

Методика преподавания информатики.

Лекции для студентов 4-го курса педагогического потока ММФ (февраль — июнь 2009года, 34/34 часа)

Введение.

Понятие информатики.

Прежде чем говорить о содержании учебного предмета “Информатика”, попытаемся определить область, именуемую этим словом.

До настоящего времени нет строгого определения термина „информатика”. В учебниках приведено, например, следующее определение: „Информатика — наука, исследующая законы и методы переработки и накопления информации”. Далее можно прочитать следующее: „... Эта наука позволяет не только понять принципы работы и возможности использования ЭВМ, но и даёт представление о законах и методах представления информации при общении людей и в жизни общества.”

В литературе часто можно встретить следующие признаки информатики: сбор, передача, переработка, хранение информации. Но с помощью подстановки вместо слова “информация” какого-нибудь другого слова (например, “руда” или “лесоматериалы”) такое определение легко превратить в определение соответственно металлургии или лесного хозяйства.

Дать определение, отражающее специфику информатики, не просто. Представляется удачным рабочее определение этого термина с помощью трёх компонент:

1) *HARDWARE* – “твёрдые”, аппаратные средства (иногда говорят “железо”). Дословно переводится как аппаратура.

2) *SOFTWARE* – “мягкие”, программные средства, к которым можно и нужно отнести не только программы, но и обрабатываемые данные. Дословно переводится как программное обеспечение.

3) *BRAINWARE* – алгоритмическая, в буквальном переводе – мозговая составляющая (*brain* - мозг). Это не просто алгоритмы обработки данных, что снова вело бы к *SOFT*, а интеллектуальная составляющая, знание, которое воплощено и в программах, и в данных, и в аппаратуре.

Это определение информатики удивительно перекликается с наивно-фундаментальным представлением древних о существовании четырёх стихий – первоэлементов:

0) Земля – *HARD*

0) Вода – *SOFT*; данные и программы не постоянны, текучи, изменчивы;

0) Воздух – это та сфера, в которой происходят информационные процессы (инфосфера В.И.Вернадского);

0) Огонь – конечно же, *BRAIN*.

Слово „информатика” возникло во французском языке в начале 60-х годов и является сочетанием двух слов: ИНФОРмация и автоМАТИКА (или ИНФОРМАция и автомаТИКА). Этот термин обозначает область автоматизированной обработки информации, при которой львиную долю операций над ней выполняют ЭВМ и другие технические средства, а окончательные решения по управлению принимает человек.

Анализ различных определений этого термина позволяет выделить следующие аспекты его содержания:

— совокупность средств автоматизированной информационной техники и технологии;

- особая отрасль экономики, включающая всю сферу автоматизированной обработки и использования информации;
- отрасль научного знания, изучающая процессы передачи и обработки информации и средства её автоматизированной обработки (техническое, математическое и программное обеспечение);
- теория научной информации и научно-информационной деятельности с акцентом на средства автоматизации.

Из истории преподавания курса школьной информатики.

Ограничимся перечислением основных этапов становления школьной информатики в прошлом веке в бывшем Советском Союзе и в нашей Республике Беларусь.

В 50-е годы практиковалось изучение программирования в ряде школ г. Новосибирска. Главным идеологом этого нововведения был академик А.П.Ершов.

В 60-е годы осуществлялась подготовка программистов в московских школах с математической специализацией, а позже, в 70-е годы, готовили школьников по специальностям, связанным с ЭВМ (Москва, Ленинград, Новосибирск).

В конце 70-ых годов началось массовое производство микро ЭВМ, что расширило области применения и доступность ЭВМ. Началась разработка концепции школьной информатики (А.П.Ершов, Г.А.Звенигородский, Ю.А.Первин). В 1982 году соответствующими министерствами СССР было принято решение о введении калькуляторов в учебный процесс школы.

В 1984 году были разработаны основные направления реформы общеобразовательной и профессиональной школы. Решено ввести в школе новый курс, который сначала назывался “Основы информатики и вычислительной техники” (ОИиВТ). В 1985 г. разработана первая экспериментальная программа этого предмета, в 1985—1986 гг. издано в двух частях пробное пособие для средних учебных заведений под редакцией А.П.Ершова. С 1 сентября 1985 г. началось преподавание основ информатики и вычислительной техники в массовой школе и подготовка учителей информатики в пединститутах по новым учебным планам.

С 1986 г. начинает издаваться всесоюзный журнал “Инфо” (“Информатика и образование”). Во второй половине 80-х годов на его страницах была проведена оживленная полемика вокруг содержания предмета ОИиВТ. Была разработана конкурсная программа учебника. В соответствии с ней было издано первое поколение учебников по информатике и соответствующие пособия для учителей авторскими коллективами под руководством А.П.Ершова, А.Г.Кушниренко (Г.В.Лебедев, Р.А.Сворень), В.Г.Житомерского (А.Г.Гейн), В.А.Каймина и др.

Необходимость введения нового предмета объяснялась широким внедрением вычислительной техники и особенно персональных компьютеров во все области человеческой деятельности, в том числе в сферу быта и систему народного образования. Бурное развитие микроэлектроники и на её базе производства мини- и микро-ЭВМ, а позже персональных компьютеров, совершенствование их программного обеспечения и средств общения с ними способствовало компьютеризации школы и других средних учебных заведений и введению предмета ОИиВТ.

В первой половине 90-х годов продолжалась оживленная дискуссия по вопросу преподавания информатики в Республике Беларусь, были разработаны новые концепции информатизации образования. В 1994 г. Основной курс информатики перенесли в базовую школу в 8 – 9 классы. Этот двухгодичный курс был обязательным для всех типов учебных заведений. В 10 -11 классах курс продолжался на углубленном и повышенном уровнях в специальных классах, школах, лицеях и гимназиях. Были разработаны новые программы, учебники и учебные пособия белорусских авторов, таких, как Ю.А.Быкадоров, В.М. Ко-

тов, О. И. Мельников, А.Т.Кузнецов, А.И.Павловский и др. Были переделаны и переизданы первые учебники российских авторов.

В настоящее время общие вопросы методики преподавания информатики в основном понятны и сформулированы, ответы на них в значительной степени найдены. Детально разработана и методика преподавания ряда конкретных тем. В то же время постоянная открытость информатики новому ставит более сложную и важную задачу – научить студента самостоятельно разрабатывать методику, научить методическому творчеству, передать ему опыт подобной творческой деятельности.

Начиная с 2004/05 учебного года, в рамках реализации реформы общеобразовательной школы обязательный курс информатики изучается не два, а четыре года в 7(6), 8(7), 9(8) и 10(9) классах (*в скобках указана старая нумерация классов согласно 11-летнему сроку обучения*). Разработана сквозная четырёхлетняя программа этого предмета. Подготовлены и изданы четыре учебника заведующего кафедрой информатики Академии Последипломного Образования Пупцева А. Е. и соответствующие методические пособия для учителей. Начиная с 2006/2007 учебного года, информатика изучается ещё на один год раньше, с 6-го класса. Сквозной пятилетней программы изучения этого предмета пока нет. Действует временная программа для 6-го класса. Эту программу, электронный вариант готовящегося учебного пособия для 6-го класса и соответствующего методического пособия для учителей, можно найти на сайте Главного Информационно-Аналитического Центра Министерства Образования (ГИАЦМО) Республики Беларусь:

<http://www.giac.unibel.by>

Центр находится в Минске по адресу: ул. Захарова, д. 59 (информатика в комнате 308). На этом же сайте регулярно размещается другая полезная информация (программы, инструктивные письма, указания и т. п.), связанная с преподаванием не только информатики, а и других общеобразовательных предметов в средней школе.

Тема 1. Методика преподавания информатики в системе педагогических знаний

Одновременно с введением информатики в школьное образование была проведена массовая переподготовка уже работающих учителей и выпускников педвузов, а в 1985 году введены новые учебные планы для систематической подготовки учителей информатики.

Методику преподавания информатики (МПИ) как новую дисциплину начали преподавать в пединститутах в 1987/88 учебном году. На тот момент существовало лишь фрагментарное видение предмета, отраженное в ряде статей, докладов, дискуссий. Поэтому преподавание курса началось одновременно с его разработкой. Курс МПИ вначале строился как методика решения задач на алгоритмизацию в безмашинном варианте.

1.1. Цели изучение курса МПИ.

Курс МПИ рассматривается как единая система целей, содержание, методов, форм и средств. В результате изучения курса будущий учитель должен подготовиться не только к преподаванию информатики, а и к полноценной работе в компьютеризированной школе. Для этого ему необходимо:

1) увидеть место и значение курса информатики в общем образовании школьника, понять и вскрыть связи этого предмета с другими дисциплинами;

2) освоить содержание курса, проведя сравнительный анализ действующих и новых учебников и программ;

0) овладеть средствами изучения курса, освоить классические и новые методы обучения, управлять умственной деятельностью учащихся, научиться развивать и закреплять у учащихся интерес к предмету;

0) освоить и научиться использовать на практике различные организационные формы занятий;

0) развить и закрепить логико-алгоритмический и системно-комбинаторный стиль мышления, который является признаком профессионализма.

Изучив курс МПИ, студент должен знать:

- теоретические основы курса: историю, место и роль дисциплины; цели, методы и формы преподавания информатики; использование принципов дидактики;

- содержание реформы в преподавании школьной информатики, программ, существующих и новых основных учебников для базовой школы, а также учебных пособий для классов с углубленным и повышенным уровнем изучением информатики;

- язык программирования C++ Builder в визуальном режиме в объёме, необходимом для изучения в основном курсе школьной информатики и (или) на факультативных и других дополнительных формах занятий в школах, лицеях и других учебных заведениях;

- методику изучения основ алгоритмизации и программирования;

- методику изучения других основных разделов школьной информатики.

Кроме этого, студент должен уметь:

- разработать учебную и рабочую программу по информатике для любого класса;

- составить план урока по любой теме и провести по нему занятие;

- составить и отладить программу на языке C++ Builder, который не изучался в основном курсе методов программирования

Согласно концепции информатизации РБ, МПИ не может и не должна сводиться к преподаванию только информатики. Она обеспечивает реализацию следующей группы целей этой концепции:

- изучение новых областей знаний, связанных с информатикой (через учебное моделирование);

- приобретение навыков использования компьютерных технологий;

- более полное выявление и развитие способностей учащихся специальными методами.

Будучи первой дисциплиной методического цикла, связанного с ЭВМ, МПИ играет корректирующую роль в компьютеризации образования в целом. Проблемы информатизации образования лежат не только и не столько в области техники. Намного важнее наличие хороших программных средств. Но даже полное обеспечение системы образования мощными современными компьютерами с дружественным программным обеспечением ещё не означает автоматически, что проблемы информатизации образования решены. Большую роль играет степень психологической готовности и потребность педагогов и управленцев в пересмотре средств и способов своей деятельности в связи с использованием ЭВМ.

1.2. Связь МПИ с другими предметами.

Основная особенность курса МПИ – связь с другими, прежде всего методического цикла, предметами. С методикой преподавания математики она связана, можно сказать, генетически. Понятие алгоритма пришло из математики. Возможна такая аналогия: множество/элемент = алгоритм/команда. С другой стороны, многие доказательства в математике имеют алгоритмическую структуру, и существует задача научить выявлять эту алгоритмическую составляющую в доказательствах.

Время от времени возникала и продолжает возникать тенденция объединить информатику, например, с математикой. В лучшем варианте, у сильного учителя, могут раскрыться межпредметные связи. Но сама информатика может проиграть в этом случае в силу математизации задач и связанного с этим неизбежного роста их трудности. Это противоречит тенденции разгрузки начала изучения информатики как раз от математики (решение квадратных уравнений). При этом могут быть утрачены и связи информатики с другими дисциплинами, и предметы будут по-прежнему изучаться обособленно. Не связано с математикой и умение пользоваться готовыми ПС.

Компьютер – достаточно дидактически мощное средство, которое заставляет пересматривать и содержание, и формы, и методы обучения, всю систему в конкретных дисциплинах. Вот некоторые примеры влияния компьютера на курс математики:

0) Традиционное повышенное внимание школьной математики к логарифмической функции связано с упрощением вычислений вручную при переходе от умножения чисел к сложению их логарифмов. При использовании компьютеров исчезает основание для приоритета логарифмов.

0) Исследование графика функции может быть начато с его быстрого построения на экране и резко упрощается. Если функция в точке не существует, компьютер выдаёт сообщение об ошибке типа деление на ноль.

0) С помощью компьютера легче понять смысл итерационных методов для решения тех или других задач (вычисление квадратного корня, решение уравнений и др.).

Оптимальным решением является интеграция информатики с другими дисциплинами, но не в учебных часах, а на уровне решаемых задач, системы знаний. Можно ожидать взаимного обогащения понятий, разнообразия связей и как следствие прочности и полезности знаний в целом.

В идеале возможности применения ЭВМ должны практически раскрывать учителя-предметники. Пока этот процесс идет крайне медленно, курс школьной информатики и кабинет ВТ остаются в определенной изоляции. Не будем останавливаться на причинах этого. Они в основном очевидны. В оправдание этого можно сказать, что сфера образования вообще инерционна в большей степени, чем общество. Потери от игнорирования компьютеров при образовании школьников слишком велики.

Сходство МПИ с методиками преподавания физики, химии проявляется в склонности информатики к опыту, эксперименту. Действительно, запуск программы на компьютере — это своеобразный эксперимент. С точки зрения физики компьютер — это прибор, поведение которого можно исследовать. У методики физики можно позаимствовать методы выполнения подобных опытов.

Область пересечения интересов МПИ с психологией – это прежде всего проблемы общения. Особая форма общения – компьютерная игра. Методика информатики уже предполагает и использует органическое включение игровых элементов в учебную деятельность с компьютером. Можно, например, рассматривать отладку программы как игру с компьютером – “кто умнее”.

Своеобразной является связь с методикой изучения иностранного языка. Перевод, например, алгоритма на конкретный формальный язык (C, *Pascal*, алгоритмический язык и др.) – это вполне языковая, речевая деятельность, иногда и непростая проблема.

1.3. Другие особенности МПИ

Сильные внутри предметные связи информатики порождают нетривиальную задачу поиска оптимального порядка изучения материала в соответствии с принципами последовательности и доступности изложения. При буквальном понимании последовательности предполагается, что учебный материал выстраивается в логическую цепочку или может быть представлен в виде дерева, где нет логических кругов, и повторение идет лишь как закрепление материала. В информатике это, увы, или к счастью, невозможно. Сильные

внутри предметные связи между различными темами не позволяют “выпрямить материал”, как это имеет место, например, при изучении истории, литературы и некоторых других предметов.

В информатике нельзя некоторые темы (например, оператор цикла) изучить сразу полностью, за один или несколько подряд идущих уроков. Особенно это касается раздела “Основы алгоритмизации и программирования”. Поэтому еще академиком А.П.Ершовым была предложена реализация принципа последовательности в форме *цикличности*. Это означает, что сначала изучается некоторый элемент в простейшем, наиболее распространенном на практике виде. Затем это понятие повторяется, обогащаясь новыми возможностями, дополнительными особенностями. Если для других дисциплин это желательный путь, то для информатики многие разделы или темы по-другому изучить просто невозможно.

Особенностью МПИ является *динамический, изменяющийся характер самой информатики* и как науки, и как учебного предмета, ее нестабильность, постоянное развитие и совершенствование как технических, так и особенно программных средств. В этих условиях вынужденным и плодотворным решением является максимальная опора на результаты общей дидактики, на конкретные методики близких дисциплин – математики и физики. Необходимо руководствоваться принципом единого классического образования школьников в области информатики, не зависящего, или в малой степени зависящего от типов компьютеров и ПО. Надо больше внимания уделять наиболее общим, фундаментальным знаниям. Следует по возможности минимизировать или полностью избегать машинно-зависимых знаний и умений, которые могут оказаться бесполезными, а, возможно, и вредными, при работе на другом типе компьютера, с другой операционной системой или другой версией языка.

Основная особенность МПИ – связь предмета с *использованием компьютера*, который обладает несравненно большей “самостоятельностью”, чем любой другой прибор, например, в физике. Общение с компьютером требует развития особых черт мышления и поведения, адекватных методов обучения и воспитания. Достаточно сказать, что цена ошибки в обычном тексте и в программе для компьютера различна. Компьютер является таким средством обучения, мощность, эффективность и дидактические возможности которого потенциально превосходят то, что доступно иным средствам обучения. Это средство не может не вести к обновлению методик преподавания и других дисциплин.

При изучении МПИ на лабораторных занятиях студент “играет” три роли и соответственно выполняет три вида деятельности:

0) роль учащегося – для лучшего понимания материала “изнутри” с позиции ученика и освоения тех учебников, которые появились в школе после поступления студента в вуз;

0) роль учителя – разработка инструкций, заданий, вопросов для контроля и других материалов для учащегося, управление с рабочего места для преподавателя работой класса ПЭВМ;

0) роль методиста-предметника – разработка методических материалов, прежде всего, для себя как учителя, а фактически и для других учителей.

Итоговый результат работы студента: учебно-методический комплекс (УМК) по определенной теме курса, который включает:

- одну или несколько лекций;
- подробный план проведения лабораторных (практических) занятий;
- дидактический материал: упражнения, задачи, задания с решениями и комментариями; или без них;
- контрольные вопросы;
- набор тестов с ответами и комментариями для самостоятельной работы или без них.

Тема 2. Цели преподавания школьной информатики.

2.1. Уровни работы с компьютером

0) Пассивный пользователь, не работая непосредственно с ЭВМ, пользуется компьютерными “благами”: получает в банкоматах зарплату, покупает билет на поезд и т.д. Минимальные сведения о компьютерах получает из средств массовой информации. Профессиональные навыки работы на ЭВМ отсутствуют.

0) Активный (параметрический) пользователь работает с готовыми программами, подставляя в них свои параметры: добавляет базы данных, осуществляет её корректировку и поиск нужной информации и т.п. Такой пользователь способен длительное время обходиться без помощи программиста. С позиции школьной информатики это наиболее массовая фигура в настоящем и ближайшем будущем.

0) Программирующий пользователь способен внести небольшие изменения в готовую программу. Например, он может вставить свою формулу в готовую универсальную программу построения графика функции. Он может программировать небольшие задачи в простых средах (формулы в электронных таблицах типа *Excel*), способен грамотно поставить задачу программисту, выбрать необходимое готовое программное обеспечение для решения своих задач. Эффективность деятельности активного пользователя возрастает, если он подтягивается до уровня программирующего пользователя, так как уменьшается зависимость от программистов. Подготовка всех учащихся до этого уровня — посильная для школы и социально необходимая задача.

0) Парапрограммист (“настройщик”) работает на языках сверхвысокого уровня: в СУБД, электронных таблицах и др. Основное его умение: с одной стороны формализовать прикладные задачи пользователя и довести их до состояния работающей программы, и наоборот, видеть конкретные применения готовых программ. Он в состоянии настроить готовые программные продукты общего назначения на конкретные нужды пользователя. Такой специалист перебрасывает мостик между реальностью и компьютером, что требует системно-комбинаторного, прикладного образа мышления. От него в конечном счёте зависит, дойдет ли разработка программиста до практического применения, сможет ли пользователь с ней работать. Такой “настройщик” пока необходим, так как рост друженности программ происходит в первую очередь по отношению к программистам.

0) Программист разрабатывает средства для парапрограммистов и для пользователей. То есть предполагается, что он разрабатывает прежде всего общее программное обеспечение. Но нельзя согласиться с автором книги [1], который утверждает, что программист прикладные задачи обычно не решает. А кто тогда их программирует, если нет готовых программ? Программист использует достаточно мощные и сложные для массового освоения системы и языки программирования: Delphi, Paskal, Builder, C++, Visual C++, Java и др.

0) Наконец, системный программист обеспечивает эффективность работы на всех предыдущих уровнях.

2.2. Цели преподавания информатики как единство образования, развития и воспитания

Определение целей преподавания (целеполагание) – наиболее трудная часть практической методики. Цели обучения в общей дидактике рассматриваются в единстве образования, развития и воспитания. Информатика подчеркивает и практический аспект — подготовку молодого человека к полноценной жизни в компьютеризованном обществе.

Прикладной компонент **образования** проявляется в умении извлекать практическую пользу из общения с ЭВМ (запросы к поисковым системам, Интернет, небольшие расчеты и т.п.). Это определяет уровень, качество жизни.

Общее образование заключается в формировании мировоззрения, понимании принципов работы, возможностей и ограничений ЭВМ и достигается лишь на основе сис-

темных знаний, выводящих за рамки прагматичного, потребительского подхода к ЭВМ. Признаки общего образования при изучении информатики:

- 0) понимание общих принципов работы ЭВМ (а не частностей!);
- 0) представление о сути деятельности программиста и других специалистов в области информационных технологий;
- 0) знания о типах информации и способах ее обработки;
- 0) представление о типах алгоритмов, способах их записи (в частности о программах), о сложности алгоритмов;
- 0) знания о возможностях и ограничениях автоматизации мышления.

Цели **развития** в основном сводятся к формированию двух взаимодополняющих стилей мышления.

- 0) Логико-алгоритмическое мышление проявляется в умении:
 - строить логические утверждения о свойствах данных и запросы к поисковым системам;
 - мыслить индуктивно и дедуктивно при анализе своих затруднений;
 - формализовать свои намерения вплоть до записи на алгоритмическом языке или языке программирования.
- 0) Признаки системно - комбинаторного мышления:
 - видение предметов и явлений в целостности, взаимосвязях;
 - умение строить несколько взаимодополняющих точек зрения на одну и ту же проблему;
 - умение комбинировать понятийные и орудийные средства из различных дисциплин при построении моделей. Например, с точки зрения алгебры функция есть соответствие, с точки зрения геометрии – кривая, а с точки зрения информатики – алгоритм вычисления результатов по заданным аргументам.

Заметим, что системно–комбинаторный стиль мышления, к сожалению, в основном чужд практике средней и высшей школы. Предметы расчленены на высшие и низшие, теоретические и практические. Не всегда видна связь между ними. Поэтому и студенту, школьнику сквозь обилие частных не видно интегральных, целостных знаний.

Информатика должна выполнять роль межпредметной, интегрирующей дисциплины. Если ее рассматривать просто как локальный предмет, не связанный с другими, если ее изолировать в кабинете ВТ, то такой путь является тупиковым, сковывающим творчество и учителей, и школьников.

Особенностью системно-комбинаторного мышления является также и то, что работа с компьютером — это своеобразная языковая деятельность. Как пользователи, так и особенно программисты излагают свои мысли и намерения в формализованном виде, оперируя правилами специальных языков.

При использовании компьютера, особенно при разработке программ, у школьника развивается умение и склонность к наблюдению за собственным мышлением. В компьютерной деятельности, как ни в одной другой, можно проследить за следующей цепочкой: «что я хотел (алгоритм) – что сделал (текст программы) – что получилось и почему (результат выполнения программы)». Компьютер овеществляет мышление, делает наглядными наши намерения.

При преподавании информатики достигаются и также и **цели воспитания**. При этом формируются следующие черты и качества личности:

- объективное отношение к результатам компьютерных вычислений, т. е. критичность и самокритичность в оценке своих способностей;
- бережное отношение к технике, и к информации, как к своей так и чужой; этическое, нравственное неприятие компьютерного вандализма и вирусотворчества;
- личная ответственность за результаты своей работы на компьютере, за возможные свои ошибки, а также за решения, принимаемые на основе компьютерных данных;

- потребность и умение работать в коллективе при решении больших задач, забота о пользователе продуктов своего труда.

2.3. Компьютерная грамотность, образованность, культура

Под **компьютерной грамотностью** можно понимать умение считать, читать, писать, рисовать, искать информацию с помощью ЭВМ. Такое определение было дано в одном из первых учебников по школьной информатике. Признаком высокой, сформировавшейся грамотности – самостоятельность и эффективность работы с применением ЭВМ. Это первая характеристика качества обучения школьника информатике.

Иногда трактуют такую грамотность излишне широко. Видимо, нужна иная, более точная категория — **компьютерная образованность** как вторая характеристика качества обучения. Ее основные признаки вытекают из обычных представлений об образованном человеке, взятых в контексте информатики. К ним можно отнести:

- 4) регулярное чтение литературы по информатике;
- 4) широкий кругозор в компьютерной области, понимание возможностей и ограничений ЭВМ;
- 4) ориентирование в многообразии программных средств: знать их назначение, качественные характеристики, умение выбрать оптимальные программные средства для конкретной работы, для решения конкретных задач.

Школа не может обеспечить такую образованность, но заложить основы и сформировать потребность в ней — должна!

Третья характеристика качества образования школьника — **информационная культура**, под которой, прежде всего, понимается этика использования компьютера в контексте общечеловеческих ценностей. Можно конкретно говорить в стиле религиозных заповедей, веками регулирующих нормальное человеческое поведение.

- „Не убий” чужую информацию случайно или умышленно. Отсюда неприятие вирусотворчества и небрежности.
- „Не укради” чужую программу или данные.
- „Не сотвори себе кумира”, что означает отказ от компьютерного фанатизма, от излишнего заикливания на программах, от попыток сужения реального мира до компьютерной среды. Особенно это актуально в связи с широким развитием сети Internet.
- „Не лжесвидетельствуй”, т.е. не обсуждай те программы и данные, которые недостаточно хорошо знаешь (скромность).
- «Возлюби ближнего как самого себя»: помогай товарищам и коллегам, не занимай лишнего места в общей памяти, удаляя ненужные устаревшие файлы.

В информационной культуре есть и эстетическое содержание – умение видеть, ценить и создавать красивое: оригинальный, понятный, эффективный алгоритм, красивое оформление сценария работы программы и её результатов (окна, кнопки и другие элементы экрана, использование графики, цвета).

Частью информационной культуры является также самодисциплина и аккуратность: наличие комментариев, структурная запись текстов алгоритмов, стройная система в обозначениях, продуманная организация хранения информации во внешней памяти и др.

Таким образом, можно сделать следующие выводы:

- 5) цели практического образования служат формированию компьютерной грамотности;
- 5) цели общего образования и умственного развития связаны с компьютерной образованностью;
- 5) цели воспитания служат формированию информационной культуры.

Заметим, что между уровнем освоения работы на ЭВМ, грамотностью, образованностью и культурой не всегда существуют однозначные связи. Например, пользователь, не умеющий программировать, может быть весьма эрудированным, и наоборот, фанатичный программист может ничего не знать кроме, скажем, своего языка или используемой

системы. С этой точки зрения создатель вирусов грамотен и, возможно, образован, но культуры, разумеется, у него нет.

Проведенный анализ системы целей позволяет предложить ответы на некоторые дискуссионные вопросы преподавания информатики:

3) Нужно ли всем школьникам изучать программирование? — Да, но лишь в той мере, в какой это обеспечивает подготовку программирующих пользователей, формирует логико-алгоритмическое мышление. Чрезмерное увлечение этим чревато потерей времени, нужного для повышения уровня компьютерной образованности. Признаком профессионализма для программиста является избегание программирования в случаях, когда можно использовать готовые программы. С другой стороны, квалифицированный пользователь не должен бояться программирования небольших, нестандартных задач.

3) Какой язык изучать или использовать? — Неважно какой. Главное, чтобы с его помощью можно было изучить основные типы алгоритмов, основы современных методов и технологий программирования. Не обязательно использовать простую систему программирования. Можно, например, выбрать профессиональный язык C++, но некоторые его достаточно сложные возможности не изучать.

3) Нужно ли будущему пользователю знать устройство ЭВМ? — Да, в той мере, в какой это способствует компьютерной образованности и успеху в практическом использовании компьютера.

3) Сложным, носящим межпредметный характер, является следующий вопрос: нужно ли в информатике заниматься моделированием? — Не так важно, где моделировать, но именно через моделирование формируется системно-комбинаторное мышление, умение решать реальные задачи из разных областей, формируется научная картина мира.

Тема3. Принципы дидактики и преподавание информатики.

При отборе содержания, средств и методов преподавания курса на помощь учителя информатики должны прийти общие принципы дидактики.

3.1. Принцип научности

Этот принцип требует, чтобы в содержании образования нашли отражение новейшие достижения соответствующей области знаний с адаптацией на познавательные возможности учащихся. Эта задача упрощается тем, что пока нет деления на высшую информатику и низшую, как это имеет место, например, в математике. Любое понятие из “большой” информатики находит свои аналогии в школьной информатике.

Фундаментальными являются понятия “информация”, “алгоритм”, “исполнитель”. Исполнитель выполняет несколько функций:

- это дидактическое средство для придания процессу исполнения алгоритмов наглядности (Робот, Чертежник и др.);
- это понятие, позволяющее с единых позиций трактовать многие вопросы: Робот — исполнитель над графикой; Редактор — исполнитель над текстами; Операционная Система — исполнитель для файлов; Принтер — исполнитель для листа бумаги и т.д.

Компьютерную модель всякого исполнителя можно понимать в терминах объектно-ориентированного программирования как модуль или объект.

Научность обучения подразумевает также современность методов обучения, что применительно к информатике означает, прежде всего, моделирование в широком смысле, а также исследовательскую деятельность учащегося.

3.2. Сознательность усвоения и деятельности

В традиционном смысле сознательность — это полное понимание учащимся содержания и средств своей деятельности. Но компьютер, будучи сложным устройством, зара-

нее вынуждает ограничивать эту сознательность целями обучения. Можно ли за ограниченное время полно, детально рассказать обо всех процессах, происходящих в компьютере, например, при нажатии клавиши “Enter”. Все это можно и не знать. Конструктор ЭВМ, программист и пользователь имеют различные точки зрения на такое событие. Надо сформировать у учащегося взаимодополняющих точек зрения на подобные ситуации, что и дает многостороннее знание.

Здесь решающее значение имеет уровень знаний учителя и, самое главное, умение отобрать, ограничить материал. Таким невольным вынужденным отбором содержания можно объяснить нередкий парадоксальный успех среднего, не очень эрудированного студента на педпрактике.

3.3. Доступность и наглядность содержания

Принцип доступности реализуется через выделение уровней обучения и работы за компьютером. Например, самый низкий уровень: простое использование готового программного обеспечения. Это доступно всем учащимся. В разделе “алгоритмизация и программирование” предлагаемые задачи можно разделить на 3 уровня, чтобы школьник мог выбрать доступную для себя задачу (на 5 баллов, среднего уровня на 7 баллов, сложную задачу на 9 баллов). На 10 баллов надо предложить нестандартную оригинальную задачу.

Наглядность достигается с помощью:

- блок – схем алгоритмов;
- структурной (с отступами) записи текстов программ (алгоритмов);
- использования цвета;
- демонстрации готовой программы, процесса ее выполнения.

3.4. Активность и самостоятельность

Активность учащегося при изучении информатики имеет следующую особенность. Если при изучении других дисциплин педагог работает в прямом контакте с обучаемыми, видит их реакцию, сам реагирует, то здесь возможна работа ученика один на один с компьютером. В информатике активность учащегося является не только целью, но и необходимым условием успешности обучения.

Формы проявления активности различны, например, самоконтроль; контроль над работой товарища, оказание реальной ему помощи; модификация готовых и разработка собственных алгоритмов и программ. Активность следует из интереса к предмету, но учителю важно четко сформулировать, что является контролируемым результатом обучения, то есть, что нужно «сдать». Активизировать работу, особенно в начале обучения, можно, практикуя работу учащихся по двое за одним компьютером, даже если их достаточное количество в классе. В таком случае уменьшается неуверенность, возникает диалог, происходит взаимное обучение. Но при этом возникает вопрос: вместе дать возможность работать двум сильным ученикам, хорошо успевающий должен быть в паре со слабым, или оба должны быть с невысокими способностями?

Самостоятельность учащегося, как цель и условие успешного изучения информатики, следует за активностью. Возможные этапы нарастания самостоятельности: от полного управления учителем, через дозированную помощь к самоуправлению познавательной деятельностью с помощью компьютера. Полностью самостоятельность реализуется при переходе к творческой деятельности, при выполнении индивидуальных заданий. Когда ученик обращается за помощью к учителю, к товарищу — это, понятно, проявление не самостоятельности, а активности.

Значение самостоятельности: она ведет к большей результативности обучения, учит находить выходы из затруднительных, иногда, казалось бы, тупиковых ситуаций. Особенно это касается отладки программ, поиска ошибок. Самостоятельность учит также искать нужную литературу, пользоваться ею, а также учит использовать компьютерные средства помощи (Help).

3.5. Прочность и системность знаний

Прочность знаний тесно связана с их системностью, основанной на поиске, построении и учете внутри- и межпредметных связей и ассоциаций. Обычная, например, для зоологии структура изучаемого предмета в виде дерева обязательна: в информатике должна быть дополнена «паутиной» связей между листьями-понятиями, их взаимным обогащением в их комбинациях.

Содержание информатики и как науки, и как учебного предмета в виде одного дерева представить невозможно. Скорее всего, это лес с переплетенными кронами, растущий из таких фундаментальных для этой дисциплины понятий, как «информация», «алгоритм», «исполнитель» и т.д.

3.6. Индивидуализация и коллективность обучения

Индивидуализация и коллективность обучения дополняют друг друга, особенно в информатике. В этом отношении компьютер — дидактически двойственный инструмент. Тиражируя обучающие или готовые программы, он способствует организации единообразной, фронтальной групповой деятельности, но способ работы учащегося с программой — все же “один на один”, со своим индивидуальным темпом, своими путями преодоления трудностей.

Индивидуализация возможна:

- через выполнение индивидуальных, а не общих, одинаковых для всех, заданий, классифицированных по уровню сложности;
- через гибкую настройку обучающей программы (например, на тип мышления учащегося);
- через освобождение времени педагога для индивидуальной работы при автоматизации рутинной части педагогического труда.

При работе учащихся вдвоем за компьютером, что может оказаться весьма полезным, могут сложиться устойчивые отношения типа “работник—указчик”. Поэтому время от времени учащихся надо менять местами и ролями.

3.7. Эффективность учебной деятельности

Эффективность предполагает оптимизацию усилий педагога и ученика для обеспечения наибольшего их КПД, отношение результат/усилие. Это требует, прежде всего, отсутствия постороннего содержания в их деятельности. Например, блок-схемы, наглядные и удобные для малых задач, могут превратить информатику в черчение при более сложных алгоритмах. Паскаль и другие учебные языки, безусловно, эффективны для изучения алгоритмизации или программирования, но мало приспособлены для вычисления выражений типа $693/13$. Здесь лучше использовать калькулятор.

При дефиците машинного времени, а также для сохранения зрения и здоровья эффективность работы за дисплеем должна обеспечиваться предварительной подготовкой учащегося, изучением инструкций. Эффективным является непосредственное редактирование текстов программ (особенно Cut, Copy, Paste), отладка программ (пошаговое выполнение, точки останова, окно Watch и др.)

3.8. Связь теории и практики

Путь от теории, от приобретения знаний до их применения, то есть до практики, в информатике очень короткий, короче даже чем, например, на уроках труда. Учащийся может решить задачу, полезную учителю, классу или школе. Например, он может создать небольшую базу данных «успеваемость», или по школьному оборудованию, для библиотеки, и т.п.

Понятия теории и практики в информатике обнаруживают полную параллельность с общенаучными категориями. Как и в «большой» науке, теория есть средство прогноза,

предсказания или объяснения свойств, поведения компьютерного мира, а практика — средство проверки теории и, с другой стороны, источник гипотез для неё.

Теоретическим методом выглядит доказательство алгоритма без ЭВМ. Но такой способ приемлем для небольших по объему несложных алгоритмов. Эффективнее, конечно, тестирование программы на практике с использованием компьютера. Но в тоже время минимальная система тестов, которая охватывала бы максимум возможных ситуаций (ветвей программы) строится даже для конкретной задачи теоретически.

Другой аспект — это связь теории и практики при изучении непосредственно информатики. Как и в других науках, теория объясняет или предсказывает результат опыта (запуск алгоритмов на компьютере), а практика (работа на компьютере) служит средством проверки теории и источником гипотез, например, о поведении программы. Эти два вида деятельности тесно переплетаются на уровне мышления учащегося.

Комбинированным методом работы является отладка программ. Здесь тесно переплетаются практическое проявление ошибки при компиляции и (или) выполнении программ, теоретический анализ ее причин, их поиск с привлечением определенных знаний из соответствующей области и практическая проверка на компьютере внесённых исправлений.

Тема 4. Методы преподавания и изучения информатики и их реализация на практике

4.1 Обзор методов изучения информатики

Метод — это способ деятельности, направленный на достижения определенной цели. В общей дидактике, как известно, следует различать следующие понятия:

- учение — учебная деятельность учащегося, например, отладка программ самостоятельно;
- преподавание — деятельность учителя, например, разработка инструкций, текстов, заданий для индивидуальной работы и т.д.;
- обучение — их совместная деятельность, например, защита учеником индивидуальных заданий.

Особенность информатики в этом смысле заключается в следующем. Компьютер как посредник между учителем и учеником увеличивает объем относительно независимых видов деятельности учащегося и учителя и сокращает объем их совместной работы. Это связано с тем, что целью курса является в идеале независимость обучаемого при работе с ЭВМ от педагога, а затем и от программиста. Но это не означает выведение педагога от учебной деятельности. Его опыт и знания никогда не заменят никакие обучающие программы.

Как отмечалось раньше, учащийся, работающий за компьютером, более самостоятельно, имеет локальные собственные цели.

Метод преподавания в общем смысле можно понимать как метод управления познавательной деятельностью учащегося.

Метод обучения можно понимать как метод познания обучаемым действительности в специально созданной учебной ситуации.

Известно множество оснований для классификации методов обучения:

- по содержанию обучения;
- по способу восприятия информации: словесные, практические и др.;
- по способу получения знаний (теория и(или) практика);
- по способу реализации обратной связи (контроль учителя, самоконтроль, контроль с помощью компьютера).

4.2 Традиционные методы обучения и информатика

В свое время классическая педагогика осознала недостаточность словесного образования, указав на необходимость приобретения информации не из книг и лекций, а через наблюдение, познание самих предметов и явлений

Предметным методом применительно к информатике является наблюдение учащегося за работой товарища или учителя, а также *наглядные методы*: иллюстрация и демонстрация. Полезно иллюстрировать, прежде всего, то, что «не видно»: например, модель всей памяти ЭВМ или одной ячейки с записью и перезаписью в неё значений. Учитель может демонстрировать работу за компьютером в принципе и молча («делай, как я»). Любой современный компилятор демонстрирует место, и даже смысл ошибки. Учащийся может демонстрировать свои результаты, как учителю, так и товарищу. Особенность компьютерной демонстрации – динамичность и управляемость наглядными образами. Есть возможность вмешаться в процесс демонстрации для индивидуализации темпа или повторений. К демонстрации можно также отнести современные средства презентации (например, PowerPoint).

4.3 Словесно-фронтальные методы

Рассказ. Характерный признак рассказа – яркое, занимательное, эмоциональное повествование. Оно выполняется, как правило, в следующей последовательности: вступление, изложение, заключение. Знания при этом приобретаются теоретически, логика рассуждений здесь нестрогая: аналогии, примеры. Обратная связь при этом по степени внимания. Можно попросить учащегося воспроизвести услышанное.

Основное назначение: передача конкретных сведений. Подходящие для такого метода темы: области применения ЭВМ, история развития вычислительной техники и компьютерных технологий, вирусы. Это можно поручить и «сильным» ученикам, так как рассказ имеет смысл использовать для изучения несложных тем, которые носят информационный характер.

Лекция. Основные черты этого метода: объемность материала, сложность, логичность, взаимосвязь отдельных вопросов. Поэтому критерий выбора тем для лекции: трудность в изучении; отсутствие хорошей, понятной и доступной для школьника литературы и др. Поэтому подходящими темами могут быть, например, такие: устройство ЭВМ, системы счисления и представления информации в памяти ЭВМ, вспомогательные алгоритмы. Примерами неудачных тем являются, например, следующие: клавиатура, подготовка и редактирование текстов в текстовом редакторе и др. Дидактическая функция аналогична, как и для рассказов: передача знаний, сведений. Логика рассуждений — в основном дедуктивная, строгая, но могут использоваться и такие умственные операции, как аналогия и сравнение. Обратная связь во время лекции обычно слабая и чаще всего отсроченная, так как в виду сложности материала надо дать время на обдумывание.

Классический вид лекции как диктовка, пересказ основных положений, фактов для информатики мало эффективен. Необходимо, чтобы лекция носила проблемный характер, по возможности использовала раздаточный материал (например, перечень и вид стандартных процедур и функций по определенной теме), компьютерные средства демонстрации.

Беседа. Особенность этого метода – системы управляющих вопросов, ведущих обучаемого к заранее намеченной преподавателем цели. Как правило, обсуждаются наиболее важные вопросы до и после практики. Для беседы можно выбирать также темы, по которым учащиеся уже что-нибудь знают: начальные сведения по ПЭВМ, понятия алгоритма. В тоже время учащиеся получают новые знания, систематизируют то, что было им известно. Поэтому дидактическая функция беседы — приобретения и упорядочивание знаний. Метод носит в основном теоретический характер, но может опираться на опередившую практику.

Инструктаж. Черты этого метода – краткость, повелительная форма. Логика почти не привлекается, развернутых рассуждений нет. Примеры тем: техника безопасности в компьютерном кабинете, начальные сведения о работе с клавиатурой, порядок работы со стандартным или другим программным обеспечением. Дидактическая функция — усвоение сведений и некоторых стандартных способов действия. Инструктаж может сопровождаться показом образца действия. Основная цель - подготовка к практике. Обратная связь выполняется, прежде всего, через практику, а также через устный контроль.

4.4. Мыслительные операции и работа на ЭВМ

Рассмотрим, как реализуются умственные логические операции в связи с ЭВМ. Если учитель учтет их, ему будет проще реализовать конкретную комбинацию методов обучения.

Анализ и синтез в информатике имеют следующую специфику. Например, целью анализа может быть выяснение причин ошибки. При этом процесс выполнения программы расчленяется на шаги. Если поиск ошибки выполняется в диалоге с ЭВМ, путем трассировки программы, то это «грубый» анализ – разложение. Если ошибка ищется за столом, то это мысленное выделение частей.

Примером сложного синтеза может являться создание учащимся модели компьютерной среды, (например, электронные таблицы) на основе понимания отдельных команд и наблюдений за реакцией среды на эти команды.

Сравнение имеет место, когда сопоставляются, например, близкие элементы языка программирования (операторы *Если* и *Выбор*, на *Pascal* операторы *while* и *repeat*, функции и процедуры, статические и динамические массивы и т.п.). Эту же операцию можно использовать при изучении устройства ЭВМ (сравнение оперативной и внешней памяти), какой-нибудь системы (перемещения и копирование текста в *Word* или ячеек в *Excel*) и т.п.

Сравнение – достаточно мощный дидактический прием. С его помощью легче вводить новые понятия. С помощью этого метода сначала указываем на сходство нового элемента с уже изученным, а затем на различие. Используя этот прием, можно более объективно проверить знания учащихся. Одно дело, рассказать в отдельности, например, про массив, и отдельно про файл с типом на языке *Pascal*. Учащийся должен более глубоко знать материал, если необходимо сравнить эти два типа данных: найти общие характеристики и различия, особенности по отношению друг к другу.

Классификация как мыслительная операция имеет место при освоении учащимся достаточно большого объема материала. Особенно актуально это в настоящее время, когда необходимо осваивать большое количество стандартных средств (классов и их методов, самостоятельных функций, компонент и свойств в системах типа *Delphi* и т.п.). К сожалению, в книгах часто такого рода элементы описываются в алфавитном порядке. Такая классификация вполне уместна не при изучении, а при использовании такого пособия в качестве справочного при условии, что материал уже освоен. При изучении, особенно на начальном этапе, удобнее, если такие элементы будут классифицированы по назначению, по важности, по частоте использования на практике, или по другим критериям (например, строковые функции в языке *C++* по типам возвращаемых значений).

Обобщение, индукция и дедукция. Ярким классическим приемом обобщения в информатике является переход от представления об отдельных повторениях похожих действий к команде цикла.

Развернутой формой обобщения является индукция. Имеется в виду не математическая, а неполная индукция. Пример. Учащийся впервые сел за компьютер и в текстовом редакторе нажал клавишу → «стрелка вправо». Если спросить, что произойдет, если нажать эту клавишу, ответ будет однозначным и даже недоуменным. Но не принят во внимание случай, когда курсор уже находится у правого края. В этом примере ход рассужде-

ний учащегося, скорее всего, неосознанный, представляет собой неполную индукцию, причем очень неосторожную.

Несмотря на, казалось бы, неправильный результат в этом примере, такая схема часто используется и ведет к верным умозаключениям, подтверждаются практикой по следующим причинам.

Комбинаторно неисчерпаемое множество возможных состояний компьютера устроено очень регулярно: однотипны ячейки памяти, схемы, регистры, структуры алгоритмов (так называемые базовые управляющие структуры: следование, ветвление и повторение).

Современные программные средства, особенно работающие в среде *Windows*, всегда придерживаются определенных стандартов (меню, кнопки и другие элементы приложения, панель инструментов, строка состояний и т.д.). При разработке своих программ также надо соблюдать общепринятые требования.

Индукция проявляется и как обобщение команд: тело цикла получается из серий частных повторяющихся действий при конкретных значениях параметра.

Индуктивным является и умозаключение о правильности программы на основании конечного числа тестов. Несмотря на не строгость этого вывода, так программирует весь мир. Популярно в информатике вводить новые понятия, отталкиваясь от примеров. Приведем несколько имен (идентификаторов), например, в языке *Pascal*: *MyMin*, *R1*, *r2*, *Cout_of_Max*. И наоборот, запишем имена, которые не являются идентификаторами: Число, $2\max$, $R+1$,... . Анализ этих примеров позволяет сформулировать общеизвестные правила построения идентификаторов. Следует индуктивное обобщение: считать сделанные утверждения истинными не для данных примеров, а вообще для всех имен.

Для успешности такого индуктивного подхода система примеров должна обладать рядом свойств:

- представительностью: примеры представляют наиболее важные случаи;
- минимальностью: примеры не должны повторять одинаковые правила;
- ортогональности: примеры должны быть независимы;
- распространенностью.

В идеале система примеров должна минимально представлять все частные случаи.

Но не рекомендуется оглашать индуктивное заключение типа «Ученик имеет плохие способности», если он не справился с некоторым заданием. И, наоборот, даже единственный, частный успех является основанием для высказывания гипотезы о том, что он вообще способный.

Роль дедукции в школьной информатике скромнее, чем индукции, в силу опережающей практики возможности тут же проверять верность умозаключения опытом, прямо на компьютере.

Дедуктивным является, например, поиск ошибки в программе, если она ищется пошаговым исполнением. Но и в этом случае заключительный этап понимания, «схватывания» ошибки происходит очень быстро, практически неосознанно. Есть все основания говорить в этом случае о так называемой «свернутой» дедукции, идущей от теста, текста программы, дидактического и ожидаемого результата. Заметим, что рекомендуется вести протокол ошибок, особенно сложных, над которыми долго думали, на которые потратили много времени. Это полезно делать как ученику, так и учителю.

Аналогия и перенос. Аналогия – это осторожная индукция. Схема рассуждений следующая. Пусть при обстоятельствах *B* имеет место факт *F*. Обстоятельства *B1* в каком-то смысле подобны обстоятельствам *B*. Значит, можно ожидать, что и при них будет иметь место факт *F*.

Конкретизируем это на таком примере. Пусть в некоторой системе сочетание клавиш (например, *Ctrl+Del*) при редактировании удаляет строку. Пусть есть другая система той же фирмы. Тогда и в ней удаление строки должно выполняться аналогично. Тем са-

мым умение переносится в близкую обстановку более обоснованно, чем при неполной индукции.

Аналогия и перенос имеют место и при составлении и использовании описания команд (или операторов) в общем виде, подобном формулам Бэкуса-Наура. Например, в некоторой системе, чтобы посмотреть программу на рабочем месте учащегося с номером N надо набрать команду, общий вид которой следующий:

$_SEND(<N>)$

Здесь расчет на использование операции переноса этой команды применительно к конкретному получателю. Но при этом есть следующая опасность. Учащийся с РМУ номер 5 может набрать

$_SEND(<5>)$ и даже $_SEND(<N>)$

и будет прав. Так написано в инструкции. Логичнее и понятнее будет, если включим конкретный частный случай:

$_SEND(2)$

и доверимся рассуждению по аналогии тех, кто сидит не на РМУ номер 2.

Абстракция и конкретизация. Эти операции связаны прежде всего с компьютерным *моделированием*. Исходная задача всегда ставится конкретно, в терминах области знания, где она возникла, и её перевод на язык информатики – самостоятельная и трудная проблема. Суть дела – в переводе с конкретного языка на абстрактный. Затем полученные результаты должны быть представлены пользователю или для себя на языке постановки задачи, т.е. выполнен обратный перевод.

Переводом задачи с формального языка на понятный, содержательный является написание *комментариев*, что является, в общем, не простой задачей. Это искусство своего рода, творчество, хотя и не очень сложное. Главное здесь, найти «золотую середину», чтобы было не много пояснений, но в то же время достаточно для понимания. Особенно это актуально для учебных программ и алгоритмов.

Тема 5. Организационные формы обучения

5.1. Общие сведения

Организационные формы обучения (ОФО) являются внешним, видимым проявлением структуры информационных связей между педагогом и учащимися.

В общей дидактике конкретные ОФО различаются по числу участников совместной деятельности (фронтальная, бригадная или звеньевая, индивидуальная, парная работа) и по роли участников учебного процесса (управляет учитель или учащиеся).

С древних времен начался монотонный отход от индивидуального обучения. Индивидуальное обучение стало расточительным по отношению к усилиям учителя. Традиционный учебник представлял собой первый шаг к синтезу фронтального и индивидуального обучения.

В информатике новым формообразующим элементом этой структуры является компьютер. С одной стороны, появление компьютера возрождает индивидуальные формы обучения, доступные ранее лишь для детей из высших классов. Благодаря обучающей программе или электронному учебнику ученик может воспринимать заложенные в них знания и опыт в своем индивидуальном темпе. Более того, все чаще предлагается не жесткий единообразный алгоритм обучения, а спектр вариантов обучения. Что и в каком порядке познавать, определяет учащийся в зависимости от способностей, уровня знания и других личных качеств.

С другой стороны, за счет тиражирования информации сохраняется и преимущество фронтальных форм: возможность учиться у лучших учителей.

Методика преподавания информатики указывает на две новые ситуации: работа ученика под управлением обучающей программы либо “обучение” компьютера учеником. И

то и другое представляет собой самоуправление познавательной деятельностью при немедленном отражении её результатов на экране.

Организационные формы обучения определяются целями, содержанием и методами преподавания, которые тесно взаимосвязаны. На организационную форму влияет содержание обучения. Так как метод обучения определяется содержанием, то он принимает свою форму. Применительно к информатике демонстрация учителем образца деятельности за компьютером (метод обучения) с целью передачи опыта такой деятельности (содержание обучения) наибольшему числу учащихся приобретает адекватную, фронтальную форму занятий.

В информатике в связи с возможностью использования компьютера в пределах одного урока могут комбинироваться многие формы. Особенно это касается сдвоенных уроков, которые более эффективны, так как сокращаются потери времени на подготовку вычислительной техники и вхождение класса в работу.

5.2. Число участников учебного процесса и формы обучения

Фронтальные формы. Как и в других предметах, они применяются при усвоении всеми учащимися одного и того же содержания или вида деятельности. Примеры: Лекция о системах счисления, семинар с выступлением учащегося о применении ЭВМ в медицине, демонстрация нового типа алгоритма или программы (отлаженной первым из решивших задачу) на доске или, лучше, на проекторе.

Достаточно сложной является фронтальная форма работы с использованием всеми учениками компьютеров. Особенно это касается сложных и не очень интересных тем. При этом надо сделать все возможное, чтобы учащиеся не перешли на игру, в *Internet* и т. д. Учитель должен решительно прервать начавшуюся индивидуальную самостоятельность и восстановить единое состояние компьютерной среды на всех ПЭВМ. Надо уметь вовремя перейти к парной или индивидуальной работе.

Обучение в составе бригады (звена) полезна с той точки зрения, что она при умелой организации может отражать реальное разделение труда в коллективе программистов, работающих над одной большой задачей. Типичным примером учебной задачи для использования бригадной формы может являться рисование в графическом режиме большой “картины” состоящей из несколько простых элементов (дом, деревья, солнышко т. д.). Сначала эти части рисуются каждым членом бригады отдельно в виде, например, подпрограмм, а затем собираются на один компьютер.

Основные преимущества бригадной формы: интенсивное взаимное обучение, помощь друг другу а, значит, современная ликвидация пробелов в знаниях. Чтобы уменьшить возможные недостатки такой формы, надо правильно ответить на следующие вопросы. Какой должен быть состав бригады по способностям, по отношению к работе Все должны быть одинакового уровня (например, все “сильные” в одной бригаде, а все “слабые” в другой) ? При этом должны быть и задачи разной сложности. Какое должно быть оптимальное количество членов бригады? Кто должен ею руководить? Постоянным ли должен быть состав бригады? Предлагается подумать над этими вопросами.

Парная работа на ЭВМ является разновидностью бригадной, и сформировалась, казалось бы, из-за нехватки компьютеров. Но было замечено, что и при достаточном их количестве такая форма может быть полезной в начале обучения при освоении или закреплении новой сложной темы.

Учащийся, работающий один за компьютером, может по разным причинам (стеснителен, не у кого спросить, самолюбив и т.д.) не обратиться за помощью. Когда за одним компьютером работают двое, то вероятность незнания одного и того же обоими сразу уменьшается. Смелее обратятся к учителю. С другой стороны, в целом класс будет реже отвлекать его и будет возможность учителю больше внимания уделять слабым ученикам.

При парной работе могут возникать такие же вопросы, как и при бригадной форме. Оба должны быть одного уровня или слабый ученик должен заниматься с сильным? По-

стоянными ли должны быть пары? Как оценивать результат парной работы? Как правильно распределить роли двух учащихся, работающих за одним компьютером? На эти и другие подобные вопросы предлагается ответить самостоятельно.

Заметим, что втроем работать за компьютером менее эффективно и нецелесообразно. Это может быть оправдано вот в каком случае. Когда большой класс или его половина (больше 10 человек) или школьники «тяжелые» в смысле успеваемости, то имеет смысл уменьшить количество одновременно работающих компьютеров, которые используются при объяснении нового материала. Тогда учителю легче руководить синхронной, под диктовку, работой, например, на 5-ти, а не на 12-ти компьютерах, даже если их достаточное количество. Надо учитывать, что такая форма занятия очень сложная. Легче сначала объяснить без компьютера, если, конечно, это целесообразно и в принципе возможно, а затем воспроизвести на компьютере.

Тогда, а также в других случаях, возможна работа **один на один с компьютером**. Отличие этой формы от классической самостоятельной в том, что в виде программы присутствует знание, обладающее собственной активностью. Надо учитывать, что поведение программы зависит от действий учащегося. В этом случае возобновляется фронтальное обучение, но с индивидуальными темпом и способами усвоения.

Один и без компьютера. Для осмысления того, что происходило за компьютером, бывает полезно от него удалиться во время урока, особенно при появлении трудных моментов или неожиданных действий ПЭВМ.

Полезно также и отстранение от компьютера тех учащихся, которые не подготовились к занятию: не изучили инструкцию по выполнению работы в диалоговом режиме; не разработали блок-схему алгоритма или не написали программу, предлагавшуюся в качестве домашнего задания, в бумажном варианте. Но, с другой стороны, такая форма не обязательно должна быть как наказание. Иногда надо объяснить, что это делается для его же блага (например, чтобы не потерял свой файл, чтобы надежно его сохранил).

5.3. Формы обучения в зависимости от того, кто управляет

Управление со стороны учителя очевидно при фронтальных формах. Как говорилось ранее, сложнее управлять индивидуальной деятельностью по понятным причинам: ситуация за каждым компьютером практически уникальна. Можно предложить следующие варианты выхода из этого:

- привлечь для помощи сильных учеников;
- «автоформализовать собственный педагогический опыт» (А.Г. Ершов) в виде обучающих программ;
- использовать педагогические знания других учителей, не обязательно в виде обучающих программ.

Работа возможна **под руководством товарища**. Иногда помощь товарища оказывается эффективнее, доступнее и понятнее, чем помощь учителя. Одна из причин этого в том, что обучаемый не боится спросить у товарища такое, что, по его мнению, спрашивать у учителя было бы стыдно. Другая причина в том, что некоторые учащиеся боятся спрашивать у учителя, так как, по их мнению, существует опасность получить плохую оценку.

Если возникнут общие для класса затруднения, то есть смысл выйти на несколько минут из класса. За это время школьники друг у друга могут выяснить все возникшие вопросы намного быстрее, чем по инструкции или с помощью учителя.

5.4. Основные организационные формы обучения.

Лекция имеет два смысла: это и форма, и метод. Лекция всегда фронтальная.

Из всего многообразия вопросов, связанных с лекцией, обратим внимание на следующее. Она может поддерживаться компьютером как средством наглядности и демонст-

рации и, если позволяет материальная база, проводиться в дисплейном классе. Упражнение при этом выполняет учитель. При наличии подготовленных на компьютере конспектов у учащихся усиливается самоуправление познавательной деятельностью. Полезно использовать конспект, в котором слева отпечатан кратко основной материал, а справа оставлено место для комментариев учащегося, куда записываются пояснения, результаты компьютерных экспериментов и т.д.

Во время компьютерной лекции не следует увлекаться демонстрацией готовых программ. Дело в том, что при изучении алгоритмизации и программирования часто бывает полезным сам процесс разработки алгоритма и программы, важно, как учитель думает, размышляет при этом.

Предлагается подумать над следующими вопросами. Какое должно быть соотношение теории и примеров в лекции? В каком порядке рассматривать те или другие вопросы? Какой оптимальный объем материала, рассматриваемый на лекции?

Семинар является переходной формой от фронтальной к индивидуальной работе и поэтому сохраняет свое значение и в изучении информатики.

Прежде чем использовать компьютер, необходимо вырабатывать ряд навыков и умений, так как, например, разрабатывать алгоритм или осваивать новую систему прямо за экраном тяжело и это могут делать не все. Работать без предварительного изучения инструкции расточительно по отношению к машинному времени.

Кроме этого, нужна адекватная форма работы для коллективного осмысления того, что сделано на компьютере, анализ того, что и почему получилось, и анализ типичных ошибок, основных затруднений и путей их преодоления. Это можно делать на семинаре в без машинном варианте.

Лабораторная работа является основной формой в компьютерном классе. Желательно, чтобы перед началом работы учитель проверил, готов ли класс выполнять общие или индивидуальные задания по соответствующей теме. Тем, кто не готов, не стоит садиться за компьютер. Поэтому кроме ЭВМ, в классе должны быть и столы для обычных занятий. Но учащийся не должен воспринимать такое отстранение от компьютера как наказание. Причиной неготовности может быть и излишняя самоуверенность, надежда разобратся за компьютером, по ходу дела.

Требования к предварительной подготовке к лабораторным занятиям повышаются, если школа арендует машинное время в другом учебном заведении. При этом время может быть выделено «залпом», в течение 2-3 недель.

Ситуация меняется при использовании встроенной помощи Help, которая рассчитана на обучение (но не с нуля!) в ходе работы.

Иногда полезно выполнять индивидуальные задания вдвоем. Но при этом возникает ряд вопросов. Например, предлагать одно задание на двоих? Если давать два задания, то они должны быть одного типа, одного уровня сложности или разные? Оба школьника должны быть одинаковы по способностям или разные? Возможны и другие вопросы.

В этом вопросе не должно быть двух крайностей: поощрение коллективной парной работы и, наоборот, полный запрет совместной работы. По мере роста уверенности и компетентности учащихся следует переходить к индивидуальной работе за ПЭВМ.

Более высокой формой работы по сравнению с фронтальными лабораторными, является **индивидуальный практикум**. Его характерные черты:

- разнотипность заданий по сложности. Проблема: найти «золотую середину», чтобы задание было не слишком простое (иначе мало пользы) и, в то же время, посильным (в противном случае ученик не получит удовлетворения от работы). Можно предложить комплекс взаимосвязанных задач (см. тему б);

- большая, по сравнению с лабораторными занятиями, самостоятельность. Связано это с разнотипностью заданий по содержанию внутри одной темы. Например, в теме «Циклические алгоритмы» это может быть и вычисление конечных и бесконечных сумм,

произведений и факториалов, оригинальный вывод, задачи целочисленной арифметики. Ещё более разнообразны задачи темам „Одномерные массивы” и „Матрицы”;

- необходимость обращаться к учебникам, к справочному материалу. Поэтому такая форма занятий, как ни одна другая, учит самостоятельности и работе с литературой.

Учителю необходимо иметь в виду, что при выполнении индивидуальных заданий со стороны учеников, особенно сильных, могут поступать разнообразные, самые неожиданные, не стандартные вопросы.

5.5. Вспомогательные организационные формы обучения

Экскурсия.

Основные её цели:

1) показать «живую» информатику, а точнее её какой-нибудь раздел (часть) в конкретной сфере;

1) скорректировать у учащихся «книжные» теоретические представления о настоящей информатике;

1) сформировать интерес к предмету (разделу), если экскурсия проводится до его изучения;

4) обобщение знаний, их систематизация, показ связи курса с реальностью, если экскурсия проводится на завершающем этапе;

5) профориентация на профессии, связанные с использованием ЭВМ.

Экскурсия не обязательно должна быть на крупное предприятие, в большую организацию. Может быть вполне достаточно, чтобы бухгалтер или секретарь, работник отдела кадров, или библиотекарь показали и прокомментировали конкретную систему в действии, на реальных данных, в «прямом эфире».

Экскурсия должна быть тщательно подготовлена:

— учитель должен договориться кто, когда и в каком объеме, в каком режиме будет показывать;

— не мешало бы предварительно учителю самому ознакомиться, хотя бы кратко, с тем, что будет показано;

— заготовить перечень вопросов, на которые учащиеся должны ответить во время экскурсии, по её окончании, или во время уроков. Например: какой объем информации хранится? Тип ЭВМ, ее характеристики? Точность расчетов? и др.;

— подготовить разъяснения по другим вопросам, на которые не обязательно должны отвечать школьники.

Итог экскурсии – ее коллективное обсуждение, анализ увиденного. Школьники должны выразить свое отношение к проведенному мероприятию.

Факультативные курсы.

Цели факультативных курсов:

— углубление знаний в конкретном разделе информатики (язык программирования, устройство ЭВМ, использование с демонстрацией и изучением конкретных систем и др.);

— показать связи информатики с другими дисциплинами;

— профориентация, в том числе и на профессию педагога;

Особое внимание надо обратить на межпредметный факультатив: информатика и математика, информатика и физика, даже информатика и литература или история и др. Такие курсы полезны по следующей причине. Образование поделено на непересекающиеся дисциплины (темы). Поэтому что учащемуся трудно привести знания, полученные при изучении разных предметов, в какую-то единую систему. Такие факультативные занятия помогают увидеть связи между предметами.

Полезен факультативный курс, связанный с управлением школой. С администрацией можно согласовать и разработать подсистемы: «Оборудование», «Библиотека», «Доку-

ментация», «Расписание» и т. п. Такая работа должна заинтересовать учащихся. Кроме этого, она будет иметь определенный воспитательный эффект.

Не обязательно ставить целью на факультативных курсах изучить углубленно конкретный язык программирования, что обычно практикуется. Отталкиваться надо от конкретных задач. Работая с библиотекаршей, зная ее задачи, можно лучше понять, например, необходимость внешней памяти, то есть файлов и средств работы с ними.

Характерные особенности факультативных занятий:

- большая самостоятельность;
- самоуправление познавательной деятельностью;
- меньшее число обучаемых.

Кружок.

Это наиболее гибкая, глубоко индивидуальная форма работы с разнообразным содержанием. Могут участвовать ученики разных возрастов. Обычно в кружке занимаются те, кто проявил повышенный интерес к предмету. По-видимому, его могут посещать и не очень способные ученики. Главное, чтобы кружок школьник посещал по собственному желанию и его участие (или не участие) не влияло на оценку по предмету.

Для старшего возраста на кружке можно решать объемные задачи, связанные целостным содержанием: разработка определенных баз данных для школы, сервисных программ для учителя информатики или обучающихся (контролирующих) программ по другим предметам.

При работе с младшими школьниками следует избегать уклона в стандартный курс информатики, который они еще не изучали. Позже будет не интересно. Возможное содержание кружка: освоение готовых ПС, способствующих закреплению знаний из других дисциплин, изучаемых параллельно, т. е. обучающих программ. Для поддержания интереса можно рекомендовать простейшие графические пакеты.

Тема 6. Изучение раздела

“Основы алгоритмизации и программирования”

6.1. Методические особенности раздела “ОАиП”

Раздел “Основы алгоритмизации и программирования” (ОАиП) имеет явно выраженную практическую направленность, что способствует усилению связи обучения с жизнью. В этом разделе, как ни в одном другом, большое внимание должно уделяться решению задач и выполнению упражнений.

При решении некоторых задач строится математическая модель и вычислительный алгоритм, требующие обоснований, выходящих за рамки школьной программы. Чтобы больше внимания уделить вопросам алгоритмизации, можно не давать обоснований используемых методов и фактов. Желательно, чтобы таких задач было как можно меньше.

Происходит расширение традиционного, известного из других школьных предметов понятия величина. Вводятся и используются не только числовые, но и литерные величины. Кроме этого, ученики знакомятся с новыми способами организации данных в форме одномерных и двумерных таблиц (массивов).

При изучении программирования школьника надо приучить абсолютно точно по определенным фиксированным правилам записывать алгоритм. Поэтому с самого начала необходимо привить школьникам такие качества, как четкость, ясность, аккуратность в записях, предельную внимательность и сосредоточенность, особенно при работе на ЭВМ.

Необходимо иметь в виду следующую методическую сложность. Алгоритмический стиль мышления, который мы должны сформировать, отличается от математического, хотя и основывается на нем. Решить задачу и составить алгоритм (программу), с помощью которой ЭВМ должна решить ту же задачу, это не всегда одно и то же, то есть требует различных способностей, различных сторон мышления. Например, даже ученики с хорошими математическими способностями не сразу понимают динамический смысл записи алгоритма. Если любое записанное действие при решении математической задачи выполняется всегда, если оно записано, то в программе, например, может выполняться только одна из двух ее ветвей. Некоторая последовательность действий может повторяться, динамически может меняться значение некоторой величины и т. п. Эти трудности возрастают в условиях безмашинного изучения алгоритмов. Как преодолеть эту сложность? Опыт изучения других дисциплин почти не помогает, так как формулы в математике, физике, химии имеют другой, статический смысл. Для лучшего понимания учениками работы алгоритма можно рекомендовать следующие приемы:

- исполнение алгоритма вручную;
- использование блок-схем ;
- выполнение программы по шагам, т. е. трассировка программы, с помощью современных средств программирования (команды *Trace Into*, *Step Over* и др. в современных интегрированных средах программирования).

6.2. Общие методические принципы обучения ОА и П

Рассматриваются принципы, которыми можно руководствоваться при изучении ОА и П как в машинном, так и в безмашинном вариантах.

6.2.1. Принцип многоуровневости

Этот принцип связан с тем, что в алгоритмизации и программировании, как ни в одной другой области деятельности, все темы, элементы языка тесно переплетены между собой. Поэтому некоторую трудность представляет определение порядка изучения понятий и конструкций языка, приемов и методов программирования. Большинство учебников построены таким образом, что сначала рассматриваются элементы выбранного языка программирования (константы и переменные, все их типы, все операции над ними, правила построения выражений и т. д.), а потом последовательно изучаются операторы, методы и технологии программирования. При этом, как правило, сразу в одном пункте приводятся все возможности изучаемого элемента языка, например, оператора, хотя некоторые из них редко используются или достаточно сложные. Аналогично при изучении какой-нибудь другой системы (не обязательно системы программирования) приводится полное описание некоторого элемента, хотя на начальном этапе достаточно знать ее простейшую форму, лишь некоторые ее возможности.

Такую общепринятую методику можно усовершенствовать следующим образом. Параллельное изучение ЯП и средств разработки программы делим на этапы (уровни). Первый из них (основы, начало, быстрое введение) включает лишь некоторые наиболее простые и часто используемые в дальнейшем элементы (типы данных (вещественные и целочисленные), операции над ними (арифметические), операторы, стандартные функции и процедуры, и т.п.), обеспечивающие возможность быстрой разработки простейших программ. После приобретения первоначальных навыков программирования школьник осваивает второй, основной уровень. На этом этапе изучаются новые элементы языка и ОС а также рассматриваются новые возможности тех же конструкций языка и ОС, основы которых были изучены на первом этапе. Первые два уровня должны быть усвоены всеми учениками класса.

Третий, заключительный уровень состоит в изучении дополнительных возможностей системы и языка, которые могут встретиться реже, при разработке более сложных программ. Изучаемые на этом этапе возможности ОС, ЯП позволяют повысить эффективность использования ЭВМ и оптимизировать программы. Такие дополнительные возможности есть смысл изучать со всеми учениками только в специализированных классах, а в обычной школе лишь на дополнительных необязательных занятиях с учениками, проявляющими повышенный интерес и способности к программированию.

6.2.2. Принцип предварительной мотивации

Рекомендуется в качестве основного принципа изучения ОА и П использовать принцип предварительной мотивации элементов выбранного ЯП. Это означает, что любую тему лучше изучать “от частного к общему”, а не наоборот. Элементы языка рекомендуется вводить по следующей схеме:

- 1) наглядная простая задача, которая требует для своего решения введения нового элемента, т. е. показывается необходимость изучаемого элемента;
- 2) пример использования нового элемента при решении этой задачи;
- 3) общий вид конструкции, ее формальное описание;
- 4) примеры на закрепление;
- 5) выполнение индивидуальных заданий на ЭВМ и (или) упражнений без использования компьютеров.

При этом пункты 1, 2, 3 лучше рассмотреть в форме лекции, 4 и 5 на практических занятиях. При этом теорию желательно максимально сокращена, основные методы, приемы программирования должны изучаться на простых наглядных примерах.

Следуя этому принципу, не обязательно всегда соблюдать одинаковый порядок форм занятий: лекция – семинарские занятия – лабораторные работы. При изучении некоторых тем, например, ввод-вывод, эффективнее сначала решить задачу с использованием ЭВМ и (или) выполнить упражнения без ЭВМ, а потом на лекции обобщить, подвести итоги, объяснить новые дополнительные возможности изучаемого элемента и обратить внимание на наиболее сложные вопросы темы.

6.2.3. Принципы сравнения и повторения

При изучении многих тем следует руководствоваться принципом сравнения, который предполагает анализ различных алгоритмов и (или) программ решения одной и той же задачи, выбор из них наилучшего.

Этот же принцип можно использовать также для того, чтобы показать необходимость той или иной конструкции языка, быстрее и лучше понять ее. Для этого можно записать несколько вариантов некоторого фрагмента программы решения одной и той же задачи, используя разные элементы. Можно привести, например, следующие такого рода элементы:

- представление положительных и отрицательных целых чисел в памяти ЭВМ;
- логические операции „и” и „или” (не важно, в каком языке);
- операторы цикла *while* и *for* в языках *Pascal* и *C++*;
- операторы цикла *while* и *Repeat* в языке *Pascal* или *while* и *do ...while* в *C++*;
- операторы ветвления *if* (если) и *Case* (выбор) в языке *Pascal* или *if* и *switch* в *C++*;
- вспомогательные алгоритмы и вспомогательные алгоритмы для вычисления функций в “школьном” алгоритмическом языке;
- процедуры и процедуры-функции в языке *Pascal*;
- функции типа *void* и функции с одним результатом в языке *C++*;
- атрибуты доступа *private* и *public* в объектно-ориентированном программировании.

Один из вариантов, как правило эффективнее и “красивее”, хотя он может быть труднее для изучения.

Благодаря такому методическому приему реализуется на практике принцип повторения. Один из сравниваемых элементов изучается обычно раньше (например, оператор *if*). Повторение при изучении ОА и П в большей степени, чем в математике, играет важную роль. Это связано с тем, что при изучении нового элемента языка, при написании программ по новой теме надо знать практически почти все ранее изученные темы. Например, выражения, операторы *if*, ввода-вывода будут использоваться при написании программ по любой новой теме.

6.2.4. Принцип индивидуальных заданий

При обучении программированию следует исходить из того, что развить алгоритмическое мышление школьника, научить его программировать и решать задачи с применением ЭВМ можно только при условии, если каждый из них будет регулярно составлять программы, выполнит самостоятельно несколько индивидуальных заданий. Каждое такое задание закрепляет и проверяет знания по одной или нескольким наиболее важным темам и предполагает выполнение следующих этапов:

- изучение постановки задачи;
- разработку алгоритма ее решения;
- запись алгоритма в виде программы;
- подготовку программы на компьютере в соответствующей системе программирования;
- отладку и тестирование программы;
- анализ результатов и подготовку отчета.

В отличие от традиционной методики, когда всему классу предлагается общая задача, при такой организации работы школьника исключается формальное переписывание друг у друга. При этом можно учесть способности школьников, варьируя уровнем сложности заданий. Важно также что при выполнении индивидуальных заданий повышается активность на занятиях и ответственность за его своевременное выполнение.

При этом эффективность обучения возрастает, если использовать следующие методические приемы:

- выполнение одного задания вдвоем (наиболее способный ученик со слабым) ;
- доклад наиболее интересных, оригинально, нестандартно выполненных заданий на семинарском занятии и его обсуждение;
- обмен готовыми программами друг с другом с объяснением;
- конкурс на лучшую программу решения одной и той же задачи;
- бригадный метод работы (одно задание на 2-4 человека). Это позволяет ученикам обмениваться идеями друг с другом, искать вместе ошибки, учит работать в коллективе.

Обсуждая этот принцип, возникает вопрос, как распределять варианты заданий. Всем давать задания примерно одинаковой сложности внутри одной темы или учитывать способности учеников? Одинаковым ли должно быть количество заданий или задания по некоторым темам рекомендовать только наиболее „сильным” ученикам? Неправильно поступают те учителя, которые всему классу в качестве домашнего задания предлагают одни и те же или одинаковые по сложности задачи. Более способным ученикам мало пользы от простых заданий, а слабые ученики вынуждены обращаться за помощью к друзьям, родителям, репетиторам и т. д.

6.2.5. Принцип параллельности

Важным принципом изучения ОА и П в машинном варианте, кроме рассмотренных, должен быть принцип параллельности изучения выбранного языка программирования (*Pascal*, *C++* и др.) и интегрированной среды разработки программ. Параллельно с этим можно изучать и операционную систему (например, *Windows*). Это позволит уже с первых уроков часть занятий проводить в дисплейных классах.

Необходимо обратить внимание на то, что не должно быть двух крайностей в изучении начал программирования при наличии в школе компьютеров. Первая из них заключается в том, что сначала в течении некоторого времени, например, одной четверти, изучаются основы ЯП в безмашинном варианте традиционными, методами с мелом у доски и лишь потом осваиваются методы разработки программ и школьников допускают к компьютерам. При таком подходе теряется интерес к программированию и многие его элементы (например, ввод-вывод), динамический смысл выполнения операторов без общения с ЭВМ остаются непонятными длительное время.

Нельзя согласиться и с теми учителями информатики, которые все уроки проводят в дисплейных классах и (или) на несколько минут отключают компьютеры для объяснения нового материала. При такой методике меньше внимания уделяется развитию алгоритмического мышления и наиболее принципиальным общим вопросам и методам программирования, что является самым важным при изучении любого ЯП.

6.3. Формы занятий

С учетом отмеченных принципов рекомендуется организовать изучение ОА и П в машинном варианте с использованием следующих форм: 1) лекция без ЭВМ; 2) семинарское занятие без ЭВМ; 3) лабораторные занятия в дисплейных классах с использованием ЭВМ; 4) индивидуальные занятия и консультации.

Основным принципом организации занятий в школе по ОА и П в условиях систематического использования ЭВМ должно являться взаимосогласованное изложение теории и практики с опережающим изложением практических вопросов во время работы на машине. При этом приоритет должен быть отдан решению задач, поэтому основными формами занятий должны быть семинарские и лабораторные, во время которых решаются задачи и выполняются упражнения. Такие занятия рекомендуется проводить с 1/2 или 1/3 частью класса (10-15 человек), чтобы каждый ученик имел возможность готовить и отлаживать на ЭВМ программы.

Лекцию можно проводить со всем классом или с двумя параллельными классами (до 40 человек) в течении не более 45 минут. Двухчасовая лекция в школе малоэффективна, несмотря на то, что такой формой занятий иногда руководители учреждений образования восхищаются в средствах массовой информации. Опыт показывает, что в этом случае после первого часа ученики с трудом воспринимают материал. Сэкономленное за счет объединения двух классов время можно использовать для индивидуальной работы. Школьникам, проявляющим повышенный интерес к программированию, можно рекомендовать предварительно изучить материал лекции по литературе. Тогда на лекции, практическом занятии он может получить ответ на возникшие во время самостоятельного изучения вопросы.

Распределение часов между разными формами занятий зависит от изучаемой темы, количества учеников на лекции и практических занятиях, уровня их подготовленности. Например, если тема имеет теоретическую направленность, то соотношение может быть таким: 1л: 1/2 сем: 1/2 ЭВМ. При таком варианте семинарское и лабораторное занятия проводятся за 1 урок в дисплейном классе. Если тема имеет практический характер, то можно рекомендовать следующее соотношение: 1/2 л: 1/2 сем: 1 ЭВМ.

Эффективными являются также сдвоенные уроки информатики. Тогда за 2 часа в дисплейном классе можно провести все формы занятий. Два часа подряд без компьютера можно с пользой работать, мастерски используя различные методические приемы и умело варьируя формами занятий (повторение в форме беседы предыдущей темы, лекция по теоретической части нового материала, выполнение небольших упражнений учителем или учениками, беседа по закреплению изученного только что материала). Но при любой организации занятий из медицинских соображений школьник не должен работать за дисплеем более 45 минут подряд в течение дня.

6.4. Изучение ОАиП с использованием компьютеров

Можно предложить несколько разнообразных форм проведения лабораторных занятий с использованием компьютеров. Они выбираются в зависимости от изучаемой темы и поставленной цели занятия, уровня подготовленности учащихся, наличия компьютеров, количества учеников и других факторов.

1) Сначала, не включая компьютеры, учитель дает возможность записать основные команды, порядок работы, а затем на этом же занятии каждый ученик, используя конспект, выполняет на ЭВМ то, о чем было рассказано.

2) Если учеников не очень много (не более 10 человек), объясняемый материал можно сразу показывать на одном (или двух) дисплеях, а затем каждый ученик закрепляет показанные элементы на своем компьютере.

3) Этот вариант обучения предполагает синхронное выполнение учениками действий на ЭВМ по командам, которые учитель записывает на доске. Недостатком этого способа является то, что выполнять команды ученики будут неравномерно, и некоторые из них должны будут ждать следующего указания учителя. Следующую команду учитель может подавать лишь после того, когда предыдущий пункт выполнен всеми учениками.

Эти три методических приема целесообразно использовать на начальном этапе обучения работе на компьютерах или при демонстрации новых возможностей ЭВМ. При этом не следует включать много компьютеров. Чем больше включено компьютеров при объяснении нового материала, тем труднее объяснять его. На первых занятиях, а также когда осваивается новая тема, за одним компьютером можно работать нескольким школьникам.

При закреплении показанных навыков, при выполнении индивидуальных заданий основной является индивидуальная форма работы учителя с учениками, при которой по мере возникновения у них вопросов учитель оказывает необходимую помощь. Теперь можно работать с большим количеством компьютеров (до 15), равным количеству учеников.

Тема 7. Решение задач при изучении ОА и П

7.1. Выбор задач

При выборе задач по ОА и П необходимо руководствоваться следующими принципами. Важно не только и не столько изучить правила конкретного языка программирования и обучить школьника записывать с его помощью алгоритм. Прежде всего, и это самое главное, задачи должны способствовать развитию алгоритмического стиля мышления, помочь овладеть общими методами и приемами программирования. С помощью задач школьник должен освоить основные типы алгоритмов (ветвления, циклы с известным и неизвестным количеством повторений), научиться использовать и обрабатывать данные различных типов (целые, вещественные, логические, символьные, строки, массивы, структуры).

Не существенно, из какой области знаний взяты задачи. Гораздо важнее, можно ли с их помощью научиться разработке качественных программ и освоить фундаментальные понятия информатики и, в частности, алгоритмизации. Но с другой стороны, с целью поддержания меж предметных связей, демонстрации возможностей применения ЭВМ, для разнообразия примеров желательно решать с помощью ЭВМ не только математические задачи. В противном случае школьник, проявляющий способности к программированию, но у которого определены проблемы с математикой, может потерять интерес к информатике.

Задачи должны быть интересными, наглядными, понятными школьнику, разнообразными не только по тематике, но и по методическим приемам, используемым при их решении. Задачи не должны требовать много времени на объяснение их содержательной постановки. Перед выбором каждой задачи должны быть поставлены определенные цели, и ее решение должно способствовать их достижению.

Все задачи по программированию можно разделить на два больших класса: задачи-программы и упражнения.

Первый из них включает задачи на составление алгоритма и полной программы на выбранном языке и их анализ, а в машинном варианте, кроме этого, и её отладку. В упражнениях требуется записать один или несколько вариантов элемента языка (записать выражение, оператор) или части программы и (или) проанализировать их. При анализе программы (ее части) надо ответить на ряд вопросов, например, есть ли ошибки, всегда ли ошибка будет проявляться, как повлияет на результат какое-нибудь изменение в программе и т.п. Первый класс задач является более важным, и его необходимо чаще практиковать, особенно для закрепления одной или нескольких тем и при выполнении индивидуальных заданий.

Но на начальном этапе изучения темы и при объяснении наиболее сложных принципиальных вопросов нельзя игнорировать упражнения, которые играют подготовительную, вспомогательную роль. Упражнения особенно полезны при работе с отстающими школьниками. Частным случаем упражнений являются тесты.

Одним из этапов обучения ОА и П, важным условием умения составлять качественные программы является понимание готовых программ, умение их “читать”. Поэтому полезно выполнять упражнения на восстановление постановки задачи, для которой записана программа или ее часть. В случае затруднения можно рекомендовать исполнить ее для одного или нескольких тестов или составить блок-схему.

Необходимо обратить внимание на такие упражнения, в которых надо сравнить несколько вариантов программы или алгоритма. При этом необходимо выбрать правильный вариант из предложенных. Полезны упражнения, в которых требуется исследовать на эффективность программу или её часть, выбрать наилучший из правильных вариантов или записать, если можно, более эффективный вариант.

7.2. Упражнения на составление и анализ блок-схем

При изучении основных типов алгоритмов, наиболее важных и сложных операторов и приемов программирования полезно использовать задачи и упражнения на составление и анализ блок-схем. Кроме простоты и наглядности, важной отличительной их особенностью и достоинством является то, что они не зависят от языка программирования. Научившись записывать алгоритмы в виде блок-схем, достаточно освоить синтаксические правила конкретного языка, чтобы можно было разрабатывать эффективные программы. Блок-схемы способствуют быстрому развитию алгоритмического мышления. Их можно использовать как в машинном, так и в без машинном вариантах изучения ОА и П. К сожалению, в последнее время в учебниках по программированию этому уделяется недостаточно внимания.

Блок-схемы можно изучать как параллельно с языком программирования, так и последовательно. В первом случае сначала в течение 1-2 уроков учащиеся дают простейшие элементы блок-схем, правила их разработки и требования к ним. При изучении методов программирования и наиболее сложных операторов языка составляется при необходимости блок-схема алгоритма, а потом с ее помощью разрабатывается программа. При таком способе, хотя и требуется больше времени, школьники лучше понимают динамический смысл выполнения программ.

Последовательное изучение блок-схем и языка программирования заключается в том; что на начальном этапе изучения ОА несколько уроков (5-7) учащихся разрабатывают и анализируют блок-схемы различных типов алгоритмов (линейных, разветвляющихся, циклических с известным и неизвестным количеством повторений, итерационных, алгоритмы с вложенными циклами). Затем эти же типы алгоритмов изучаются на уровне программ.

При этом не обязательно решать только математические задачи. Эффективным является составление и анализ блок-схем так называемых “бытовых” алгоритмов. Вот примеры таких алгоритмов, возникающих в повседневной жизни:

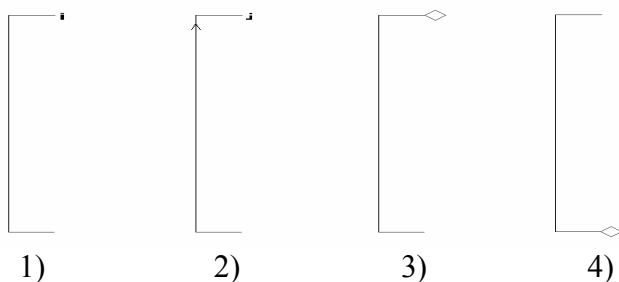
- переход улицы при различных условиях (есть (нет) светофора);
- проход в метро пассажира, пользующегося (не пользующегося) льготами;
- алгоритм, как приготовить и съесть завтрак;
- различные игровые алгоритмы и т.п.

На уровне блок-схем полезно использовать также задачи по управлению движением Робота на клетчатом поле.

Научившись разрабатывать блок-схемы основных типов алгоритмов, можно изучать основные конструкции языка программирования, основываясь на полученных навыках и умениях. Это выполняется качественнее и быстрее, чем без предварительного изучения блок-схем. При таком подходе удачно реализуется принцип повторения. Кроме этого, экономится машинное время, так как разработка блок-схем не требует использования компьютеров. На уровне блок-схем легко показать основные принципы и понятия структурного программирования: пошаговая детализация алгоритма и базовые алгоритмические (управляющие) структуры с одним входом и одним выходом.

Кроме обычных блок-схем в виде фигур, соединенных линиями, при разработке алгоритмов, представляющих вложенные и (или) последовательные циклы, полезно составить схему циклов и (или) набросок программы. Например, при изучении языка Паскаль для изображения различных типов цикла можно использовать следующие наглядные представления:

- 1) For i...to...Do 2) For j...downto...Do 3) While...Do 4) Repeat



Например, для последовательности чисел по итерационной формуле Герона найти $y = \sqrt{a}$. Признак конца ввода – число 0. Тогда схема циклов может иметь, например, вид, представленный на рис.1. Для удобства объяснений циклы можно нумеровать.

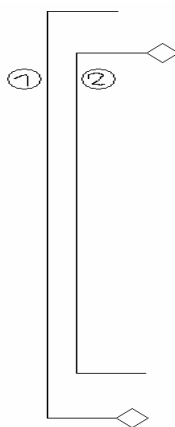


Рис. 1.

В наброске программы не используются никакие геометрические фигуры. Некоторые принципиальные операторы (цикла, if), но не обязательно все, записываются по синтаксическим правилам конкретного языка. А вместо группы некоторых операторов или других элементов записываем на русском языке, что они делают. Например, рассмотрим сле-

дующую задачу. В каждой строке матрицы $A[10,15]$ все положительные числа разделить на первое по порядку отрицательное число. Если отрицательных чисел в строке нет, оставить ее без изменения. набросок программы для этой задачи выглядит следующим образом:

```

Ввод матрицы
For i: = 1 to 10 DO
  Begin
    поиск первого отрицательного числа в i-ой строке
    if есть отрицательное число
      then преобразование i-ой строки
    Else writeln ( 'отриц. числа в', i, '-ой строке нет');
  End
Вывод матрицы.

```

Некоторые из действий, например, поиск первого отрицательного числа в i -ой строке, можно детализировать, используя в зависимости от сложности этой части программы обычную блок-схему, схему циклов, набросок этого фрагмента программы, или операторы языка.

При использовании блок-схем алгоритмов традиционным основным типом задач являются задачи: “составить блок-схему (схему циклов, набросок программы) для решения следующей задачи...”. Кроме этого, на разных этапах изучения ОА и П можно рекомендовать следующие типы задач и упражнений:

- по готовой блок-схеме составить программу (часть программы, записать оператор);
- по заданной программе или ее фрагменту составить блок-схему (схему циклов);
- по заданной схеме циклов написать одну или несколько из возможных постановок задач на определенную тему. Такие упражнения можно использовать на заключительном этапе изучения ОА и П, когда учащиеся получают определенный опыт и будут иметь некоторый набор задач разного типа и их алгоритмов.

Кроме этого, при изучении и использовании блок-схем можно рекомендовать те же типы упражнений и методические приемы, которые перечислены в других параграфах этой главы.

7.3. Тесты

Тесты можно практиковать не только во время проверки знаний (контрольные работы, зачётные занятия, экзамен и т.п.), но и во время обычных занятий, как правило, без использования компьютера. Основная их особенность в том, что проверку правильности ответов на них можно формализовать и поэтому поручить компьютеру. Тесты могут быть нескольких типов.

Основным является тест с вариантами ответов, из которых надо выбрать один или несколько правильных. Вот несколько примеров теста с вариантами ответа.

Дан код:

```

void fun (int x, int *y)
  { *y=x*10; } //1
void main()
  { int a=5,
    *b=new int (2); //2
    ????????;
    cout<< a<< “ “<< (*b); //3
    getch() ; }

```

Как вызвать функцию на месте ????????, чтобы вывести 5 50?

Варианты ответов: 1) ошибка в строке //1. 2) ошибка в строке //2. 3) ошибка в строке //3. 4) $fun6(a,\&b)$; 5) $fun6(a, b)$; 6) $fun6(a, *b)$;

Из предложенных вариантов правильных может быть несколько. Например,

Но надо иметь в виду, что иногда наличие вариантов ответа может оказать определённую помощь. Поэтому более сложным является тест, в котором надо записать ответ.....(продолжение следует).

7.4. Комплекс взаимосвязанных задач

Особое внимание следует обратить на решение комплекса взаимосвязанных задач. При этом задачи могут быть как однотипными в алгоритмическом отношении и одинаковыми по трудности, так и с последовательным возрастанием их сложности.

Пусть дана матрица $A(25,10)$, где $a[i,j]$ – оценка i -го ученика по j -му предмету за одну четверть. Примером комплекса однотипных задач являются следующие:

- 1) Для каждого ученика найти количество пятерок.
- 2) По каждому предмету найти количество пятерок.
- 3) Во всей матрице найти количество пятерок.

В алгоритме первой задачи внутренний цикл по номеру строки, а внешний – по номеру столбца, а во второй задаче наоборот. В алгоритме третьей задачи несущественно, какой цикл внутренний, какой внешний. Но в отличие от первых двух задач, в которых оператор $K5:=0$, где $K5$ – количество пятерок, записывается между операторами цикла, в третьей задаче этот оператор должен быть перед внешним циклом.

Примером последовательности задач с усложнением являются следующие задачи:

- 1) Найти наибольшее из двух чисел.
- 2) Найти наибольшее из трех чисел.
- 3) Найти наибольшее число в одномерном массиве.
 - 1) Найти наибольшее и наименьшее числа в одномерном массиве и их номера.
 - 1) Найти второе наибольшее число и количество его повторений.
- 6) Рассортировать одномерный массив, используя метод выбора (т.е. с помощью нахождения наибольшего элемента).
- 7) Найти наибольший элемент среди наименьших элементов строк матрицы $A[10,20]$.
- 8) Найти k -й наибольший элемент в одномерном массиве.

В такой последовательности задач решение следующей задачи основывается на предыдущих.

Комплекс взаимосвязанных задач можно использовать также при сравнении различных типов алгоритмов, а, значит, и различных операторов. Например, для демонстрации отличия циклов с известным и неизвестным количеством повторений можно предложить следующие три задачи. Найти сумму отрицательных чисел массива, если

- а) массив не рассортирован;
- б) массив рассортирован по возрастанию;
- в) массив рассортирован по убыванию.

В первом из предложенных вариантов условия необходимо проанализировать все элементы массива. В остальных задачах количество повторений цикла зависит от элементов массива. Как только найдём неотрицательное число, можно прекращать цикл.

Для того, чтобы лучше понять отличие двух базовых управляющих структур “цикл пока” (оператор *while*) и “цикл до” (оператор *repeat* языка *pascal*), можно решить две следующие задачи. Задан массив и число x . Найти сумму чисел, предшествующих числу x , 1) не включая это число, 2) включая его. Для каждой из них можно рассмотреть два случая: 1) число x есть в массиве, 2) числа x может не быть в массиве.

Это были примеры последовательности задач, которые используются при изучении различных тем (оператор *if*, одномерные табличные величины и оператор цикла, сложные циклы, двумерные табличные величины). Такой же комплекс задач можно решать и при изучении одной темы. Например, в теме “Строки” можно рекомендовать следующую последовательность задач с возрастающей сложностью:

- 1) В слове найти количество буквы “а”.
- 2) В слове найти, какие буквы встречаются чаще других.

3) В тексте, в котором слова разделены одним пробелом, найти слова, в которых чаще всего встречается заданная буква.

4) В тексте найти слова, в которых чаще всего встречается заданная буква, если между словами один символ-разделитель (; , ! , ? и т.п.) без пробелов.

5) В тексте найти слова, в которых чаще всего встречается заданная буква, если справа и слева от символа-разделителя слов может быть любое количество пробелов.

Использование комплекса взаимосвязанных задач позволяет учесть уровень подготовленности обучаемых. В одном классе можно остановиться, например, на третьем этапе, а в другом (или с наиболее сильными учениками) можно “дойти” и до последней задачи. Начинать также можно с любой задачи. При такой методике можно предоставить ученикам возможность самостоятельно выбирать задачи соответствующего уровня. Это эффективнее случайного набора задач, разделенных по сложности на несколько уровней. Если ученик не может справиться с задачей выбранной сложности, он может, не переключаясь на постановку принципиально другой задачи, “опуститься на этаж ниже”. И, наоборот, если задача для ученика окажется простой, он имеет возможность усложнить ее,

По каждой наиболее важной теме можно подготовить несколько таких последовательностей задач. И тогда этот принцип можно использовать при подготовке индивидуальных и домашних заданий, при проведении проверочных работ. Тогда ученик в состоянии сам оценить уровень своей подготовки. Оценка в таком случае должна зависеть от того, с задачами какого уровня справился ученик.

7.4. Методические приемы, используемые при решении задач

Для более эффективного обучения ОА и П необходимо стремиться разнообразить методические приемы, используемые при решении задач. Рассмотрим некоторые из них.

Для того чтобы показать важность места ключевых слов, можно использовать следующий методический прием. Записываем два варианта алгоритма, отличающихся местоположением ключевых слов, один из которых правильный. Например:

алг приготовление отварной картошки

нач если картошка не очищена

то очистить картошку

сварить картошку

все

съесть картошку

кон

Во втором варианте ключевое слово “все” записывается строкой выше. При анализе необходимо найти правильный вариант, объяснить, всегда ли будет неправильно работать плохой вариант.

Место ключевых слов “нц” – “кц” в циклических алгоритмах можно показать с помощью следующего примера:

алг завтрак1

нач пока в тарелке не меньше ложки каши

нц съесть ложку каши

выпить стакан чая

кц

кон

Аналогично можно поднять ключевое слово кц на строку выше.

Для сравнения с командой “если” можно привести следующий пример:

алг завтрак2

нач если в тарелке не меньше ложки каши

то съесть ложку каши

все

выпить стакан чая

кон

Исполнение и анализ следующих трех вариантов фрагмента программы помогает понять назначение ключевых слов *Begin* – *End* и их роль при совместном использовании операторов *For* и *If* языка Паскаль.

1 вариант ----- <i>For i:=1 to 10 Do</i> <i>If A[i]>0 then</i> <i>Begin k:=k+l;</i> <i>write (' k= ', k)</i> <i>End</i>	2 вариант ----- <i>For i:=1 to 10 Do</i> <i>If A[i]>0 then</i> <i>k:=k+l;</i> <i>write (' k= ', k)</i>	3 вариант ----- <i>For i:=1 to 10 Do</i> <i>Begin If A[i]>0 then</i> <i>k:=k+l;</i> <i>write (' k= ', k)</i> <i>End</i>
--	--	--

Варианты отличаются наличием и расстановкой рассматриваемых ключевых слов. В каждом из вариантов спрашивается, что будет выведено.

Для того чтобы объяснить новый элемент языка, один и тот же пример можно решить разными способами. Например, рассмотрим следующее упражнение из темы “Логические выражения”: переменной *y* присвоить значение $x-2$, если $3 < x < 5$ и 0 в остальных случаях. Можно проанализировать следующие варианты, правильно решающие данную задачу:

- 1) *If x <= 3 then y:=0*
 else If x < 5 then y:=x-2
 else y:=0;
- 2) *y:=0;*
 If x<5 then if x>3 then y:=x-2;
- 3) *If (x>3) and (x < 5) then y:=x-2*
 else y:=0;
- 4) *If (x <=3) or (x >=5) then y:=0*
 else y:=x-2;

При этом некоторые из вариантов можно предложить школьникам записать самостоятельно.

При решении наиболее сложных задач эффективно использование следующего методического приема. Записываем один или два неправильных варианта решения одной и той же задачи, которые часто приводят школьники. Например, преобразовать массив *A[10]* следующим образом: на место первого элемента поместить 10-й, на место второго – 9-й и т. д., то есть “перевернуть массив”.

1-й вариант (неправильный). ----- <i>For i:=1 to 10 Do</i> <i>A[i]:=A[11-i]</i>	2-й вариант (неправильный). ----- <i>For i:=1 to 10 Do</i> <i>begin d:=a[i];</i> <i>a[i]:=a[11-i];</i> <i>a[11-i]:=d</i> <i>end</i>
--	---

Проанализировав их выполнение на конкретном числовом массиве, школьники должны исправить ошибку во втором варианте

For i:=1 to 5 Do

т. е. записать правильный вариант.

Полезно использовать упражнения на выбор наилучшего оператора или фрагмента программы для решения одной и той же задачи. Например, записан следующий оператор *if* языка C++:

```
if (x<0) y=x*x*x*x +x*x +1;
else y=x*x +x +1;
```

Записать решение этого упражнения с помощью сокращенной формы *if*. Некоторые из учеников могут предложить следующий вариант:

```
y:=x*x+x+1;
if (x<0) y=x*x*x*x +x*x +1;
```

Но можно записать более эффективный и “красивый” вариант:

```
if (x<0) x=x*x;
y=x*x+x+1;
```

Его могут записать лишь наиболее способные ученики.

Если с задачей большинство учеников не может справиться, полезным является следующий методический прием. Необходимо дописать пропущенный оператор или его элемент в указанном месте. Например, пусть программа выполняет циклический сдвиг элементов массива на одну позицию вправо. При записи оператора цикла значения начального и конечного параметров должны дописать школьники.

```
const n=10; int a[n]={1, -2, 33, -40, 5, 6, 7, 80, 99, 100};
r=a[n-1];
for (int i=?; i<? ; i++)
a[n-1-i]=a[n-2-i];
a[0]=r;
```

Можно предложить дописать индексы элементов массива в операторе присваивания:

```
a[?]=a[?];
```

При этом заголовок оператора цикла должен быть записан полностью.

При изучении синтаксических правил языка, для иллюстрации стиля программирования и внешнего представления данных можно применять в качестве наглядного материала заранее подготовленные распечатки текстов программ, их результатов, копий экрана. Это существенно экономит время, затрачиваемое на изучение и таких, например, вопросов, как ввод-вывод информации, использование комментариев и других. Это целесообразно использовать при рассмотрении тем, в которых существенна визуальная, наглядная сторона вопроса. Особенно большое значение это имеет в условиях безмашинного обучения или с ограниченным доступом на ЭВМ.

Так как программа лучше воспринимается иногда в целом, когда она вся “перед глазами”, то можно практиковать следующее. Вместо того чтобы писать программу на доске, можно школьникам раздать заранее подготовленные распечатки программ или продемонстрировать готовую программу на компьютере и объяснить ее. Но в то же время этим нельзя злоупотреблять. Дело в том, что при изучении некоторых тем (например, сложные циклы, сортировка массивов) важен сам процесс разработки алгоритма и написания программы, а не ее готовый текст. Важно бывает продемонстрировать учащимся, как учитель думает, рассуждает, как он доходит до того или другого алгоритма, чтобы и ученики вместе с ним размышляли и участвовали в разработке программы или алгоритма.

Почти все рассмотренные выше методические приемы не зависят от изучаемого языка программирования.

Тема 8. Программа курса «Информатика» для VI-X классов (базовый уровень)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Информатика как динамично развивающаяся наука становится одной из тех отраслей знаний, которая призвана готовить современного человека к жизни в новом информационном обществе.

Учебный предмет «Информатика» как самостоятельная дисциплина является образовательным компонентом общего среднего образования. Вместе с тем, выражая общие идеи формализации, он пронизывает содержание многих других предметов и, следовательно, становится дисциплиной обобщающего, методологического плана. Основное назначение базового курса состоит в выполнении социального заказа современного общества, направленного на подготовку подрастающего поколения к полноценной работе в условиях глобальной информатизации всех сторон общественной жизни.

Основные цели учебного предмета «Информатика»:

- формирование основ научного мировоззрения, включающего инвариантные фундаментальные знания в области информатики;
- подготовка учащихся к активной полноценной жизни в условиях информационного общества;
- развитие логико-алгоритмического, пооперационного и системного мышления учащихся;
- формирование у учащихся этических и нравственных норм использования компьютера и компьютерных сетей, бережного отношения к вычислительной технике.

Задачи учебного предмета «Информатика»:

- формирование знаний учащихся в области теории информации, компьютерного моделирования, основ алгоритмизации и программирования;
- подготовка учащихся в области основных содержательных компонентов информационной культуры: технологического, коммуникативного, алгоритмического;
- формирование и развитие интереса учащихся к компьютерным технологиям и методам обработки информации;
- формирование у учащихся умений использования компьютерных информационных технологий в повседневной жизни информационного общества;
- формирование у учащихся умений использования компьютерного моделирования для решения практических задач из различных предметных областей;
- подготовка учащихся к совместной деятельности для решения различных практических задач.

В основу учебного предмета положены фундаментальные понятия информатики: «информация», «компьютер», «алгоритм», «модель». Компьютер рассматривается как средство (инструмент) обработки информации при решении практических задач.

Программа построена по принципу дидактической спирали. На первом этапе (VI-VIII классы) осуществляется первоначальное знакомство учащихся с компьютером, формируются основы компьютерной грамотности. При этом материал VI класса носит пропедевтический характер и направлен на развитие логического и алгоритмического мышления. На втором этапе (IX-X классы) изученный материал рассматривается на более высоком уровне, изучаются современные информационные технологии, а также систематизируются и классифицируются основные понятия.

Основное назначение предмета состоит в формировании основ информационной культуры. При этом предполагается формирование умений и навыков работы на компьютере, а также способностей решения задач с его использованием, опираясь на компьютерное моделирование.

В результате изучения материала базового курса учащиеся должны иметь представление о следующем:

- информации и информационных процессах;
- месте и роли информатики в современном обществе;
- кодировании информации в компьютере, измерении ее объема и хранении;

- аппаратном и программном обеспечении компьютера, основных этапах их создания и развития;
- алгоритмах и их исполнителях;
- технологии обработки текстовой и графической информации;
- технологии обработки информации в электронных таблицах и системах управления базами данных;
- мультимедийных и сетевых технологиях;
- основных этапах построения компьютерной модели и использовании компьютерного моделирования для решения задач.

Данная программа учитывает разнообразие типов компьютерной техники в общеобразовательных учреждениях Республики Беларусь. Название тем и их содержание даны в общем виде, что позволяет легко адаптировать программу к используемым компьютерной технике и программному обеспечению.

Приведенное в программе распределение часов по темам может быть изменено учителем в зависимости от технического и программного обеспечения компьютерного класса с обязательным выполнением требований, предъявляемых программой.

ПРОГРАММА (170 часов)

VI КЛАСС (34 часа)

3. ВВЕДЕНИЕ (2 ч)

Информация. Виды информации. Носители информации.

Введение в учебный предмет «Информатика». Техника безопасности, правила и порядок работы в кабинете информатики.

Основная цель:

сформировать первоначальные представления об информации, месте предмета «Информатика» в школьном курсе.

Учащиеся должны иметь общее представление об информации. Основные понятия темы рассматриваются качественно на примерах деятельности человека с учетом возрастных особенностей учащихся.

Учащиеся должны знать:

смысл понятий «информация», «виды информации», «носитель информации»; «информатика», «информационная технология», правила работы в кабинете информатики.

Учащиеся должны уметь:

приводить примеры информации различных видов, носителей информации.

3. ПЕРВОНАЧАЛЬНЫЕ ПРИЕМЫ РАБОТЫ С КОМПЬЮТЕРОМ (14 ч)

Основные устройства компьютера. Хранение информации в компьютере. Понятие о файле, папке, диске.

Знакомство с клавиатурой. Техника работы с клавиатурой.

Упражнения для снятия утомления при работе за компьютером.

Знакомство с интерфейсом операционной системы. Управление компьютером с помощью мыши. Управление компьютером с помощью меню.

Запуск программы.

Программа для обработки числовой информации «Калькулятор». Назначение и структура программы. Последовательность выполнения операций. Вычисление выражений. Понятие памяти и использование ее при выполнении вычислений. Получение информации о назначении кнопки. Справочная система программы.

Программа воспроизведения звуковой информации: назначение, основы работы.

Программа воспроизведения видео информации: назначение, основы работы.

Программа обработки текстовой информации. Назначение текстового редактора. Загрузка текста. Редактирование текста. Ввод текста. Сохранение текста.

Программа обработки графической информации. Назначение графического редактора. Загрузка изображения. Создание и редактирование изображения. Сохранение изображения.

Основные цели:

сформировать первоначальное представление об основных устройствах компьютера; сформировать начальные умения работы с клавиатурой и мышью; ознакомить с интерфейсом операционной системы и ее стандартными приложениями для обработки информации.

В теме перечисляются и кратко характеризуются основные устройства компьютера. Знакомство с файлами и каталогами осуществляется в объеме, достаточном для сохранения и считывания информации.

Для освоения клавиатуры рекомендуется использовать клавиатурные тренажеры.

Упражнения для снятия утомления при работе за компьютером необходимо проводить в соответствии с действующими санитарными нормами.

Рассмотрение перечисленных стандартных приложений операционной системы осуществляется в объеме необходимом для знакомства учащихся с возможностью про-

грамм при обработке различных видов информации. В качестве элементов редактирования текста достаточно рассмотреть удаление символов. В качестве элементов редактирования изображения достаточно рассмотреть инструмент «Ластик».

Учащиеся должны знать:

смысл понятий «файл», «папка», «диск», «программа» («приложение»); назначение основных устройств компьютера; структуру клавиатуры.

Учащиеся должны уметь:

включать и выключать компьютер; запускать программу на выполнение; вводить информацию с клавиатуры; производить вычисления с помощью программы «Калькулятор», уметь воспроизводить файлы с аудио- и видеoinформацией; выполнять простейшие операции с текстом и изображением: создание, элементы редактирования, загрузка из файла, сохранение в файле.

Контрольная работа по темам 1-2 (1 ч)

3. УЧИМСЯ РАССУЖДАТЬ (15 ч)

Множество и его элементы. Подмножества. Сравнение множеств: больше, меньше, равны. Операции над множествами: пересечение, объединение, разность.

Шифрование (кодирование и декодирование) информации.

Знакомство с понятием «граф».

Высказывание. Истинность высказывания. Логические связки: отрицание (НЕ), конъюнкция (И), дизъюнкция (ИЛИ), импликация (ЕСЛИ...ТО).

Понятие алгоритма. Запись алгоритма. Исполнитель алгоритма. Среда обитания, система команд, система ограничений исполнителя.

Компьютерный исполнитель.

Основная цель:

развить логическое мышление у учащихся, сформировать представления об алгоритме и его исполнителе.

Для формирования алгоритмического мышления следует обращать внимание не на механическое использование компьютерных исполнителей, а на описания алгоритмов. Формирование умений и навыков составления и записи алгоритмов рекомендуется осуществлять на компьютерных исполнителях с минимальной системой команд.

Учащиеся должны знать:

смысл понятий «множество», «элемент множества», «признак множества», «подмножество», «операция над множествами»; «кодирование информации», «декодирование информации»; «граф», «вершина графа», «ребро графа», «маршрут»; «высказывание», «истинность высказывания», «сложное высказывание», «логическая связка», «отрицание», «конъюнкция», «дизъюнкция», «импликация»; «алгоритм», «исполнитель алгоритма», «среда обитания», «команда исполнителя», «система команд исполнителя», «система ограничений исполнителя»; способы задания множества, способы записи алгоритмов.

Учащиеся должны уметь:

задавать множества, выделять подмножества по некоторым признакам, сортировать элементы множества по признаку, сравнивать множества, выполнять операции над множествами, решать задачи с использованием множеств; решать задачи с использованием кодирования и декодирования информации; строить графы для решения задач; формировать высказывания, используя логические связки, определять истинность высказываний, решать задачи, используя логику высказываний; характеризовать исполнителя алгоритма, составлять и выполнять алгоритмы для решения задач (линейной конструкции).

Контрольная работа по теме 3 (1 ч)

Обобщающее занятие (1 ч)

VII КЛАСС (34 часа)

Повторение (1 ч)

3. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ (2 ч)

Система объектов. Элементы системы и связи между ними.

Понятие о процессе. Информационные процессы: передача, обработка, хранение информации.

Основная цель:

сформировать первоначальные представления о системе объектов, об информационных процессах.

Учащиеся должны иметь общее представление об информационных процессах. Основные понятия темы рассматриваются качественно на примерах деятельности человека с учетом возрастных особенностей учащихся.

Учащиеся должны знать:

смысл понятий «система», «процесс», «информационный процесс».

Учащиеся должны уметь:

приводить примеры информационных процессов.

3. ОСНОВЫ РАБОТЫ С ГРАФИЧЕСКОЙ ИНФОРМАЦИЕЙ (8 ч)

Понятие графической информации. Графический редактор. Создание и редактирование изображений.

Операции над фрагментом изображения. Использование текста в графическом редакторе.

Сохранение и загрузка изображения.

Основная цель:

сформировать первоначальные умения работы с графическим редактором.

Понятие графической информации вводится на конкретных примерах. Изучаются принципы и приемы работы с графическим редактором. Рассматриваются операции копирования и перемещения фрагментов изображения.

В зависимости от возможностей используемого редактора рекомендуется рассмотреть и другие операции (масштабирование, повороты и пр.).

Учащиеся должны знать:

смысл понятия «графическая информация»; назначение графического редактора и основы работы с ним; назначение основных операций редактирования изображения.

Учащиеся должны уметь:

запускать графический редактор, создавать и редактировать изображения; выполнять операции над фрагментами; выполнять надписи на изображении; сохранять и загружать изображения.

Контрольная работа по темам 4-5 (1 ч)

3. ОСНОВЫ РАБОТЫ С ТЕКСТОВОЙ ИНФОРМАЦИЕЙ (8 ч)

Понятие текстовой информации. Текстовый редактор.

Ввод и редактирование текста. Операции над фрагментом текста.

Сохранение и загрузка текста.

Основная цель:

сформировать первоначальные умения работы с текстовым редактором.

Понятие текстовой информации объясняется на конкретных примерах. Изучаются простейшие операции редактирования текста: вставка, удаление и замена символов; вставка и удаление пустой строки; разрезание и «склеивание» строк. С целью формирова-

ния устойчивых навыков работы с клавиатурой на данном этапе следует уделять внимание вводу текста.

Более детальное изучение текстового редактора будет продолжено в IX классе.

Учащиеся должны знать:

смысл понятия «текстовая информация»; назначение текстового редактора и основы работы с ним.

Учащиеся должны уметь:

запускать текстовый редактор, вводить текст, выполнять простейшие операции редактирования текста, выполнять операции над фрагментом текста, сохранять и загружать текст.

3. ОСНОВЫ АЛГОРИТМИЗАЦИИ (8 ч)

Линейные алгоритмы. Вспомогательные алгоритмы.

Основная цель:

сформировать представления о вспомогательном алгоритме.

Вспомогательные алгоритмы рассматриваются без параметров.

Учащиеся должны знать:

смысл понятий «вспомогательный алгоритм».

Учащиеся должны уметь:

использовать вспомогательные алгоритмы.

Контрольная работа по темам 6-7 (1 ч)

3. ОБЩЕНИЕ ПОСРЕДСТВОМ КОМПЬЮТЕРА (4 ч)

Сети компьютеров. Обмен сообщениями между пользователями компьютеров: электронная почта, чат.

Программы для отправки (получения) электронной почты и общения в сети в режиме реального времени.

Основные цели:

сформировать представление о компьютерных сетях, научить обмениваться сообщениями в сети.

В этой теме понятие о сети компьютеров рассматривается в общем виде. Возможна классификация на локальные сети и глобальные.

В качестве возможности использования сети компьютеров рассматривается использование сетей для передачи сообщения от одного пользователя к другому: использование электронной почты и общение в сети в режиме реального времени (чат).

При изучении электронной почты достаточно рассмотреть формат построения адреса в виде «Имя_пользователя»@«Составное_имя_компьютера». При изучении технологии работы с почтовым клиентом рекомендуется рассмотреть создание и отправку письма, получение и прочтение письма, вложение файла (при наличии времени).

В качестве программ для общения в режиме реального времени рекомендуется использовать программы, позволяющие обмениваться сообщениями в локальной сети (например, WinChat, NetMeeting).

Учащиеся должны знать:

смысл понятий «компьютерная сеть», возможности использования компьютерной сети для обмена сообщениями между пользователями.

Учащиеся должны уметь:

использовать программу почтового клиента для создания, получения и прочтения сообщения; пользоваться программой для общения в сети в режиме реального времени.

Обобщающее занятие (1 ч)

VIII КЛАСС (34 ч)

Повторение (1 ч)

3. ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В КОМПЬЮТЕРЕ (3 ч)

Представление информации в компьютере.

Системы счисления. Двоичная система счисления.

Количество информации. Измерение информации.

Основная цель:

познакомить учащихся со способами представления информации в компьютере.

В теме рассматриваются общие вопросы представления информации в компьютере и единицы ее измерения. Способы кодирования информации разбираются позднее при изучении технологий ее обработки.

Введение основных понятий системы счисления рекомендуется давать в общем виде (система с основанием p). В качестве примеров можно рекомендовать наиболее распространенные системы: двоичную, восьмеричную, шестнадцатеричную.

Перевод целых чисел из двоичной системы счисления в десятичную и наоборот показывается на конкретных примерах.

Учащиеся должны знать:

смысл понятий «система счисления», «позиционная система счисления», «непозиционная система счисления»; единицы измерения информации.

Учащиеся должны уметь:

определять объем информации, переводить целые числа из десятичной системы счисления в двоичную и наоборот.

3. ОСНОВЫ РАБОТЫ С ФАЙЛОВОЙ СИСТЕМОЙ (8 ч)

Программное обеспечение компьютера. Классификация программного обеспечения.

Операционная система. Взаимодействие пользователя с операционной системой.

Понятие о файловой системе. Имя файла. Тип файла. Указание пути к файлу. Полное имя файла.

Основные операции с файлами и каталогами (папками). Запуск программы.

Основные цели:

сформировать представление о программном обеспечении компьютера, основных функциях операционной системы; сформировать умения работы с файловой системой.

Классификация программного обеспечения компьютера приводится в общем виде (системные, прикладные, инструментальные программы) без подробной характеристики каждой группы. В качестве примеров можно использовать изученные ранее программы.

При изучении понятия операционной системы следует рассматривать ее назначение и функции.

В данной теме рассматриваются основные операции с файлами и каталогами (папками): копирование, удаление и перемещение файлов; создание и удаление каталогов; поиск файлов; навигация по файловой системе. Рекомендуется использовать специальные программы (системные оболочки, файловые менеджеры и т.д.). Целесообразно кратко познакомить учащихся с выполнением операций над файлами и каталогами в командном режиме.

Учащиеся должны знать:

смысл понятий «операционная система», «файловая система», «имя файла», «тип файла», «путь к файлу», «полное имя файла»; назначение программного обеспечения; основные функции операционной системы; основные операции работы с файлами и каталогами (папками).

Учащиеся должны уметь:

выполнять основные операции при работе с файлами и каталогами (папками).

Контрольная работа по темам 9-10 (1 ч)

3. ОСНОВЫ АЛГОРИТМИЗАЦИИ (16 ч)

Типы алгоритмов.

Алгоритмы с повторениями. Алгоритмы с ветвлениями.

Простые и составные условия.

Основная цель:

сформировать умения записи, исполнения и отладки алгоритмов различных типов.

В теме рекомендуется рассмотреть различные способы записи алгоритмов: словесное описание, блок-схема, программа.

Учащиеся должны знать:

типы алгоритмов; основные команды компьютерного исполнителя и правила их записи; правила записи основных алгоритмических конструкций, составных условий.

Учащиеся должны уметь:

составлять и записывать алгоритмы различных типов; осуществлять выполнение и отладку алгоритмов.

Контрольная работа по теме 11 (1ч)

3. ОСНОВЫ РАБОТЫ В ЛОКАЛЬНОЙ КОМПЬЮТЕРНОЙ СЕТИ (3 ч)

Представление о компьютерных сетях. Понятие локальной сети.

Локальные и общие ресурсы сети. Совместное использование общих ресурсов.

Культура совместной работы в сети.

Основная цель:

сформировать представление о локальных компьютерных сетях и их возможностях.

Данная тема рассматривается как логическое продолжение предыдущей. Основное внимание уделяется работе с файлами в локальной сети, запуску программ с общедоступных дисков, правилам использования общих ресурсов. В качестве общих ресурсов достаточно рассмотреть диски и принтеры.

Учащиеся должны знать:

смысл понятий «компьютерная сеть», «ресурсы сети», «общий ресурс», «локальный ресурс».

Учащиеся должны уметь:

работать в локальной сети.

Обобщающее занятие (1 ч)

IX КЛАСС (34 ч)

Повторение (1 ч)

3. АППАРАТНО-ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА (4 ч)

Объекты и системы.

Состав персонального компьютера. Аппаратное обеспечение современного компьютера: процессор, устройства хранения, устройства ввода-вывода.

Объекты операционной системы. Обмен информацией между приложениями.

Хранение и защита информации. Архивация. Антивирусная защита.

Правовая защита программного обеспечения.

Основная цель:

сформировать представление об аппаратном и программном обеспечении компьютера.

Изучение аппаратных и программных средств современного компьютера проводится на базе углубления и расширения материала VII-VIII классов. Характеризуются группы программного обеспечения.

При рассмотрении системных программ делается обзор современных операционных систем и сервисных программ (драйверов устройств, программ-оболочек и т.п.).

При возможности вопросы хранения и защиты информации рекомендуется изучать на конкретных примерах применения современных архиваторов и антивирусных программ.

Возможно рассмотрение других классификаций программных средств.

Учащиеся должны знать:

состав современного компьютера и назначение его устройств; классификацию и назначение программного обеспечения; назначение, возможности и порядок работы с антивирусными программами и программами-архиваторами.

Учащиеся должны уметь:

определять место предлагаемого программного обеспечения в выбранной классификации.

3. ОСНОВЫ РАБОТЫ С БАЗАМИ ДАННЫХ (6 ч)

Базы данных, системы управления базами данных. Типовые задачи обработки информации в базах данных. Информационно-поисковые системы.

Структура базы данных. Поиск и сортировка. Вывод информации из базы данных.

Основная цель:

сформировать первоначальные умения работы с информацией в базах данных.

Основное внимание уделяется работе с готовой базой данных (поиск и вывод информации). В структуре базы данных достаточно рассмотреть понятия «запись», «поле» (не затрагивая «тип поля»). Вопросы создания базы данных и изменения ее структуры в теме не рассматриваются.

По возможности рекомендуется продемонстрировать работу одной из информационно-поисковых систем, например, телефонного справочника, карты.

Учащиеся должны знать:

понятия «база данных», «система управления базой данных», «запись», «поле»; назначение и возможности систем управления базами данных; понятие о структуре базы данных.

Учащиеся должны уметь:

загружать базу данных; осуществлять поиск и сортировку информации в базе данных.

Контрольная работа по темам 13-14 (1 ч)

3. ТЕХНОЛОГИЯ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ (10 ч)

Кодирование текстовой информации в компьютере. Типовые задачи обработки текстовой информации. Назначение текстовых редакторов.

Создание и редактирование текстового документа.

Форматирование текстового документа (символов, абзацев).

Подготовка документа к печати. Параметры страницы.

Сохранение и загрузка документа.

Оформление таблиц.

Вставка объектов.

Основные цели:

познакомить учащихся с типовыми задачами обработки текстовой информации; сформировать умения работы с текстовыми документами.

Изучение материала базируется на углублении и развитии темы «Основы работы с текстовой информацией».

Оформление таблиц осуществляется с использованием возможностей, предоставляемых изучаемым редактором.

При отсутствии в применяемом редакторе возможности вставки объектов, рекомендуется познакомить учащихся с этой технологией теоретически.

Учащиеся должны знать:

понятия «текстовый редактор», «редактирование», «форматирование»; типовые задачи обработки текстовой информации; кодирование текстовой информации; элементы текстового документа; основные операции редактирования, форматирования и печати текстового документа; оформление таблиц; вставка объектов.

Учащиеся должны уметь:

запускать текстовый редактор; создавать и редактировать текстовый документ; сохранять, открывать и готовить к печати текстовый документ.

3. ТЕХНОЛОГИЯ ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ (5 ч)

Кодирование графической информации в компьютере. Растровая и векторная графика. Типовые задачи обработки графической информации.

Программы обработки графической информации и их использование.

Основные цели:

познакомить учащихся с типовыми задачами обработки графической информации; сформировать умения работы с графической информацией.

Основные принципы работы с графическим редактором были изучены в VII классе, поэтому в данной теме рассматриваются инструменты и методы обработки графической информации. Рекомендуется познакомить учащихся с векторным графическим редактором. Рассмотрение кодирования графической информации следует проводить в сравнении с кодированием текстовой.

При отсутствии векторного редактора следует использовать другие средства для создания чертежей, схем и других графических объектов, а освободившиеся часы перенести в тему «Мультимедийные технологии».

Учащиеся должны знать:

понятия «растровая графика», «векторная графика»; типовые задачи обработки графической информации, кодирование графической информации.

Учащиеся должны уметь:

решать типовые задачи обработки графической информации.

Контрольная работа по темам 15-16 (1 ч)

3. МУЛЬТИМЕДИЙНЫЕ ТЕХНОЛОГИИ (5 ч)

Мультимедиа. Состав мультимедиа: текст, графика, звук, анимация, видео. Применение мультимедийных технологий. Аппаратные и программные средства мультимедиа-технологий.

Использование программного обеспечения для подготовки презентации.

Основная цель:

познакомить учащихся с мультимедийными технологиями и их использованием.

Понятие презентации в данной теме рассматривается в широком смысле: как представление информации с использованием имеющихся технических средств, в виде наглядного материала и т.п. Поэтому рекомендуется рассматривать вопросы подготовки элементов презентации с помощью изученных программных средств (текстового и графического редакторов, информационно-поисковых систем и пр.). По возможности используются специализированные программные средства для подготовки презентации.

Учащиеся должны знать:

понятие «мультимедиа»; аппаратные и программные средства мультимедиа-технологий.

Учащиеся должны уметь:

создавать простую компьютерную презентацию.

Обобщающее занятие (1 ч)

X КЛАСС (34 ч)
Повторение (1 ч)

**3. ТЕХНОЛОГИЯ ОБРАБОТКИ ИНФОРМАЦИИ
В ЭЛЕКТРОННЫХ ТАБЛИЦАХ (9 ч)**

Электронные таблицы. Типовые задачи обработки информации в электронных таблицах.

Структура таблицы. Ячейки, столбцы и строки. Ссылки. Типы данных. Ввод и редактирование данных. Форматирование. Ввод и редактирование формул. Копирование ячеек. Функции. Построение диаграмм.

Подготовка таблицы к печати.

Сохранение и загрузка таблицы.

Основные цели:

познакомить учащихся с типовыми задачами обработки информации в электронных таблицах; сформировать умения работы с электронными таблицами.

Знакомство с работой в электронных таблицах следует проводить на примерах решения расчетных задач из различных предметных областей. Рекомендуется рассмотреть применение простейших вычислительных методов.

Учащиеся должны знать:

понятия «электронная таблица», «ячейка», «столбец», «строка», «формат числовых данных», «абсолютная ссылка» и «относительная ссылка»; назначение и возможности электронных таблиц; правила ввода и редактирования чисел, текста и формул; копирование ячеек; порядок подготовки таблицы к печати.

Учащиеся должны уметь:

запускать программное средство; вводить и редактировать данные, выполнять операции над ячейками; сохранять и загружать таблицу.

Контрольная работа по теме 18 (1 ч)

3. ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ (13 ч)

Понятие модели. Моделирование. Виды моделей.

Этапы моделирования. Методы моделирования. Компьютерная реализация модели.

Статические и динамические модели. Моделирование случайных событий.

Исследование моделей из разных предметных областей.

Основная цель:

сформировать умения использовать компьютерные модели для решения практических задач из разных предметных областей.

Предполагается использование готовых компьютерных моделей, а также построение моделей в электронных таблицах.

Учащиеся должны знать:

понятия «модель», «моделирование», «компьютерная модель»; виды моделей; основные этапы и методы моделирования.

Учащиеся должны уметь:

использовать методы компьютерного моделирования при решении практических задач.

Контрольная работа по теме 19 (1 ч)

3. КОМПЬЮТЕРНЫЕ СЕТИ (8 ч)

Локальные и глобальные компьютерные сети. Архитектура компьютерных сетей. Аппаратное и программное обеспечение.

Подключение к компьютерным сетям. Модем.

Глобальная компьютерная сеть Internet. История возникновения и развития. Основные возможности Internet. Основные службы Internet. Принципы адресации в Internet.

Служба World Wide Web (WWW). Основные понятия: «гипертекст», «гипертекстовая ссылка», «Web-страница». Основные возможности браузеров.

Электронная почта. Адрес электронной почты. Программа для работы с электронной почтой. Стиль и этикет при переписке.

Поиск информации в Internet. Принцип построения запроса. Анализ результатов поиска.

Правовые и этические нормы работы в Internet.

Основная цель:

познакомить учащихся с организацией и возможностями сети Internet, с методами и способами поиска и передачи информации.

Для формирования требуемых умений по данной теме необходимо наличие современных компьютеров, объединенных в локальную сеть (доступ в сеть Internet может быть необязателен). При отсутствии технической возможности данная тема излагается теоретически (с использованием имеющихся традиционных средств обучения). При этом объем выделяемого на тему времени может быть сокращен до 4 часов.

Учащиеся должны знать:

понятия «компьютерная сеть», «сервер», «клиент», «Internet», «ресурсы Internet», «World Wide Web», «Web-страница», «гипертекст», «гиперссылка»; «электронная почта», «почтовый сервер», «почтовый клиент», «адрес электронной почты»; аппаратное и программное обеспечение сети; порядок подключения к компьютерной сети; историю возникновения и развития сети Internet; принципы адресации в Internet; службы Internet; назначение, основные возможности браузеров, почтовых программ и порядок работы с ними; правовые и этические нормы работы в Internet.

Учащиеся должны уметь:

использовать программы для навигации в Internet и для работы с электронной почтой; осуществлять поиск информации в Internet.

Примечание. Перечень умений учащихся уточняется в зависимости от возможностей компьютерного класса.

3. ИСТОРИЯ СОЗДАНИЯ И РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ (1 ч)

Вычислительные устройства. Вычислительные машины. Персональные компьютеры. Программное обеспечение.

Основная цель:

познакомить учащихся с основными этапами становления и перспективами развития вычислительной техники и программного обеспечения.

Учащиеся должны знать:

основные этапы становления и перспективы развития вычислительной техники и программного обеспечения.

Программа курса «Информатика» для IX-X классов (повышенный уровень)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебный предмет «Информатика» как самостоятельная дисциплина является образовательным компонентом общего среднего образования. Вместе с тем, выражая общие идеи формализации, он пронизывает содержание многих других предметов и, следовательно, становится дисциплиной обобщающего, методологического плана. Основное назначение учебного курса состоит в выполнении социального заказа современного общества, направленного на подготовку подрастающего поколения к полноценной работе в условиях глобальной информатизации всех сторон общественной жизни.

Учебная программа построена на основе базовой учебной программы для 12-летней школы и предполагает изучение информатики на повышенном уровне. Данная программа расширяет и углубляет знания и умения учащихся в области компьютерных информационных технологий и имеет технологическую направленность.

Основными целями учебной программы являются: формирование у учащихся основ научного мировоззрения в области информатики; этических основ и нравственных норм использования компьютера и компьютерных информационных технологий; развитие логического и алгоритмического мышления, системных подходов к решению задач; подготовка учащихся к активной полноценной жизни в условиях современного общества.

Основными задачами учебной программы являются: формирование знаний учащихся в области основ информационной культуры, компьютерного моделирования; умений использования компьютера для решения различных практических задач; развитие их интересов к компьютерным информационным технологиям и методам обработки информации.

В результате изучения материала учебного курса учащиеся должны иметь представление о следующем:

- технологии обработки текстовой и графической информации;
- технологии обработки информации в системах управления базами данных;
- мультимедийных и сетевых технологиях;
- технологии построения Web-страниц;
- технологии работы в системе компьютерной математики;
- основных этапах построения компьютерной модели и использовании компьютерного моделирования для решения задач.

Данная программа рассчитана только на использование IBM-совместимой техники в общеобразовательных учреждениях Республики Беларусь.

Приведенное в программе распределение учебных часов по каждой теме может быть изменено учителем в объеме до 25 процентов от рекомендуемого в программе объема с обязательным выполнением требований, предъявляемых программой.

ПРОГРАММА IX-X КЛАССЫ (136 ч)

IX КЛАСС (68 ч)

Повторение (1 ч)

1. АППАРАТНОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА (6 ч)

Состав персонального компьютера.

Аппаратное обеспечение современного компьютера: процессор, устройства хранения, устройства ввода-вывода.

Программное обеспечение и его классификация: системное, прикладное, инструментальное.

Хранение и защита информации. Архивация. Антивирусная защита. Правовая защита программного обеспечения.

Бесклавиатурный ввод информации в компьютер. Работа со сканером.

Основная цель:

сформировать представление об аппаратном и программном обеспечении компьютера.

Изучение аппаратных и программных средств современного компьютера проводится на базе углубления и расширения изученного ранее материала. Характеризуются группы программного обеспечения. Возможно рассмотрение других классификаций программных средств.

При рассмотрении системных программ делается обзор современных операционных систем и сервисных программ (драйверов устройств, программ-оболочек и т.п.).

При возможности вопросы хранения и защиты информации рекомендуется изучать на конкретных примерах применения современных архиваторов и антивирусных программ.

Учащиеся должны знать:

состав современного компьютера и назначение его устройств; классификацию и назначение программного обеспечения; назначение, возможности и порядок работы с антивирусными программами, программами-архиваторами, программами работы со сканером.

Учащиеся должны уметь:

определять место предлагаемого программного обеспечения в выбранной классификации; работать с программами-архиваторами и антивирусными программами, программами работы со сканером.

1. ОСНОВЫ РАБОТЫ С БАЗАМИ ДАННЫХ (9 ч)

Базы данных, системы управления базами данных. Типовые задачи обработки информации в базах данных. Информационно-поисковые системы.

Структура базы данных. Поиск и сортировка. Фильтрация данных.

Создание таблиц в базе данных.

Вывод информации из базы данных.

Основная цель:

сформировать первоначальные умения работы с информацией в базах данных.

Основное внимание уделяется работе с готовой базой данных (поиск и вывод информации). В структуре базы данных следует рассмотреть понятия «запись», «поле», «тип поля», которые используются при создании таблиц в базе данных.

Рекомендуется продемонстрировать работу одной из информационно-поисковых систем, например телефонного справочника, карты территории и др.

Учащиеся должны знать:

смысл понятий «база данных», «система управления базой данных», «запись», «поле», «тип поля»; назначение и возможности систем управления базами данных; понятие о структуре базы данных.

Учащиеся должны уметь:

загружать базу данных; осуществлять поиск и сортировку информации в базе данных, выводить базу данных, создавать таблицы в базе данных.

Контрольная работа по темам 1-2 (1 ч)

1. ТЕХНОЛОГИЯ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ (12 ч)

Кодирование текстовой информации в компьютере. Типовые задачи обработки текстовой информации. Назначение текстовых редакторов.

Создание и редактирование текстового документа.

Представление о размере, гарнитуре и стиле шрифта.

Форматирование текстового документа (символов, абзацев).

Подготовка документа к печати. Параметры страницы.

Сохранение и загрузка документа.

Оформление таблиц.

Вставка объектов: фигурный текст, символ, буквица, рисунок, формула, диаграмма, файл.

Слияние документов (письма, конверты, наклейки).

Основные цели:

познакомить учащихся с типовыми задачами обработки текстовой информации; сформировать умения работы с текстовыми документами.

Изучение материала базируется на углублении и развитии навыков учащихся по обработке текстовой информации полученных в VII классе.

При слиянии документов следует обратить внимание на то, что документ-источник оформляется в виде таблицы.

Учащиеся должны знать:

понятия «текстовый редактор», «редактирование», «стиль шрифта», «гарнитура шрифта», «форматирование»; типовые задачи обработки текстовой информации; кодирование текстовой информации; элементы текстового документа; основные операции редактирования, форматирования и печати текстового документа; оформление таблиц; порядок вставки объектов; порядок слияния документов.

Учащиеся должны уметь:

запускать текстовый редактор; создавать и редактировать текстовый документ; сохранять, открывать и готовить к печати текстовый документ, вставлять в документ различные объекты (фигурный текст, символ, буквица, рисунок, формула, диаграмма, файл), выполнять слияние документов (создавать письма, конверты, наклейки).

1. ТЕХНОЛОГИЯ ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ (21 ч)

1.0. Представление графической информации в компьютере (1 час)

Кодирование графической информации в компьютере. Растровая и векторная графика. Типовые задачи обработки графической информации

Основная цель:

познакомить учащихся с типовыми задачами обработки графической информации, возможностями растровой и векторной графики.

Рассмотрение кодирования графической информации следует проводить в сравнении с кодированием текстовой. При изучении темы рекомендуется больше времени уделять практической работе.

Учащиеся должны знать:

понятия «растровая графика», «пиксель», «векторная графика», «направленный отрезок–вектор»; типовые задачи обработки графической информации, кодирование графической информации.

1.0. Технология обработки графической информации в растровой графике (10 час)

Основные приемы рисования и редактирования изображений растровой графики.

Средства выделения и трансформирования изображений.

Использование слоев при создании изображений.

Применение фильтров к изображению.

Учащиеся должны знать:

назначение основных инструментов создания и редактирования изображений; понятие трансформации, возможности выделения изображения в растровой графике и его трансформирования; понятие «слой»; операции со слоями.

Учащиеся должны уметь:

создавать, закрашивать и редактировать простейшие растровые графические изображения; выделять, перемещать и копировать графические изображения и трансформировать изображения; выполнять операции со слоями; применять к графическому изображению фильтры.

1.0. Технология обработки графической информации в векторной графике (10 час)

Основные приемы рисования и редактирования изображений векторной графики.

Средства выделения объектов. Создание и изменение трансформ.

Создание, просмотр и редактирование слоев в векторной графике.

Учащиеся должны знать:

назначение основных инструментов рисования и редактирования изображений векторной графики, средства выделения объектов и изменения трансформ; понятия «служебные слои» и «слой-маски».

Учащиеся должны уметь:

создавать, закрашивать и редактировать простейшие векторные графические изображения; создавать и редактировать символы; выделять, перемещать и копировать графические объекты; создавать трансформы, выполнять операции со слоями.

Контрольная работа по темам 3-4 (1 ч)

7. МУЛЬТИМЕДИЙНЫЕ ТЕХНОЛОГИИ (9 ч)

Мультимедиа. Состав мультимедиа: текст, графика, звук, анимация, видео. Применение мультимедийных технологий. Аппаратные и программные средства мультимедиа-технологий.

Сканирование изображений.

Текст в мультимедиа. Графика и анимация в мультимедиа.

Звук в мультимедиа. Системные звуковые сигналы в мультимедиа. MIDI звук и цифровой звук. Проигрывание мелодий со звуковых компакт-диск с помощью универсальных программных средств. Запись речи в цифровой файл и прослушивание речи.

Использование программного обеспечения для подготовки компьютерных презентаций.

Основная цель:

познакомить учащихся с мультимедийными технологиями и их использованием при подготовке компьютерных презентаций.

Создание компьютерных презентаций выполняется с использованием специализированных программных средств.

Учащиеся должны знать:

понятие «мультимедиа»; аппаратные и программные средства мультимедиа-технологий, возможности применения мультимедийных технологий в различных сферах деятельности человека; компоненты мультимедиа; отличия MIDI-звука от цифрового звука; принципы работы видео.

Учащиеся должны уметь:

создавать простую компьютерную презентацию с использованием специализированных программных средств; сканировать изображения; проигрывать мелодии звуковых файлов на компьютере; записывать разговорную речь и воспроизводить ее; использовать эффекты анимации при создании презентаций.

7. СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ И АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ (7 ч)

6.1. Представление о системах автоматического управления (2 ч)

Назначение и возможности систем автоматического управления. Управляющий алгоритм. Обратная связь. Понятие замкнутых и разомкнутых систем управления.

Виды алгоритмов процессов управления.

Основная цель:

сформировать представление о системах автоматического управления и их использовании.

При изучении темы рекомендуется приводить примеры использования систем автоматического управления из различных областей деятельности человека: экономики, производства, быта и т.д.

Учащиеся должны знать:

смысл понятий «управление», «управляющий алгоритм», «обратная связь», «замкнутые и разомкнутые системы управления»; виды алгоритмов процессов управления.

7.1. Графика в системах автоматизированного проектирования (5 ч)

Место систем автоматизированного проектирования (САПР) в автоматизации производственных процессов. Классификация САПР.

Использование систем автоматизированного проектирования для решения практических задач.

Основная цель:

сформировать представление о системах автоматизированного проектирования и их возможностях.

Учащиеся должны знать:

назначение, состав и классификацию САПР, основное аппаратное и программное обеспечение САПР.

Учащиеся должны уметь:

использовать систему автоматизированного проектирования для построения простейших чертежей, схем.

Обобщающее занятие (1 ч)

X КЛАСС (68 ч)
Повторение (1 ч)

7. ЭЛЕКТРОННЫЕ ТАБЛИЦЫ (12 ч)

Электронные таблицы. Типовые задачи обработки информации в электронных таблицах.

Структура таблицы. Ячейки, столбцы и строки. Ссылки. Типы данных. Ввод и редактирование данных. Форматирование. Ввод и редактирование формул. Копирование ячеек. Функции. Построение диаграмм. Списки и работа с ними (сортировка, фильтры, итоги и сводная таблица).

Использование функции ЕСЛИ для вычислений с условиями.

Подготовка таблицы к печати.

Сохранение и загрузка таблицы.

Обмен информацией между электронными таблицами, текстовым и графическим редакторами.

Основные цели:

познакомить учащихся с типовыми задачами обработки информации в электронных таблицах; сформировать умения работы с электронными таблицами.

Знакомство с работой в электронных таблицах следует проводить на примерах решения расчетных задач из различных предметных областей. Рекомендуется рассмотреть применение простейших вычислительных методов.

Учащиеся должны знать:

понятия «электронная таблица», «ячейка», «столбец», «строка», «формат числовых данных», «абсолютная» и «относительная» ссылки, «фильтры»; назначение и возможности электронных таблиц; правила ввода и редактирования чисел, текста и формул; копирование ячеек; порядок подготовки таблицы к печати; возможности обмена информацией между электронными таблицами и текстовыми документами.

Учащиеся должны уметь:

запускать программное средство; вводить и редактировать данные, выполнять операции над элементами электронной таблицы; сохранять и загружать таблицу; выполнять обмен информацией между электронными таблицами, текстовым и графическим редакторами.

11. СИСТЕМЫ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ (10 часов)

Системы компьютерной математики, их назначение и возможности.

Создание и редактирование документов, содержащих математические расчеты, таблицы и графики, в системе компьютерной математики.

Вычисления с произвольным количеством цифр результата, действия с рациональными дробями, линейные и квадратные уравнения, их решения в аналитическом виде, графическое отделение корней и их нахождение в численном виде.

Построение и сравнение графиков функций $y=ax+b$, $y=ax^2$, $y=a/x$, $y=\sqrt{x}$ и других, изучаемых в курсе математики.

Знакомство с работой в системе компьютерной математики следует проводить на примерах решения различных задач из курсов математики и физики. Результаты рекомендуется представлять графиками, для сравнения и проверки найденных решений использовать встроенные функции системы.

Основная цель:

сформировать начальные умения практической работы в универсальной системе компьютерной математики.

Учащиеся должны знать:

смысл понятий «система компьютерной математики», «символьные вычисления», «численные расчеты».

Учащиеся должны уметь:

производить в системе компьютерной математики простейшие символьные преобразования (подстановки, упрощения, расширение выражений), решать в аналитическом и численном видах простые алгебраические уравнения и системы, выполнять конкретные практически полезные математические расчеты и вычисления, строить графики простых функций, сохранять и распечатывать результаты своей работы.

Контрольная работа по темам 7-8 (1 ч)

11. КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ (14 ч)

Понятие модели. Моделирование. Виды моделей.

Этапы моделирования. Методы моделирования. Компьютерная реализация модели.

Статические и динамические модели. Моделирование случайных событий.

Исследование моделей из разных предметных областей.

Основная цель:

сформировать умения использовать компьютерные модели для решения практических задач из разных предметных областей.

Предполагается использование готовых компьютерных моделей, а также построение моделей в электронных таблицах.

Учащиеся должны знать:

понятия «модель», «моделирование», «компьютерная модель»; виды моделей; основные этапы и методы моделирования.

Учащиеся должны уметь:

использовать методы компьютерного моделирования при решении практических задач.

Контрольная работа по теме 9 (1 ч)

11. КОМПЬЮТЕРНЫЕ СЕТИ (12 ч)

Локальные и глобальные компьютерные сети. Архитектура компьютерных сетей. Аппаратное и программное обеспечение.

Подключение к компьютерным сетям. Модем.

Совместная работа с графическими редакторами в локальной сети.

Ведение диалога в локальной сети.

Передача файлов в локальной сети, совместная работа с приложениями.

Представление о назначении и возможностях сети Internet.

Глобальная компьютерная сеть Internet. История возникновения и развития. Основные возможности Internet. Основные службы Internet. Принципы адресации в Internet.

Служба World Wide Web (WWW). Основные понятия: гипертекст, гипертекстовая ссылка, Web-страница. Основные возможности браузеров.

Электронная почта. Адрес электронной почты. Программа для работы с электронной почтой. Стиль и этикет при переписке.

Поиск информации в Internet. Принцип построения запроса. Анализ результатов поиска. Работа с образовательными ресурсами и электронными каталогами библиотек.

Правовые и этические нормы работы в Internet. Назначение смайликов.

Основные цели:

сформировать представления учащихся о совместной работе в локальной сети с различными программами;

познакомить учащихся с организацией и возможностями сети Internet, с методами и способами поиска и передачи информации.

Изучение материала базируется на углублении и развитии материала, изученного в VII-VIII классах.

Для формирования требуемых умений по данной теме необходимо наличие современных компьютеров, объединенных в локальную сеть, и обязателен доступ в сеть Internet.

Учащиеся должны знать:

понятия «компьютерная сеть», «сервер», «клиент», «Internet», «ресурсы Internet», «World Wide Web», «Web-страница», «гипертекст», «гиперссылка»; «электронная почта», «почтовый сервер», «почтовый клиент», «адрес электронной почты»;

аппаратное и программное обеспечение сети; порядок подключения к компьютерной сети;

возможности совместной работы с различными программами в локальной сети;

назначение и возможности сети Internet;

историю возникновения и развития сети Internet; принципы адресации в Internet; службы Internet;

образовательные ресурсы в Internet по различным предметным областям и электронные каталоги библиотек;

назначение, основные возможности браузеров, почтовых программ и порядок работы с ними;

правовые и этические нормы работы в Internet, назначение смайликов.

Учащиеся должны уметь:

выполнять совместную работу в графическом редакторе, вести диалог, отправлять и принимать файлы в локальной сети;

использовать программы для навигации в Internet и для работы с электронной почтой; осуществлять поиск информации в Internet.

11. ТЕХНОЛОГИЯ СОЗДАНИЯ WEB-СТРАНИЦ (14 ч)

Общий обзор редакторов Web-страниц. Создание Web-страницы средствами Microsoft Office: выбор оформления, работа с гиперссылками, сохранение в формате HTML.

Работа с редактором визуального проектирования Web-страниц. Особенности оконного интерфейса, приемы форматирования, свойства страницы, работа с текстом, таблицами, гиперссылками, списками, фреймами.

Разработка Web-страницы и подключение ее к Internet.

Основная цель:

познакомить учащихся с приемами создания Web-страниц средствами редактора визуального проектирования.

Учащиеся должны знать:

назначение редакторов Web-страниц и их возможности, возможности создания Web-страницы средствами Microsoft Office.

Учащиеся должны уметь:

создать Web-страницу в Microsoft Office: выбрать оформление, стиль, шрифт, работать с гиперссылками, сохранять в формате HTML; и подключить Web-страницу к сети Internet.

11. ИСТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ (2 ч)

Вычислительные устройства. Вычислительные машины. Персональные компьютеры. Программное обеспечение.

Основная цель:

познакомить учащихся с основными этапами становления и перспективами развития вычислительной техники и программного обеспечения.

Учащиеся должны знать:

основные этапы становления и перспективы развития вычислительной техники и программного обеспечения.

Обобщающее занятие (1 ч)

Программа курса «Информатика» для IX-X классов (углубленный уровень)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебный предмет «Информатика» как самостоятельная дисциплина является образовательным компонентом общего среднего образования. Основное назначение учебного курса состоит в выполнении социального заказа современного общества, направленного на подготовку подрастающего поколения к полноценной работе в условиях глобальной информатизации всех сторон общественной жизни. Традиционно наука «Информатика» рассматривается как совокупность методов хранения, обработки и передачи информации. Программа углубленного изучения делится на три части:

- информационные технологии (базовый курс);
- методы алгоритмизации;
- программирование.

Учебная программа построена на основе базовой учебной программы для 12-летней общеобразовательной школы. Учебная программа предназначена для учащихся, изучающих информатику и математику на углубленном уровне.

Программа имеет системно-информационный характер. Она нацелена на формирование у учащихся системно-информационного взгляда на мир, включающего абстрагирование, моделирование, и алгоритмическое мышление, а также навыков владения компьютером как на уровне "грамотного пользователя", так и начинающего программиста. В рамках этих задач значительное внимание уделено изучению программирования и методам алгоритмизации, освоение учащимися основных информационных технологий.

Программа предусматривает расширение и углубление знаний и умений учащихся в области теории построения и записи алгоритмов, знакомство с методологией и технологией программирования. Программа имеет алгоритмическую направленность. Она не привязана к конкретному языку программирования.

Основные цели курса:

- обеспечение подготовки учащихся с ориентацией на профили, углубленно изучающих методы и средства информационных технологий решения задач и программирование;
- реализация в наиболее полной мере возрастающего интереса учащихся к углубленному изучению программирования через совершенствование их алгоритмического и логического мышления;
- раскрытие основных приёмов и методов обработки информации разной структуры и организации с помощью разнообразных систем обработки информации и языка программирования;
- развитие у учащихся информационной культуры.

Приведенное в программе распределение учебных часов по каждой теме может быть изменено учителем в объеме до 25 процентов от рекомендуемого в программе объема с обязательным выполнением требований, предъявляемых программой.

ПРОГРАММА (204 часа)

IX КЛАСС (102 ч)

РАЗДЕЛ «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ» (34 ч)

Повторение (1 ч)

3. АППАРАТНО-ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА (4 ч)

Объекты и системы.

Состав персонального компьютера. Аппаратное обеспечение современного компьютера: процессор, устройства хранения, устройства ввода-вывода.

Объекты операционной системы. Обмен информацией между приложениями.

Хранение и защита информации. Архивация. Антивирусная защита.

Правовая защита программного обеспечения.

Основная цель:

сформировать представление об аппаратном и программном обеспечении компьютера.

Изучение аппаратных и программных средств современного компьютера проводится на базе углубления и расширения материала VII-VIII классов. Характеризуются группы программного обеспечения.

При рассмотрении системных программ делается обзор современных операционных систем и сервисных программ (драйверов устройств, программ-оболочек и т.п.).

При возможности вопросы хранения и защиты информации рекомендуется изучать на конкретных примерах применения современных архиваторов и антивирусных программ.

Возможно рассмотрение других классификаций программных средств.

Учащиеся должны знать:

состав современного компьютера и назначение его устройств; классификацию и назначение программного обеспечения; назначение, возможности и порядок работы с антивирусными программами и программами-архиваторами.

Учащиеся должны уметь:

определять место предлагаемого программного обеспечения в выбранной классификации.

3. ОСНОВЫ РАБОТЫ С БАЗАМИ ДАННЫХ (7 ч)

Базы данных, системы управления базами данных. Типовые задачи обработки информации в базах данных. Информационно-поисковые системы.

Структура базы данных. Поиск и сортировка. Вывод информации из базы данных.

Основная цель:

сформировать первоначальные умения работы с информацией в базах данных.

Основное внимание уделяется работе с готовой базой данных (поиск и вывод информации). В структуре базы данных достаточно рассмотреть понятия «запись», «поле» (не затрагивая «тип поля»). Вопросы создания базы данных и изменения ее структуры в теме не рассматриваются.

По возможности рекомендуется продемонстрировать работу одной из информационно-поисковых систем, например, телефонного справочника, карты.

Учащиеся должны знать:

понятия «база данных», «система управления базой данных», «запись», «поле»; назначение и возможности систем управления базами данных; понятие о структуре базы данных.

Учащиеся должны уметь:

загружать базу данных; осуществлять поиск и сортировку информации в базах данных.

Контрольная работа по темам 1-2 (1 ч)

3. ТЕХНОЛОГИЯ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ (10 ч)

Кодирование текстовой информации в компьютере. Типовые задачи обработки текстовой информации. Назначение текстовых редакторов.

Создание и редактирование текстового документа.

Форматирование текстового документа (символов, абзацев).

Подготовка документа к печати. Параметры страницы.

Сохранение и загрузка документа.

Оформление таблиц.

Вставка объектов.

Основные цели:

познакомить учащихся с типовыми задачами обработки текстовой информации; сформировать умения работы с текстовыми документами.

Изучение материала базируется на углублении и развитии темы «Основы работы с текстовой информацией».

Оформление таблиц осуществляется с использованием возможностей, предоставляемых изучаемым редактором.

При отсутствии в применяемом редакторе возможности вставки объектов, рекомендуется познакомить учащихся с этой технологией теоретически.

Учащиеся должны знать:

понятия «текстовый редактор», «редактирование», «форматирование»; типовые задачи обработки текстовой информации; кодирование текстовой информации; элементы текстового документа; основные операции редактирования, форматирования и печати текстового документа; оформление таблиц; вставка объектов.

Учащиеся должны уметь:

запускать текстовый редактор; создавать и редактировать текстовый документ; сохранять, открывать и готовить к печати текстовый документ.

3. ТЕХНОЛОГИЯ ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ (5 ч)

Кодирование графической информации в компьютере. Растровая и векторная графика. Типовые задачи обработки графической информации.

Программы обработки графической информации и их использование.

Основные цели:

познакомить учащихся с типовыми задачами обработки графической информации; сформировать умения работы с графической информацией.

Основные принципы работы с графическим редактором были изучены в VII классе, поэтому в данной теме рассматриваются инструменты и методы обработки графической информации. Рекомендуется познакомить учащихся с векторным графическим редактором. Рассмотрение кодирования графической информации следует проводить в сравнении с кодированием текстовой.

При отсутствии векторного редактора следует использовать другие средства для создания чертежей, схем и других графических объектов, а освободившиеся часы перенести в тему «Мультимедийные технологии».

Учащиеся должны знать:

понятия «растровая графика», «векторная графика»; типовые задачи обработки графической информации, кодирование графической информации.

Учащиеся должны уметь:

решать типовые задачи обработки графической информации.

Контрольная работа по темам 3-4 (1 ч)

3. МУЛЬТИМЕДИЙНЫЕ ТЕХНОЛОГИИ (5 ч)

Мультимедиа. Состав мультимедиа: текст, графика, звук, анимация, видео. Применение мультимедийных технологий. Аппаратные и программные средства мультимедиа-технологий.

Использование программного обеспечения для подготовки презентации.

Основная цель:

познакомить учащихся с мультимедийными технологиями и их использованием.

Понятие презентации в данной теме рассматривается в широком смысле: как представление информации с использованием имеющихся технических средств, в виде наглядного материала и т.п. Поэтому рекомендуется рассматривать вопросы подготовки элементов презентации с помощью изученных программных средств (текстового и графического редакторов, информационно-поисковых систем и пр.). По возможности используются специализированные программные средства для подготовки презентации.

Учащиеся должны знать:

понятие «мультимедиа»; аппаратные и программные средства мультимедиа-технологий.

Учащиеся должны уметь:

создавать простую компьютерную презентацию.

РАЗДЕЛ «АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ» (68 ч)

3. ЭТАПЫ РЕШЕНИЯ ЗАДАЧ С ПОМОЩЬЮ ЭВМ (2 ч)

Разработка алгоритма. Обоснование. Представление. Анализ.

Основная цель:

сформировать первоначальные знания об этапах решения задач на ЭВМ.

Учащиеся должны знать:

основные этапы решения задач на ЭВМ.

3. ОСНОВЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ (8 ч)

Алфавит языка, общая структура программы, величины и их идентификаторы, стандартные типы данных, выражения и стандартные арифметические функции, оператор присваивания, операторы ввода-вывода.

Назначение и порядок работы в среде программирования.

Составление и реализация линейных алгоритмов. Отладка программы.

Использование текстовых файлов для организации ввода-вывода данных.

Основные цели:

сформировать первоначальные знания о языке программирования; сформировать умения и навыки записи текстов программ и их отладку на этом языке.

Учащиеся должны знать:

структуру программы на языке программирования; базовые элементы языка программирования; операторы ввода-вывода, арифметические функции.

Учащиеся должны уметь:

составлять простейшие программы с использованием арифметических функций.

3. УПРАВЛЯЮЩИЕ СТРУКТУРЫ ЯЗЫКА (14 ч)

Логические величины, операции, выражения. Логические выражения в качестве условий.

Программирование алгоритмов с использованием условного оператора и оператора выбора. Программирование циклических алгоритмов.

Поиск НОД, НОК натуральных чисел.

Составление смешанных алгоритмов с использованием ветвлений и повторений.

Основная цель:

сформировать умения программировать с использованием управляющих структур языка.

Учащиеся должны знать:

операторы и методы программирования алгоритмов с ветвлениями и повторениями.

Учащиеся должны уметь:

составлять программы с ветвлениями и повторениями.

Контрольная работа по темам 6-8 (1ч)

3. ПРОЦЕДУРЫ И ФУНКЦИИ (8 ч)

Порядок организации и описания процедур и функций, фактические и формальные, локальные и глобальные параметры. Рекурсия.

Основные цели:

сформировать умения использовать пользовательские процедуры и функции при решении задач.

Учащиеся должны знать:

формат и порядок описания процедур и функций пользователя, их место в структуре программы; понятия фактического и формального, глобального и локального параметров; отличительные особенности функций от процедур; определение рекурсивных процессов.

Учащиеся должны уметь:

самостоятельно создавать процедуры и функции пользователя; реализовывать алгоритмы с использованием рекурсивных подпрограмм.

3. ОДНОМЕРНЫЕ МАССИВЫ (12 ч)

Работа с одномерными массивами: описание массивов, заполнение и вывод элементов массива, поиск, замена элементов с заданными свойствами, подсчет количества или сумм искоемых элементов в массиве, поиск максимального и минимального элемента массива, перестановка элементов массива, удаление или вставка элемента в массиве.

Упорядочение элементов массива. Методы сортировок. Задачи на использование сортировок.

Бинарный поиск.

Основная цель:

формировать умения решать задачи с использованием массива.

Учащиеся должны знать:

формат описания одномерного массива; методы сортировки одномерных массивов.

Учащиеся должны уметь:

создавать, преобразовывать, вести поиск элементов с заданными свойствами; сортировать элементы массива.

3. СИМВОЛЫ И СТРОКИ (8 ч)

Процедуры и функции обработки строк и символов. Поиск, замена, удаление, преобразование, подсчет символов и слов в тексте.

Основная цель:

формировать умения решать задачи с использованием символьной и строковой информации.

Учащиеся должны знать:

формат описания строк в программе; назначение основных стандартных процедур и функций работы со строками и символами.

Учащиеся должны уметь:

пользоваться стандартными процедурами и функциями работы со строками; описывать и реализовывать алгоритмы, моделирующие процессы обработки строковой информации: поиск, удаление, преобразование, замена.

3. АЛГОРИТМЫ ЦЕЛОЧИСЛЕННОЙ АРИФМЕТИКИ (10 ч)

Разложение на простые множители, поиск делителей числа, простые числа, выделение цифр числа, системы счисления, перевод из одной системы счисления в другую, делимость чисел.

Основная цель:

формировать умения строить алгоритмы целочисленной арифметики.

Учащиеся должны знать:

отдельные понятия целочисленной арифметики для решения задач: представление чисел, обработка цифр числа, делимость чисел и т.д.

Учащиеся должны уметь:

реализовывать алгоритмы целочисленной арифметики.

Контрольная работа по темам 9-12 (1 ч)

3. АЛГОРИТМЫ ДЛИННОЙ АРИФМЕТИКИ (4 ч)

Арифметические операции над «длинными» числами.

Основная цель:

формировать умения работать с «длинными» числами.

Учащиеся должны знать:

организацию хранения в программе чисел, которые не вмещаются в стандартные типы; арифметические операции над «длинными» числами.

Учащиеся должны уметь:

реализовывать алгоритмы длинной арифметики.

X КЛАСС (102 ч)

РАЗДЕЛ «АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ» (68 ч)

Повторение (1 ч)

3. МНОЖЕСТВА (6 ч)

Описание, основные операции над множествами, использование множеств при обработке данных.

Основная цель:

формировать умения работать с множествами.

Учащиеся должны знать:

формат описания множества как структурированного типа данных, операции над элементами множеств и самими множествами.

Учащиеся должны уметь:

описывать и реализовывать алгоритмы обработки информации с использованием множеств.

3. ТИПИЗИРОВАННЫЕ И ТЕКСТОВЫЕ ФАЙЛЫ (8 ч)

Запись как структурированный тип данных. Способы описания записей.

Файловый тип данных. Типизированные и текстовые файлы. Процедуры и функции работы с типизированными и текстовыми файлами: порядок организации связи с файлом, открытие и закрытие файла, чтение и запись информации.

Основная цель:

формировать умения работать с записями, типизированными и текстовыми файлами.

Учащиеся должны знать:

порядок записи типизированных и текстовых файлов; назначение основных процедур и функций работы с типизированными и текстовыми файлами.

Учащиеся должны уметь:

пользоваться основными процедурами и функциями работы с файлами; программировать алгоритмы с использованием данных из файлов.

3. ДВУМЕРНЫЕ МАССИВЫ (8 ч)

Описание, заполнение, вывод, подсчет количества или сумм элементов, строк или столбцов, поиск и замена элементов с заданными свойствами, перестановка строк и столбцов матрицы, удаление или вставка строк и столбцов, заполнение матрицы по правилу. Использование текстовых файлов при работе с массивами.

Основная цель:

формировать умения решать задачи с использованием двумерных массивов.

Учащиеся должны знать:

формат описания двумерного массива и основные способы работы с двумерными массивами.

Учащиеся должны уметь:

выполнять основные действия над элементами массивов, использовать текстовые файлы при работе с массивами.

3. РЕКУРРЕНТНЫЕ СООТНОШЕНИЯ (12 ч)

Поиск решения задачи с использованием подзадач. Рекуррентные соотношения и их программирование. Приведение рекуррентных соотношений к замкнутому виду. Поиск решения задачи с помощью расчетной таблицы (одномерной или двумерной).

Основная цель:

формировать понятия «подзадача», «поиск решения задачи с использованием решений подзадач», «рекуррентное соотношение», дать навыки использования расчетной таблицы для записи решений подзадач и поиска решения задачи с помощью расчетной таблицы.

Учащиеся должны знать:

понятия «подзадача» и «рекуррентное соотношение».

Учащиеся должны уметь:

использовать решения подзадач для решения задачи, составлять рекуррентные соотношения, использовать расчетные таблицы для записи решений подзадач.

Контрольная работа по темам 14-17 (1 ч)

3. КОМБИНАТОРИКА (12 ч)

Понятие комбинаторной задачи. Комбинаторные объекты: перестановки, сочетания, размещения, размещения с повторением, перестановки с повторением, сочетания с повторением., подмножества. Операции над комбинаторными объектами.

Основные цели:

формировать понятия «комбинаторные объекты», «перестановка элементов», «сочетание элементов», «генерация комбинаторных объектов»; рассмотреть операции над комбинаторными объектами.

Учащиеся должны знать:

понятия «комбинаторный объект»; алгоритмы нахождения числа перестановок, сочетаний и размещений.

Учащиеся должны уметь:

использовать готовые алгоритмы для генерации комбинаторных объектов и выполнять операции над ними.

3. СЛОЖНОСТЬ И ЭФФЕКТИВНОСТЬ АЛГОРИТМОВ (4 ч)

Понятие размера задачи и вычислительная сложность алгоритма. Поиск эффективных алгоритмов. Нахождение вычислительной сложности алгоритмов из школьных курсов математики и информатики.

Основная цель:

формировать представление об эффективности алгоритмов.

Учащиеся должны знать:

методы разработки эффективных алгоритмов.

Учащиеся должны уметь:

определять сложность алгоритма, использовать на практике методы разработки эффективных алгоритмов.

3. ГЕОМЕТРИЧЕСКИЕ ЗАДАЧИ (10 ч)

Уравнения прямой, размещение точек плоскости относительно заданной прямой. Расстояние между точками и от точки до прямой. Точка пересечения отрезков. Взаимное расположение отрезков.

Понятие многоугольника. Площадь многоугольника. Взаимное расположение фигур.

Основная цель:

формировать понятия о методах представления геометрических объектов, развивать пространственное мышление.

Учащиеся должны знать:

способы представления геометрических объектов; размещение точек плоскости относительно заданной прямой; определение расстояния между точками и от точки до прямой; Нахождение точки пересечения отрезков, взаимное расположение отрезков; понятие многоугольника, порядок вычисления площади многоугольника; взаимное расположение фигур.

Учащиеся должны уметь:

реализовывать геометрические задачи на языке программирования.

Контрольная работа по темам 18-20 (1 ч)

3. ПРОГРАММИРОВАНИЕ ГРАФИКИ (5 ч)

Основные процедуры ввода графической информации. Построение отрезков, прямоугольников, окружностей. Работа с цветом.

Основная цель:

познакомить учащихся с графическими возможностями языка программирования.

Учащиеся должны знать:

основные процедуры для работы с графикой.

Учащиеся должны уметь:

составлять программы, содержащие графические объекты.

РАЗДЕЛ «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ» (34 ч)

3. ТЕХНОЛОГИЯ ОБРАБОТКИ ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ (9 ч)

Электронные таблицы. Типовые задачи обработки информации в электронных таблицах.

Структура таблицы. Ячейки, столбцы и строки. Ссылки. Типы данных. Ввод и редактирование данных. Форматирование. Ввод и редактирование формул. Копирование ячеек. Функции. Построение диаграмм.

Подготовка таблицы к печати.

Сохранение и загрузка таблицы.

Основные цели:

познакомить учащихся с типовыми задачами обработки информации в электронных таблицах; сформировать умения работы с электронными таблицами.

Знакомство с работой в электронных таблицах следует проводить на примерах решения расчетных задач из различных предметных областей. Рекомендуется рассмотреть применение простейших вычислительных методов.

Учащиеся должны знать:

понятия «электронная таблица», «ячейка», «столбец», «строка», «формат числовых данных», «абсолютная ссылка» и «относительная ссылка»; назначение и возможности электронных таблиц; правила ввода и редактирования чисел, текста и формул; копирование ячеек; порядок подготовки таблицы к печати.

Учащиеся должны уметь:

запускать программное средство; вводить и редактировать данные, выполнять операции над ячейками; сохранять и загружать таблицу.

Контрольная работа по теме 22 (1 ч)

3. ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ (13 ч)

Понятие модели. Моделирование. Виды моделей.

Этапы моделирования. Методы моделирования. Компьютерная реализация модели.

Статические и динамические модели. Моделирование случайных событий.

Исследование моделей из разных предметных областей.

Основная цель:

сформировать умения использовать компьютерные модели для решения практических задач из разных предметных областей.

Предполагается использование готовых компьютерных моделей, а также построение моделей в электронных таблицах.

Учащиеся должны знать:

понятия «модель», «моделирование», «компьютерная модель»; виды моделей; основные этапы и методы моделирования.

Учащиеся должны уметь:

использовать методы компьютерного моделирования при решении практических задач.

Контрольная работа по теме 23 (1 ч)

3. КОМПЬЮТЕРНЫЕ СЕТИ (8 ч)

Локальные и глобальные компьютерные сети. Архитектура компьютерных сетей. Аппаратное и программное обеспечение.

Подключение к компьютерным сетям. Модем.

Глобальная компьютерная сеть Internet. История возникновения и развития. Основные возможности Internet. Основные службы Internet. Принципы адресации в Internet.

Служба World Wide Web (WWW). Основные понятия: «гипертекст», «гипертекстовая ссылка», «Web-страница». Основные возможности браузеров.

Электронная почта. Адрес электронной почты. Программа для работы с электронной почтой. Стиль и этикет при переписке.

Поиск информации в Internet. Принцип построения запроса. Анализ результатов поиска.

Правовые и этические нормы работы в Internet.

Основная цель:

познакомить учащихся с организацией и возможностями сети Internet, с методами и способами поиска и передачи информации.

Для формирования требуемых умений по данной теме необходимо наличие современных компьютеров, объединенных в локальную сеть (доступ в сеть Internet может быть необязателен). При отсутствии технической возможности данная тема излагается теоретически (с использованием имеющихся традиционных средств обучения). При этом объем выделяемого на тему времени может быть сокращен до 4 часов.

Учащиеся должны знать:

понятия «компьютерная сеть», «сервер», «клиент», «Internet», «ресурсы Internet», «World Wide Web», «Web-страница», «гипертекст», «гиперссылка»; «электронная почта», «почтовый сервер», «почтовый клиент», «адрес электронной почты»; аппаратное и программное обеспечение сети; порядок подключения к компьютерной сети; историю возникновения и развития сети Internet; принципы адресации в Internet; службы Internet; назначение, основные возможности браузеров, почтовых программ и порядок работы с ними; правовые и этические нормы работы в Internet.

Учащиеся должны уметь:

использовать программы для навигации в Internet и для работы с электронной почтой; осуществлять поиск информации в Internet.

Примечание. Перечень умений учащихся уточняется в зависимости от возможностей компьютерного класса.

3. ИСТОРИЯ СОЗДАНИЯ И РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ (1 ч)

Вычислительные устройства. Вычислительные машины. Персональные компьютеры. Программное обеспечение.

Основная цель:

познакомить учащихся с основными этапами становления и перспективами развития вычислительной техники и программного обеспечения.

Учащиеся должны знать:

основные этапы становления и перспективы развития вычислительной техники и программного обеспечения.

УЧЕБНАЯ ПРОГРАММА ДИСЦИПЛИНЫ МЕТОДИКА ПРЕПОДАВАНИЯ ИНФОРМАТИКИ

Предисловие

1 РАЗРАБОТАНА Белорусским государственным университетом

ИСПОЛНИТЕЛИ:

Аленский Н.А. – старший преподаватель кафедры Численных методов и программирования ММФ БГУ

ВНЕСЕНА Кафедрой Численных методов и программирования механико-математического факультета Белорусского государственного университета

ОДОБРЕНА Научно-методическим советом Белорусского государственного университета (Протокол от №)

2 УТВЕРЖДЕНА И ВВЕДЕНА В ДЕЙСТВИЕ приказом Ректора Белорусского государственного университета от № с 200 г .

© БГУ (Электронный документ)

Настоящий руководящий документ (учебная программа дисциплины) не может быть тиражирован и распространен без разрешения Белорусского государственного университета

Издан на русском языке

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Курс «Методика преподавания информатики» (МПИ) для студентов специальности Н – 01.01.00 – Математика (педагогический поток) связан с дисциплиной “Методы программирования и информатика”, который изучался на 1-ом и 2-ом курсах, “Методика преподавания математики” и с другими дисциплинами компьютерного и психолого-педагогического циклов.

Основной целью преподавания курса «Методика преподавания информатики» является подготовить студентов к преподаванию информатики в базовой школе и в средних учебных заведениях. Кроме этого, **цели курса** МПИ следующие:

увидеть место и значение курса школьной информатики в общем образовании школьника, понять и вскрыть связи этого предмета с другими изучаемыми в школе дисциплинами, подготовиться к полноценной работе в компьютеризованной школе;

освоить содержание курса, проведя сравнительный анализ действующих и новых учебников и программ;

овладеть средствами изучения курса, освоить классические и новые методы обучения и организационные формы занятий и научиться применять их на практике;

развить и закрепить логико-алгоритмический и системно-комбинаторный стиль мышления, который является признаком профессионализма преподавателя информатики.

Методика изучения основного, самого сложного раздела школьной информатики "Основы алгоритмизации и программирования" предлагается изучать на примере языка C++.

Учебный курс предназначен для студентов четвёртого курса дневного педагогического отделения механико-математического факультета Белгосуниверситета. **Объём курса:** учебный курс рассчитан на 1 семестр общим объёмом **68 часов**, в том числе **34 часа** лекций и **34 часа** лабораторных занятий.

Программа дисциплины «Методика преподавания информатики» 4 курс.

Введение. Понятие информатики, история преподавания школьной информатики, её современное состояние.

0. *Методика преподавания информатики в системе педагогических знаний.* Цели изучения курса МПИ, связь с другими предметами, особенности курса.

0. *Цели преподавания школьной информатики.* Уровни работы с компьютером. Цели преподавания школьной информатики как единство образования, развития и воспитания. Компьютерная грамотность, образованность и культура.

0. *Принципы дидактики и их реализация в преподавании информатики.* Принцип научности. Сознательность усвоения и деятельности. Доступность и наглядность содержания. Активность и самостоятельность. Индивидуализация и коллективность обучения. Эффективность учебной деятельности. Связь теории и практики.

0. *Методы преподавания и изучения школьной информатики.* Характеристика методов, реализация их на практике. Традиционные и словесно-фронтальные методы. Мыслительные операции и работа на ЭВМ.

0. *Организационные формы обучения (ОФО).* ОФО в зависимости от числа участников. ОФО в зависимости от того, кто управляет обучением. Основные ОФО: лекция, семинар, лабораторные работы, индивидуальный практикум. Вспомогательные ОФО: экскурсия, факультативные занятия, кружок, олимпиады.

0. *Анализ программ и школьных учебных пособий* по информатике. Основные учебные пособия для базовой школы. Учебные пособия для классов с углубленным изучением информатики. Учебные пособия для классов с повышенным уровнем изучения информатики.

0. *Методика изучения основ алгоритмизации и программирования* (на примере C++). Методические особенности и принципы изучения операторов и операций, простых и составных структур данных, подпрограмм и современных технологий программирования. Выбор задач, комплекс взаимосвязанных задач, методические приёмы, используемые при решении задач по программированию.

0. *Изучение устройства ЭВМ, принципов работы и областей использования.* Цели и мотивы изучения. Методические особенности изучения особенностей машинной арифметики, команд и работы процессора, современных устройств ввода-вывода. Методика изучения областей использования ЭВМ.

9. *Методика изучения программного обеспечения.* Методические особенности и принципы изучения файловой системы и работы с файлами, СУБД, электронных таблиц, текстового и графического редакторов.

9. *Учебное моделирование в курсе информатики.* Методические особенности и принципы изучения раздела.

Программа лабораторных занятий по курсу
«Методика преподавания информатики»
4 курс.

1. Методика преподавания информатики в системе педагогических знаний. Цели преподавания школьной информатики.

1. Принципы дидактики и их реализация в преподавании информатики. Методы и организационные формы преподавания и изучения школьной информатики.

1. Анализ программ и школьных учебных пособий по информатике.

1. Изучение устройства ЭВМ, принципов работы и областей использования.

1. Методика изучения программного обеспечения.

1. Учебное моделирование в курсе информатики.

1. Язык C++ (6 час)

1.4 Основы языка C++: операторы, операции, ввод, вывод.

1.4 Функции, введение в объектно-ориентированное программирование, принцип инкапсуляции.

1.4 Указатели, связь указателей с массивами и строками, массивы указателей.

1.4 Структуры, объединения, поля битов, списки.

8. Основы визуального программирования (на примере *Builder*).

7.0. Среда Builder:

7.0. Ввод, вывод, организация диалога. Компоненты TEdit, TLabel. Функции InputBox, ShowMessage, MessageDlg. Задание величин с помощью компонент (TTrackBar (движок), TUpDown (счетчик), TScrollBar (полоса прокрутки)). Вывод числовой и текстовой информации с помощью TCanvas. Компонента StringGrid, ввод, вывод и работа с матрицами.

7.0. Разработка основных элементов современного приложения под Windows. Главное (TMainMenu) и контекстное (TPopupMenu) меню, строка состояния (TStatusBar) и панель инструментов (TToolBar).

7.0. Классификация событий. Обработка событий, связанных с мышью и клавиатурой.

7.0. Обзор и классификация компонент. Работа с текстом. Управляющие и другие компоненты. Использование стандартных диалоговых окон. Работа с датой и временем: тип дата-время, компоненты TMonthCalendar и TDateTimePicker, стандартные процедуры и функции. Компонент TTimer.

7.0. Форма: добавление новой формы, её виды, свойства и события.

7.0. Графические возможности: технология вывода графики, рисование элементарных фигур, задание параметров изображений, вывод текста с помощью методов класса TCanvas. Компоненты TImage, TShape возможности их оформления и анализа. Программное построение графика функций. Вывод стандартных иллюстраций. Мультипликация, использование компоненты TTimer для организации движения.

9. Методика изучения основ алгоритмизации и программирования.

Литература.

Теоретические вопросы методики преподавания информатики:

1. Аленский Н. А. Методические рекомендации по спецкурсу “Информатика в средней школе”. — Мн. : БГУ, ротاپринт, 1992. — 42с.
2. А.И. Бочкин. Методика преподавания информатики: Учеб. пос. Мн.: Выш. шк. , 1998.— 431 с.
3. М.П.Лапчик. Методика преподавания информатики. Учеб. пос. М. — 2001.

Основные учебники для базовой школы по четырёхлетней программе по информатике(7 – 10 класс):

4. Пупцев А.Е. Информатика: учеб. пособие для 7-го (6) кл. учреждений, обеспечивающих получение общ. сред. образования, с рус. яз. обучения, с 12-летним сроком обучения / А.Е. Пупцев, П.Л. Гращенко, А.И. Лапо. — Мн.: Нар. асвета, 2004 (в 2003 – 6 кл.) — 158 с.: ил.
5. Пупцев А.Е. Информатика: учеб. пособие для 8-го кл. учреждений, обеспечивающих получение общ. сред. образования, с рус. яз. обучения, с 12-летним сроком обучения / А.Е. Пупцев, А.И. Лапо. — Мн.: Нар. асвета, 2004. — 168 с.: ил.
6. Пупцев А.Е. Информатика: учеб. пособие для 9-го кл. учреждений, обеспечивающих получение общ. сред. образования, с рус. яз. обучения, с 12-летним сроком обучения / А.Е. Пупцев, А.И. Лапо. Л. А. Вишневская — Мн.: Нар. асвета, 2005. — 197 с.: ил. (2-е изд. 2006).
7. Пупцев А.Е. Информатика: Учеб. пособие для 10-го кл. учреждений, обеспечивающих получение общ. сред. образования, с рус. яз. обучения, с 12-летним сроком обучения — Мн.: Нар. асвета, 2006.

„Старые” учебники для базовой школы (8 – 9 класс):

8. Быкадоров Ю.А., Кузнецов А.Т. Информатика. 8 класс. Учебное пособие для 8 класса. — Мн.: “Народная асвета”, — 2000, 2001, 2002г.г.
9. Быкадоров Ю.А., Кузнецов А.Т. Информатика. 9 класс. Учебное пособие для 9 класса. — Мн.: “Народная асвета”, — 2000, 2001, 2002г.г.

Методические и дидактические материалы к учебникам:

10. Пупцев А.Е. Информатика, 7 класс. Опорные конспекты. Поурочные тематические задания. — Мн.: Новое знание. 2004.
11. Пупцев А.Е. Информатика, 7 класс. Контрольные и самостоятельные работы — Мн.: Новое знание. 2004.
12. Пупцев А.Е. Информатика, 8 класс. Опорные конспекты. Поурочные тематические задания. — Мн.: Новое знание. 2005.
13. Пупцев А.Е. Информатика, 8 класс. Контрольные и самостоятельные работы. — Мн.: Новое знание. 2005.

Учебные пособия для классов с углубленным изучением информатики и с повышенным уровнем изучения информатики.

14. Котов В.М. и др. Методы алгоритмизации. Учебное пособие для 8 – 9 классов с углубленным изучением информатики. — Мн.: “Народная асвета”, — 2000г.
15. Котов В.М., Мельников О.И. Информатика. Методы алгоритмизации. Учебное пособие для 10 – 11 классов с углубленным изучением информатики. — Мн.: “Народная асвета”, — 2000г.

16. Павловский А.И., Пупцев А.Е., Гращенко П.Л. Информатика. 10 класс. Учебное пособие для 10 класса с углубленным изучением информатики. — Мн.: “Народная асвета”, — 2000г.

17. Павловский А.И. и др. Информатика. 11 класс. Учебное пособие для 11 класса с углубленным изучением информатики. — Мн.: “Народная асвета”, — 2001г.

18. Быкадоров Ю.А. и др. Информатика и вычислительная математика. Учебное пособие для 10 - 11 классов с повышенным уровнем изучения информатики. — Мн.: “Народная асвета”, — 2000г.

Замечание.

Учреждением ГИАЦ Министерства образования РБ подготовлен электронный вариант методических рекомендаций на сайте www.giac.unibel.by в разделе “Информатика в школе”.

Литература по языку C++ и визуальным системам Builder и Delphi.

14. Керниган Б. И др. Язык программирования Си. — М.: Финансы и статистика, 1992.

15. Н. А. Аленский. Основы программирования на языке C++: учеб.пособие. — Мн.: АПО, 2005, — 148с.

16. Н. А. Аленский. Практическое руководство по языку C++: учеб.пособие. — Мн.: АПО, 2007, — 278с.

17. Основы алгоритмизации и программирования. Язык Си. Пособие для студентов БГУИР / Демидович Е.Н. — Мн., 2001 – 440с.

18. В.В. Подбельский, С.С. Фомин. Программирование на языке Си. —М.: Финансы и статистика, 1999 – 600с.

19. В.В. Подбельский. Язык C++. Учебное пособие. – М.: Финансы и статистика, 2000. –560с.

20. Г.Шилд. Программирование на BORLAND C++ для профессионалов. —Мн.: ООО «Попури», 1999. — 800с.

21. Г. Шилдт. Самоучитель C++, 3-е издание: — СПб.: BHV — Санкт-Петербург, 1998. — 688с.

22. Х.М. Дейтел, П. Дж. Дейтел. Как программировать на C++. — М.: Бином , 1998. — 1020с.

23. Крэйг Арнуш. Освой самостоятельно Borland C++5. — М.: Бином, 1997. — 719с.

24. Фаронов В.В. Delphi 5. Учебный курс. — М.: ‘Нолидж’, 2000. — 608с.

25. С. Бобровский. Delphi 5. Учебный курс. — СПб: Издательство ‘Питер’, 2000. — 640 с.

26. Архангельский А. Я. Программирование в C++Builder 6 --- М: ЗАО “Издательство БИНОМ”, 2002.

27. Архангельский А. Я. Справочное пособие по C++Builder 6. Книга 1. Язык C++ --- М: ЗАО “Издательство БИНОМ ”, 2002.

28. Архангельский А. Я. C++Builder 6. Справочное пособие. Книга 2. Классы и компоненты. --- М: ЗАО “Издательство БИНОМ ”, 2002. --- 528с.

29. Архангельский А. Я. Решение типовых задач в C++Builder 6.--- М: ЗАО “Издательство БИНОМ ”, 2003.

Сборники задач по программированию:

30. Абрамов С.А. и др. Задачи по программированию. — М.: Наука, 1988, –224с.

31. Н.А. Аленский и др. Задачи и методические рекомендации по программированию. — Мн.: БГУ, 1990, –67с.

32. Н. А. Аленский. Сборник задач по программированию на языке C++. Учебно-методическое пособие для студентов специальностей G 31 03 01 “Математика”, G 31 03 03 “Механика”. – Мн.: БГУ, 2005. – 48с.

33. Заданні і метадычныя рэкамендацыі па вылічальнай практыцы па курсу “Метады праграмавання і інфарматыка”: вучэбна-метадычны дапаможнік для студ. мех.-мат. фак. /Аут.-склад. Г.А.Расолька, А.С.Шыбут, Л.Л.Кузняцова. – Мн.: БДУ, 2001. – 48с.

3. Касьянов В. Н., Сабельфельд В.К. Сборник заданий по практикуму на ЭВМ. — М.: Наука., 1986. — 272 с.

3. Кравчук А. И., Кравчук А.С. Сборник лабораторных работ и примеров решения задач по алгоритмизации и программированию на языке СИ: учебно-методическое пособие для студентов высших технических учебных заведений. Минск, : УП ”Технопринт”., 2002. — 116с.

Содержание.

Введение.....	1
Понятие информатики.....	1
Из истории преподавания курса школьной информатики.....	2
Тема 1. Методика преподавания информатики в системе педагогических знаний... 4	4
1.1. Цели изучения курса МПИ.....	4
1.2. Связь МПИ с другими предметами.....	5
1.3. Другие особенности МПИ.....	6
Тема 2. Цели преподавания школьной информатики.....7	7
2.1. Уровни работы с компьютером.....	7
2.2. Цели преподавания информатики как единство образования, развития и воспитания.....	8
2.3. Компьютерная грамотность, образованность, культура.....	10
Тема 3. Принципы дидактики и преподавание информатики..... 11	11
3.1. Принцип научности.....	11
3.2. Сознательность усвоения и деятельности.....	11
3.3. Доступность и наглядность содержания.....	11
3.4. Активность и самостоятельность.....	12
3.5. Прочность и системность знаний.....	12
3.6. Индивидуализация и коллективность обучения.....	13
3.7. Эффективность учебной деятельности.....	13
3.8. Связь теории и практики.....	13
Тема 4. Методы преподавания и изучения информатики14	14
4.1 Обзор методов изучения информатики.....	14
4.2 Традиционные методы обучения и информатика.....	14
4.3 Словесно-фронтальные методы.....	15
4.4. Мыслительные операции и работа на ЭВМ.....	15
Тема 5. Организационные формы обучения..... 17	17
5.1. Общие сведения.....	17
5.2. Число участников учебного процесса и формы обучения.....	18
5.3. Формы обучения в зависимости от того, кто управляет.....	20
5.4. Основные организационные формы обучения.....	20
5.5. Вспомогательные организационные формы обучения.....	21
Тема 6. Изучение раздела “Основы алгоритмизации и программирования” 23	23
6.1. Методические особенности раздела “ОАиП”.....	23
6.2. Общие методические принципы обучения ОА и П.....	24
6.2.1. Принцип многоуровневости.....	24
6.2.2. Принцип предварительной мотивации.....	24
6.2.3. Принципы сравнения и повторения.....	25
6.2.4. Принцип индивидуальных заданий.....	25
6.2.5. Принцип параллельности.....	26

6.3. Формы занятий.....	27
6.4. Изучение ОАиП с использованием компьютеров.....	27
Тема 7. Решение задач при изучении ОА и П.....	28
7.1. Выбор задач.....	28
7.2. Упражнения на составление и анализ блок-схем.....	29
7.3. Тесты.....	31
7.4. Комплекс взаимосвязанных задач.....	32
7.5. Методические приемы, используемые при решении задач.....	33
Программа курса «Информатика» для VI-X классов (базовый уровень).....	35
VI КЛАСС (34 часа).....	37
VII КЛАСС (34 часа).....	39
VIII КЛАСС (34 часа).....	41
IX КЛАСС (34 часа).....	42
X КЛАСС (34 часа).....	45
Программа курса «Информатика» для X-XI классов (повышенный уровень).....	47
IX КЛАСС (68 ч).....	48
X КЛАСС (68 ч).....	53
Программа для X-XI классов (углубленный уровень).....	56
IX КЛАСС (102 ч).....	56
X КЛАСС (102 ч).....	60
УЧЕБНАЯ ПРОГРАММА ДИСЦИПЛИНЫ	
МЕТОДИКА ПРЕПОДАВАНИЯ ИНФОРМАТИКИ.....	65
Литература.....	68