана на использовании учащимися объективных условий, необходимых для выполнения определенной деятельности. Исполнительная сторона сводится к последовательности

этапов преобразования объекта деятельности, а контрольная — требует от учащегося наблюдения за ходом деятельности и сопоставления ее результатов с соответствующими образцами, а при обнаружении расхождений — соответствующей корректировки ориентировочной и исполнительной составляющих этой деятельности. Вся деятельность не является самоцелью, а вызвана неким мотивом этой деятельности, в состав которой он входит. Когда цель задания совпадает с мотивом, действие становится деятельностью. Иначе говоря, деятельность — это процесс решения задач, вызванный желанием достичь цели, что может быть обеспечено с помощью этого процесса. Роль мотивации П. Я. Гальперин оценивает так высоко, что наряду с пятью основными этапами в процессе овладения новыми действиями в последних своих работах он рекомендует учитывать еще один этап — формирование соответствующей мотивации у учащихся.

Концепцию поэтапного формирования умственных действий наряду с ее создателем развивают его ученики, среди которых важную роль играет

Н. Ф. Талызина, автор многочисленных исследований и работ по этой теме. Познакомимся с характеристикой вышеуказанных этапов в ее понимании.

Этап создания схемы ориентационной основы деятельности. На этом этапе учащиеся получают данные о цели деятельности и ее предмете, что обеспечивает им предварительное ознакомление с деятельностью и условиями ее выполнения. Таким образом, учащиеся узнают, что является предметом учения и в какой последовательности они должны выполнять ориентационные, исполнительные и контрольные действия. Это еще не сама деятельность, а лишь система указаний на то, как учащийся должен осуществлять эту деятельность.

Этап формирования материальной деятельности. Здесь уже сами учащиеся выполняют действия (например, измеряют размеры школьного класса), но только во внешней форме — материальной или материализованной. Материальная форма имеет место тогда, когда учащиеся сталкиваются с самими предметами, например с объектами природы, а материализованная — когда они выполняют действия на геометрических или физических моделях. Эти действия охватывают как ориентационную, так и исполнительную и контрольную деятельность, причем внимательное отношение ко всем трем сторонам деятельности имеет на данном этапе особое значение. Одновременно происходит название всех этих действий.

Этап внешней речи. Здесь действие подвергается дальнейшему обобщению благодаря его полной вербализации в устной или письменной речи. Таким образом, действие усваивается в форме, оторванной от конкретики, т.е. обобщенной.

Этап внутренней речи. Так же как и на предыдущем этапе, действие проявляется в обобщенном виде, однако его вербальное освоение происходит без участия внешней речи. После получения мысленной формы действие начинает быстро редуцироваться, приобретая форму, идентичную образцу, и подвергаясь автоматизации.

Этап интериоризации действия. Действие становится здесь внутренним процессом, максимально автоматизированным, становится актом мысли, ход которого закрыт, а известен только конечный «продукт» этого процесса. Таким образом, поэтапно учащиеся приходят к овладению мысленными действиями, совсем не похожими на действия внешние, но являющимися их производными.

В теории П. Я. Гальперина использованы достижения кибернетики. Ее главный принцип состоит в управлении процессом обучения, которое он понимает шире, чем другие, считая его главной формой деятельности учащейся молодежи. Управление здесь доведено до своего совершенства, что и позволяет преобразовывать внутренние действия во внешние и тем самым ускорять развитие учащихся. Такие их способности, как мышление, память или внимание, остаются в тени ориентационных, исполнительных и конкретных действий, а вся эмоциональная жизнь учащегося сводится к заботе о формировании мотивации к обучению. При всем этом данная система является весьма молодой, к тому же находящейся в стадии продолжающегося развития, на основе немногочисленного опыта ее трудно оценить, что было возможно в случае рассмотрения других, уже долгое время применяющихся на практике систем. Апробируем эту теорию в ходе разработанных уроков [10].

Глава 2. Разработка методического материала

2.1. Методические рекомендации по обучению

Переходя от теории к практике, применим теорию П.Я. Гальперина и Н.Ф. Талызиной конкретно к разработанным урокам информатики. На предварительном этапе надо сформировать мотивацию у учеников. Это будет происходить на первом или втором уроке. Надо будет объяснить им, как знания, которые им предлагаются, необходимы им в дальнейшем. Скорее всего, нужно будет ориентировать на удачное поступление в ВУЗ.

На первом этапе необходимо заинтересовать учащихся, поэтому следует рассказать об алгоритмах их свойствах и типах, привести примеры задач, решаемых с помощью алгоритмов определенного типа. Второй этап реализуется при помощи наглядности. Это презентации, в которых приведены: определения различных типов алгоритмов, блок-схемы, примеры задач, способы их решения (при этом представлены как блок-схемы алгоритмов, так и фрагменты программного кода, написанного на языке Visual Basic). Следующий этап – внешняя речь. Для лучшего запоминания материала, учащиеся должны проговаривать его. С целью реализации этого процесса необходимо проведение уроков-бесед и семинарских занятий. На четвертом этапе – нужно чтобы изученный материал еще раз проговорили про себя. Данный процесс достаточно сложно проконтролировать. Поэтому следует повысить мотивацию к изучению данного материала с помощью решения занимательных задач. И на последнем этапе учащимся предстоит применить знания в практических работах (при создании проектов в среде Visual Basic), тестах, самостоятельных и контрольных работах, а так же во время подготовки к семинарскому занятию. Для наибольшей эффективности будут предложены задачи для самостоятельного решения.

2.1.1 Средство обучения

Средством обучения будет являться специально разработанный блок презентаций на тему «Алгоритмы».

Первая презентация «Алгоритмы» содержит формализованные задачи, решение которых возможно с помощью применения алгоритмов.

Прежде чем дать определение алгоритма, необходимо сказать ученикам о месте алгоритмов в жизни людей. Привести примеры алгоритмов, не называя их алгоритмами, например, что все люди с детства привыкли следовать тем или иным правилам, выполнять разнообразные инструкции и указания. Затем необходимо привести такие примеры, на которых ученики смогут понять, что же такое алгоритм, а затем дать

непосредственно определение алгоритма. Важно заметить, что в любом классе, независимо от возраста и специализации, желательно в начале изучения тем все примеры приводить из жизни. Это связано как с отношением многих учеников к информатике, так и с восприятием информации вообще. После того, как было дано определение алгоритма, следует закрепить его на конкретных примерах, желательно, также взятых из жизни. Важно в самом начале не усложнять понимание алгоритмизации строгими правилами записи алгоритма. Достаточно будет писать алгоритмы в привычном для учеников виде. Например, «Покупка продуктов» и «Приготовление еды». Решение поставленных задач возможно с помощью построения блок-схем [приложение2].

Аналогично, в виде последовательности действий можно описать процессы решения многих задач, с которыми учащиеся имеют дело в школе.

Например:

1. Вычисление периметра многоугольника;
2. Проведение фонетического разбора слова;
3. Нахождение значения выражения;
4. Составление плана сочинения;

При решении подобных заданий необходимо четко придерживаться определенного порядка действий, а это и есть алгоритм. В итоге формулируем определение алгоритма - это конечная последовательность команд (предписаний) исполнителю совершить конечную последовательность действий, которая направлена на достижение определённой цели.

Параллельно с введением понятия алгоритма можно привести учащимся краткую историческую справку возникновения этого понятия. Слово алгоритм произошло от algorithm - латинского написания слова аль - Хорезми, под которым в средневековой Европе знали величайшего математика из Хорезма (город в современном Узбекистане) Мухамеда бен Мусу, жившего в 783-850 гг. в своей книге "Об индийском счете" он изложил правила записи натуральных чисел с помощью арабских цифр и правила действия над ними "столбиком", знакомые теперь каждому школьнику. В XII веке эта книга была переведена на латынь и получила широкое распространение в Европе.

После того как ученики усвоят, что такое алгоритм надо дать им понятие формального исполнителя. Можно сказать, что сознание алгоритма доступно исключительно живым существам, а долгое время считалось, что только человеку. Другое дело - реализация уже имеющегося алгоритма. Её можно поручить субъекту или объекту, который не обязан вникать в существо дела, а возможно, и не способен его понять. Такой субъект или объект принято называть формальным исполнителем. Затем

можно привести пример формального исполнителя, примером может служить стиральная машина - автомат, которая неукоснительно исполняет предписанные ей действия, даже если вы забыли положить в неё бельё или насыпать стиральный порошок.

Далее необходимо рассказать о свойствах алгоритмов. Первое свойство – дискретность. *Дискретность* алгоритма означает, что он исполняется по шагам: каждое действие, предусмотренное алгоритмом, исполняется только после того, как закончилось исполнение предыдущего. Но здесь важно заметить, что увлекаться детализацией алгоритма, разбиением его на все более и более мелкие шаги не стоит.

Другое свойство принято называть *определённостью*. Оно означает, что на каждом шаге однозначно определено преобразование объектов среды исполнителя, полученных на предшествующих шагах алгоритма. К примеру, в одном из кулинарных рецептов сказано: «Слегка потрясите, чтобы смесь стала комковатой. Подогрейте молоко в маленькой кастрюльке и влейте её в смесь». Перед учениками можно поставить вопрос: сможет ли формальный исполнитель выполнить данный алгоритм? Ведь формальному исполнителю здесь неясно, требуется ли трясти смесь, пока она вся не станет комом, и какой всё-таки величины кастрюля. Большая или маленькая? И до какой температуры надо подогреть коньяк. Так что такой алгоритм любому исполнителю выполнить довольно трудно, практически невозможно. Нужно сказать, что в алгоритме не должны присутствовать не определённые слова: немного, чуть-чуть, слегка и т. д. В качестве примеров неопределённых понятий можно привести «солить по-вкусу», «довести до готовности».

Третье свойство, которое надо дать ученикам - *результативность* алгоритма. Это свойство подразумевает, что каждый шаг (и алгоритм в целом) после своего завершения даёт среду, в которой все имеющиеся объекты однозначно определены. Если это по каким-либо причинам невозможно, то алгоритм должен сообщать, что решение задачи не существует. Классическим примером может служить алгоритм решения квадратного уравнения. В том случае, когда дискриминант принимает отрицательное значение, важно получить сообщение об отсутствии действительных корней, что и будет результатом работы алгоритма.

Надо сказать, что алгоритм должен быть понятен не только автору, но и исполнителю. Например, если мы предложим какому-нибудь мальчику испечь торт, то, как правило, у него ни чего не получится, потому что он этого делать не умеет. Но если мы составим подробный алгоритм работы, разобьем его на элементарные шаги, такие, что он без труда поймёт и сможет выполнить каждый шаг, то сможет успешно испечь

любой торт. Каждый шаг алгоритма обязательно представляет собой какое-либо допустимое действие исполнителя. Это свойство алгоритма называют **понятностью**.

Наконец, еще одно свойство алгоритма - **массовость**. Оно означает, что имеется некоторое множество данных, которые могут обрабатываться алгоритмом. Массовость алгоритма тесно связана с понятностью, в качестве примера можно разобрать пример с тортом, и сказать, что чем подробнее будет описан алгоритм приготовления, тем больше вероятности, что торт будет испечен.

Алгоритмы, исполнителем которых является человек, удобно записывать в словесной форме; изображать с помощью блок-схем. В последнем случае для обозначения шагов алгоритма используются геометрические фигуры.

Наиболее естественной формой представления (восприятия) информации является графический образ – рисунок, чертеж, схема и т.д. К этой форме человек прибегает всякий раз (возможно неявно для себя), когда необходимо решать (описывать, формулировать) действительно сложные задачи. Эффективное оперирование наглядными образами, быстрое установление смысловой связи между ними – является сильной стороной человеческого мышления. Поэтому использование блок-схем является необходимым инструментом образовательного процесса.

С помощью построения блок-схем можно доступно и наглядно рассказать о типах алгоритмов. В зависимости от порядка выполнения команд, выделяют три типа алгоритмов: линейный, циклический и алгоритмы с ветвлением.

Линейный алгоритм - это алгоритм, в котором команды выполняются последовательно, в порядке их записи (рис.1). Наглядным примером являются алгоритмы: «Покупка продуктов», «Приготовление еды».

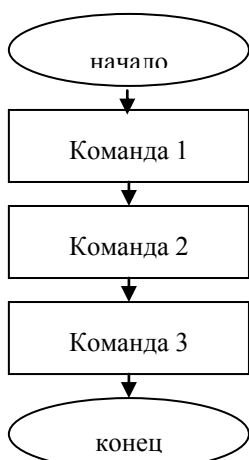


Рис.1 Блок-схема линейного алгоритма

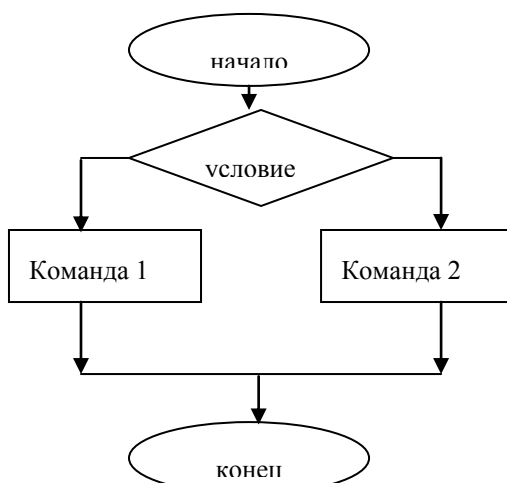


Рис.2 Блок-схема алгоритма ветвления

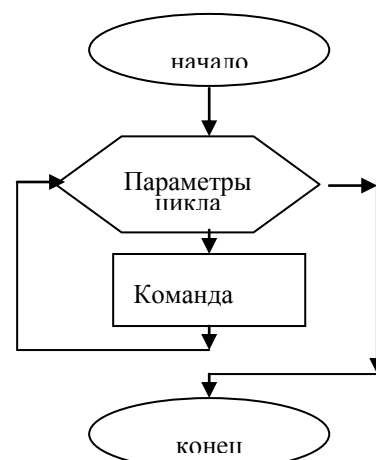


Рис.3 Блок-схема циклического алгоритма

В алгоритмической структуре «ветвление» выполняется та или иная серия команд в зависимости от истинности условия (рис. 1). Пример: «Одеться по погоде». Условие: если на улице идет дождь, то «да», надеваю плащ, и беру зонт, «нет» иду в костюме, зонт не беру.

Циклический алгоритм - это алгоритм, в котором серия команд (тело цикла) повторяется многократно (рис. 3). *Пример:* «Выполнение домашнего задания (из четырех предметов)». Пока все задания не выполнены, цикл является незавершенным. Только после выполнения задания по алгебре, выполнения задания по русскому, выполнения задания по истории, выполнения задания по физике осуществляется выход из цикла.

2.1.2 Visual Basic for Application

В следующих презентациях рассмотрены примеры решения задач с помощью построением блок-схем и написанием программного кода в среде Visual Basic for Application. Прежде чем составлять программы на Visual Basic for Application необходимо познакомить учащихся с основными элементами данной среды, синтаксисом, лексикой и семантикой данного языка программирования.

Visual Basic for Application (VBA) - доступный и в то же время мощный язык макросов для приложений Microsoft Office (Word, Excel и т.п.). Язык VBA встроен в приложения Office, и код на языке VBA можно хранить внутри документов приложений Office — документах Word, книгах Excel, презентациях PowerPoint и т. п.

В настоящее время VBA встроен:

- во все главные приложения MS Office — Word, Excel, Access, PowerPoint, Outlook, FrontPage, InfoPath;
- в другие приложения Microsoft, например, Visio и Microsoft Project;
- в более чем 100 приложений третьих фирм, например, CorelDraw и CorelWordPerfect Office 2000, AutoCAD и т.п.

У VBA есть также множество других преимуществ:

VBA — универсальный язык.

VBA изначально был ориентирован на пользователей, а не на профессиональных программистов, поэтому создавать программы на нем можно научиться быстро и легко. Кроме того, в Office встроены мощные средства, облегчающие работу пользователя: подсказки по объектам и по синтаксису, макрорекордер и т.п.

При создании приложений на VBA не нужно заботиться об установке и настройке специальной среды программирования и наличии нужных библиотек на компьютере пользователя — MS Office есть практически на любом компьютере.

Изучив приёмы программирования на VBA, можно без особых усилий изучать другие языки, такие как Pascal, C++ и др. Так же для операционной системы Macintosh существует язык Real Basic, причем синтаксис, лексика и семантика, которого очень похожи на VBA. Поэтому VBA является тем языком, который включен в курс школьной программы.

Среда Visual Basic for Application может с успехом использоваться начинающими пользователями для познания секретов программирования и увлекательных занятий по созданию несложных (поначалу) приложений и, в то же время, предоставляет мощные инструменты разработки опытным программистам. Развитая справочная система позволит при построении приложения и работе в Visual Basic for Application найти выход из любой ситуации и получить ответ на любой вопрос. Начинать работать с VBA можно практически с любым уровнем подготовки.

Структура проекта VBA

Объекты, методы, свойства, события

Объектам VBA присуща функциональность - они действуют определенным образом и могут откликаться на определенные ситуации. При этом если свойства объекта определяют его внешний вид и состояние, методы объекта определяют те задачи, которые может выполнить данный объект. Методы, по сути дела, представляют собой сегмент программного кода, внедренный в объект. Существует определенный формат программного кода, задающего установку свойства и использование метода:

Объект. Свойство = ЗначениеСвойства

Объект. Метод [Параметр1 [...]]

Здесь объект - имя настраиваемого объекта; Свойство - это характеристика, которую нужно изменить; Метод - это команда, которая используется для изменения объекта; Значение - это новая установка свойства; Параметр - это аргумент, используемый методом. Следующий пример показывает, как можно изменить текст в строке заголовка окна приложения посредством установки нового значения свойству Caption объекта Application:Application.Caption = "Пример установки свойства» Как и свойства, различные типы объектов могут иметь различные методы. Следующий пример показывает использование метода CountA (подсчет количества не пустых строк) объекта Application для определения номера последней не пустой строки активного рабочего листа:

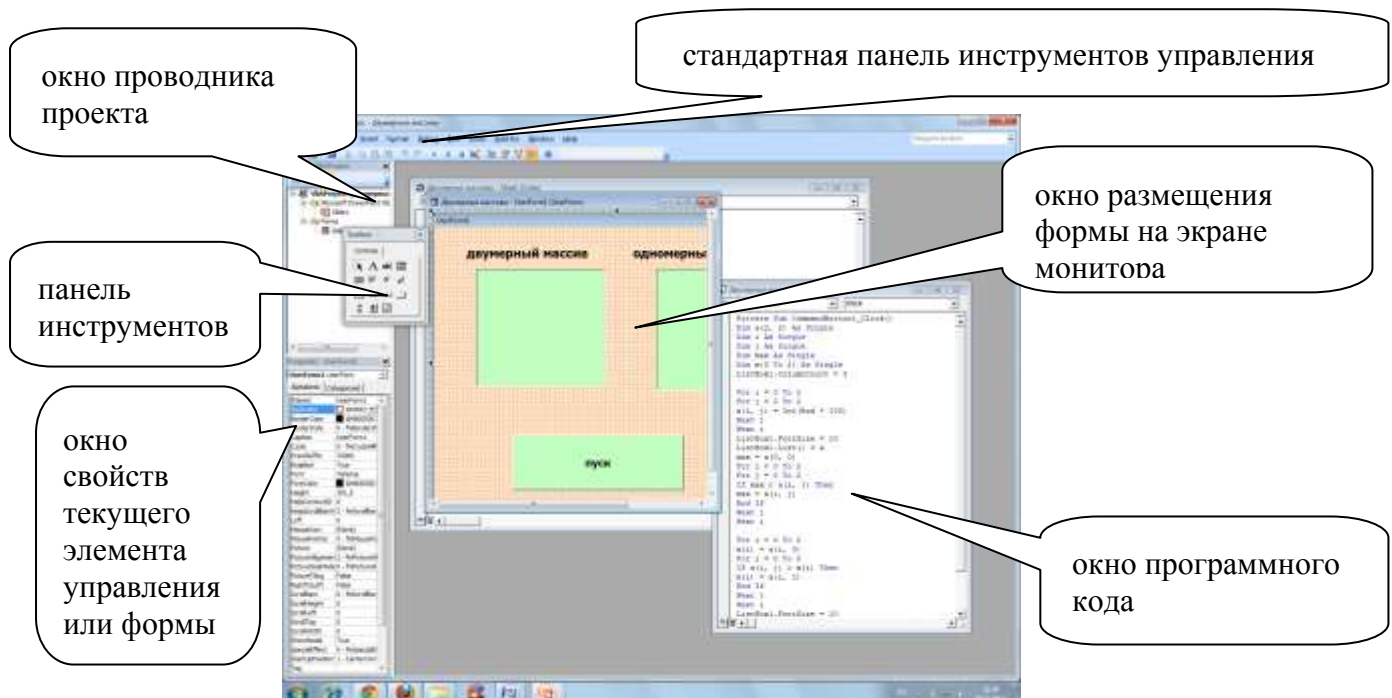
НомерСтроки= Application.CountA(ActiveSheet.Columns(1))

Объекты могут реагировать на события - действия пользователя или другие внешние действия, например, щелчок по кнопке, изменение текста, нажатие клавиши и

др. Событие представляет собой действие, распознаваемое объектом, для которого можно запрограммировать отклик. Иногда свойства и методы объекта оказываются связанными в том смысле, что выполнение некоторого метода приводит к изменению свойств объекта. В свою очередь, изменение некоторых свойств может вызвать наступление событий. Программа может обрабатывать два основных типа событий: инициируемые пользователем и генерируемые системой. События, инициируемые пользователем, возникают в результате его действий: нажатие клавиши, щелчки кнопками мыши. Но есть события, являющиеся следствием действий пользователя. Таким образом, любое из действий пользователя может вызвать целый набор событий, и порядок их вызова может быть важным. Основными действиями пользователя, генерирующими вызов событий в программе, являются следующие:· запуск программы; нажатие клавиши; щелчок кнопкой мыши;· перемещение мыши;· выход из программы. Суть программирования на VBA заключается в двух понятиях: событие и отклик на него. Если пользователь производит какое-то воздействие на систему, допустим, щелкает на кнопке, тогда в качестве отклика выполняется код созданной пользователем процедуры. Этот специальный вид процедур, генерирующих отклик на события, называется процедурами обработки событий. Если такой отклик не создан (не написана соответствующая процедура), то система никак не будет реагировать на это событие. Таким образом, задачей пользователя является разработка программного кода процедур для обработки различных событий, которые являются важными с точки зрения функционирования программы.

Среда разработки VBA

Рис.1. Окно среды программирования Visual Basic.



Главное преимущество объектно-ориентированного программирования в том, что объекты обеспечивают программный код, который уже не требуется писать разработчику. Ему просто нужно установить свойства объекта и вызвать методы объекта, чтобы побудить объект выполнить требуемые функции.

2.1.2 Решение задач в среде VBA

Для того чтобы учащиеся смогли закрепить полученные знания и навыки предлагается решение задач в среде VBA. Приведем примеры решения некоторых задач:

1. «Найти периметр и площадь треугольника (по формуле Герона) по заданным трем сторонам (A, B, C)». Во-первых, составим алгоритм решения поставленной задачи. В приведенном примере данная задача решается с использованием алгоритма ветвления. Условием является сравнение длины каждой стороны с суммой двух оставшихся, т.е.

$A + B > C$, $B + C > A$, $C + A > B$. Если условие выполняется, то вычисляем периметр и площадь треугольника по формуле Герона. Если нет, то появляется диалоговое окно, говорящее об ошибке «Сумма двух сторон треугольника должна быть больше третьей стороны». Во-вторых, составляется блок-схема алгоритма.

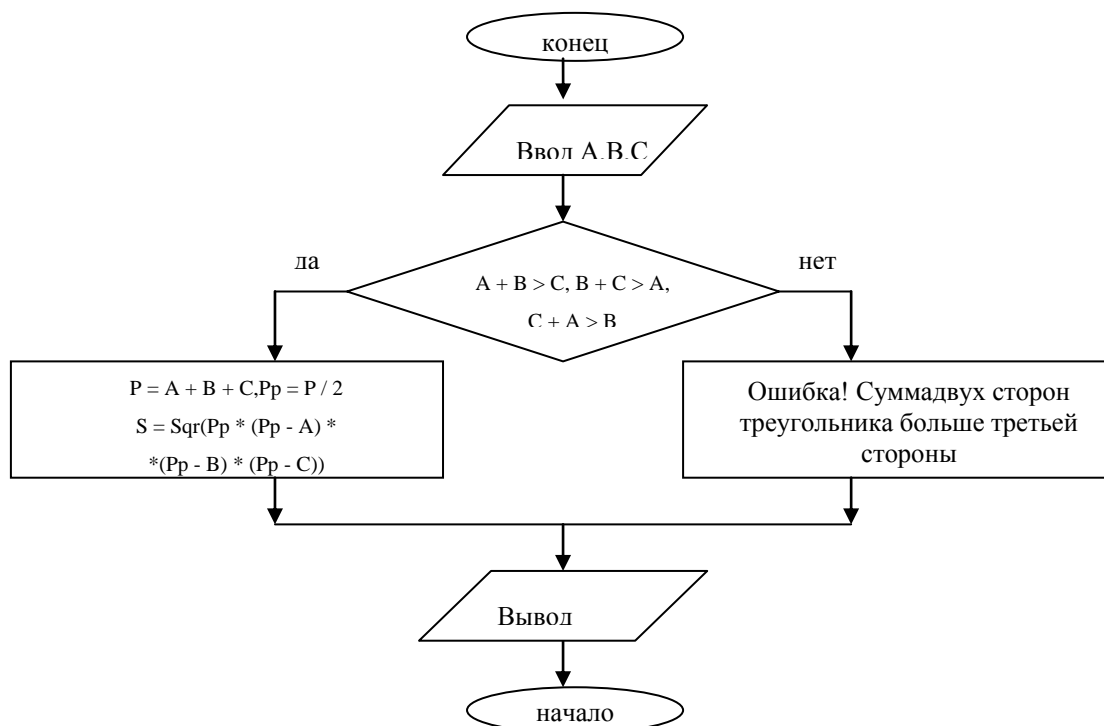


Рис. 5. Блок-схема к задаче «Нахождение площади и периметра треугольника по известным трем сторонам A, B, C»

В качестве наглядного примера, учащимся приводится фрагмент программного кода:

```

If (A + B > C) And (B + C > A) And (C + A > B) Then
P = A + B + C
Pr = P / 2
  
```

```

S = Sqr(Pp * (Pp - A) * (Pp - B) * (Pp - C))
TextBox4.Text = Str(P)
TextBox5.Text = Str(S)
Else
Text1.Text = " Ошибка! Суммадвух сторон треугольника больше третьей стороны"
Text2.Text = ""
Text3.Text = ""
End If

```

- «Нахождение корней квадратного уравнения $ax^2+bx+c=0$ ». Представленная задача решается с помощью алгоритма вложенного ветвления. Во-первых, делаем проверку значения дискриминанта – это первое условие. Если $d>0$, то по ветви «нет» ставится следующее условие: равен ли дискриминант нулю. Если условие выполняется, то корень уравнения единственный и вычисляется по формуле: $x = -b / (2 * a)$ (конечно, нужно рассматривать случай при котором $a \neq 0$, иначе выражение не имеет смысла). Если дискриминант меньше нуля, то появляется надпись «Нет корней». В случае выполнения
- первого условия, т.е. по ветви «да» существует два корня квадратного уравнения, и они рассчитываются по формулам: $x_1 = (-b + \text{Sqr}(d)) / (2 * a)$; $x_2 = (-b - \text{Sqr}(d)) / (2 * a)$. Во-вторых, выполняется построение блок-схемы.

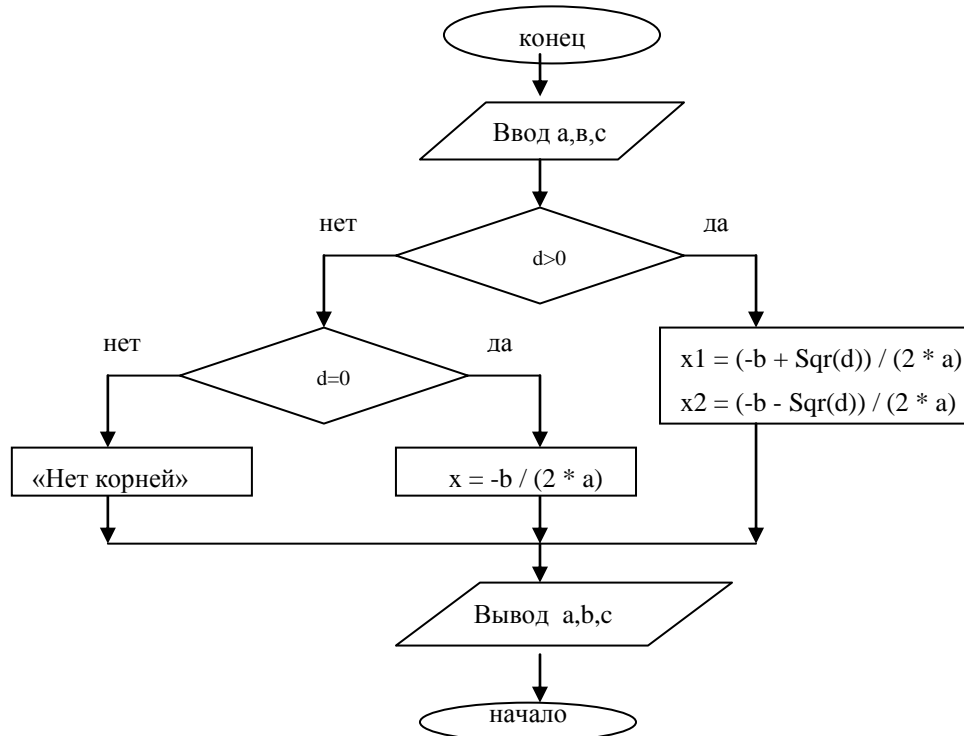


Рис. 5. Блок-схема к задаче «Нахождение корней квадратного уравнения $ax^2+bx+c=0$ ». В третьих, приводится фрагмент программного кода:

```

If d > 0 Then
x1 = (-b + Sqr(d)) / (2 * a)

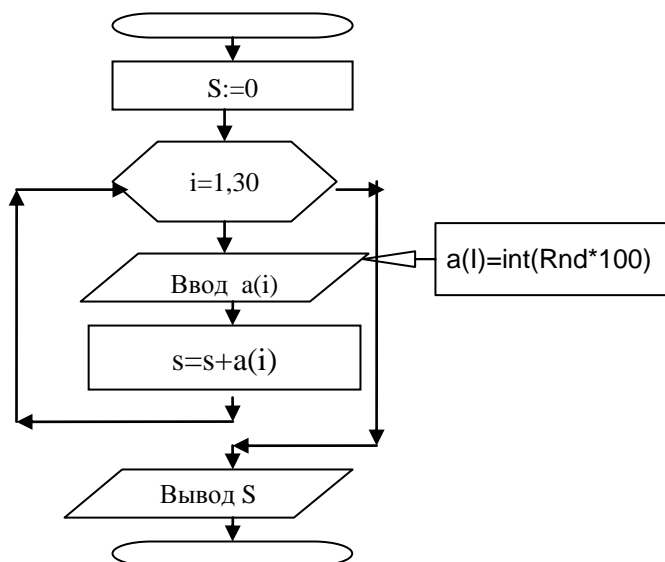
```

```

x2 = (-b - Sqr(d)) / (2 * a)
Label1.Caption = x1
Label2.Caption = x2
Else
If d = 0 Then
    x2 = -b / (2 * a)
    Label2.Caption = x2
    Label1.Caption = "Один корень"
Else
    Label1.Caption = "Нет корней"
    Label2.Caption = "Нет корней"
End If
End If

```

3. Найти сумму 30 целых чисел. Пусть значения этих чисел от 25 до 54. используем для решения этой задачи одномерный массив, которому дадим имя А, каждое из чисел будет иметь свой порядковый номер и к каждому значению можно обращаться по порядковому номеру элемента массива, например: А[3]. Значение третьего элемента массива А[3]=27. Массивы представляют собой фиксированное количество элементов одного типа, объединенных одним именем, причем каждый элемент имеет свой уникальный номер, и номера элементов идут подряд. Существуют одномерные и двумерные массивы. Одномерные массивы называют одно-индексными переменными, двумерные - двух индексными переменными. В данном случае задан одномерный массив. Строим блок-



схему:

Вводить значения элементов массива можно не только с клавиатуры, но считывая данные из файла, или заполняя массив случайными числами, считывая их с генератора случайных чисел - Randomize. Затем приводим фрагмент программного кода для поставленной задачи:

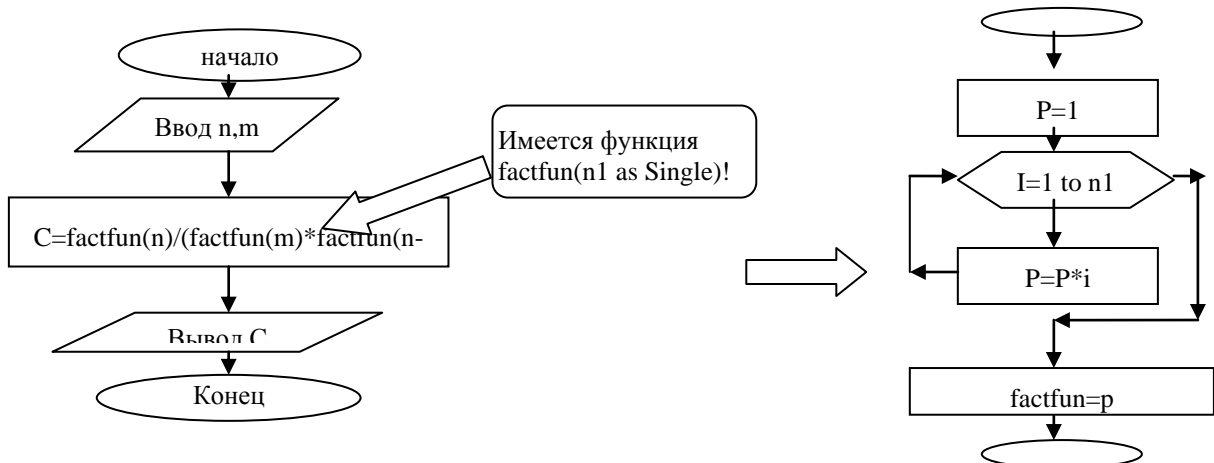
```

Dim a(1 to 30) as Single
Dim s as Single
Dim p as Single
Dim I as integer
For i = 1 To 10
a(i) = Int(Rnd * 10) + 1
List1.additem a(I)
Next i
s = 0
p = 1
For i = 1 To 10 Step 1
s = s + a(i)
p = p * a(i)
Next i
Label1.caption=s
Label2.caption=p

```

4. «Рассчитать число сочетаний из N по M: $C_n^m = \frac{n!}{m!(n-m)!}$ »

Данную задачу решаем с помощью вспомогательного алгоритма. Нередко в программах встречаются повторяющиеся или похожие фрагменты. Имеется возможность оформлять такие фрагменты специальным образом - выделять в подпрограммы, к которым можно обращаться из любого места основной программы указывая их имя, подобно тому, как мы это делаем, вызывая стандартные процедуры и функции (sq(x), abs(x), sin(x) и пр.). В VB имеется два вида подпрограмм: общие процедуры и функции пользователя. Эти вспомогательные алгоритмы оформляются в опции **Insert командой Procedure...** при открытом окне программного кода. Блок-схема для решения представленной задачи (основной алгоритм):



Далее приводим пример вспомогательного алгоритма

Factfun(n1 as Single):

Функция имеет формальные параметры, которые при обращении к ней получают фактические значения.

Затем приводим пример программного кода.

```
Private Sub Command2_Click()  
n = Val(Textbox3.Text)  
m = Val(TextBox4.Text)  
c = factfun(n) / (factfun(m) * factfun(n - m))  
Label6.Caption = c  
End Sub  
Public Function factfun(n1 As Single)  
Dim i As Single  
factfun = 1  
For i = 1 To n1  
factfun = factfun * i  
Next i  
End Function
```

2.1.3 Задачи для самостоятельного решения

Обязательным условием закрепления полученных знаний, умений и навыков учащимися является самостоятельное решение задач. Далее приводятся примеры условий десяти задач:

1. Определите, является ли человек пенсионером. Если мужчине назначается пенсия старше 60 лет, а женщине больше 55 лет, или стаж по выслуге лет человека превысил 25 лет.
2. Составьте алгоритм – меню, печатающее расписание уроков в вашем классе в определенный день недели.
3. Вычислите, что больше $\sin(a-3)$ или $\text{tg}(a-10)$.
4. Составьте программу, выбирающую из трех чисел то, которое лежит между двумя другими.
5. Начальный вклад в сбербанк составил А рублей под Р процентов годовых. Через сколько лет он станет больше В рублей?
6. Сбербанк начисляет Р процентов годовых. Какой станет сумма в рублях А, положенная на N лет?

7. В первый день тренировок спортсмен пробежал 10 км. В каждый следующий день он увеличивал норму на 10% от предыдущего дня. Через сколько дней он будет пробегать ежедневно больше 20 км? Какое расстояние он пробежит за 10 дней? Через сколько дней он пробежит суммарный путь 100 км?
8. Составить программу, чтобы она считала факториал числа N: $N! = 1*2*3*...*N$.
9. Составить программу, чтобы производился расчет суммы: $S=1+1/2^2+1/3^3+...+1/N^N$
10. Дано действительное число x. Вычислить:

$$\frac{x}{x^2 + \frac{2}{4}}$$

$$x^2 + \frac{\quad}{8}$$

$$x^2 + \frac{\quad}{x^2 + \frac{256}{x^2}}$$

2.2 Содержание обучения

Обязательный минимум содержания учебного модуля «Алгоритмы» определён Федеральными Государственными стандартами образования Российской Федерации третьего поколения.

Изучение информатики в профильной школе (10-11классы) реализуется на двух уровнях:

1. базовый (общеобразовательный) уровень — **уровень А**, из расчёта 34 учебных часов (1 час в неделю) в старшей школе
2. профильный уровень — **уровень Б**, 136 учебных часов (4 часа в неделю) на профильном уровне.

Изучение информатики и информационных технологий в старшей школе на профильном уровне направлено на достижение следующих целей:

По модулю «Алгоритмы» предусматривается изучение следующих вопросов:

- Алгоритм.
- Свойства алгоритмов.
- Способы записи алгоритмов;
- Блок-схемы.
- Алгоритмические конструкции.
- Разбиение задачи на подзадачи, вспомогательный алгоритм.

В результате изучения ученик должен знать:

- основные конструкции языка программирования;

- свойства алгоритмов и основные алгоритмические конструкции; тезис о полноте формализации понятия алгоритма;

Уметь:

- строить информационные модели объектов, систем и процессов, используя для этого типовые средства (язык программирования, таблицы, графики, диаграммы, формулы и т.п.); [1].

Использовать приобретенные знания и умения в практической деятельности и повседневной жизни для:

- поиска и отбора информации, в частности, связанной с личными познавательными интересами, самообразованием и профессиональной ориентацией;

В соответствии задачами ФГОСов третьего поколения разработано планирование по курсу «Основы алгоритмизации и программирование» (62 ч.)

Основы алгоритмизации и программирование	
1,2 (4 час)	Повт. Программное обеспечение ПК. Назначение и характеристика языков программирования. Интегрированная среда разработки языка Visual Basic for Application . Программа на VBA. Проект, форма. Сохранение программы. Графический интерфейс, назначение окон и панелей.(2 час) Примеры разработки простых визуальных проектов.(2час)
3 (2 час)	Объект, свойства, методы и события. Разработка проекта
4 (2 час)	Событийная процедура, формат записи. Разработка проекта
5 (2 час)	Объект Image, его свойства, свойства объекта Label. Разработка проекта №3
5,6 (4 час)	Арифметические, строковые и логические выражения. Объявление переменных и присваивание им значений. Добавление формы в проект, датчик случайных чисел, условный алгоритм
7 (2 час)	Понятие функции в языке VBA. Математические и строковые функции. Примеры. Функции преобразования типов данных. Функции даты и времени. Разработка простых проектов, связанных с переменными рассмотренных типов.
8 (2 час)	Понятие алгоритма, блок-схема алгоритма. Свойства алгоритмов. Типы алгоритмических структур.
9 (2 час)	Линейный алгоритм. Разработка проектов

10 (2 час)	Разработка блок-схемы программы решения квадратного уравнения. Алгоритмическая структура «Ветвление - полное и неполное» в VBA.Создание проекта
11 (2 час)	Алгоритмическая структура «Вложенное ветвление» в VBA.Создание проекта
12 (2 час)	Метки и их свойства. Множественный выбор Case. Разработка блок-схемы программы «калькулятор» в одном окне.
13,14,15 (6 час)	Алгоритмическая структура «Цикл» в VBA. Блок- схемы циклов со счетчиком, предусловием и постусловием. Разработка типовых проектов.
16 (2 час)	Редактор меню. Разработка проекта
17 (2 час)	Создание диалоговых программ. Включение в проект тестового задания по типам алгоритмов.
18 (2 час)	Вспомогательный алгоритм. Событийные и общие процедуры. Область видимости процедур Разработка проекта
19 (2 час)	Модульный принцип построения проекта и программного кода
20 (2 час)	Разработка проектов с общими процедурами. Решение задач по теме, разработка проектов
21 (2 час)	Реализация вспомогательного алгоритма в виде функции Решение задач по теме, разработка проектов
22 (2 час)	Понятие массива. Работа с массивами. Заполнение массива, поиск в массивах.
23 (2 час)	Сортировка массива. Разработка и отладка проекта.
24 (2 час)	Разработка проектов и решение задач на VBA по теме «Массивы»
25 (2 час)	Двумерные массивы и вложенные циклы.
26 (2 час)	Разработка проектов и решение задач на обработку двумерных массивов. Массивы и общие процедуры Отладка проектов нахождения максим. и мин эл-тов строк и столбцов.
27 (2 час)	Вспомогательный алгоритм и массивы Решение задач по теме, разработка проектов
28 (2 час)	Технология решения задач на компьютере. Основные этапы. Принципы создания диалоговых программ.
29, 30, 31	Решение задач формата ЕГЭ

2.2.1 Содержательные аспекты темы «Алгоритмы и алгоритмические структуры»

С алгоритмами связана вся история математики, это одно из основных ее понятий. На протяжении продолжительного времени люди пользовались интуитивным понятием «алгоритм», которое близко по смыслу понятиям «правило», «инструкция». [И.А. Смольникова].

Человек знакомится с понятием алгоритма, обучаясь в средней, а затем в высшей школе. Ценность алгоритмов для обучения состоит в том, что они приводят к решению задачи возможно более коротким путем. С помощью алгоритма решение разбивается на простые операции и осуществляется автоматически, если следовать указаниям алгоритма шаг за шагом. [И.А. Смольникова].

Слово «Алгоритм» происходит от algorithmi - латинского написания имени аль-Хорезми, под которым в средневековой Европе знали величайшего математика из Хорезма (город в современном Узбекистане) Мухаммеда бен Мусу, жившего в 783-850 гг. В своей книге «Об индийском счете» он сформулировал правила записи натуральных чисел с помощью арабских цифр и правила действий над ними столбиком. [Кузнецов А.А. и др. Основы информатики. - М.: Дрофа, 1998]. В дальнейшем алгоритмом стали называть точное предписание, определяющее последовательность действий, обеспечивающую получение требуемого результата из исходных данных. Алгоритм может быть предназначен для выполнения его человеком или автоматическим устройством. Создание алгоритма, пусть даже самого простого, - процесс творческий. Он доступен исключительно живым существам, а долгое время считалось, что только человеку. Другое дело - реализация уже имеющегося алгоритма. Ее можно поручить субъекту или объекту, который не обязан вникать в существо дела, а возможно, и не способен его понять. Такой субъект или объект принято называть формальным исполнителем. Примером формального исполнителя может служить стиральная машина-автомат, которая неукоснительно исполняет предписанные ей действия, даже если вы забыли положить в нее порошок. Человек тоже может выступать в роли формального исполнителя, но в первую очередь формальными исполнителями являются различные автоматические устройства, и компьютер в том числе. Каждый алгоритм создается в расчете на вполне конкретного исполнителя. Те действия, которые может совершать исполнитель, называются его допустимыми действиями. Совокупность допустимых действий образует систему команд исполнителя. Алгоритм должен содержать только те действия, которые допустимы для данного исполнителя.

2.2.2 Свойства алгоритмов

Данное выше определение алгоритма нельзя считать строгим - не вполне ясно, что такое «точное предписание» или «последовательность действий, обеспечивающая получение требуемого результата». Поэтому обычно формулируют несколько общих свойств алгоритмов, позволяющих отличать алгоритмы от других инструкций.

Такими свойствами являются:

Дискретность (прерывность, раздельность) - алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов. Каждое действие, предусмотренное алгоритмом, исполняется только после того, как закончилось исполнение предыдущего.

Определенность - каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.

Результативность (конечность) - алгоритм должен приводить к решению задачи за конечное число шагов.

Массовость - алгоритм решения задачи разрабатывается в общем виде, то есть, он должен быть применим для некоторого класса задач, различающихся только исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

Правила выполнения арифметических операций или геометрических построений представляют собой алгоритмы. При этом остается без ответа вопрос, чем же отличается понятие алгоритма от таких понятий, как «метод», «способ», «правило». Можно даже встретить утверждение, что слова «алгоритм», «способ», «правило» выражают одно и то же (т.е. являются синонимами), хотя такое утверждение, очевидно, противоречит «свойствам алгоритма».

Само выражение «свойства алгоритма» не совсем корректно. Свойствами обладают объективно существующие реальности. Можно говорить, например, о свойствах какого-либо вещества. Алгоритм – искусственная конструкция, которую мы сооружаем для достижения своих целей. Чтобы алгоритм выполнил свое предназначение, его необходимо строить по определенным правилам. Поэтому нужно говорить все же не о свойствах алгоритма, а о правилах построения алгоритма, или о требованиях, предъявляемых к алгоритму.

Первое правило – при построении алгоритма, прежде всего, необходимо задать множество объектов, с которыми будет работать алгоритм. Формализованное (закодированное) представление этих объектов носит название данных. Алгоритм

приступает к работе с некоторым набором данных, которые называются входными, и в результате своей работы выдает данные, которые называются выходными. Таким образом, алгоритм преобразует входные данные в выходные.

Это правило позволяет сразу отделить алгоритмы от “методов” и “способов”. Пока мы не имеем формализованных входных данных, мы не можем построить алгоритм.

Второе правило – для работы алгоритма требуется память. В памяти размещаются входные данные, с которыми алгоритм начинает работать, промежуточные данные и выходные данные, которые являются результатом работы алгоритма. Память является дискретной, т.е. состоящей из отдельных ячеек. Поименованная ячейка памяти носит название переменной. В теории алгоритмов размеры памяти не ограничиваются, т. е. считается, что мы можем предоставить алгоритму любой необходимый для работы объем памяти.

В школьной «теории алгоритмов» эти два правила не рассматриваются. В то же время практическая работа с алгоритмами (программирование) начинается именно с реализации этих правил. В языках программирования распределение памяти осуществляется декларативными операторами (операторами описания переменных). В языке Бейсик не все переменные описываются, обычно описываются только массивы. Но все равно при запуске программы транслятор языка анализирует все идентификаторы в тексте программы и отводит память под соответствующие переменные.

Третье правило – дискретность. Алгоритм строится из отдельных шагов (действий, операций, команд). Множество шагов, из которых составлен алгоритм, конечно.

Четвертое правило – детерминированность. После каждого шага необходимо указывать, какой шаг выполняется следующим, либо давать команду остановки.

Пятое правило – сходимость (результативность). Алгоритм должен завершать работу после конечного числа шагов. При этом необходимо указать, что считать результатом работы алгоритма.

Итак, алгоритм – неопределяемое понятие теории алгоритмов. Алгоритм каждому определенному набору входных данных ставит в соответствие некоторый набор выходных данных, т. е. вычисляет (реализует) функцию. При рассмотрении конкретных вопросов в теории алгоритмов всегда имеется в виду какая-то конкретная модель алгоритма.

2.2.3 Виды алгоритмов и их реализация

Алгоритм применительно к вычислительной машине – точное предписание, т.е. набор операций и правил их чередования, при помощи которого, начиная с некоторых исходных данных, можно решить любую задачу фиксированного типа.

Виды алгоритмов как логико-математических средств отражают указанные компоненты человеческой деятельности и тенденции, а сами алгоритмы в зависимости от цели, начальных условий задачи, путей ее решения, определения действий исполнителя подразделяются следующим образом:

Механические алгоритмы, или иначе детерминированные, жесткие (например, алгоритм работы машины, двигателя и т.п.);

Гибкие алгоритмы, например стохастические, т.е. вероятностные и эвристические.

Механический алгоритм задает определенные действия, обозначая их в единственной и достоверной последовательности, обеспечивая тем самым однозначный требуемый или искомый результат, если выполняются те условия процесса, задачи, для которых разработан алгоритм.

Вероятностный (стохастический) алгоритм дает программу решения задачи несколькими путями или способами, приводящими к вероятному достижению результата.

Эвристический алгоритм (от греческого слова “эврика”) – это такой алгоритм, в котором достижение конечного результата программы действий однозначно не предопределено, так же как не обозначена вся последовательность действий, не выявлены все действия исполнителя. К эвристическим алгоритмам относят, например, инструкции и предписания. В этих алгоритмах используются универсальные логические процедуры и способы принятия решений, основанные на аналогиях, ассоциациях и прошлом опыте решения схожих задач.

Линейный алгоритм – набор команд (указаний), выполняемых последовательно во времени друг за другом.

Разветвляющийся алгоритм – алгоритм, содержащий хотя бы одно условие, в результате проверки которого ЭВМ обеспечивает переход на один из двух возможных шагов.

Циклический алгоритм – алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными. К циклическим алгоритмам сводится большинство методов вычислений, перебора вариантов.

Цикл программы – последовательность команд (серия, тело цикла), которая может выполняться многократно (для новых исходных данных) до удовлетворения некоторого условия.

1. **Вспомогательный** (подчиненный) алгоритм (процедура) – алгоритм, ранее разработанный и целиком используемый при алгоритмизации конкретной задачи. В некоторых случаях при наличии одинаковых последовательностей

указаний (команд) для различных данных с целью сокращения записи также выделяют вспомогательный алгоритм. Вычислительная техника и программирование. [Вычислительная техника и программирование. Под ред. А.В.Ретрова. - М.: Высшая школа, 1990].

2.2.4 Методы изображения алгоритмов

На практике наиболее распространены следующие формы представления алгоритмов:

словесная (записи на естественном языке);

графическая (изображения из графических символов);

псевдокоды (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);

программная (тексты на языках программирования).

2.2.5 Словесное описание алгоритма

Данный способ получил значительно меньшее распространение из-за его многословности и отсутствия наглядности.

Рассмотрим пример на алгоритме нахождения максимального из двух значений:

Определим форматы переменных X , Y , M , где X и Y – значения для сравнения, M – переменная для хранения максимального значения;

получим два значения чисел X и Y для сравнения;

сравним X и Y .

если X меньше Y , значит большее число Y .

Поместим в переменную M значение Y .

Если X не меньше (больше) Y , значит большее число X .

Поместим в переменную M значение X .

Словесный способ не имеет широкого распространения по следующим причинам:

такие описания строго не формализуемы;

страдают многословностью записей;

допускают неоднозначность толкования отдельных предписаний.

2.2.6 Блок-схема алгоритма

А этот способ оказался очень удобным средством изображения алгоритмов и получил широкое распространение в научной и учебной литературе.

Структурная (блок-, граф-) схема алгоритма – графическое изображение алгоритма в виде схемы связанных между собой с помощью стрелок (линий перехода) блоков –

графических символов, каждый из которых соответствует одному шагу алгоритма. Внутри блока дается описание соответствующего действия.

Графическое изображение алгоритма широко используется перед программированием задачи вследствие его наглядности, т.к. зрительное восприятие обычно облегчает процесс написания программы, ее корректировки при возможных ошибках, осмысливание процесса обработки информации.


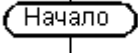

Можно встретить даже такое утверждение: «Внешне алгоритм представляет собой схему – набор прямоугольников и других символов, внутри которых записывается, что вычисляется, что вводится в машину и что выдается на печать и другие средства отображения информации «Здесь форма представления алгоритма смешивается с самим алгоритмом.

Принцип программирования «сверху вниз» требует, чтобы блок-схема поэтапно конкретизировалась и каждый блок «расписывался» до элементарных операций. Но такой подход можно осуществить при решении несложных задач. При решении сколько-нибудь серьезной задачи блок-схема «расползется» до такой степени, что ее невозможно будет охватить одним взглядом.

Блок-схемы алгоритмов удобно использовать для объяснения работы уже готового алгоритма, при этом в качестве блоков берутся действительно блоки алгоритма, работа которых не требует пояснений. Блок-схема алгоритма должна служить для упрощения изображения алгоритма, а не для усложнения.

В таблице приведены наиболее часто употребляемые символы.

Название символа	Обозначение и пример заполнения	Пояснение
Процесс		Вычислительное действие или последовательность действий
Решение		Проверка условий
Модификация		Начало цикла
Предопределенный процесс		Вычисления по подпрограмме, стандартной подпрограмме

Ввод-вывод		Ввод-вывод в общем виде
Пуск-останов		Начало, конец алгоритма, вход и выход в подпрограмму
Документ		Вывод результатов на печать

Блок «процесс» применяется для обозначения действия или последовательности действий, изменяющих значение, форму представления или размещения данных. Для улучшения наглядности схемы несколько отдельных блоков обработки можно объединять в один блок. Представление отдельных операций достаточно свободно.

Блок «решение» используется для обозначения переходов управления по условию. В каждом блоке «решение» должны быть указаны вопрос, условие или сравнение, которые он определяет.

Блок «модификация» используется для организации циклических конструкций. (Слово модификация означает видоизменение, преобразование). Внутри блока записывается параметр цикла, для которого указываются его начальное значение, граничное условие и шаг изменения значения параметра для каждого повторения.

Блок «предопределенный процесс» используется для указания обращений к вспомогательным алгоритмам, существующим автономно в виде некоторых самостоятельных модулей, и для обращений к библиотечным подпрограммам.

2.2.7 Псевдокод

Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов. Он занимает промежуточное место между естественным и формальным языками.

С одной стороны, он близок к обычному естественному языку, поэтому алгоритмы могут на нем записываться и читаться как обычный текст. С другой стороны, в псевдокоде используются некоторые формальные конструкции и математическая символика, что приближает запись алгоритма к общепринятой математической записи.

В псевдокоде не приняты строгие синтаксические правила для записи команд, присущие формальным языкам, что облегчает запись алгоритма на стадии его проектирования и дает возможность использовать более широкий набор команд,

рассчитанный на абстрактного исполнителя. Однако в псевдокоде обычно имеются некоторые конструкции, присущие формальным языкам, что облегчает переход от записи на псевдокоде к записи алгоритма на формальном языке. В частности, в псевдокоде, так же, как и в формальных языках, есть служебные слова, смысл которых определен раз и навсегда. Они выделяются в печатном тексте жирным шрифтом, а в рукописном тексте подчеркиваются. Единого или формального определения псевдокода не существует, поэтому возможны различные псевдокоды, отличающиеся набором служебных слов и основных (базовых) конструкций.

Примером псевдокода является школьный алгоритмический язык в русской нотации, описанный в учебнике А.Г. Кушниренко и др. «Основы информатики и вычислительной техники».

Пример записи алгоритма на школьном алгоритмическом языке:

алг Сумма квадратов (арг цел n , рез цел S)

дано | $n > 0$

надо | $S = 1*1 + 2*2 + 3*3 + \dots + n*n$

нач цел i

ввод n ; $S:=0$

нц для i от 1 до n

$S:=S+i*i$

кц

вывод "S = ", S

кон

2.2.8 Программное представление алгоритма

При записи алгоритма в словесной форме, в виде блок-схемы или на псевдокоде допускается определенный произвол при изображении команд. Вместе с тем такая запись точна настолько, что позволяет человеку понять суть дела и исполнить алгоритм.

Однако на практике в качестве исполнителей алгоритмов используются специальные автоматы — компьютеры. Поэтому алгоритм, предназначенный для исполнения на компьютере, должен быть записан на «понятном» ему языке. И здесь на первый план выдвигается необходимость точной записи команд, не оставляющей места для произвольного толкования их исполнителем.

Следовательно, язык для записи алгоритмов должен быть формализован. Такой язык принято называть языком программирования, а запись алгоритма на этом языке — программой для компьютера.

2.2.9 Порядок разработки иерархической схемы алгоритма

К основным методам структурного программирования относится, прежде всего, отказ от бессистемного употребления оператора непосредственного перехода и преимущественное использование других структурированных операторов, методы нисходящего проектирования разработки программы, идеи пошаговой детализации и некоторые другие соглашения, касающиеся дисциплины программирования.

Всякая программа, в соответствии с структурным подходом к программированию, может быть построена только с использованием трех основных типов блоков.

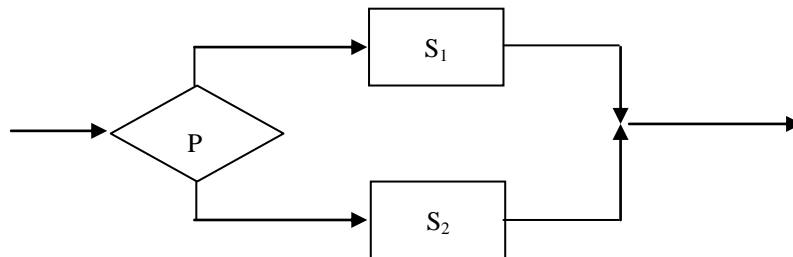
1. Функциональный блок, который на блок-схеме изображается в виде прямоугольников с одним входом и одним выходом:



Функциональному блоку в языках программирования соответствуют операторы ввода и вывода или любой оператор присваивания.

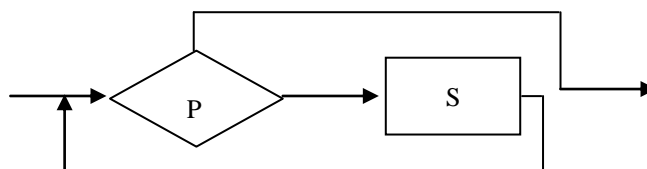
В виде функционального блока может быть изображена любая последовательность операторов, выполняющихся один за другим, имеющая один вход и один выход.

2. Условная конструкция. Этот блок включает проверку некоторого логического условия (P), в зависимости от которого выполняется либо один (S1), либо другой (S2)



операторы:

3. Блок обобщенного цикла. Этот блок обеспечивает многократное повторение выполнения оператора S пока выполнено логическое условие P:



При конструировании программы с использованием рассмотренных типов блоков эти блоки образуют линейную цепочку так, что выход одного блока подсоединяется к входу следующего. Таким образом, программа имеет линейную структуру, причем порядок следования блоков соответствует порядку, в котором они выполняются.

Такая структура значительно облегчает чтение и понимание программы, а также упрощает доказательство ее правильности. Так как линейная цепочка блоков может быть

сведена к одному блоку, то любая программа может, в конечном итоге, рассматриваться как единый функциональный блок с одним входом и одним выходом.

При проектировании и написании программы нужно выполнить обратное преобразование, то есть этот блок разбить на последовательность подблоков, затем каждый подблок разбить на последовательность более мелких блоков до тех пор, пока не будут получены «атомарные» блоки, рассмотренных выше типов. Такой метод конструирования программы принято называть нисходящим («сверху вниз»).

При нисходящем методе конструирования алгоритма и программы первоначально рассматривается вся задача в целом. На каждом последующем этапе задача разбивается на более мелкие подзадачи, каждая подзадача, в конечном итоге на еще более мелкие подзадачи и так до тех пор, пока не будут получены такие подзадачи, которые легко кодируются на выбранном языке программирования. При этом на каждом шаге уточняются все новые и новые детали («пошаговая детализация»).

В процессе нисходящего проектирования сохраняется строгая дисциплина программирования, то есть разбиение на подзадачи осуществляется путем применения только рассмотренных типов конструкций (функциональный блок, условная конструкция, обобщенный цикл), поэтому, в конечном итоге, получается хорошо структурированная программа.

Разработка алгоритмов как творческий процесс требует профессиональной эрудиции и изобретательности. Алгоритмист должен разбираться в постановке решаемой задачи и одновременно четко представлять специфику языка программирования и возможности компьютера, на котором впоследствии будет реализован алгоритм.
[А.И.Смольникова]

Следует отметить, что роль алгоритмов весьма значительна в учебном процессе при изучении естественнонаучных и технических дисциплин. Особенно эта роль возрастает при подготовке и повышении квалификации кадров в области информатики.

Однако общей **основой обучения** для трех целевых направлений служит **«алгоритмический» стиль мышления** обучаемых. Если для первой и третьей целей формирование алгоритмического стиля мышления является непосредственной предпосылкой эффективного решения задач на ЭВМ, то всем при работе с компьютером часто приходится пользоваться алгоритмизацией хотя бы на уровне обыденного сознания: разделять работу на отдельные этапы; ставить выполнение того или иного этапа в зависимость от выполнения предыдущего этапа; обеспечивать повторность выполнения этапов в зависимости от истинности или ложности критерия и т.п. По обучению этот достаточно низкий уровень алгоритмического мышления должен трансформироваться в более развитый в методологическом смысле.

Необходимо также отметить возрастание роли алгоритмического стиля как средства коммуникации и обмена идеями, данными планами действий и совмещения расписаний между людьми, как неотъемлемой части интеллектуального богатства современного общества. Таким образом, можно констатировать, что формирование алгоритмического стиля мышления является элементом методологической взаимосвязи и гармонизации трех названных выше подходов к подготовке и повышению квалификации кадров. Даже если учебный курс ставит достаточно утилитарную задачу ликвидации компьютерной неграмотности специалистов, то и в этом случае развитие алгоритмического стиля мышления вносит определенный вклад в становление информационной культуры специалистов, подготовку их к жизнедеятельности в информационном обществе. [А.И.Смольникова].

2.3 Организационные аспекты занятий в компьютерном классе

Для каждого школьного кабинета ИВТ существуют общепринятые санитарно-гигиенические требования. ниже приводятся требования СанПиН.

2.3.1 Санитарно-гигиенические требования

1. Помещения кабинета ИВТ должны иметь естественное и искусственное освещение в соответствии с СанПиН 2.2.2.542-96.
2. Основной поток естественного света должен быть слева. Ориентация оконных проемов должна быть на север или на северо-восток. Не допускается направление основного светового потока естественного света сзади и спереди работающего на ПЭВМ. При двухстороннем освещении при глубине помещения кабинета более 6м обязательно устройство правостороннего подсвета, высота которого должна быть не менее 2,2м от пола.
3. В осветительных установках кабинета ИВТ должна быть использована система общего освещения, выполненная потолочными или подвесными люминесцентными светильниками, равномерно размещенными по потолку рядами в виде сплошных линий с двух сторон от рабочего стола с ПЭВМ или ВДТ. Светильники, а также оконные светопроемы не должны отражаться на экранах ПЭВМ или ВДТ.
4. 2.10.1.4. Освещенность поверхности ученических столов при искусственном освещении должна быть в пределах 300-500 лк. Светильники должны иметь светорассеивающую арматуру.

5. В качестве источников света рекомендуется использовать люминесцентные лампы мощностью 40Вт, 58Вт или энергосберегающие мощностью 36Вт типа ЛБ, ЛХБ как наиболее эффективные и приемлемые с точки зрения спектрального состава.
6. Для учебных помещений с ПЭВМ и ВДТ следует применять светильники серии ЛПО36 с высокочастотными пускорегулируемыми аппаратами (ВЧПРА). Можно допустить применение светильников без ВЧПРА в модификации "кососвет".
7. В помещениях с ПЭВМ по причине загрязнения воздуха антропогенными веществами органической природы и диоксидом углерода рекомендуется иметь приточно-вытяжную вентиляцию, обеспечивающую оптимальный температурно-влажностный режим для всех климатических зон.
8. При отсутствии приточно-вытяжной вентиляции можно организовать кондиционирование воздуха с помощью бытовых кондиционеров.
9. Расчет кондиционеров должен быть проведен инженером по вентиляции в зависимости от их производительности, количества теплоизбытков от машин, людей, солнечной радиации и источников искусственного освещения.
10. Кабинет ИВТ должен быть оборудован умывальником с подводкой горячей и холодной воды.
11. Электроснабжение кабинета должно быть выполнено в соответствии с требованиями ГОСТ 28139-89 и ПУЭ.
12. Подводка электрического напряжения к столам обучающихся и учителя должна быть стационарной и скрытой.
13. Расположение электрощита и Устройства защитного отключения должно давать учителю возможность мгновенного отключения системы электроснабжения. Рекомендуемое размещение - слева или справа от классной доски.
14. Для обеспечения пожарной безопасности кабинет МВТ должен быть укомплектован 2-мя углекислотными огнетушителями (типа ОУ-2).
15. Для окраски стен и панелей должны быть использованы светлые тона красок ($\rho=0,5-0,6$). Состав красок должен исключать возникновение известковой пыли.
16. Поверхности ограждающих конструкций кабинета, классной доски, рабочих столов должны быть матовыми.
17. Поверхность пола должна быть ровной, без выбоин, нескользкой, удобной для очистки и влажной уборки, обладать антистатическими свойствами.

18. Содержание вредных химических веществ в воздухе помещений с использованием видеодисплейных терминалов (ВДТ) и персональных электронно-вычислительных машин (ПЭВМ) не должно превышать среднесуточных концентраций для атмосферного воздуха.
19. Для внутренней отделки интерьера помещений с ПЭВМ и ВДТ не разрешается применять синтетические материалы, выделяющие в воздух вредные химические вещества и соединения. К ним можно отнести древесно-стружечные плиты, слоистый бумажный пластик, моющиеся обои, рулонные синтетические покрытия и др.
20. Уровень шума на рабочем месте во всех учебных помещениях с ВДТ и ПЭВМ не должен превышать 50 дБА. (Санитарные нормы допустимого шума в помещениях жилых и общественных зданий и на территории жилой застройки N 3077-84, п.7.2).

2.3.2 Требования к помещениям кабинета ИВТ

1. Кабинет информатики и вычислительной техники (МВТ) организуется как учебно-воспитательное подразделение средней общеобразовательной и профессиональной школы, учебно-производственного комбината, оснащенное комплектом учебной вычислительной техники (КУВТ), учебно-наглядными пособиями, учебным оборудованием, мебелью, оргтехникой и приспособлениями для проведения теоретических и практических, классных, внеклассных занятий по курсу "Основы информатики и вычислительной техники" (ОИВТ) как базовому, так и профильным. Кроме того, КИВТ может использоваться в преподавании различных учебных предметов, трудовой подготовки.
2. Площадь помещений кабинета ИВТ определяется в соответствии с требованиями нормативного документа "Учебно-материальная база образовательного учреждения общего среднего образование" ч. I. "Нормы и требования к учебным зданиям и пришкольным участкам", а также СанПиН 2.2.2.542-96.
3. Размещение КИВТ во всех учебных заведениях в цокольных и подвальных помещениях не допускается.
4. Минимальная площадь, приходящаяся на одну ПЭВМ, должна быть не менее 6 кв.м., а объем - не менее 24,0 куб.м. при высоте не менее 4 м. При меньшей высоте учебного помещения рекомендуется увеличить площадь на одно рабочее место.

5. При кабинете ИВТ должна быть организована лаборантская площадью не менее 18 кв.м. Лаборантское помещение должно иметь два выхода: в учебное помещение и на лестничную площадку или в рекреацию.
6. Площадь кабинета должна позволять расставить в нем мебель с соблюдением санитарно-гигиенических норм.
7. Передняя стена КИВТ оборудуется классной доской для фломастеров, экраном, шкафом для хранения учебно-наглядных пособий и носителей информации.
8. При входе в кабинет ИВТ должны быть предусмотрены встроенные или пристенные шкафы (полки) для портфелей.
9. Слева от доски, в рабочей зоне учителя, на стене должен быть закреплен электрораспределительный щит с пультом управления электроснабжением рабочих мест учителя и учащихся.
10. Под доской или отдельно под стендами устанавливают ящики для таблиц. На верхней кромке доски крепятся держатели (или планка с держателями) для подвешивания таблиц.
11. На стене, противоположной окнам, размещаются экспозиционные щиты с постоянной и временной информацией.
12. Вдоль задней стены возможна установка секционного шкафа для хранения учебного оборудования и носителей информации в зависимости от площади кабинета.
13. Верхняя часть задней стены кабинета должна быть предназначена для экспонирования пособий, необходимых для изучения отдельных тем программы.

2.4 Разработка уроков

- Раздел программы: «Основы алгоритмизации и программирование».
- Тема урока: «Алгоритмическая структура «Цикл» в VBA. Блок - схемы циклов со счетчиком, предусловием и постусловием».
- Тип урока: урок-лекция.
- Вид: комбинированный урок.

Цель урока:

1. Образовательная: научиться работать с алгоритмической структурой «Цикл» в среде VBA, строить блок-схемы циклов со счетчиком, предусловием и постусловием, решать задачи с использованием данных циклов.

2. Воспитательная: развитие научного мировоззрения; формирование навыков работы в группе.
3. Развивающая: развитие единого подхода к изучению информационных процессов в живой природе, обществе, технике, обоснование общности процессов восприятия, эффективного оперирования наглядными образами, быстрого установления смысловой связи между ними.

Ход урока:

I. Организационный момент.

Проверка домашнего задания.

1. Изучение нового материала с использованием презентации, выполненной в Power Point «Циклы».

[1 слайд] **Цикл-это многократное повторение последовательности действий по некоторому условию.**

Различают три вида циклов:

- ✓ **Цикл с параметром;**
- ✓ **Цикл с предусловием;**
- ✓ **Цикл с постусловием.**

[2 слайд] **Основные понятия:**

- **Управляющая переменная цикла**
- **Начальное значение управляющей переменной**
- **Конечное значение управляющей переменной**
- **Шаг цикла**
- **Тело цикла**

[3 слайд] **Цикл с параметром (счетчиком)**

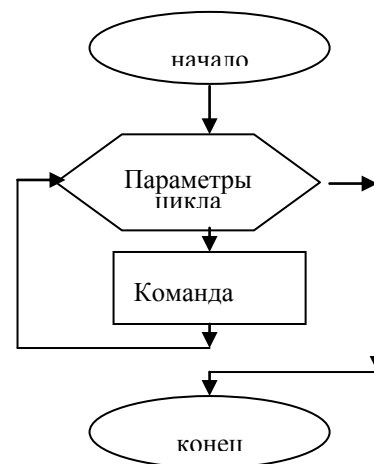
Счетчик – управляющая переменная

Начало – начальное значение счетчика (управляющей переменной)

Конец – конечное значение счетчика (управляющей переменной)

Шаг – шаг изменения значения счетчика (по умолчанию 1)

Блок-схема для алгоритма цикл с параметром:



Операторы цикла с параметром (для VB):

For счетчик = начало **To** конец [**Step** шаг]

Next счетчик

[4 слайд] **Циклы с параметром. Задача:** Найти сумму цифр целого неотрицательного числа. Для решения данной задачи используется алгоритм цикла с параметром. На первом этапе определяется число символов в строке. Затем организуется цикл, начальным и конечным параметрами которого являются i и k символ, а в теле цикла находится сумма цифр заданного числа: $S=S+\text{val}(\text{mid}(N,i,1))$. На втором этапе приводится блок-схема алгоритма представленной задачи.

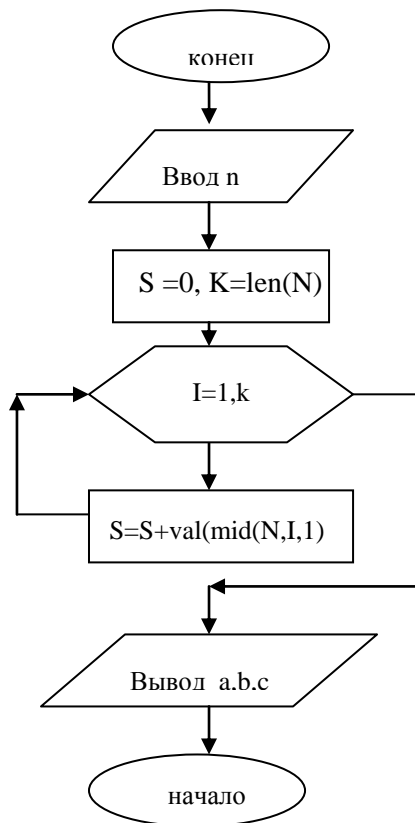


Рис. 5. Блок-схема к задаче «Найти сумму цифр целого неотрицательного числа». На третьем этапе, приводится программный код:

```
Dim n As String
Dim s As Single
Dim k As Single
Dim i As Single
Private Sub CommandButton1_Click()
n = TextBox1.Text
```

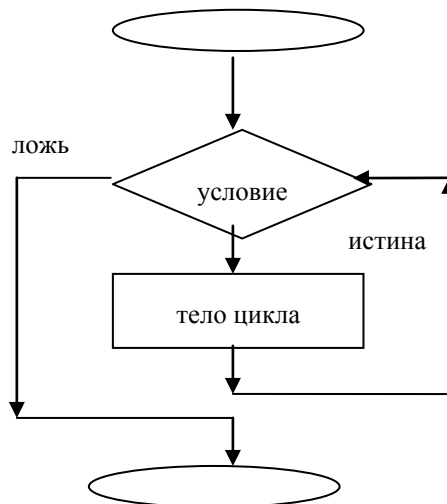
```

k = Len(n)
s = 0
For i = 1 To k
    s = s + Val(Mid(n, i, 1))
Next i
TextBox2.Text = Str(s)
End Sub

```

[5 слайд] **Цикл с предусловием «Пока».** Цикл с предусловием используется в том случае, когда число шагов цикла точно не определено, а выход из цикла осуществляется при проверке истинности условия. Если условие примет значение «ложь», то выполнение цикла закончится.

Блок-схема алгоритма:



Операторы цикла с предусловием (2 варианта)

Do While Условие

Тело цикла

[Exit Do]

Loop

Do Until Условие

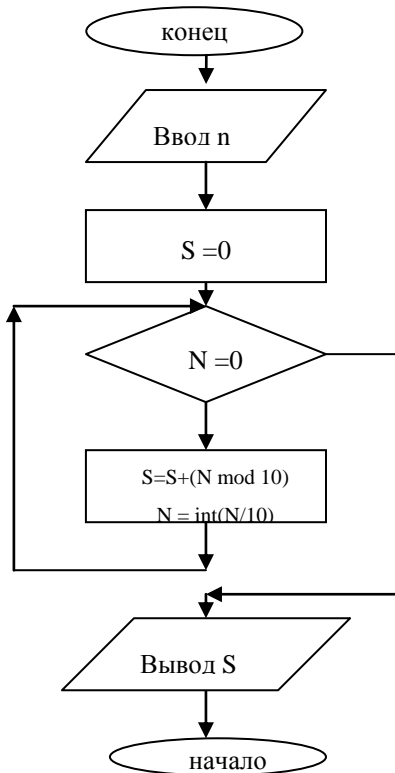
Тело цикла

[Exit Do]

Loop

[6 слайд] **Циклы с предпроверкой условия. Задача:** Найти сумму цифр целого неотрицательного числа. Данную задачу можно также решить с помощью циклического алгоритма с предпроверкой условия. Пока количество цифр в числе не станет равным нулю, выполняется деление на десять, выделение целой части числа, а также расчет суммы цифр в числе.

Блок-схема к задаче «Найти сумму цифр целого неотрицательного числа»:



Для данного способа решения приводится фрагмент программного кода:

```

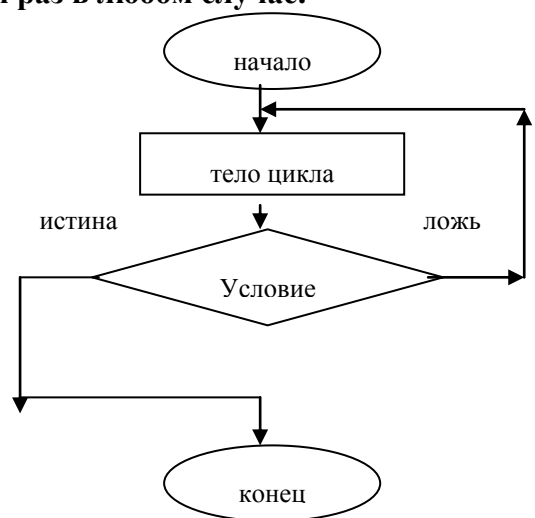
Dim n as single
Dim s as single
n=val(TextBox1.text)
s=0
Do Until n=0
  s=s+(n mod 10)
  n=int(n/10)
Loop
TextBox2.text=str(s)
  
```

[7 слайд] Цикл с постусловием «До»

Цикл с постусловием используется в том случае, когда число шагов цикла точно не определено, а выход из цикла осуществляется при проверке истинности условия. Если условие примет значение «истина», то выполнение цикла закончится.

Цикл с постусловием выполняется хотя бы один раз в любом случае.

Блок-схема алгоритма:

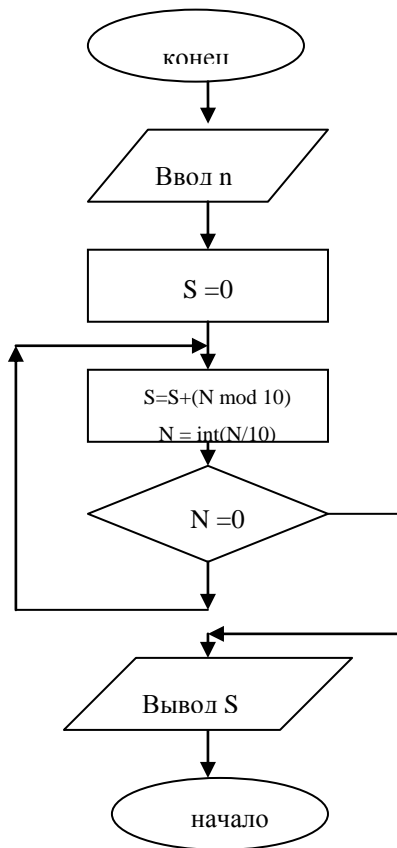


Операторы цикла с постусловием (2 варианта)

Do	Do
Тело цикла	Тело цикла
[Exit Do]	[Exit Do]
Loop While Условие	Loop Until Условие

[8 слайд] **Циклы с постпроверкой условия.** Задача: Найти сумму цифр целого неотрицательного числа. Поставленную задачу можно решить, применяя циклический алгоритм с постпроверкой условия. Указанное число делится на десять, выделяется целая

часть числа, а также производится расчет суммы цифр до тех пор, пока количество цифр в числе не станет равным нулю. Блок-схема данного способа решения.



Циклический алгоритм с постпроверкой условия. Для этого способа решения приводится фрагмент программного кода:

```

Dim n as single
Dim s as single
n=val(TextBox1.text)
s=0
Do
  s=s+(n mod 10)
  n=int(n/10)
Loop Until n=0
Text2.text=str(S)
  
```

[9 слайд] **Функции для работы со строковыми переменными.**

<i>функция</i>	<i>действие</i>	<i>Тип возвращаемого значения</i>
Len(a)	длина строковой переменной a	число
Left(a,k)	Вырезать слева k символов	string
Right(a,k)	Вырезать справа k символов	string
Mid(a,i,k)	Вырезать начиная с i символа в количестве k	string

III. Практическая работа (отработка ЗУН по данной теме)

Далее предлагаются задания для работы в классе. Можно предложить учащимся совместное решение задач (в группах по 2 человека), либо поработать индивидуально.

а) Для начала учащимся предлагается выполнение проектов в среде VB с использованием готовых программных кодов, приведенных в презентации.

б) Затем совместное или в группах решение задач, представленных на слайдах №10 (задача №4) и №11 (задача №4). Составляем блок-схемы к приведенным задачам и программный код.

[10 слайд] *Задачи* для самостоятельного решения.

1. Начальный вклад в сбербанк составил A рублей под P процентов годовых. Через сколько лет он станет больше B рублей?
2. Сбербанк начисляет P процентов годовых. Какой станет сумма в рублях A , положенная на N лет?
3. Одноклеточная амеба каждые три часа делится на 2 клетки. Сколько клеток будет через 5 дней?
4. В первый день тренировок спортсмен пробежал 10 км. В каждый следующий день он увеличивал норму на 10% от предыдущего дня. Через сколько дней он будет пробегать ежедневно больше 20 км? Какое расстояние он пробежит за 10 дней? Через сколько дней он пробежит суммарный путь 100 км?

[11 слайд] *Задачи* (циклы с параметром).

$$1. S = \frac{1}{2} + \frac{1}{4} + \frac{1}{6} + \dots + \frac{1}{n}$$

$$2. S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots + \frac{1}{n}$$

$$3. S = 1 + \frac{1}{2} - \frac{1}{3} + \frac{1}{4} - \frac{1}{5} + \dots + \frac{-1^n}{n}$$

$$4. S = 1 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \dots + \frac{1}{n!}$$

$$5. S = \sqrt{2 + \sqrt{2 + \sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}}}$$

IV. Закрепление знаний, умений, навыков. Домашнее задание: § 3.2, 3.2.2 стр. 117-123. Составить блок-схемы к задачам: №1 и №2 (слайд 10) и №3 (слайд 11).

Выводы по главе 2

Известно, что информатика (computer science) является не только естественнонаучной (теория и практика), но и инженерной (технология) дисциплиной. Она требует активного обучения и приложениям. Поэтому в выборе стратегий обучения информатика имеет большие степени свободы, чем другие традиционные дисциплины.

Если анализировать эволюцию учебных планов по информатике, рассматривать существующие международные рекомендации, то можно заметить, что в последние годы происходят определенные изменения в подходах к обучению информатике в средних и высших учебных заведениях. Эти изменения явно прослеживаются, например, в методике изучения одного из основных разделов информатики – алгоритмизации и программирования. Вообще, при изучении этого раздела можно применить два подхода: *алгоритмический* и *технологический*. Первый из них основывается на принципе “Сначала думать, а потом делать” – программирование как творческая активность. Второй поддерживает стиль “Сначала попробовать, а затем посмотреть, что же получилось” – программирование как ремесло. (С.С. Кобиров, Самаркандский государственный университет).

В последнее время при изучении информатики главное внимание уделяется технологическому подходу. Это означает, что основной упор делается на знание инструмента (текстовые и графические редакторы, электронные таблицы, интегрированные пакеты, программы браузеры), а не на умение грамотно решить поставленную задачу. Такое усиление технологического подхода связано с внедрением информатики в другие предметы (как математики на все остальные) и его применение пригодно в основном при подготовке пользователей ЭВМ, т.е. специалистов различных предметных областей. Однако, технологический подход мало пригоден для профильной подготовки учащихся. Это обосновывается следующими соображениями.

Во-первых, для профессиональной подготовки основное внимание необходимо уделять выработке алгоритмического мышления обучаемого.

Во-вторых, при изучении и применении готовых программных продуктов главным является знание синтаксиса пользовательского интерфейса, но и есть семантическая сторона – теоретические знания и практические навыки.

В-третьих, специалист должен не только изучить и применить чужие инструменты, но и создать свои, а для этого нужно разработать алгоритмы и программы.

Из сказанного следует, что для непрерывной (школа, колледж, ВУЗ) профессиональной подготовки учащихся необходимо применить алгоритмический подход и обучать их конструированию алгоритмов (программ) методично и систематически. Отсюда вытекает еще одна важная задача. Обучаемый намного быстрее и легче разберется в каком либо языке, если ему дать возможность самому составить блок-схему алгоритма, посмотреть как он (алгоритм) будет выполняться, проследить изменения значений переменных, а затем посмотреть, как выглядит исходный текст непосредственно на изучаемом языке.

Наиболее естественной формой представления (восприятия) информации является графический образ – рисунок, чертеж, схема и т.д. К этой форме человек прибегает всякий раз (возможно неявно для себя), когда необходимо решать (описывать, формулировать) действительно сложные задачи. Эффективное оперирование наглядными образами, быстрое установление смысловой связи между ними – является сильной стороной человеческого мышления.

Еще во времена становления программирования, когда программы писались на внутреннем языке ЭВМ – машинном коде (ассемблере), неотъемлемой частью разработки программ было использование блок-схем. Их применение значительно облегчало восприятие и анализ программы. Двумерное представление программы более ясно отражало ее структуру. Применение блок-схем позволяло быстрее и качественнее разрабатывать и отлаживать программы, а также облегчалось их сопровождение.

Сохранение двух различных форм представления программ – самого текста и блок-схемы необходимо для лучшего изучения темы «Алгоритмы и алгоритмические структуры». Естественным развитием данной ситуации является объединение двух подходов в описании программ: в виде текста и блок-схемы. Результатом такого объединения является понятие визуального программирования. Под ним понимается способ описания алгоритма решения задачи в графическом виде, соединяющий достоинства текста и блок-схем программ.

Глава 3. Организация и результаты эксперимента

3.1 Цель и гипотеза эксперимента

Цель эксперимента: проверка эффективности предполагаемого содержания и разработанной педагогической методике в экспериментальном курсе «Алгоритмы и алгоритмические структуры» по сравнению со стандартным базовым курсом для подготовки школьников к применению данной темы во время сдачи экзаменов, ЕГЭ и в будущей профессиональной деятельности.

Гипотеза эксперимента заключалась в следующем:

Применение научно-обоснованного содержания и методики подготовки школьников:

- повысит уровень подготовки учащихся к применению данной темы в развитии алгоритмического мышления;
- познакомит учащихся с различными аспектами применения алгоритмов в социальной жизни;

- сформирует профессиональную готовность к использованию алгоритмов;
- увеличит практическую направленность и творческую активность учащихся по применению алгоритмического мышления в будущей профессиональной деятельности.

Результаты эксперимента были обработаны с помощью методов математической статистики [Грабарь М.И 21], [Ительсон Л.Б. 35], [Маслак], [Немов Р.С. 71], [Нурмухамедов], [Сироткина].

Обработка результатов велась по 4-м этапам (I-IV).

3.2. Доступность теста и сравнение знаний групп

В ходе эксперимента проводилось тестирование для определения уровня подготовки алгоритмического мышления в будущей профессиональной деятельности.

Тест из 22-ух вопросов «Определение уровня подготовки школьников к применению алгоритмического мышления», входной контроль (до изучения темы «Алгоритмы и алгоритмические структуры»).

Определим доступность этого теста из 22-х вопросов.

Для контрольной группы:

Таблица 1. Результаты теста.

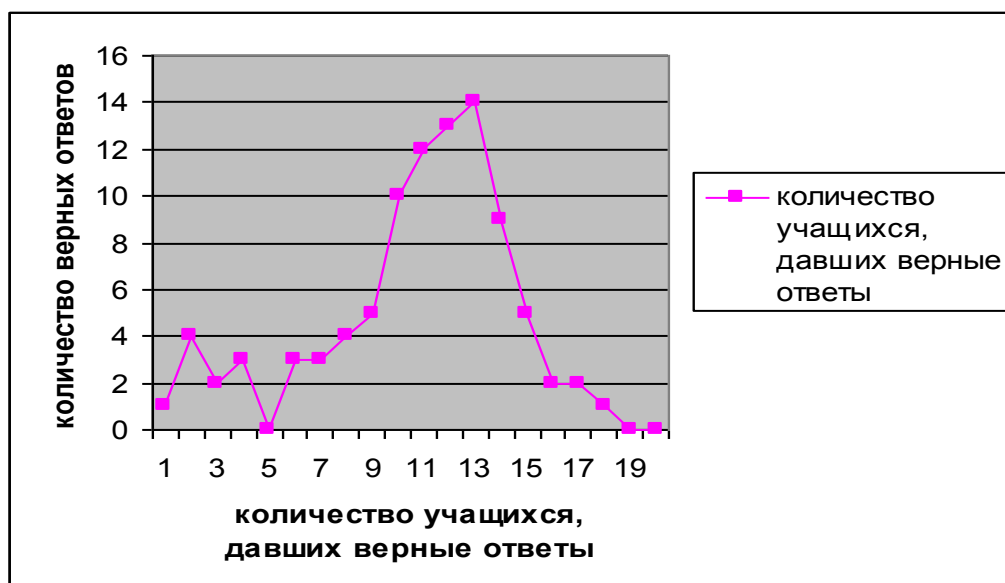
I	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
II	1	4	2	3	0	3	3	4	5	10	12	13	14	9	5	2	2	1	0	0

Здесь строка

I - количество правильных ответов;

II - количество учащихся, давших правильные ответы.

Соответствующий график 1 «Результаты теста в контрольной группе».

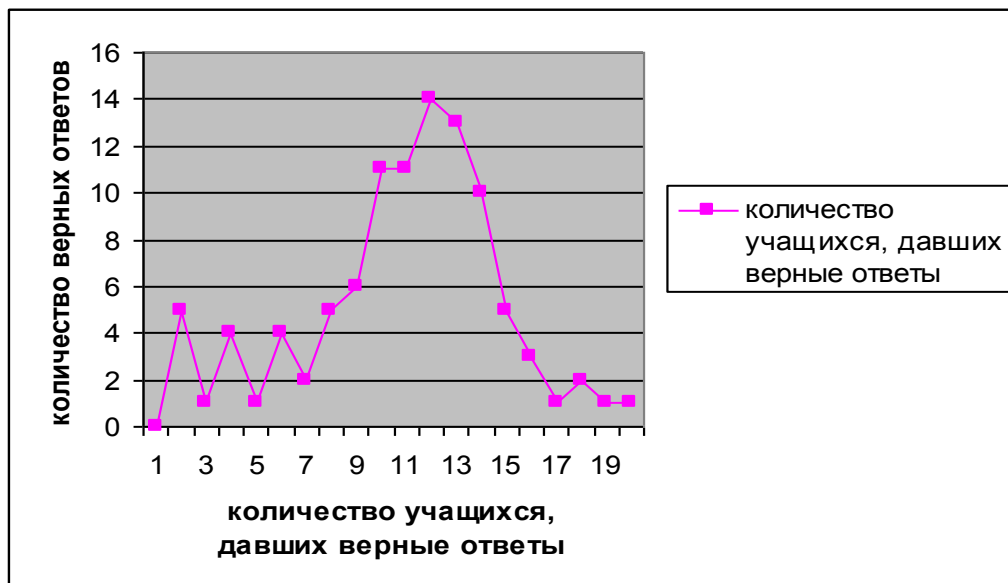


Для экспериментальной группы:

Таблица 2. Результаты теста.

I	3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
II	0 5 1 4 1 4 2 5 6 11 11 14 13 10 5 3 1 2 1 1

Соответствующий график 2 «Результаты теста в экспериментальной группе».



Тест из 22-ух вопросов «Определение уровня подготовки школьников к применению алгоритмического мышления», выходной контроль (после изучения темы «Алгоритмы и алгоритмические структуры»):

Определим доступность этого теста из 22-х вопросов.

Для контрольной группы:

Таблица 1. Результаты теста.

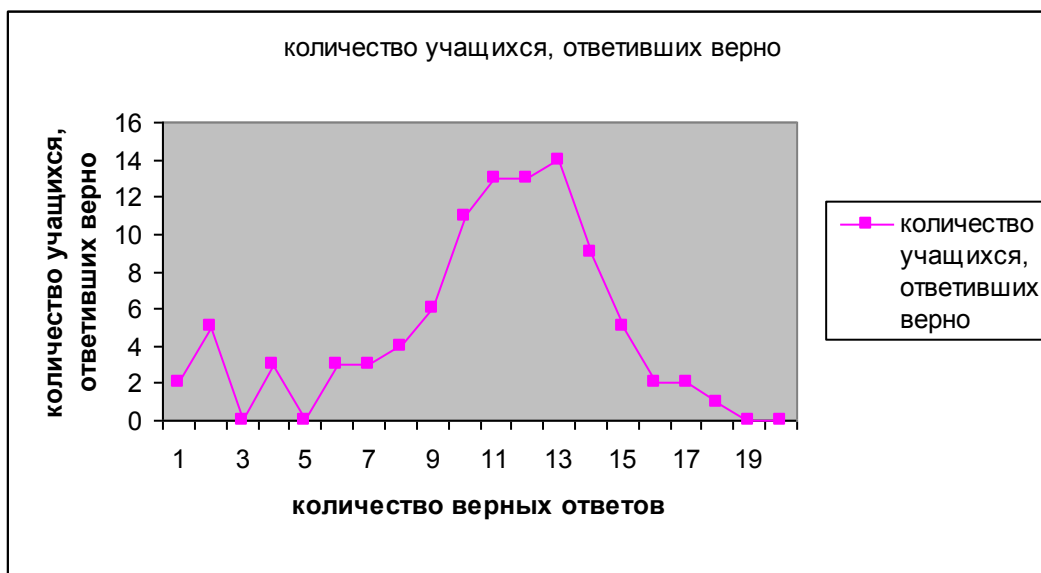
I	3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
II	2 5 0 3 0 3 3 4 6 11 13 13 14 9 5 2 2 1 0 0

Здесь строка

I - количество правильных ответов;

II - количество учащихся, давших правильные ответы.

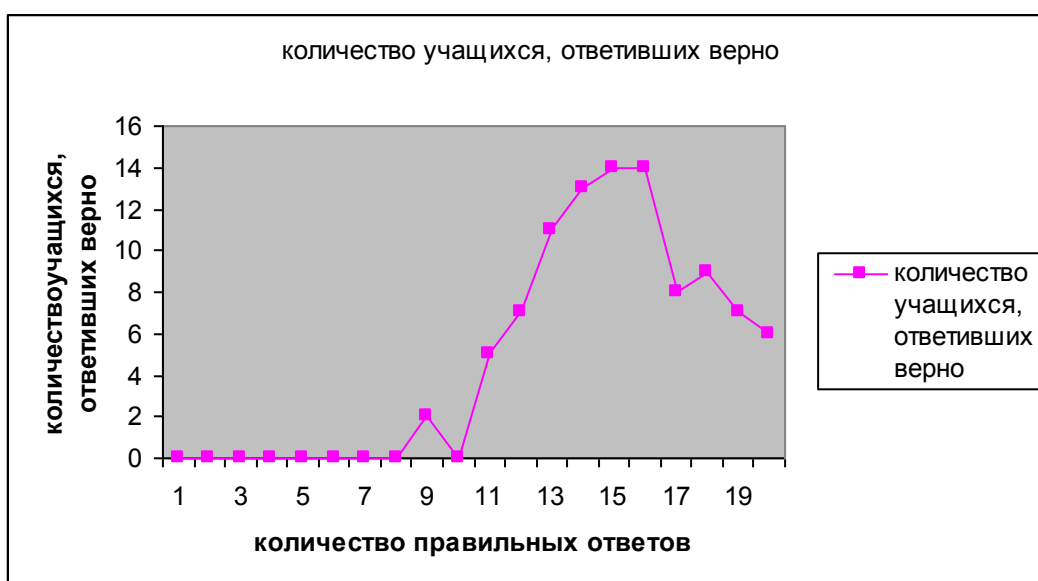
Соответствующий график 3 «Результаты теста в контрольной группе».



Для экспериментальной группы:

Таблица 2. Результаты теста.

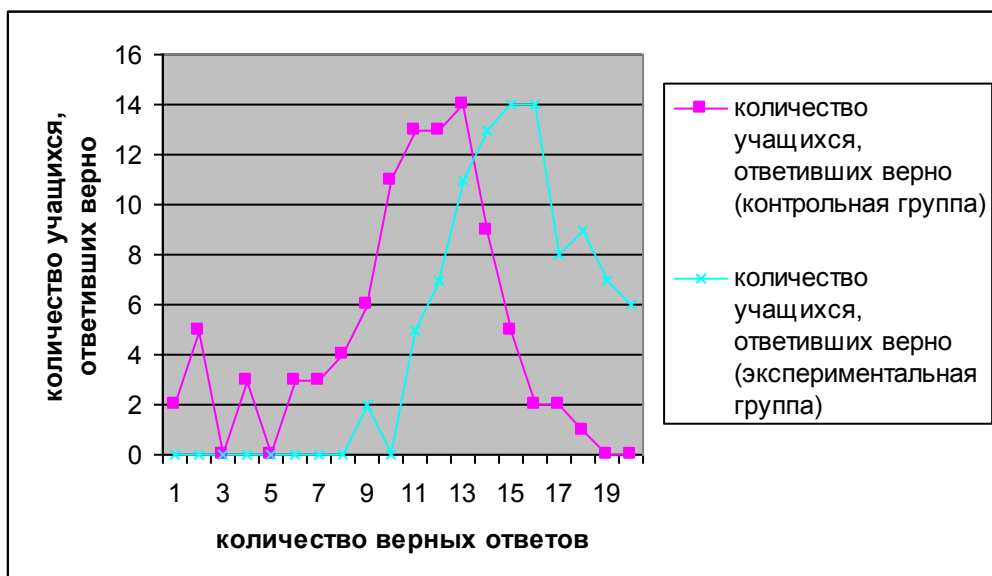
I	3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
II	0 0 0 0 0 0 0 0 0 2 0 5 7 11 13 14 14 8 9 7 6



Соответствующий график 4 « Результаты теста в экспериментальной группе».

Определение уровня подготовки учащихся проводилось в двух группах: в экспериментальной и в контрольной группах, не изучающих экспериментальный курс (проводился только традиционный курс "Основы алгоритмизации и программирование"). В экспериментальной группе уровень верных ответов колебался от 62% до 97%, тогда как в контрольной - от 14% до 66%.

График сравнения количества верных ответов в экспериментальной и контрольной группах.



Проверим нулевую гипотезу: изучение курса "Основы алгоритмизации и программирование" стохастически [Грабарь М.И. с.27] увеличивает количество верных ответов на тест и, таким образом, дает лучшую подготовку по данной теме.

Кроме того, были рассчитаны **описательные** статистические параметры из п.3 [Жорн., с.545] – в таблице ниже.

<i>Статистические параметры</i>	<i>количество учащихся, давших верные ответы в контрольной группе</i>	<i>Статистические параметры</i>	<i>количество учащихся, давших верные ответы в экспериментальной группе</i>
Среднее	4,8	Среднее	4,8
Стандартная ошибка	1,047804723	Стандартная ошибка	1,180098123
Медиана	3	Медиана	3,5
Мода	0	Мода	0
Стандартное отклонение	4,685925174	Стандартное отклонение	5,277559244
Дисперсия выборки	21,95789474	Дисперсия выборки	27,85263158
Экссесс	-0,460936728	Экссесс	-1,152490979
Асимметричность	0,917593885	Асимметричность	0,580713865
Интервал	14	Интервал	14
Минимум	0	Минимум	0
Максимум	14	Максимум	14
Сумма	96	Сумма	96
Счет	20	Счет	20

Наибольший(1)	14	Наибольший(1)	14
Наименьший(1)	0	Наименьший(1)	0
Уровень надежности(95,0%)	2,193080484	Уровень надежности(95,0%)	2,469973752
<i>корреляция</i>	<i>Столбец 1</i>	<i>Столбец 2</i>	
Столбец 1	1		
Столбец 2	0,260444473	1	

Ниже по наиболее чувствительному критерию Вилкоксона-Манна-Уитни [Грабарь М.И. с.83, 87-89; 122] была рассчитана степень различия уровней подготовки у школьников контрольной и экспериментальной групп. Этот критерий предназначен для выявления различий в распределении изучаемого свойства у объектов двух совокупностей на основе сравнения результатов изучения данного свойства у членов независимых выборок, сделанных у этих совокупностей по шкале порядка [Грабарь М.И. с.16, 19,118].

Пусть случайная переменная x - число правильных ответов студентов первой выборки (экспериментальной), а случайная переменная y - число правильных ответов второй выборки (контрольной). Объем первой выборки $n_1=15$, объем второй $n_2=10$, т.е. объем выборок небольшой [Грабарь М.И. с.118].

Таким образом, имеем две серии независимых [Грабарь М.И. с.39] наблюдений над случайными переменными x и y со значениями, равными количеству верных ответов:

x : 15,17,16,15,12,12,10,11,19,14,17,10,14,16,19.

y : 8,4,12,13,12,17,6,16,9,10.

Упорядочим выборки по невозрастанию:

Экспериментальная группа x_i : 19,19,17,17,16,16,15,15,14,14,13,13,11,10,10.

Контрольная группа y_j : 17,16,13,12,12, 10,9,8,6,4.

Объединим обе серии наблюдений в одну выборку с количеством членов $N=25$, члены этой выборки запишем в ряд по убыванию значений. Затем этот ряд ранжируем, т.е. припишем каждому значению x_i и y_j ранг R_m , численно равный месту, на котором стоит это значение в ряду; укажем $K_m > 1$ - количество одинаковых рангов. Для одинаковых значений x и y , ранг будет дробный при четном K_m и одинаков и, поэтому первоочередность x или y не играет роли.

Значение переменных x и y , и соответствующий им ранг R и количество одинаковых рангов запишем в форме таблицы 2.

Таблица 2. «Ранжирование объединённой выборки экспериментальной (x) и контрольной (y) групп».

№	x_i	y_j	R_m	K_m	№	x_i	y_j	R_m	K_m
1.		4	1		14.	14		14.5	
2.		6	2		15.	14		14.5	2
3.		8	3		16.	15		16.5	
4.		9	4		17.	15		16.5	2
5.		10	6	3	18.	16		19	
6.	10		6		19.	16		19	
7.	10		6		20.		16	19	3
8.	11		8		21.	17		22	
9.		12	9.5	2	22.		17	22	
10.		12	9.5		23.	17		22	3
11.	13		12	3	24.	19		24.5	
12.	13		12		25.	19		24.5	2
13.		13	12						

На основе данных таблицы 2 подсчитаем значение статистики критерия T по формуле [Грабарь М.И. с.93];

$$T = S - \frac{n \cdot (n+1)}{2}, \text{ где } n = \min(n_1, n_2) = 10:$$

где S - сумма рангов, приписанных членам только y-выборки (меньшего объема): n

$$S = \sum_{i=1} R_i .$$

Для таблицы 2 имеем:

$$S = 88; T = 33.$$

Предположим, что значение переменной x в среднем выше, чем значение переменной y. Для доказательства этой гипотезы используем односторонний критерий [Грабарь М.И. с.87-89]. Проводим отклонение на уровне значимости $\alpha=0,05$ нулевой гипотезы H_0 : вероятность $P(x < y) \geq 0,5$ (при наблюдаемом значении $T > W_\alpha$), либо принятие при альтернативе H_1 : $P(x > y) \geq 0,5$ с достоверностью $1-\alpha=0,95$ (при $T < W_{1-\alpha}$),

$$W_\alpha = \frac{n_1 \cdot n_2}{2} + x_\alpha \cdot \sqrt{\frac{n_1 \cdot n_2 (n_1 + n_2 + 1) - \sum (K^3 - K)}{12}},$$

где $X_{0,05} = 1,64$, $K_m = (3,2,3,2,2,3,3,2)$ – число членов ряда с одинаковым рангом R_m ; $KR=8$ – их общее количество; сумма $\sum (K^3 - K)$ по всем 8-ми одинаковым R_m $\Sigma=24+6+24+6+6+24+24+6 = 120$, $W_\alpha = 104.107$ при $\alpha = 0,05$, $W_{1-\alpha} = n_1 n_2 - W_\alpha = 150 - 104.107 = 45.893$, т.е. $T = 33 < 45.893 = W_{1-\alpha}$.

Следовательно, с достоверностью не ниже 95% значение результатов тестирования в экспериментальной группе выше, чем в контрольной группе.

3.3 Критерии оценки итоговых программ на VBA

Одной из основных целей проведения курса «Основы алгоритмизации и программирование» была развитие алгоритмического мышления с целью дальнейшего применения при решении алгоритмических задач, построения к ним блок-схем и написания программ, а так же применения в будущей профессиональной деятельности.

Для измерения критериев оценки алгоритмического мышления был произведен анализ качества знаний учащихся, составлена таблица сравнения усвояемости основных понятий курса «Основы алгоритмизации и программирование» в контрольной и экспериментальной группах.

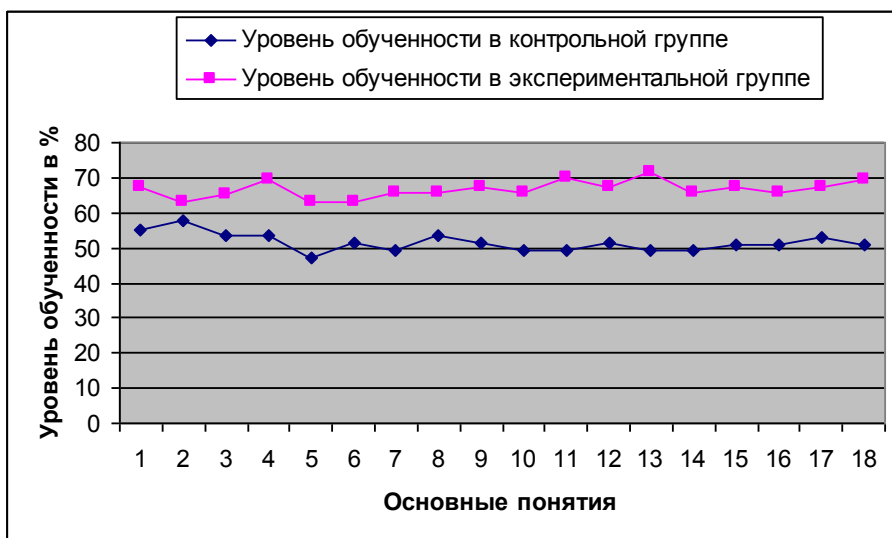
Таблица сравнения усвоения основных понятий курса «Основы алгоритмизации и программирование» в контрольной и экспериментальной группах.

Основные понятия	Количество учащихся, получивших оценки (группы по 15 человек, всего 30 человек)							
	Контрольная				Экспериментальная			
	«5»	«4»	«3»	«2»	«5»	«4»	«3»	«2»
Основные алгоритмические структуры	1	8	6	0	3	10	2	0
Линейные алгоритмы	2	7	6	0	2	10	3	0
Объект Image, его свойства, свойства объекта Label.	1	7	7	0	2	11	2	0
Арифметические, строковые и логические выражения	1	7	7	0	3	11	1	0
Объявление переменных и присваивание им значений. Датчик случайных чисел	0	6	9	0	2	10	3	0
Алгоритмическая структура «Ветвление» в VBA.	1	6	8	0	2	10	3	0
Понятие функции в языке VBA. Математические и строковые функции	0	7	8	0	3	9	3	0
Функции преобразования типов данных. Функции даты и времени.	1	7	7	0	3	9	3	0
Алгоритмическая структура «Вложенное ветвление» в VBA.Создание проекта	1	6	8	0	3	10	2	0
Метки и их свойства. Множественный выбор Case.	0	7	8	0	3	9	3	0
Алгоритмическая структура «Цикл» в VBA. Блок-схемы циклов со счетчиком, предусловием и постусловием.	0	7	8	0	4	9	2	0
Создание диалоговых программ.	1	6	8	0	3	10	2	0
Вспомогательный алгоритм. Событийные и общие процедуры. Область видимости процедур	0	7	8	0	4	10	1	0
Понятие массива. Работа с массивами. Заполнение массива, поиск в массивах	0	7	8	0	3	9	3	0
Сортировка массива	0	8	7	0	3	10	2	0
Двумерные массивы и вложенные циклы	0	8	7	0	3	9	3	0
Вспомогательный алгоритм и массивы	0	9	6	0	3	10	2	0
Решение задач формата ЕГЭ	0	8	7	0	3	11	1	0

Рассчитаем качество знаний и уровень обученности по каждому основному понятию в контрольной и экспериментальной группах, построим графики и проанализируем результат.

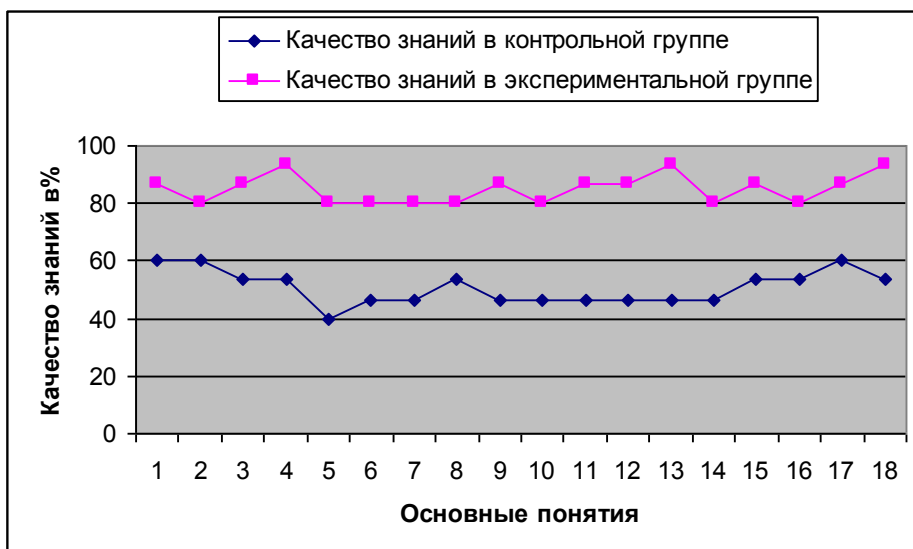
Следовательно, уровень обученности и качество знаний в экспериментальной группе выше, чем в контрольной. Полученные данные являются доказательством развития алгоритмического мышления у учащихся экспериментальной группы в результате применения данной авторской методики «Основы алгоритмизации и программирование».

График сравнения уровня обученности в контрольной и экспериментальной группах



Средний уровень обученности в экспериментальной группе - 66,68%, а в контрольной – 51,40%. Среднее значение качества знаний в экспериментальной группе – 84,81%, а в контрольной – 50,74%. Следовательно, в среднем качество знаний и уровень обученности выше в экспериментальной по сравнению с контрольной.

График сравнения качества знаний в контрольной и экспериментальной группах



3.4 Отношение к ИКТ

Поскольку одной из основных целей проведения курса "Алгоритмы и алгоритмические структуры" было научить учащихся практически применять алгоритмическое мышление в будущей профессиональной деятельности, одним из важных показателей эффективности предлагаемого курса является критерий сформированности практической направленности знаний школьников по их использованию в дальнейшей работе. К критериям сформированности практической направленности знаний учащихся относятся:

- положительное отношение школьников к учебным занятиям по курсу "Алгоритмы и алгоритмические структуры" и отдельным ее формам,
- удовлетворенность студентов знаниями, полученными на занятиях и осознание их необходимости;
- деятельное отношение школьников к овладению знаниями, умениями и навыками по использованию алгоритмического мышления в дальнейшей деятельности, выборе профессии;
- интерес и удовлетворенность учащихся самостоятельной работой в ходе изучения курса "Алгоритмы и алгоритмические структуры";
- самооценка школьников своей самостоятельной работы по курсу "Алгоритмы и алгоритмические структуры";
- отношение учащихся к заданиям.

Для определения критериев практической направленности знаний учащихся была адаптирована соответствующая методика [Термиркулова 107], разработана анкета, позволяющая определять коэффициент практической направленности знаний школьников профильных классов при изучении курса "Алгоритмы и алгоритмические структуры", который определяется на основе оценки положительных поведенческих и эмоциональных отношений школьников к вышеназванному курсу. В этой анкете вопросы № 2,3,5,11 характеризуют положительное отношение школьников к курсу, вопросы № 1,6,9,10 поведенческое отношение, вопросы № 4,7,8 - эмоциональное.

Анкета состоит из двух колонок. В левой колонке содержатся вопросы, раскрывающие содержание практической направленности знаний студентов, в правой - четыре типа ответов на каждый вопрос. Выбор ответов первого типа отражает наиболее положительную самооценку (++), второго - скорее положительную, чем отрицательную (+-), третьего - скорее отрицательную, чем положительную (-+), четвертого типа –

отрицательную (--). Приведем данные анкетирования в виде таблиц, которые показывают количество студентов выбравших данный тип ответа на данный вопрос.

Анкета определения практической направленности знаний школьников при подготовке к применению знаний в учебной деятельности.

№	Вопросы анкеты	Ответы	Тип
1	Нравится ли Вам предмет информатика?	Очень нравится	++
		Скорее нравится, чем не нравится	+-
		Скорее не нравится, чем нравится	-+
		Не нравится	--
2	Нравятся ли Вам занятия?	Очень нравится	++
		Скорее нравится, чем не нравится	+-
		Скорее не нравится, чем нравится	-+
		Не нравится	--
3	Какое значение имеет для Вас занятия в плане подготовки в ВУЗ?	Очень большое	++
		Достаточно большое	+-
		Не достаточное	-+
		Почти никакого	--
4	Доставляют ли Вам удовлетворение занятия?	Очень большое	++
		Достаточно большое	+-
		Недостаточное	-+
		Почти никакого	--
5	Можно ли использовать полученные знания при поступлении в ВУЗ?	Да	++
		Скорее да, чем нет	+-
		Скорее нет, чем да	-+
		Нет	--
6	Много ли времени Вы уделяете изучению достижений в области алгоритмики?	Очень много	++
		Достаточно много	+-
		Недостаточно	-+
		Почти не уделяю	--
7	С интересом ли Вы выполняете контрольные работы?	Делаю сразу же	++
		Делаю постепенно, но в срок	+-
		Делаю за день до сдачи	-+
		Делаю после напоминания	--

8	Чувствуете ли Вы удовлетворение от контрольных работ?	Да	++
		Скорее да, чем нет	+-
		Скорее нет, чем да	-+
		Нет	--
9	Как часто Вам требуется помощь при выполнении контрольных заданий?	Никогда не требуется	++
		Редко	+-
		Скорее часто, чем редко	-+
		Часто	--
10	Как вы относитесь к индивидуальным заданиям?	Очень нравится	++
		Скорее нравится, чем не нравится	+-
		Скорее не нравится, чем нравится	-+
		Не нравится	--
11	Помогают ли Вам занятия изучению других школьных дисциплин?	Да	++
		Скорее да, чем нет	+-
		Скорее нет, чем да	-+
		Нет	--

Таблица 5. Результаты анкетирования (количество ответов 36-и школьников) экспериментальных групп.

Тип ответа № вопроса	I ++	II +-	III -+	IV --
1	11	22	2	1
2	18	10	1	-
3	16	18	2	-
4	24	11	-	1
5	30	6	-	-
6	17	18	1	-
7	16	18	-	2
8	17	16	2	1
9	10	12	9	5
10	13	17	4	2
11	19	13	2	2
	191 (48,2%)	161 (40,6%)	23 (5,8%)	14 (3,5%)

Таблица 6. Результаты анкетирования (количество ответов 36-и школьников) контрольных групп.

Тип ответа № вопроса	I ++	II +-	III -+	IV --
1	11	13	10	2
2	12	9	12	3

3	12	6	9	9
4	13	9	10	4
5	8	6	15	7
6	13	18	3	2
7	6	17	8	5
8	9	14	7	6
9	10	11	12	3
10	8	15	10	3
11	9	13	6	8
	111 (30,8%)	131 (36,4%)	102(28,3%)	52(14,4%)

Данные таблиц 5 и 6 показывают, что количество ответов первого типа, выбранных учащимися экспериментальной группы (48,2%) больше, чем выбранных учащимися контрольной группы (30,8%). Напротив, количество ответов четвертого типа, выбранных учащимися групп, где проводились занятия по курсу "Алгоритмы и алгоритмические структуры" меньше (3,5%), чем выбранных учащимися контрольных групп (14,4%). Это свидетельствует о более высокой практической направленности знаний учащихся в экспериментальной группе, чем в контрольной. Кроме того, были рассчитаны **описательные** статистические параметры из п.3 [Жорн., с.545] – в таблице ниже.

Следовательно, можно считать, что сравниваемые значения двух выборок действительно статистически достоверно различаются с вероятностью 95% допустимой ошибки.

Таблица 7. Данные статистической обработки результатов по первому типу вопроса.

Статистические параметры	Контрольная группа	Статистические параметры	Экспериментальная группа
Среднее	10,09090909	Среднее	17,36363636
Стандартная ошибка	0,693535677	Стандартная ошибка	1,717676693
Медиана	10	Медиана	17
Мода	12	Мода	16
Стандартное отклонение	2,30019762	Стандартное отклонение	5,696889103
Дисперсия выборки	5,290909091	Дисперсия выборки	32,45454545
Эксцесс	-0,946965671	Эксцесс	1,471638067
Асимметричность	-0,255468861	Асимметричность	1,037472213
Интервал	7	Интервал	20
Минимум	6	Минимум	10
Максимум	13	Максимум	30
Сумма	111	Сумма	191
Счет	11	Счет	11
Наибольший(1)	13	Наибольший(1)	30
Наименьший(1)	6	Наименьший(1)	10
Уровень надежности(95,0%)	1,545293781	Уровень надежности(95,0%)	3,827222159

Таблица 8. Данные статистической обработки результатов по второму типу вопроса.

Статистические	Контрольная	Статистические	Экспериментальная
----------------	-------------	----------------	-------------------

параметры	группа	параметры	группа
Среднее	11,90909091	Среднее	14,63636364
Стандартная ошибка	1,231809795	Стандартная ошибка	1,396571954
Медиана	13	Медиана	16
Мода	13	Мода	18
Стандартное отклонение	4,085450904	Стандартное отклонение	4,631905165
Дисперсия выборки	16,69090909	Дисперсия выборки	21,45454545
Эксцесс	-1,04495066	Эксцесс	-0,337304893
Асимметричность	-0,11376431	Асимметричность	-0,351041017
Интервал	12	Интервал	16
Минимум	6	Минимум	6
Максимум	18	Максимум	22
Сумма	131	Сумма	161
Счет	11	Счет	11
Наибольший(1)	18	Наибольший(1)	22
Наименьший(1)	6	Наименьший(1)	6
Уровень надежности(95,0%)	2,744643251	Уровень надежности(95,0%)	3,111756217

Таблица 9. Данные статистической обработки результатов по третьему типу вопроса.

Статистические параметры	Контрольная группа	Статистические параметры	Экспериментальная группа
Среднее	9,272727273	Среднее	2,090909091
Стандартная ошибка	0,982491352	Стандартная ошибка	0,779913112
Медиана	10	Медиана	2
Мода	10	Мода	2
Стандартное отклонение	3,258555173	Стандартное отклонение	2,586679163
Дисперсия выборки	10,61818182	Дисперсия выборки	6,690909091
Эксцесс	0,515883022	Эксцесс	5,465722472
Асимметричность	-0,233491443	Асимметричность	2,166198634
Интервал	12	Интервал	9
Минимум	3	Минимум	0
Максимум	15	Максимум	9
Сумма	102	Сумма	23
Счет	11	Счет	11
Наибольший(1)	15	Наибольший(1)	9
Наименьший(1)	3	Наименьший(1)	0
Уровень надежности(95,0%)	2,189127143	Уровень надежности(95,0%)	1,737754699

Таблица 10. Данные статистической обработки результатов по четвертому типу вопроса.

Статистические параметры	Контрольная группа	Статистические параметры	Экспериментальная группа
Среднее	4,727272727	Среднее	1,272727273
Стандартная ошибка	0,739667113	Стандартная ошибка	0,449057783
Медиана	4	Медиана	1
Мода	3	Мода	0

Стандартное отклонение	2,453198284	Стандартное отклонение	1,489356176
Дисперсия выборки	6,018181818	Дисперсия выборки	2,218181818
Эксцесс	-1,053872576	Эксцесс	3,447661919
Асимметричность	0,583327132	Асимметричность	1,651059991
Интервал	7	Интервал	5
Минимум	2	Минимум	0
Максимум	9	Максимум	5
Сумма	52	Сумма	14
Счет	11	Счет	11
Наибольший(1)	9	Наибольший(1)	5
Наименьший(1)	2	Наименьший(1)	0
Уровень надежности(95,0%)	1,648081025	Уровень надежности(95,0%)	1,000563089

Выводы по главе 3

В ходе проведения эксперимента по определению эффективности предложенной подготовки учащихся к применению алгоритмического мышления в будущей профессиональной деятельности получены следующие результаты:

1) уровень подготовки школьников в экспериментальной группе выше (62% - 97% верных ответов) и соответствует III и IV уровням подготовки (п. 1.2 гл. 1), чем в контрольной (14% - 66%) – I-III уровням. Рассчитанный уровень различий позволяет утверждать с достоверностью 95% [Грабарь М.И., с. 31-33], что результаты тестирования в экспериментальной группе выше, чем в контрольной;

2) отсутствует зависимость между результатами тестирования и успеваемостью учащихся как в экспериментальной, так и в контрольной группах;

3) полученные результаты самооценки по карте-анкете являются достаточно высокими для 10-балльной шкалы, что свидетельствует о заинтересованности, удовлетворенности работой, хорошем самочувствии, об осознании учащимися практической ценности применения алгоритмического мышления в будущей профессиональной деятельности;

4) данные анкетирования по определению сформированности практической направленности знаний школьников к использованию алгоритмического мышления в учебной деятельности свидетельствуют о более высокой практической направленности знаний учащихся в экспериментальных группах, чем в контрольных. Сравнимые значения двух выборок действительно статистически достоверно различаются с вероятностью 95% допустимой ошибки.

Заключение

1. На основе анализа состояния знаний учащихся, необходимости подготовки к выбору профессии и дальнейшей профессиональной деятельности показана педагогическая целесообразность внедрения профильного курса теоретической информатики.

2. Сформулированы уровневые характеристики обучения учащихся профильных классов применению средств ИКТ, определяющие уровни начальных (для выравнивания обучения школьников) и требуемых выпускных знаний, умений и навыков использования средств ИКТ в дальнейшей профессиональной деятельности. Определены также условия, в соответствии с которыми строится процесс обучения учащихся с учетом уровня предыдущих знаний, умений и навыков в сфере ИКТ и возможным оснащением аппаратными и программными средствами задействованных подразделений учреждения образования.

3. Для реализации отобранных принципов подбора содержания курса применен тезаурусный метод и сетевое моделирование: осуществлен синтез системы основных понятий теоретической информатики в виде взаимосвязанного описания понятий и отношений между ними в сочетании со списком определений терминов и комплектом формально-логических схем для содержания программы учебной дисциплины «Основы алгоритмизации и программирование» и основной темы изучения «Алгоритмы и алгоритмические структуры», а также разработана блочно-модульная структура программы теоретического курса «Основы алгоритмизации и программирование».

4. Для реализации отобранных принципов и требований к уровням обучения подобрана методика и определен состав программного обеспечения курса «Применение ИКТ в учебной деятельности»

5. В процессе преподавания экспериментальных профессионально-ориентированных курсов: теоретического «Основы алгоритмизации и программирование» и технологического "Применение ИКТ в учебной деятельности" собирались данные, статистическая обработка которых показала эффективность предложенных в работе подходов к обучению (подготовке) учащихся профильных классов.

Практические рекомендации: использовать данную методическую разработку для учащихся профильных классов в средних учреждениях образования.

Список источников

Литература

1. Российская Федерация. Законы. Федеральный государственный образовательный стандарт основного общего образования [Текст] : [федер. закон: Утвержден приказом Министерства образования и науки Российской Федерации от «17» декабря 2010 г. № 1897]. – М. : Просвещение, 2011. – 194с.
2. *Андреев, А.А., Каплан, С.Л.* и др. Основы открытого образования [Текст]. – М.:НИИЦ РАО, 2009. – Т.1. – 676 с.; Т.2.- 680 с.
3. *Башмако, А.И., Башмаков, И.А.* Разработка компьютерных учебников и обучающих систем [Текст]. – М.: Филинь. – 612 с.
4. *Васильев, В.И., Тягунова, Т.Н.* Теория и практика формирования программно-дидактических тестов [Текст]. – : МЭСИ, 2001. – 130 с.
5. *Вострокнутов, И.Е.* Технология оценки программных средств образовательного назначения [Текст]. – М.: 2002, -672 с.
6. *Горячев, А.В.* и др. Информатика в играх и задачах [Текст]/Тетради для учащихся (по четвертям) и методические рекомендации для учителя по каждому классу начальной школы.
7. *Грабарь, М.И.* Измерение и оценка результатов обучения [Текст] – М.: ИОСО РАО, 2000. – 93 с.
8. *Зимняя, Н.А.* Педагогическая психология [Текст] : учебное пособие для вуза. – Изд.2-е, перераб. и доп. – М. : Логос, 2006. – 264с.
9. Интернет-порталы: содержание и технологии [Текст]: Сб.науч. ст. Вып. 1 /Редкол.: А.Н.Тихонов (пред.) и др.; ГНИИ ИТТ «Информика» – М.: Просвещение, 2009.– 720 с., ил.
10. «Информационные и коммуникационные технологии в образовании» [Текст].– ж.Вестник образования. – 2003. - №2.
11. Кодирование информации. Информационные модели: 9-10 класс [Текст]: пособие для общеобразовательных учебных заведений / Кушниренко А.Г., Леонов А.Г., Эпиктетов М.Г. и др. – М.: Дрофа, 2000. – 208 с.
12. Информатика [Текст]: Учебник для педвузов / Под ред. Могилева А.В., Пака Н.И., Хеннера Е.К. – М.: Академия, 2000. – 768 с.
13. *Кравцова, А.Ю.* Опыт использования информационных технологий в школах Великобритании [Текст] – М.: Информатика и образование, 1997.

14. *Краснова, Г.А., Беляев, М.И., Соловов, А.В.* а) Технология создания электронных обучающих средств [Текст]. – М.: РУДН, 2002; б) Теория и практика создания образовательных электронных изданий [Текст]. – М.: РУДН, 2003. -241 с.
15. *Лапчик, М.П., Семакин, И.Г., Хеннер, Е.К.* Методика преподавания информатики [Текст]: Учеб. пособие для студ. пед. вузов /Под ред. М.П. Лапчика — М.: ИЦ «Академия», 2001.
16. *Мартынов, Д.В.* Совершенствование подготовки в области информатики студентов факультетов социальной работы: дисс.... кандидата педаг. наук. - М, РАО, 2000. – 344с.
17. *Симонович, С.В., Алексе,в А.Г.* Общая информатика [Текст]: Учебное пособие для средней школы. – М.: АСТ-ПРЕСС, 2003, - 592 с.
18. *Смольникова, И.А.* Информационные технологии в образовании. - Международные конференции-выставки «ИНО-97», «ИТО-97» - «ИТО-98», г. Информатика; ж. ИНФО,1997, №2, с. 78-83, бюллетень Информатизация ВШ, 1997, №4, с. 71-83; ж. Информационные и коммуникационные технологии в образовании, 1998, №1.
19. *Угринович, Н.Д.* Информатика и информационные технологии [Текст]. - М.: Лаборатория Базовых Знаний, 2001. – 464 с. с CD.

Основные положения и результаты работы отражены в следующих публикациях автора:

20. (в соавторстве) Тема. – Изд-во, год, с.

Электронные ссылки

21. Федеральный портал образования [Электронный ресурс]. – <http://www.edu.ru>.
22. Информационные технологии в образовании [Электронный ресурс]. - Международные конференции-выставки ежегодно с 1992 г. Сборники, CD-ROM и архивы на сайтах <http://ito.edu.ru>, ito.edu.su.
23. Телеконференция по образованию [Электронный ресурс] на сайте <http://relcom.edu.ru>.
24. Педагогический энциклопедический словарь [Электронный ресурс] – <http://dictionary.fio.ru/article.php?it=1/>.
25. Сайт для тех, кто проспал школу [Электронный ресурс] - subscribe.ru.
26. Для московских детей -инвалидов [Электронный ресурс] с бесплатной арендой ПК, сканера и микроскопа <http://www.9151394.ru> .
27. Общеобразовательный педагогический портал [Электронный ресурс] – <http://www.school.edu.ru>

28. Лаборатория информатики МИОО [Электронный ресурс] – <http://iit.metodist.ru>.
29. Домашний компьютер и школа [Электронный ресурс] -<http://sch0000.dol.ru/KUDITS>
30. Компьютерная азбука [Электронный ресурс] -<http://www.tl.ru/~gimnl3/docs/ivt/inf.htm>
31. Как устроен Интернет [Электронный ресурс] -
<http://www.irnet.ru/olezhka2/winterne.shtml>
32. Задачи школьных олимпиад по информатике [Электронный ресурс] –
<http://avalex.chat.ru>
33. Инфо2000 [Электронный ресурс] - <http://schools.keldysh.ru/info2000>
34. Информатика [Электронный ресурс] - <http://informatics.nm.ru/>
35. Информатика в школе [Электронный ресурс] - <http://www.infoschool.narod.ru> –
36. Зрательный словарь по информатике [Электронный ресурс] - <http://school.ort.spb.ru/library/examhelp/slovar/slovar.htm>
37. Информационно-поисковая система «Задачи» [Электронный ресурс] -
<http://zadachi.mccme.ru:8101/>
38. Игры с Муравьем <http://www.uic.ssu.samara.ru/~ant/index.html>.
39. *Смольникова, И.А.* Информационные технологии в образовании. - НИИ «Информика» [Электронный ресурс] -. <http://www.informika.ru/ИТ/публикации>.(25 марта 2007).
40. *Смольникова, И.А.* Информационные технологии в образовании. Файлы лекций, методических рекомендаций для проведения занятий с учениками по информатике и шаблоны обучающе-контролирующих презентаций.

ПРИЛОЖЕНИЯ

Приложение 1. Термины и сокращения

1. **ИКТ** = информационные и коммуникационные технологии автоматизации информационной деятельности.
2. **ИР** = информационные ресурсы – хранимые систематизированные знания.
3. **КУВТ** = класс учебно-вычислительной техники с ЛВС.
4. **ПМК** = программно-методический комплекс – комплекс, состоящий из различных типов ИР, ПО УН и инструментов для их создания и модификации, а также методических рекомендаций по его использованию в УВП.
5. **ПО УН** = программное обеспечение учебного назначения – компьютерные демонстрационные, обучающие, моделирующие, консультирующие, тестирующие, игровые программы и среды, в сетевых версиях для КУВТ.
6. **УО** – учреждение образования.