

**ПРОГРАММА
ИНТЕГРАТИВНОГО КУРСА ИНФОРМАТИКИ
ДЛЯ ФИЗИКО-МАТЕМАТИЧЕСКОГО ПРОФИЛЯ
МНОГОПРОФИЛЬНОЙ ШКОЛЫ**

И. Р. Дединский
МФТИ, Кафедра информатики
<http://ded32.ru>

Содержание

Пояснительная записка	5
Цель и задачи курса. Принципы технологии преподавания.....	6
Построение курса. Требования к знаниям и умениям учащихся.....	8
Список литературы	12
Цитируемая литература.....	12
Методическая литература	13
«Информационные технологии» для 5-6 классов.....	14
Пояснительная записка.....	14
Тематическое планирование	16
Часть 1. Основы работы в MS Windows (5 класс).....	16
Часть 2. Основы работы в MS Word (5-6 класс).....	17
Часть 3. Основы работы в Corel Draw (6 класс).....	21
Часть 4. Основы работы в MS Excel (7 класс).....	24
Часть 5. Основы работы персонального компьютера (7 класс).....	25
Часть 6. Основы работы с операционной системой (7 класс)	26
Список рекомендованной учебной и методической литературы	27
Учебная литература (для учеников).....	27
Консультативная и методическая литература (для учителя).....	27
«Алгоритмика и основы программирования» для групп 8 классов с углубленным изучением программирования.....	28
Пояснительная записка.....	28
Тематическое планирование курса.....	30
Часть 1. «Тупой Художник» (первая половина 1 семестра).....	30
Часть 2. Компьютер и математика (2 половина 2 семестра)	34
Часть 3. «Возня с данными» (классика) (2 семестр).....	36
Список рекомендованной учебной и методической литературы	40
Учебная литература (для учеников).....	40
Консультативная и методическая литература (для учителя).....	40
Приложение. Материалы к экзамену по информатике	40
1. Вопросы к экзамену	40
2. Задачи к экзамену	42
«Технология объектно-ориентированного программирования» для 9 мат. классов	44
Пояснительная записка.....	44
Тематическое планирование	45
Список рекомендованной учебной и методической литературы	46
Учебная литература (для учеников).....	46
Консультативная и методическая литература (для учителя).....	47
«Алгоритмические задачи» для 9 мат. классов.....	49

Пояснительная записка.....	49
Тематическое планирование	49
Часть 1. Элементарные вычислительные методы.....	49
Часть 2. Нечисленные и получисленные алгоритмы	51
Список рекомендованной учебной и методической литературы	51
Учебная литература (для учеников).....	51
Консультативная и методическая литература (для учителя).....	52
«Алгоритмы и структуры данных» для 10 мат. классов	53
Пояснительная записка.....	53
Тематическое планирование	54
Часть 1. Структуры данных и алгоритмы над ними	54
Часть 2. Функциональное и логическое программирование	57
Часть 3. Элементы машинной трансляции	58
Список рекомендованной учебной и методической литературы	58
Учебная литература (для учеников).....	58
Консультативная и методическая литература (для учителя).....	59
Примерный список задач-проектов к теме «Элементы машинной трансляции»	60
Словообразование.....	60
Эмуляторы одноадресных процессоров.....	61
Эмуляторы регистровых процессоров	62
Эмуляторы всяческих процессоров.....	63
«Программирование в ОС Windows» для 11 мат. классов	64
Пояснительная записка.....	64
Тематическое планирование	64
Список рекомендованной учебной и методической литературы	65
Учебная литература (для учеников).....	65
Консультативная и методическая литература (для учителя).....	65
«Технология обучения и проектирования» для 8-9 мат. классов	67
Пояснительная записка.....	67
Тематическое планирование	68
Список рекомендованной учебной и методической литературы	70
Учебная литература (для учеников).....	70
Консультативная и методическая литература (для учителя).....	70
«Введение в язык ассемблера» для 9–10 мат. классов.....	71
Пояснительная записка.....	71
Тематическое планирование	71
Список рекомендованной учебной и методической литературы	73
Учебная литература (для учеников).....	73
Консультативная и методическая литература (для учителя).....	73
Приложение 1. Информатика — Q&A (некоторые мнения).....	74
Приложение 2. Журналы результатов	79

Пояснительная записка

Предлагаемая программа предназначена для 5–11 классов физико-математической школы, лицея или гимназии. В соответствии с концепциями об универсальности и профилизации обучения, построение программы проводится по принципам, несколько отличным от традиционных подходов к планированию курсов информатики в средней школе [1, 2]. Основная направленность курса заключается не в рассмотрении информатики как простого продолжения математических дисциплин, отработке чисто компьютерных программистских или пользовательских навыков, а в обучении универсальным и эффективным методам работы с информацией в любой области знания и технологии, в том числе и при изучении любого школьного предмета. При этом компьютер рассматривается не как самоцель обучения, а лишь как средство усиления способностей человека к обработке информации, а также как партнер в процессе информационного обмена.

Специфика информатики как синтетической науки, включающей элементы математики, философии, стилистики, психологии и инженерии, состоит именно в том, что преобладающий способ реализации ее собственных знаний — это работа со знаниями более «конкретных» дисциплин. Как школьная математика дает общий базис для наук, имеющих дело с числами, так информатика оперирует сходными понятиями, появляющимися внешне по-разному в различных областях знания, включая и саму численную математику. Класс наук, так или иначе базирующихся на информатике, необычайно широк, от филологии и общественных дисциплин до физики и геометрии, что иллюстрируется хотя бы тем фактом, что зачастую многие школьные курсы, особенно на начальных (и, следовательно, основных) этапах неявно включают в себя элементы информатики в приложении к конкретной области знания. Характерна также «практичность» информатики, в том смысле, что большинство ее методов направлено на практическое устранение затруднений информационного характера (перегрузка информацией, недостаточная скорость ее усвоения, беспорядочность знаний, неспособность самостоятельно извлечь информацию). Эти особенности предмета позволяют соответствующим образом построить и само обучение, как ответ на информационную перегрузку учащихся школы.

Сочетание обучения методам, а не фактам, работы на примерах из широкого класса школьных дисциплин, универсальности большинства изучаемых приемов превращает, таким образом, информатику в общеобразовательную дисциплину, максимально интегрирующую предметы школьного курса. Как представляется авторам программы, такая концепция должна привести к повышению эффективности процесса обучения в целом, так как поможет ученикам более успешно справляться с обработкой огромного информационного потока школьного образования.

Необходимо подчеркнуть, что в ряде курсов наиболее существенным приобретением ученика становятся не сколько знания, умения и навыки, а *метауме-*

ния и *метанавыки*, определяющие правильное формирование подхода к проблеме, способа ее рассмотрения, поиска решения и реализации, а в конечном счете и организацию знания, получаемого ребенком. Это является следствием системного подхода, применяемого как непосредственно в процессе обучения, так и при составлении настоящей программы. В конечном счете, речь идет о выработке у ученика элементов системной методологии мышления. В таких случаях *метакомпоненты* специально прописаны в программах в графе «Ученик должен (метауровень)». При полном отсутствии этих компонент получается стандартное выучивание и воспроизведение, при полном наличии — в идеале становится ненужным традиционный учебный процесс (ребенок способен получать и организовывать материал самостоятельно). В реальной жизни истина лежит где-то посередине. Понятно, что абсолютное овладение этими компонентами не является строго обязательным и степень его определяется индивидуально для каждого ребенка, в зависимости от его мотиваций и склонностей.

Содержание курса соответствует рекомендациям IFIP (The International Federation for Information Processing) UNESCO [3] и полностью включает Московский региональный общеобразовательный стандарт [4]. Курс обеспечен компьютерными программными средствами (прикладными и обучающими программами). Существующие учебники по информатике могут использоваться как дополнительный материал.

Общеобразовательная часть курсов разрабатывались в рамках создания системы учебных программ «Информатика как общеобразовательная дисциплина», профильная – во взаимодействии с кафедрой информатики МФТИ. Они были успешно опробованы как автором, так и другими учителями в различных учебных заведениях в 1995-2012 годах.

Цель и задачи курса. Принципы технологии преподавания

Цель предлагаемого курса информатики:

Формирование универсального стиля научного и практического мышления, основанного на системном подходе, направленного на самостоятельное творческое познание и исследование окружающего мира.

Задачи курса:

1. Организация процессов мышления и познания.

- Обучение наиболее общим приемам работы с информацией, пригодных для использования в любой предметной области.
- Интеграция программ школьного курса, взаимное усиление информатики и других школьных предметов по принципу: технология работы с информацией — из информатики, конкретные примеры и задачи — из смежных предметов.

2. Развитие самостоятельности в работе и творческой активности.
3. Обучение навыкам работы с компьютером на уровне, минимально необходимом для самостоятельной работы с современными типами программного обеспечения, но достаточном для их дальнейшего самостоятельного изучения.

Принципы технологии преподавания:

1. Принцип самостоятельного исследования объектов и явлений с точки зрения системного подхода. Методы подачи материала:
 - Обучение навыкам самостоятельной работы с информацией;
 - Техника и технология информационного поиска («где что?»);
 - Определение структуры системы понятий («что к чему относится?»);
 - Поиск базовой системы понятий («что главное?»);
 - Связь с другими предметными областями («где еще?»);
 - Исследование «черных ящиков» (приборов, программ, учебных тем);
 - Использование знаний об аналогах и более общих объектах;
 - Выявление общих принципов устройства, работы;
 - Выявление характерных свойств объектов и иерархия этих свойств.
2. Принцип творчества, для:
 - Закрепления полученных теоретических навыков;
 - Эмоционального подкрепления в ходе творческой работы;
 - Организации и планирования проектов, обучения планированию;
 - Взаимодействия учеников в коллективных проектах.
3. Принцип эмоционального подкрепления, для:
 - Увеличения эффективности усвоения;
 - Развития уверенности в своих силах;
 - Развития эстетики.
4. Принцип технологичности:
 - Постоянное стремления повышать КПД своей работы;
 - Умение работать над конкретной задачей в контексте ее возможного развития;
 - Развитие технология решения задачи вместе с ее собственно решением;
 - Обучение технологии на сравнении «плохих» и «хороших» примеров (от первых — ко вторым);
 - Формирование технологически корректной системы ценностей ученика на уровне «люблю работать высокоэффективно» и «не люблю рутинные и ненадежные действия, ищу, как сделать это лучше»;
5. Принцип смежных дисциплин:
 - Разбор материала из смежных предметов, работа на примерах из смеж-

- ных областей;
- Взаимное усиление информатики и смежных дисциплин;
- Интеграция смежных дисциплин.

6. Принцип открытой педагогической технологии:

- Открытая и ясная система ценностей, установок;
- Подчеркнутая расстановка акцентов в подаче материала;
- Понимание «критических мест» в ходе изучения и способов их преодоления;
- Понимание связи материала и методики его подачи;
- Активное использование взаимодействия учеников и взаимного обучения.

Построение курса. Требования к знаниям и умениям учащихся

Курс информатики строится из трех основных ступеней: технологического курса (I ступень), основного курса (II ступень) и углубленного изучения (III ступень) с учетом профилизации в старших классах (рис. 1).

Курс полностью включает Московский региональный общеобразовательный стандарт [3] и большей частью обеспечен компьютерными пособиями (прикладными и обучающими программами). Существующие учебники по информатике [4, 5, 6, 7] могут быть использованы ограниченно. В базовом курсе интенсивно используется учебник и учебное пособие по компьютерным технологиям [8, 9].

I. Технологический курс охватывает 5–6 классы и включает изучение компьютерных технологий на неформальном уровне, однако достаточном для того, чтобы прошедший курс ученик смог самостоятельно использовать компьютер для решения основных учебно-практических задач. Основным принципом преподавания является его открытость, то есть не столько сообщение ребенку сведений о возможностях конкретной программы и тренировка определенных умений, сколько обучение принципам работы и изучения программного средства. Упор делается на совмещение логики и интуиции, экспериментирование с программой, что дает ребенку возможность расширять свои знания самостоятельно. Важной стороной обучения начинающего пользователя является развитие внимания и самоконтроля при применении средств программы. Ученик должен понять, что важно достигать цели минимальными средствами, но на максимальном уровне. Для этого он должен уметь организовывать процесс работы максимально технологично, иметь понятие о стилистике, не допускать «винегрета», говорящего о том, что автор умеет пользоваться программой, но не имеет вкуса.

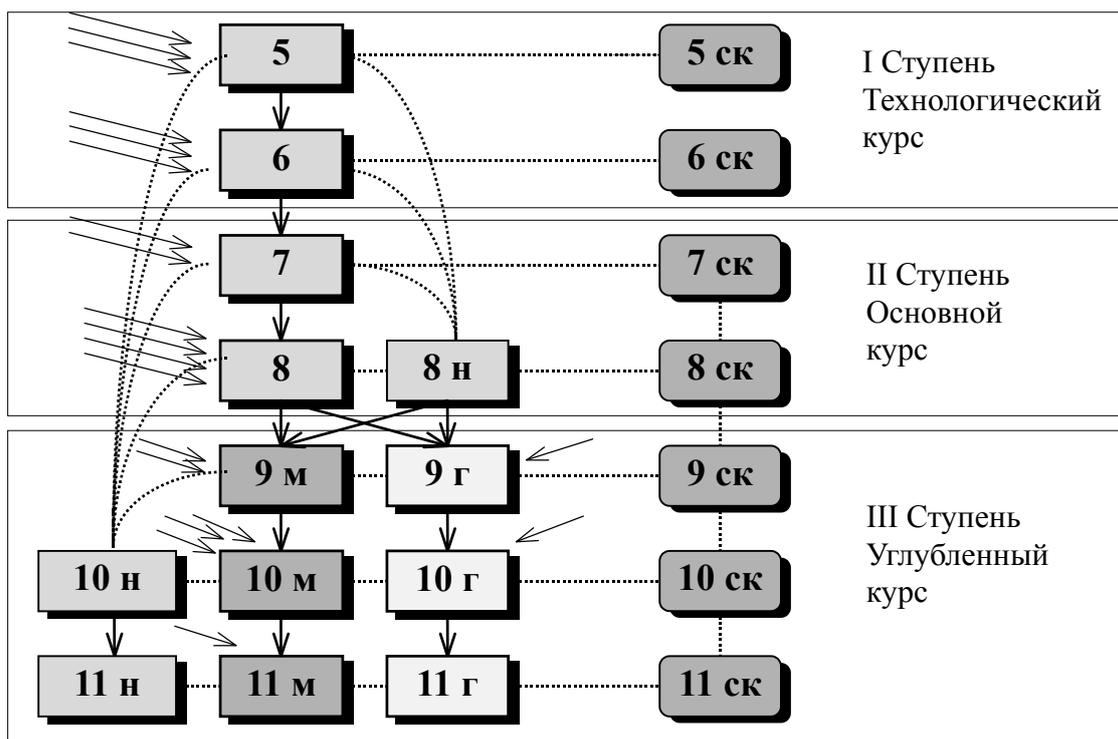


Рис. 1. Структура предлагаемого курса информатики на примере многопрофильной школы (гимназии, лицея).

М — физико-математический профиль, Г — гуманитарный и биолого-химический профиль, Н — наборные классы, СК — спецкурс (элективные курсы, спецкурсы, мастерские, факультативы). Тонкими косыми стрелками показано относительное количество межпредметных связей.

В первой половине курса (5 класс) рассматриваются основные принципы работы с компьютером и управления системой Windows. Рассмотрение идет в виде серии рассказов учителя, практических работ, компьютерных экспериментов, в течение которой вырабатываются универсальные умения, необходимые для работы в любой оконной системе стандарта GUI. Вырабатывается технология самостоятельного изучения программ, работы с системой помощи. В ходе начального курса Windows учащиеся проходят графический редактор Paint-Brush, однако он рассматривается не как самоцель, а как прообраз всех Windows-программ. Далее на достаточно подробном уровне рассматривается редактор MS Word, изучение которого продолжается в I половине 6 класса. К концу обучения ученик должен свободно владеть редактором MS Word на уровне, достаточном для грамотного оформления хорошего школьного реферата. Во II полугодии 6 класса изучается векторный графический редактор Corel Draw, что позволит учащемуся впоследствии изготавливать рисунки и схемы для решения учебно-практических задач.

В ходе технологического курса учащиеся получают представление об основных понятиях теоретической информатики, предусмотренные стандартом. Однако каждой теоретической теме, после отработки ее содержания, соответ-

вует практическая работа в каком-либо редакторе. Таким образом, вместе с изучением нового материала закрепляются технологические умения, а сам теоретический материал дополнительно повторяется и получает «образную» форму. В результате ученики лучше запоминают сведения, а также соотносят их с практикой, поскольку компьютерная работа напрямую связана с использованием важнейших теоретических и технических понятий.

По прохождении технологического курса ученик:

- Знает об основных принципах работы с компьютером в системе Windows;
- Умеет самостоятельно исследовать программы методом компьютерного эксперимента и изучения системы помощи;
- Осмысленно реагирует на ошибки и соотносит их со своими действиями, делая соответствующие выводы;
- Владеет редакторами PaintBrush, MS Word и Corel Draw в объеме, достаточном для грамотной подготовки реферата большого объема с таблицами, схемами и иллюстрациями;
- Знает правила верстки текстов, соблюдает стилистику оформления, знает основные особенности верстки книг, журналов, газет, объявлений;
- Знает основные понятия теории информации, имеет представление о видах информации, ее кодировании, единицах измерения, передаче;
- Имеет представление о строении компьютера, принципах его организации, двоичного кодирования данных и программ;
- Знает устройство компьютера и периферийных устройств в рамках, оговоренных в Образовательном стандарте [3], умеет сравнивать модели компьютеров, знать минимальные требования, предъявляемые к качеству компьютерных систем;
- Не боится компьютера, а воспринимает его как мощное средство помощи для решения разнообразных задач, которые нужно вначале уметь поставить и придумать план действий для решения.

II. Основной курс информатики (II ступень, 7-8 классы) включает в себя технологию работы с файловой системой Windows, изучение математических основ теории информации (системы счисления и булева алгебра). Из пользовательских программ изучается MS Excel. В основном курсе ученик получает необходимые знания о детальном устройстве компьютера, операционных системах и оболочках на примере Far Manager (Total Commander), их принципах работы, командах и меню. Обучение строится с использованием лабораторных работ, методических разработок и уроков-консультаций, в максимально самостоятельном режиме.

Во второй половине (8 класс) изучаются основы алгоритмизации и одного из языков программирования (желателен Си, возможен также Паскаль), как реальных представителей современных процедурных языков. Упор здесь делается на технологию компьютеризации задачи — приспособление ЭВМ под решение

задачи из некомпьютерной области. Изучение тем проводится в виде выполнения ряда проектов и лабораторных работ.

Вариант сокращенной программы для 8-х наборных классов включает в себя основы технологии Windows и алгоритмизации в объеме, достаточном для продолжения обучения в 9-11 классах.

По прохождении основного курса ученик:

- Свободно ориентируется в файловой системе Windows, основных меню и командах Far Manager, знает основные команды консоли Windows, умеет пользоваться основными антивирусными, сетевыми, архивными программами-утилитами;
- Владеет основными понятиями теории информации и ее кодирования, алгебры логики, указанными в Образовательном стандарте [3];
- Владеет основными конструкциями одного из языков программирования (Си или Паскаль), умеет использовать его для записи алгоритмов и в ходе работы над компьютерными проектами;
- Умеет использовать компьютер как помощь в решении задач по некомпьютерным предметам, знает возможности компьютера, границы и способы его применения, технологию «компьютеризации» задачи.

III. Углубленный курс (9-11 классы, III ступень) строится с учетом профилизации в старших классах школы. Предпочтение отдается прикладным задачам, типичным для той или иной специализации. Таким образом, III этап продолжает направленность предыдущего этапа на активность знания, умение использовать информационные технологии для решения задач «из жизни». Для нематематических (биолого-химических, гуманитарных) классов предполагаются курсы по основным типам программного обеспечения (углубленное изучение текстовых редакторов, изучение электронных таблиц, баз данных). Для групп математического профиля предполагается курс основных алгоритмов решения машинных задач, разбитый на курсы технологии объектно-ориентированного программирования (9 класс), вычислительных алгоритмов (9-10 класс), и структур данных и нечисленных алгоритмов (10-11 класс) Изучение основных типов программного обеспечения производится с помощью методических разработок и уроков-консультаций, максимально самостоятельно.

Вариант программы для 10-го и 11-го наборных классов включает в себя элементы пользовательских и информационно-поисковых навыков, минимально необходимых для обучения в институте и элементарной работы с основными современными пакетами программ.

По прохождении углубленного курса ученик нематематического (биолого-химического и гуманитарного) профиля:

- Владеет основными типами современного программного обеспечения (электронными таблицами, базами данных, текстовыми и графическими

- редакторами) на уровне, достаточном для полного самостоятельного изучения во внеучебное время;
- Квалифицированно владеет текстовым редактором в среде Windows на уровне, достаточном для эффективного использования редактора для написания дипломной работы с таблицами, рисунками в тексте, литературными ссылками, разбиением на подзаголовки и созданием оглавления;
 - Умеет эффективно применять существующее программное обеспечение для решения задач в своей области специализации;
 - Умеет оценить требования к компьютерной системе и программному обеспечению, исходя из условий конкретной задачи.

В дополнение к этому, ученик физико-математического профиля:

- Знает основные методы компьютерного программирования, базовые численные и нечисленные алгоритмы и работу с основными структурами данных;
- Умеет использовать полученные алгоритмические и программистские знания для решения нестандартных задач, не решаемых на типовых программных продуктах;
- Имеет представление об эффективности алгоритмов, умеет подбирать оптимальный алгоритм решения задачи, модифицировать его для своих нужд;
- Владеет основными объектно-ориентированными конструкциями одного из языков программирования (Си++ или Паскаль), умеет использовать их для записи алгоритмов и в ходе работы над компьютерными проектами.

Список литературы

Цитируемая литература

1. Основы информатики и вычислительной техники. Программы средней школы. Москва, «Просвещение», 1992.
2. Программа непрерывного курса информатики для средней школы. МИПКРО, 1993.
3. Informatics for Secondary Education. A Curriculum for Schools. UNESCO, 1994.
4. Московский региональный общеобразовательный стандарт по курсу информатики. МИПКРО, 1995 г.
5. А. Г. Кушниренко, Г. В. Лебедев, Р. А. Сворень. Основы информатики и вычислительной техники. Пробный учебник для средних учебных заведений. 2-е издание. Москва, «Просвещение», 1991.
6. В. А. Каймин, А. Г. Щеголев, Е. А. Ерохина, Д.П. Федюшин. Основы информатики и вычислительной техники. Пробный учебник для 10-11 классов средней школы. Москва, «Просвещение», 1989.
7. С.А. Абрамов, Е.В. Зима. Начала информатики. Москва, «Наука», 1990.
8. Ю. Шафрин. Основы компьютерной технологии. М., АБФ, 1998.
9. О. Ефимова, В. Морозов. Практикум по компьютерной технологии. М., АБФ, 1998.

Методическая литература

10. В. Д. Валединский “Информатика. Шаг за шагом. Рабочая тетрадь”. М., Издательство гимназии “Открытый мир”, 1995.
11. В. Дайитбегов, Е.А. Черноусов “Основы алгоритмизации и алгоритмические языки”. М., “Финансы и статистика”, 1992.
12. Информатика. Энциклопедический словарь для начинающих. М., Педагогика-пресс, 1994.
13. Н. Николь, Р. Альбрехт “Excel 5.0. Электронные таблицы”. М., ЭКОМ, 1996
14. А. Никольская “Азбука рассуждений”. Учебное издание. М., 1996
15. И. Ратбон “Windows для “чайников”. Киев, “Диалектика”, 1994
16. И. Ратбон “Еще о Windows для “чайников”. Киев, “Диалектика”, 1995
17. В. Фигурнов “IBM PC для пользователя”. М 1995
18. Д. Фойц “Windows 3.1 для пользователя”. Киев, ВНУ, 1993.
19. А. Харвей “Excel 5.0 для “чайников”. Киев, “Диалектика”, 1994
20. Microsoft Word. User’s Guide. Microsoft Corporation, 1993-1994
21. Д. Грис “Конструирование компиляторов для цифровых вычислительных машин”. М., “Мир”, 1975.
22. М. Мартин, У. Уэйт, А. Прата. Язык Си для начинающих. М., Мир, 1988.
23. Б. Керниган, Д. Ритчи. Язык программирования Си. М., Радио и связь, 1977.
24. С. Прата. Язык программирования Си. Киев, ДиаСофт, 2001.
25. С. Прата. Язык программирования C++. Киев, ДиаСофт, 2001.
26. Б. Страуструп. Язык программирования Си++. М., Финансы и статистика, 1984.
27. В. Фигурнов. Программирование на языке Турбо Паскаль. М., 1990.
28. Г. Нешков и др. Множества. Отношения. Числа. Величины. М. 1978.
29. Упражнения по логике. Под ред. В.И. Кириллова. М. 1993.
30. А.Н. Колмогоров, А.Г. Драгалин. Введение в математическую логику. М. 1982.
31. Г. Кук, Г. Бейз. Компьютерная математика М. 1990.
32. Н. Парницкий. Основы статистической информатики. М. 1981. Дж. Элти, М. Кумбс. Экспертные системы. Концепции и примеры. М. 1987.
34. Ж.-Л. Лорьер. Системы искусственного интеллекта. М. 1991.
35. Дж. Доорс и др. ПРОЛОГ — язык программирования будущего. М. 1990.
36. А. Моль. Теория информации и эстетическое восприятие. М 1966.
37. А. Яглом. Современная культура и компьютеры. М. 1990.

Программа курса
«Информационные технологии»
для 5-6 классов
(2 часа в неделю, всего 128 часов)

Пояснительная записка

Курс охватывает 5–7 классы и включает изучение компьютерных технологий на неформальном уровне, однако достаточном для того, чтобы прошедший курс ученик смог самостоятельно использовать компьютер для решения основных учебно-практических задач. Основным принципом преподавания является его открытость, то есть не сколько сообщение ребенку сведений о возможностях конкретной программы и тренировка определенных умений, сколько обучение принципам работы и изучения программного средства. Упор делается на совмещение логики и интуиции, экспериментирование с программой, что дает ребенку возможность расширять свои знания самостоятельно. Важной стороной обучения начинающего пользователя является развитие внимания и самоконтроля при применении средств программы. Ученик должен понять, что важно достигать цели минимальными средствами, но на максимальном уровне. Для этого он должен уметь организовывать процесс работы максимально технологично, иметь понятие о стилистике, не допускать «винегрета», говорящего о том, что автор умеет пользоваться программой, но не имеет вкуса.

В начале курса (5 класс) рассматриваются основные принципы работы с компьютером и управления системой Windows. Рассмотрение идет в виде серии рассказов учителя, практических работ, компьютерных экспериментов, в течение которой вырабатываются универсальные умения, необходимые для работы в любой оконной системе стандарта GUI. Вырабатывается технология самостоятельного изучения программ, работы с системой помощи. В ходе начального курса Windows учащиеся проходят графический редактор PaintBrush, однако он рассматривается не как самоцель, а как прообраз всех Windows-программ. Далее на достаточно подробном уровне рассматривается редактор MS Word, изучение которого продолжается в I половине 6 класса. К концу обучения ученик должен свободно владеть редактором MS Word на уровне, достаточном для грамотного оформления стандартного школьного реферата. Во II полугодии 6 класса изучается векторный графический редактор Corel Draw (в качестве примера мощного векторного редактора), что позволит учащемуся впоследствии изготавливать рисунки и схемы для решения учебно-практических задач.

6 класс включает в себя технологию работы с файловой системой Windows, изучение математических основ теории информации (системы счисления и булева алгебра). Из пользовательских программ изучается MS Excel. В основном курсе ученик получает необходимые знания о детальном устройстве компьютера, операционных системах и оболочках на примере Far Manager (Total Commander), их принципах работы, основных командах и меню. Обучение строится с использованием лабораторных работ, методических разработок и уроков-консультаций, в максимально самостоятельном режиме.

Важнейшей частью курса является формирование системы профессиональных ценностей (предпочтений) ученика. В конечном счете, это формирование и есть основная инвариантная методологическая задача курса, так как все остальное — технология и будет неотвратно изменяться с течением времени. Это аспект отражен в тематическом планировании в графе «Ученик должен (*метауровень*)».

Курс излагается с опорой на развитие самостоятельного мышления учащихся, самообразование. Для выполнения учащимися выбираются задания, допускающие решения в эстетически привлекательной форме, имеющие игровой компонент. Формы занятий — лекция, семинар и практическая работа. Формы контроля — сдача практических работ (основная), письменных контрольных работ, проверка конспектов.

Тематическое планирование

Часть 1. Основы работы в MS Windows (5 класс)

№	Тема / ученик должен знать / ученик должен уметь	Ученик должен уметь (метауровень)
1.	Основные части компьютера, основные термины. Правила поведения в компьютерном классе. Вход в сеть.	
2.	Операционная оболочка (система) MS Windows, ее назначение и возможности, место среди аналогичных программных средств. Краткая история версий Windows. Требования Windows к ресурсам компьютера. Запуск Windows. Окна Windows и их свойства. Части окна. Виды окон Windows и их свойства. Управление окнами.	Понять, что: 1) изучаемая область гораздо более интересна, чем трудна; 2) материал предмета позволит интереснее и полезнее работать не только в школе (например, и дома).
3.	Система помощи Windows. Работа с системой помощи. Понятие гипертекста и гипертекстовой ссылки. Понятие о контекстно-зависимой помощи. Самостоятельное изучение программ с использованием системы помощи.	Действовать «от задачи»: Активно пользоваться системой помощи для самостоятельного решения вопросов.
4.	Практическая работа «Работа с окнами и системой помощи Windows». Самостоятельное изучение какой-либо темы и использование знаний на практике.	
5.	Графический редактор Paintbrush. Основные инструменты Paintbrush. Выбор цветов рисования. Выделение, копирование и перемещение частей рисунка. Команда «Отмена». Режим «Приблизить». Инструмент «Текст». Сохранение и открытие документов. Понятие об эксперименте с программой и его осмысление.	Действовать «от возможностей программы»: Самостоятельно изучать программу в режиме эксперимента. Анализировать управляющие элементы программы, прогнозировать их назначение. Записывать и оформлять результаты изучения в «лабораторный журнал». Сочетать эксперимент с использованием системой помощи.
6.	Графический редактор Paintbrush (практическая работа). 1) Рисунок с повторениями частей изображения через перерисовывание (мартышкин труд).	Пользоваться результатами предыдущей работы (копирование частей рисунка, отражение, поворот, перекрашивание — вместо перерисовывания).

№	Тема / ученик должен знать / ученик должен уметь	Ученик должен уметь (метауровень)
7.	Графический редактор Paintbrush (практическая работа). 2) Рисунок с повторениями частей изображения через копирование и трансформацию изображения (по-человечески). Изучение специальных инструментов (кривая, цветной ластик).	

Часть 2. Основы работы в MS Word (5-6 класс)

№	Тема / ученик должен знать / ученик должен уметь	Ученик должен уметь (метауровень)
1.	Текстовый редактор MS Word. Набор текста. Специальные символы. Ошибки и их исправление. Прокрутка страницы и перемещение по документу. Выделение, копирование и удаление участков текста. Правильная технология набора текста и последствия ошибок технологии набора (яркая демонстрация).	Правильно (технологично) набирать текст (сначала набор, затем оформление). Понимать, что форматировать текст надо только средствами Word (а не пробелами и пустыми строками). Привыкнуть к отображению непечатаемых символов. Понимать и ценить эргономику управляющих действий в программах (сочетание клавиатуры и мыши, клавиатурные сокращения).
2.	Набор текста в редакторе MS Word (практическая работа).	
3.	Форматирование символов. Меню «Формат/Абзац». Инструменты форматирования символов. Шрифты и их характеристики. Читательность различных шрифтов. Стилистика шрифтового оформления текста. Семейства шрифтов. Роль форматирования символов в передаче информации, заключенной в тексте. Читательность текста, оформленного различным образом. Приемы верстки, направленные на привлечение внимания, отделение главного от второстепенного. Эстетика шрифтового оформления. Стилистические ошибки верстки. Работа в WordArt.	Правильно и, главное, технологично оформлять текст. Подчинять оформление той задаче, которая возложена на текст. Соблюдать читательность текста. Знать шрифтовые правила верстки: не более 2 гарнитур, не более 3 типоразмеров. Знать, к чему приводят нарушения этих правил. Чувствовать эстетику шрифтов и начертаний. Стараться пользоваться малым, достигая многого. Знать клавиатурные сокращения для шрифтового оформления.

№	Тема / ученик должен знать / ученик должен уметь	Ученик должен уметь (метауровень)
4.	<p>Верстка различных видов объявлений (практическая работа).</p> <p>1) Максимально плохое объявление, с нарушениями всех норм эстетики («винегрет»). 2) Максимально эстетически лаконичное объявление (строгое). 3) Оптимальное объявление.</p> <p>Анализ читабельности и эстетических качеств текста.</p>	
5.	<p>Форматирование абзацев. Характеристики абзаца. Меню «Формат/Абзац». Линейка форматирования и ее элементы. Влияние характеристик абзаца на читабельность текста. Взаимодействие с соседними абзацами. Эстетика форматирования абзаца. Понятие о единстве стиля оформления.</p>	<p>Технологично оформлять абзац. Презирать установку красной строки пробелами или табуляцией. Пользоваться абзацными отступами (отбивками), чувствовать их влияние на подачу текста (выделение вертикальными отступами, выделение отступом слева и т.д.). Копировать оформление с образца (желательно, из печатной продукции). Анализировать оформление образца. Знать клавиатурные сокращения.</p>
6.	<p>Форматирование текста с различными видами абзацев (практическая работа). Оформляется один и тот же текст в разных видах: «основной текст», «с привлечением внимания», «комментарий в тексте», «совет», «необязательный материал» и т.д.</p>	
7.	<p>Верстка таблиц. Назначение и характеристики таблиц. Вставка таблиц и работа с ними. Форматирование таблиц. Меню «Таблица». Инструменты для работы с таблицами. Разграфка таблицы через меню «Формат/Обрамление» и панель инструментов. Эстетика таблиц. Корректировка свойства абзацев, объединяемых в таблицу, для улучшения восприятия текста.</p>	<p>Свободно работать с таблицами. Знать коррекцию свойств абзацев для лучшей читаемости в таблице (отступы и т.д.). Использовать затенение фона абзаца. Использовать таблицы для решения нестандартных задач (два абзаца рядом, верстка в две колонки). Форматировать таблицы в различных стилях. Делать «дырки» в таблицах (слиянием или разграфкой). Визуально «сливать» ячейки (положением текста при отсутствии разделяющей линии).</p>
8.	<p>Форматирование таблиц (практическая работа).</p> <p>1) Неудобочитаемая таблица. 2) «Строгая» таблица. 3) Таблица по заданному образцу. 4) Свое собственное оформление: (а) мало данных, (б) много данных.</p>	

№	Тема / ученик должен знать / ученик должен уметь	Ученик должен уметь (метауровень)
9.	<p>Понятие о стиле оформления. Работа со стилями. Определение, использование и переопределение стилей. Стандартные и собственные стили. Особенности стиля «Обычный» («Нормальный»). Повышение производительности труда с помощью механизма стилей (яркая демонстрация).</p> <p>Стандартные стили. Стили «Заголовок». Понятие об уровнях заголовка. Режим Outline. Автоматический сбор оглавления.</p>	<p>Понимать, что стиль — это основа технологии, экономия времени и нервов. Свободно работать со стилями. Любить использовать стили. Бояться стиля «Обычный» («Нормальный»). Понимать согласованность стилей в документе. Разрабатывать систему стилей. Рассматривать режим Outline как «автоматический план текста».</p>
10.	<p>Разработка и использование таблицы стилей (практическая работа).</p>	
11.	<p>Комплексная практическая работа на применение стилей и работу с таблицами.</p> <p>1) Система стилей «попугай» («винегрет»). 2) «Строгая» система. 3) По образцу. 4) Своя (для рефератов). 5) Своя (для шпаргалок).</p>	
12.	<p>Вставка иллюстраций в документ. Виды вставки. Вставка различных объектов. Понятие о механизме OLE. Особенности работы с объектами OLE. Работа с ClipArt. Форматирование иллюстраций. Особенности работы с различными типами иллюстраций. Стили для иллюстраций.</p> <p>Понятие о композиции. Принцип равновесия. Принцип соизмеримости. Понятие ритма (на примере — 1-2 картинка на лист с текстом и 6-7 картинок, расставленных в определенном порядке).</p>	<p>Уметь грамотно использовать иллюстрации в документе. Следовать принципам равновесия и соизмеримости. Уметь делать рисованный заголовок.</p> <p>Уметь привлекать внимание дорисовкой. Следить за точками привязки рисованных объектов, осторожно относиться к дорисовке, привязанной к странице, не любить разрозненные (несгруппированные) объекты. Любить привязку к сетке.</p>
13.	<p>Работа с иллюстрациями в документе (практическая работа).</p> <p>1) Заведомо плохая композиция. 2) Устойчивая композиция с одной картинкой в тексте (а) в центре, (б) сбоку. 3) Устойчивая композиция с 4-5 картинками. 4) Картинка как заголовок.</p>	

№	Тема / ученик должен знать / ученик должен уметь	Ученик должен уметь (метауровень)
14.	Рисованные элементы. Панель рисования. Композиционные находки и ошибки. Роль композиции в передаче отношения автора к тексту. Понятие о колорите макета. Компьютерная графика как искусство.	
15.	Практическая работа «Дорисовка в тексте и использование ClipArt»	
16.	Макет страницы. Оформление колонтитулов. Стили колонтитулов. Понятие раздела. Форматирование разделов (различная ориентация страницы, многоколоночная верстка).	
17.	Поля и их использование в колонтитулах (Page, DocInfo, StyleRef). Построение технического нижнего колонтитула (имя файла, путь, версия, дата, время, страница, общее к-во страниц).	Свободно использовать поля. Понимать удобство технического нижнего колонтитула. Использовать поле StyleRef для автоматизации верхнего колонтитула.
18.	Практическая работа по макетированию страницы с таблицами, иллюстрациями и колонтитулами с применением стилей. 1) Плохо (никак). 2) В стиле учебника. 3) В стиле литературного издания. 3) В стиле рекламной брошюры, каталога.	Чувствовать единство оформления страницы.
19.	Анализ типов оформления различных печатных изданий. Особенности стилистики, характерные приемы верстки. Особенности оформления абзацев. Многоколоночная верстка и верстка в кадрах. Колонтитулы. Ограничения MS Word. Другие программы профессиональной верстки текста.	Понимать и чувствовать стилистику различных типов изданий, ее связь с содержанием текста. Владеть многоколоночной версткой, уметь использовать кадры. Понимать ограниченность MS Word как программы верстки.
20.	Преобразование предварительно форматированного текста. Особенности работы с большими текстами. Важность технологии набора текста для его эффективного оформления. Средства MS Word для ускорения стереотипных действий. Поиск и замена в тексте (повторение), специальные случаи	Техническая тема, связанная с импортом большого количества текста для последующих практических работ. Одновременно тренировка важных умений по автоматизации работы с текстом. Работа с метасимволами и вы-

№	Тема / ученик должен знать / ученик должен уметь	Ученик должен уметь (метауровень)
	поиска и замены (с подстановочными знаками, с включением оформления).	ражениями.
21.	Практическая работа по верстке: 1) Книжной полосы. 2) Журнальной полосы. 3) Газетной полосы (попытка).	Применять весь комплекс умений для создания законченного макета по образцу и своего собственного.
22.	Обобщающий урок. Техника и искусство компьютерной верстки.	

Часть 3. Основы работы в Corel Draw (6 класс)

№	Тема / ученик должен знать / ученик должен уметь	Ученик должен уметь (метауровень)
1.	Графическая информация. Способы представления графической информации. Растровые и векторные изображения и редакторы. Назначение и особенности растровых и векторных редакторов. Векторный редактор Corel Draw.	Понимать отличия растрового изображения от векторного и области применения этих типов. Понимать, почему рисунки и схемы надо готовить не в PaintBrush'e.
2.	Основные инструменты Corel Draw. Создание и удаление объектов. Палитра и выбор цветов. Простейшие операции поворота, масштабирования и скоса. Изменение толщины линий. Возможности рисования и композиция макета.	Использовать методики самостоятельного изучения для изучения новой программы. Любить использовать направляющие и ограничения (клавиша Shift). Ценить дублирование объектов как повышение производительности работы.
3.	Приемы ускорения работы в Corel Draw. Использование чернового просмотра. Дублирование объектов. Создание тени. Использование клавиш Shift и Control при перемещении. Направляющие. Использование направляющих для аккуратной работы с макетом.	
4.	Основные возможности Corel Draw и приемы работы. Верстка простого рисунка (практическая работа).	
5.	Работа с текстом в Corel Draw. Экспорт и импорт текста из	Ценить операцию группировки как облегчение работы

№	Тема / ученик должен знать / ученик должен уметь	Ученик должен уметь (метауровень)
	файла и буфер обмена. Команды форматирования текста. Создание текста с тенью. Группировка и разгруппировка. Группа как единое целое (на примере объекта и его тени и др.).	со сложными объектами.
6.	Верстка объявления в Corel Draw (практическая работа). 1) Много возможностей Corel Draw (винегрет). 2) Мало возможностей Corel Draw. 3) Оптимум.	Быть лаконичными в выборе изобразительных средств, не упиваться возможностями программы.
7.	Редактирование линий и заливок. Виды заливок. Градиентные заливки. Текстурные заливки. Использование заливок для придания макету сходства с естественным изображением. Цветовое решение макета и законы комбинации цветов. Колорит.	
8.	Использование заливок для улучшения макета объявления (практическая работа). 1) Много возможностей Corel Draw (винегрет). 2) Мало возможностей Corel Draw. 3) Оптимум.	
9.	Работа с кривыми в Corel Draw. Инструмент «Форма». Работа с кривыми Безье. Команды «Текст вдоль кривой» и «Огибающая». Рисование при помощи кривых. Методы обводки сканированных рисунков. Понятие о векторизации.	
10.	Верстка с использованием инструмента «Форма» и команд «Текст вдоль кривой» и «Огибающая».	
11.	Инструменты «Перспектива» и «Объем». Добавление перспективы и объема к текстовым надписям. Комбинации эффектов. Набор эффектов, колорит и композиция макета как единое целое. Чувство меры при работе с компьютерной графикой.	

№	Тема / ученик должен знать / ученик должен уметь	Ученик должен уметь (метауровень)
12.	Импорт объектов и рисунков. Работа с ClipArt. Изменение существующих объектов. Практическая работа «Рисунок в тексте».	Грамотно использовать заготовки для ускорения работы над рисунком.
13.	Практическая работа «Титульная страница».	Итоговый проект по Corel Draw.
14.	Обобщающий урок по компьютерной графике и верстке.	

Часть 4. Основы работы в MS Excel (7 класс)

№	Тема урока
1.	Электронные таблицы. Основы программы MS Excel
2.	Окно программы MS Excel
3.	Ячейка электронной таблицы и ее свойства.
4.	Практическое занятие по основам MS Excel.
5.	Как подготовить простую таблицу в MS Excel. Основные манипуляции с таблицами.
6.	Практическое занятие по созданию таблицы в MS Excel
7.	MS Excel как база данных. Разработка простейшей базы данных.
8.	Поиск информации и замена. Многозначные символы.
9.	Фильтрация данных. Пользовательские фильтры.
10.	Расчетные операции в MS Excel.
11.	Лабораторная работа «Таблица умножения».
12.	Логические значения и операции. Выражения, имеющие логические значения и их смысл. Таблицы истинности и диаграммы Венна.
13.	Применение расчетных операций для расчета таблиц истинности (лабораторная работа).
14.	Эквивалентные логические формулы. Преобразования логических формул.
15.	Доказательства эквивалентности с помощью таблиц истинности (лабораторная работа).
16.	Графическое представление информации в MS Excel. Построение гистограмм и графиков.
17.	Построение графиков функций в электронных таблицах (лабораторная работа).
18.	Статистические вычисления в MS Excel. Основные возможности.
19.	Статистические вычисления в MS Excel (лабораторная работа).
20.	Решение математических задач с помощью Excel.
21.	Решение математических задач с помощью Excel (лабораторная работа).
22.	Моделирование процессов с помощью Excel.
23.	Понятие о графическом решении систем уравнений в Excel.
24.	Итоговое занятие по Excel.

№	Тема урока
25.	Зачетное занятие по Excel.

Часть 5. Основы работы персонального компьютера (7 класс)

№	Тема урока
1.	Обзор истории вычислительной техники. Принципы архитектуры ПК. Понятие модульной и открытой архитектуры.
2.	Схема строения ПК. Системные и периферийные устройства. Применение модульного принципа.
3.	Обсуждение принципов модульной и открытой архитектур применительно к схеме строения ПК (семинар).
4.	Понятие о компьютерной шине. Различные архитектуры шин. Шина как модуль ПК. Понятие о магистрали.
5.	Понятие прерывания, работа механизма прерываний. Понятие об IRQ-конflikте.
6.	Центральный процессор и схема его строения. ЦП как модуль ПК. Понятие о системном подходе.
7.	Функционирование ЦП. Основной алгоритм работы ЦП. Понятие о системе команд процессора. Обзор различных видов ЦП. Современные тенденции в развитии ЦП.
8.	Оперативная память (RAM). Понятие о ROM и ROM BIOS. Запуск машины.
9.	Современные тенденции в развитии системных устройств. Специализация ПК.
10.	Видеокарты и мониторы. Их характеристики.
11.	Жесткие диски, их виды и характеристики. Понятие о кэшировании. Дисководы и дискеты. Другие носители информации.
12.	Порты компьютера. Подключение клавиатуры, мыши и принтера. Виды клавиатур и указующих устройств..
13.	Принтеры и их виды. Обзор технологий печати. Характеристики и основные производители принтеров.
14.	Ввод графической информации в компьютер. Сканеры и их виды. Характеристики и основные производители.
15.	Обработка графических изображений на компьютере. Сопряжение графических устройств.
16.	Понятие о компьютерных сетях. Компоненты компьютерной сети. По-

№	Тема урока
	Понятие о топологии сети.
17.	Связь компьютеров по телефону. Модемы, их характеристики. Характеристики связи.
18.	Сборка компьютера. Правила обращения с компьютерными платами. Технология сборки и тестирования.

Часть 6. Основы работы с операционной системой (7 класс)

№	Тема урока
1.	Техника безопасности при работе на компьютерах. Виды программного обеспечения.
2.	Хранение информации в ПК. Диски, каталоги и файлы. Работа в Far Manager.
3.	Кодовые таблицы символов, перекодировка файлов.
4.	Знакомство с консолью Windows. Приглашение консоли. Диск и каталог по умолчанию. Основные консольные команды Windows.
5.	Практическая работа по работе с кодовыми таблицами и приглашением Windows.
6.	Путь к файлу. Адрес файла.
7.	Полезные консольные команды Windows.
8.	Практическая работа по командам DOS.
9.	Понятие bat(cmd)-файла. Использование bat-файлов для стереотипных действий с файлами.
10.	Практическая работа на создание и использование bat-файлов.
11.	Приемы работы в Far Manager. Панели Far Manager.
12.	Работа в Windows с помощью Far Manager.
13.	Практическая работа по основным возможностям Far Manager.
14.	Выделение файлов и работа с группами файлов.
15.	Запуск программ и команд в Far Manager. Ассоциации файлов.
16.	Дополнительные функции Far Manager.
17.	Практическая работа по дополнительным возможностям Far Manager.
18.	Обобщающий урок по Far Manager.

Список рекомендованной учебной и методической литературы

Учебная литература (для учеников)

1. А. Г. Кушниренко, Г. В. Лебедев, Р. А. Сворень. Основы информатики и вычислительной техники. Пробный учебник для средних учебных заведений. 2-е издание. Москва, «Просвещение», 1991.
2. В. А. Каймин, А. Г. Щеголев, Е. А. Ерохина, Д.П. Федюшин. Основы информатики и вычислительной техники. Пробный учебник для 10-11 классов средней школы. Москва, «Просвещение», 1989.
3. С.А. Абрамов, Е.В. Зима. Начала информатики. Москва, «Наука», 1990.
4. Ю. Шафрин. Основы компьютерной технологии. М., АБФ, 1998.
5. Д. Валединский “Информатика. Шаг за шагом. Рабочая тетрадь”. М., Издательство гимназии “Открытый мир”, 1995.
6. Дайитбегов, Е.А. Черноусов “Основы алгоритмизации и алгоритмические языки”. М., “Финансы и статистика”, 1992.
7. Информатика. Энциклопедический словарь для начинающих. М., Педагогика–пресс, 1994.
8. Николь, Р. Альбрехт “Excel 5.0. Электронные таблицы”. М., ЭКОМ, 1996
9. Никольская “Азбука рассуждений”. Учебное издание. М., 1996
10. Ратбон “Windows для “чайников”. Киев, “Диалектика”, 1994
11. Ратбон “Еще о Windows для “чайников”. Киев, “Диалектика”, 1995
12. Фигурнов “IBM PC для пользователя”. М 1995
13. Фойц “Windows 3.1 для пользователя”. Киев, ВHV, 1993.
14. Харвей “Excel 5.0 для “чайников”. Киев, “Диалектика”, 1994
15. О. Ефимова, В. Морозов. Практикум по компьютерной технологии. М., АБФ, 1998.

Консультативная и методическая литература (для учителя)

16. Основы информатики и вычислительной техники. Программы средней школы. Москва, «Просвещение», 1992.
17. Программа непрерывного курса информатики для средней школы. МИПКРО, 1993.
18. Informatics for Secondary Education. A Curriculum for Schools. UNESCO, 1994.
19. Московский региональный общеобразовательный стандарт по курсу информатики. МИПКРО, 1995 г.
20. Microsoft Word. User's Guide. Microsoft Corporation, 1993-1994
21. В. Нешков и др. Множества. Отношения. Числа. Величины. М. 1978.
22. Упражнения по логике. Под ред. В.И. Кириллова. М. 1993.
23. А.Н. Колмогоров, А.Г. Драгалин. Введение в математическую логику. М. 1982.
24. Г. Кук, Г. Бейз. Компьютерная математика М. 1990.
25. А. Парницкий. Основы статистической информатики. М. 1981. Дж. Элти, М. Кумбс. Экспертные системы. Концепции и примеры. М. 1987.
27. А. Моль. Теория информации и эстетическое восприятие. М 1966.
28. И. Яглом. Современная культура и компьютеры. М.1990.

Программа курса

«Алгоритмика и основы программирования» для групп 8 классов с углубленным изучением программирования

(4 часа в неделю, всего 128 часов)

Пояснительная записка

Курс рассчитан на учащихся 8 классов (продолжающие обучение и вновь набранные классы) и основывается на той их особенности, что предварительная подготовка учащихся часто крайне разнородна, при этом часть учащихся вновь набранных классов слабо изучали информатику и совсем не изучали программирование. По этой причине преподавание программирования ведется «с нуля», в предположении, что учащийся не обладает какими-либо специальными знаниями в области программирования. Кроме того, контингент учащихся часто разнороден по своим устремлениям — если среди учащихся не было произведено специального отбора, то среди них будут не только потенциальные математики и программисты, но и компьютерные графики, и биологи, и гуманитарии. По этой причине, чтобы знания курса оказались востребованы учащимися, а также для создания комфортных условий для обучения, используются следующие принципы:

1. Во главу угла ставится задача, ее решение и, главное, путь от задачи к решению, а не программирование (кодирование) как таковое.
2. Для записи алгоритма на языке программирования выбирается минимальное подмножество средств языка, чтобы не акцентировать внимания на кодировании и для более легкого перехода на другие языки программирования.
3. Самостоятельность решения является ключевым условием, которое необходимо доказать при сдаче работы.
4. Понимание учащимся тех средств, с помощью которых он решил задачу, ставится выше уровня самих средств решения.
5. Аккуратность и надежность решения ставятся выше «программистских трюков», иногда позволяющих в отдельных случаях добиться несколько лучших результатов.
6. Задачи ставятся в нескольких вариантах различной сложности (от базового до творческого), при сдаче работы засчитывается решение на любом уровне (но удовлетворяющее принципам 2-4). Однако уровень сложности фикс-

сируется и используется как дополнительная информация к оценке, полученной учеником. В частности, для выяснения уровня профессионализма ученика (и его повышения).

7. Главным методологическим принципом является системный подход.

Важнейшей частью курса является формирование системы профессиональных ценностей (предпочтений) ученика. В конечном счете, это формирование и есть основная инвариантная методологическая задача курса, так как все остальное — технология и будет неотвратно изменяться с течением времени. Этот аспект отражен в тематическом планировании в графе «Ученик должен чувствовать (воспитание)» – по существу, это *метауровень*.

Курс излагается с опорой на развитие самостоятельного мышления учащихся, самообразование. Для выполнения учащимися выбираются задания, допускающие решения в эстетически привлекательной форме, имеющие игровой компонент. Формы занятий — лекция, семинар и практическая работа. Формы контроля — сдача практических работ (основная), письменных контрольных работ, проверка конспектов.

В качестве инструментария используются среды и Code::Blocks и Bloodshed Dev-Cpp (компилятор gcc), а также для желающих – Microsoft Visual Studio. В качестве библиотеки программной поддержки – бесплатная графическая библиотека TX Library с открытым исходным кодом, разработанная автором и доступная по адресу <http://ded32.ru>.

Результатом прохождения курса учащимися должны стать понимание основных принципов программирования и владение основными алгоритмическими конструкциями. Кроме того, определенная часть учащихся должна быть сориентирована на углубленное изучение программирования — в качестве специализации. Остальная часть не должна воспринять программирование как непреодолимо сложную дисциплину, что позволит им продолжить обучение в этой области в 9, 10 и 11 классах и успешно написать задания части «С» ЕГЭ, традиционно считающиеся «не для всех».

Тематическое планирование курса

Часть 1. «Тупой Художник» (первая половина 1 семестра)

№	Тема / ученик должен знать / ученик должен уметь (обучение)	Ученик должен чувствовать (воспитание)
1	Задача и ее решение. Стадии решения задачи. Алгоритм как решение задачи. Способ записи алгоритмов с помощью псевдокода. Исполнитель алгоритма. Множество допустимых команд исполнителя, детализация алгоритма.	<p>Осознавать, выделять и не смешивать основные стадии решения задачи. Понимать необходимость формальной записи. Понимать вторичность процесса кодирования. Уметь «сживаться» с исполнителем и его возможностями. Понимать различие свойств исполнителя и человека. Уметь анализировать и исправлять ошибки.</p> <p>Получить ярко выраженные положительные эмоции от опыта с первой программой. Любить правильно оформлять программу (основы стиля). Понимать, что плохо написанная программа плохо понимается. Уметь отделять смысловые ошибки от синтаксических.</p> <p>Занятия 1-2 проводится в игровой форме (игра в «тупого художника»), что способствует лучшему пониманию материала и снятию психологического барьера перед программированием. Эти занятия спаренные, и, насколько это возможно, с минимальным перерывом.</p> <p>Занятие 3 носит консультативный характер.</p>
2	Практический пример решения задачи. Исполнитель «Тупой художник». Запись алгоритма в нескольких видах (псевдокод, блок-схема). Ошибки в алгоритме и способе его записи. Перевод алгоритма в программу (кодирование). Компьютер как исполнитель. Понятие о языке программирования.	
3	Основные возможности графической библиотеки TX Library («команды «тупого художника»). Система координат экрана. Решение задач на составление алгоритмов и их запись в различных формах.	
4	Практика: проект «мультфильм» (линейные алгоритмы и их программирование). Работа в среде Dev-Cpp (Code::Blocks). Элементарные операции в редакторе. Набор и отладка первой программы. Ошибки кодирования. Воспроизведение алгоритмических ошибок и ошибок кодирования.	

№	Тема / ученик должен знать / ученик должен уметь (обучение)	Ученик должен чувствовать (воспитание)
5	Смысловое разделение частей алгоритма. Концепция процедурного программирования. Функции без параметров. Оформление функций. Вызов функции. Пошаговая отладка в программе с функциями. Именованые функций. Понятие стиля программирования.	Понимать, что функции облегчают программирование и понимание программы. Правильно разделять работу между функциями. Радоваться созданию новой функции. Называть функции по смыслу, понимать роль «говорящих» имен. Не любить длинных функций.
6	Практика: рефакторинг проекта «мультфильм» с применением функций без параметров.	Относиться к функции как к инструменту, который должен быть удобным и безопасным. Не писать функций «с медвежьей услугой».
7	Функции с параметрами, возможности, которые они предоставляют. Грамотное использование параметров функции (взаимная независимость). Числовой тип данных. Отладка программы с переменными и параметрами. Понятие о библиотеке функций.	Уметь гибко «настраивать» функцию с помощью параметров, не любить негибкие функции. Уметь искать «золотую середину» при составлении набора параметров. Правильно называть параметры, любить «говорящие» имена. Группировать функции в библиотеки по смыслу.
8	Практика: рефакторинг проекта «мультфильм» с применением функций с параметрами.	Понимать, как библиотеки облегчают программирование и любить их использовать.
9	Повторяющиеся действия в алгоритмах. Разбор циклических алгоритмов. Цикл while. Работа с переменными в цикле. Отладка программы с циклами. Ошибки при работе с циклами.	Понимать, что циклы облегчают программирование и делают его гораздо более интересным. Опасаться заикливания и всегда контролировать триаду «начальное значение – изменение – конечное значение». Использовать отладчик и распечатку значений на экран для понимания выполнения, поиска ошибок (и иметь опыт этого поиска).
10	Практика: доработка проекта «мультфильм» с использованием циклических алгоритмов.	

№	Тема / ученик должен знать / ученик должен уметь (обучение)	Ученик должен чувствовать (воспитание)
11	Библиотеки функций. Размещение библиотеки в отдельном файле (или множестве файлов) и подключение к главной программе с помощью директивы <code>#include</code> . Оформление библиотеки и написание документации с помощью системы <code>doxygen</code> .	Понимать, что использование библиотек позволяет написать гораздо более мощную и объемную программу. Любить создавать библиотеки с понятной документацией. Радоваться, когда библиотеку используют другие. Уметь критиковать конструктивно (в форме советов по доработке) и принимать критику. Следить за версиями создаваемых и используемых библиотек.
12	Практика: Создание библиотеки героев с документацией и обмен библиотеками между учащимися. Создание короткого мультфильма с использованием только чужих героев и написание рецензии на используемую библиотеку. Доработка собственной библиотеки, после получения рецензии на нее.	
13	Работа с мышью в TX Library. Загрузка и использование изображений. Условные операторы и их использование.	Не расходовать ресурсы программы зря: освобождать дескрипторы загруженных рисунков («холсты») и не загружать рисунки слишком часто, что приводит к торможению программы.
14	Практика: доработка проекта «мультфильм» до проекта «простая игра».	
15	Задача о функции, изменяющей значения своих параметров (на примере функции движения объекта-шарика). Передача указателей на переменные и работа с ними.	Понимать, что использование указателей позволяет разбить на функции те части программы, где переменные изменяются (и где такое разбиение было ранее невоз-

№	Тема / ученик должен знать / ученик должен уметь (обучение)	Ученик должен чувствовать (воспитание)
16	Практика: рефакторинг проекта «игра» с использованием функций, принимающих указатели.	<p>возможным). Не видеть в указателях ничего страшного.</p> <p>Примечание: глобальные переменные и указательную арифметику не рассматривать! (Первое вредно, второе плохо понятно и пока не нужно.) В случае использования глобальных переменных учеником, если ему их кто-то посоветовал – показывать, насколько сложно с ними контролировать логику работы программы. Не запрещать, а отговаривать от их использования.</p>
17	Группировка данных, принадлежащих одному объекту, в структуры. Объявление структур, инициализация и использование структурных переменных. Передача структурных переменных в функции. Использование указателей на константные структуры для ускорения передачи параметров.	<p>Любить использовать структуры для облегчения работы с объектами. При определении структуры критически подходить к вопросу, что включать в нее, а что нет. Не любить «толстые» структуры (предпочитать им вложенные структуры). Не любить нелогичные структуры с логически несвязным содержимым. Любить передачу константных объектов для собственной безопасности.</p>
18	Практика: рефакторинг проекта «игра» с использованием структур.	
19	Написание документации к игре и выпуск версии программы. Создание презентации по программе. Технология хорошего доклада.	<p>Понимать, что без документации никто не будет использовать программу. Писать документацию не «от программиста», а «от нужд пользователя». Уметь ставить себя на</p>

№	Тема / ученик должен знать / ученик должен уметь (обучение)	Ученик должен чувствовать (воспитание)
20	Практика: написание документации, создание презентации и доклад на уроке-конференции.	<p>место пользователя.</p> <p>При создании презентации не увлекаться возможностями PowerPoint. Любить концентрироваться на главном. Любить излагать мысли с помощью схем. Не любить перегруженных и недогруженных слайдов. Любить и знать правильную технологию использования «шпаргалки для докладчика».</p> <p>На докладе уметь настраиваться на позитив вне зависимости от обстоятельств и доносить свои мысли до аудитории вне зависимости от уровня ее подготовки.</p>

**Часть 2. Компьютер и математика
(2 половина 2 семестра)**

21	Построение графиков заданных элементарных функций. Задача о построении графика функции. Алгебраическая система координат. Пересчет координат из одной координатной системы в другую. Задача об исследовании элементарной функции (интервалы возрастания и убывания, точки экстремума, нарушение области определения, скорость изменения функции).	<p>Уметь использовать принципы визуального изучения для исследования абстрактных понятий (на примере математической функции). Уметь и любить использовать для этой цели программирование.</p> <p>Получить элементарные навыки исследовательской работы. Уметь и любить исследовать функции и подбирать формулы функций по заданным графикам (решать обратную задачу).</p>
22	Практика: проект «графики функций». Изучение элементарных функций с параметрами. Особенности визуального исследования.	

23	Возврат значений в функции. Функция как «черный ящик». Множество значений функции, допустимые и исключительные значения. Понятие надежности функции. Концепция защитного программирования. Функции для пересчета координат. Проект «Знакомые графики» с функциями и модулем пересчета координат.	Правильно и безопасно обрабатывать ошибки в исходных данных. Не любить использовать глобальные переменные (кроме случаев, когда это крайне необходимо). Использовать соглашения об именах для выделения глобальных переменных. Понимать единство модуля (фактически, как целостного объекта).
24	Практика: рефакторинг проекта «графики» с использованием возврата значений и функциями пересчета координат.	
25	Кнопочный интерфейс и сложности, связанные с ним. Реализация кнопок с помощью структур. Сменные (skinnable) интерфейсы.	Понимать, что если решение сложной проблемы можно описать объектами (структурами) и написать отдельный модуль для управления ими (менеджер) – то проблема перестанет быть сложной. Получать удовольствие от упрятывания таких сложностей в модули. Любить модульные программы.
26	Практика: Разработка кнопочного интерфейса и простого менеджера кнопок.	
27	Выбор одной функции из семейства родственных. Способы выбора (нумерация целыми числами, использование enum, использование указателей на функции). Запись типа переменной – указателя на функцию.	Любить использовать enum за простоту и понятность, и предпочитать его обычным целым числам. Любить указатель на функции, не бояться их. Уметь легко описывать тип переменной-указателя (отталкиваясь от прототипа соответствующей функции). Оценивать технологии программирования с точки зрения расширяемости программного проекта.
28	Практика: рефакторинги проекта «графики»: с применением enum и с применением указателей на функции.	
29	Графики неравенств (тесты на принадлежность точки заданной области). Дальнейшее использование принципа визуализации.	Уметь строить алгоритмы «от определения». Закрепить навыки моделирования и визуального изучения. В задаче о трех цветных кругах увидеть связь с диаграммами

30	Практика: проект «графики неравенств». Построение алгоритма «от определения». Вложенные циклы	Венна (логика) и изучить смешение цветов (физика и биология).
----	---	---

Часть 3. «Возня с данными» (классика) (2 семестр)

31	Массивы. Алгоритмы работы с массивами (ввод, распечатка). Размер массива. Передача массива в функцию. Ошибка выхода за границы массива и ее контроль с помощью assert. Структура «вектор» (содержащая массив и его длину, это пока не std::vector, а очень упрощенная самодельная версия). Разбор задач на алгоритмы работы с массивами.	Не бояться использовать массивы. Быть осторожным с их длиной. Любить структуру «вектор». Понимать, что в языке C массив передается «в виде оригинала» (по ссылке) и остерегаться порчи оригинала, по этой причине любить const-параметры. Понимать, как возвращать ответ в виде массива. Любить макрос assert. Тренироваться в «возне с байтами» и защитном программировании.
32	Практика: Реализация простейших функций для работы с массивами (заполнение массивов, сумма элементов, последовательный поиск элемента).	
33	Проблема проверки правильности алгоритмов. Применение автоматического тестирования (юнит-тестирования).	Любить юнит-тестирование и стараться по возможности перекладывать на него все, что связано с формальной проверкой правильности алгоритмов. Писать юнит-тесты не после завершения работы над алгоритмом, а до начала этой работы (и во время работы). Любить избыточное тестирование и не забывать про тестирование отказов. Никогда не стирать из программы юнит-тесты после завершения работы над ней.
34	Практика: реализация юнит-тестирования для уже написанных алгоритмов. Работа с файлами. Реализация библиотеки функций для работы с массивами (уже написанные функции + среднее значение, циклические сдвиги вправо-влево, обращение, двоичный поиск и т.д.)	
35	Строки как вид массивов. Максимальная и текущая длины строки. Основные алгоритмы над строками (копирование, сравнение, поиск буквы и подстроки, подсчет слов, удаление подстроки, удаление пробелов, поиск и замена подстроки).	Не путать максимальную и текущую длины строки. Бояться ошибок переполнения буфера и постоянно контролировать все, что только можно, с помощью assert.

36	Практика: проект «строковая библиотека».	
37	Алгоритмы быстрого поиска (Рабина-Карпа, Бойера-Мура). Для сильных групп – алгоритм Кнута-Морриса-Прата.	Понимать важность эффективности алгоритмов и стараться постоянно анализировать ее. Любить анализировать и придумывать алгоритмы. Любить реализовывать их эффективно и понятно. Уметь определять «узкие места» алгоритмов (в том числе и с помощью замеров времени выполнения). Понимать, как дополнительные структуры данных облегчают работу (на примере алгоритма Бойера-Мура).
38	Практика: проект «строковая библиотека», реализация алгоритмов быстрого поиска с юнит-тестированием.	
39	Перевод чисел в строковую форму и обратно в разных системах счисления. Представление отрицательных чисел. Сложение и вычитание в различных системах счисления.	Понимать, что самый скучный школьный материал становится интереснее, если применяется программирование. (Системы счисления – обязательная тема в школьной программе, как правило, при изучении довольно скучная.)
40	Практика: реализация функций перевода с юнит-тестированием. Использование функций перевода для шифрования текста.	
41	Матрицы (многомерные массивы) и массивы строк. Основные навыки работы с многомерными массивами. Представление матриц в машине (расположение в оперативной памяти). Работа с динамической памятью.	Не бояться матриц. Беречь память при использовании прямоугольных матриц. Предпочитать им массивы указателей, если это возможно для данной задачи.
42	Практика: решение задач, связанных с матрицами.	
43	Сортировка в массивах. Основные виды сортировки. Разбор задач, связанных с сортировкой. Эффективность алгоритма сортировки. Алгоритм сортировки выбором (через анализ человеческих действий).	Любить анализировать эффективность алгоритмов. Иметь опыт выбора алгоритма для решения конкретной задачи. Понимать, что знание одного алгоритма – не настоящее знание, так как всегда нужно, чтобы было с чем

44	<p>Практика: реализация сортировки выбором с юнит-тестированием и сбором статистики эффективности (количество обменов и сравнений). Построение графиков эффективности.</p> <p>Сортировка стихотворных строк со сравнением «начиная с начала» и «начиная с конца». Словари рифм.</p>	<p>сравнивать. Понимать, что идеального алгоритма часто нет и много что зависит от специфики условий задачи. Иметь опыт конкретного анализа алгоритма, сбора статистики и т.д. и сравнения с теорией.</p> <p>Для сложных объектов (строк, структур) обращать внимание на семантику операций сравнения (по каким критериям оно проводится). Понимать, что иногда элементарные операции занимают много времени (например, лексикографическое сравнение строк, обмен двух строк через копирование букв), а иногда, при удачном подборе входных данных – немного времени (например, обмен двух строк через обмен указателей на них).</p>
45	<p>Алгоритм сортировки пузырьком (через определение отсортированной последовательности). Эффективность алгоритма сортировки пузырьком. Адаптивный характер алгоритма. Шейкерная сортировка.</p>	
46	<p>Практика: реализация сортировки «пузырьком» с юнит-тестированием и сбором статистики эффективности (количество обменов и сравнений). Построение графиков эффективности.</p>	<p>Тренировать навыки осознанного выбора алгоритмов и формы представления данных.</p>
47	<p>Другие алгоритмы сортировки (простые вставки, бинарные вставки, подсчитывающая, слиянием, Шелла, быстрая). Сравнение эффективности алгоритмов.</p>	
48	<p>Практика: проект «сравнение сортировок». Реализация алгоритмов с юнит-тестированием и сбором статистики эффективности (количество обменов и сравнений). Разработка полноценной программы с интерфейсом для построения и сравнения графиков эффективности разных алгоритмов сортировки. Написание документации и создание презентации. Выступление на уроке-конференции с анализом одного из алгоритмов и демонстрацией программы.</p>	

49	Заключительное занятие. Конечный результат или возня с байтами? (размышления о будущей профессии).	Понимать, что основа любой профессии — это не только и не сколько получение результата, сколько любовь к возне с мелочами (соответствующими данной предметной области). Понять для себя, что в течение года было важнее в плане программирования и что-то решить на будущее: углубляться дальше или заняться только пользовательским курсом.
----	--	--

Список рекомендованной учебной и методической литературы

Учебная литература (для учеников)

1. Алгоритмика, 5-7 классы. Под ред. А.Л. Семенова. Дрофа, 1998.
2. Брудно А.Л., Каплан Л.И. Московские олимпиады по программированию. М., Наука, 1990.
3. Котов Ю.В., Павлова А.А. Основы машинной графики. М., Просвещение, 1993.
4. Першиков В.И., Савинков В.М. Толковый словарь по информатике. М., Финансы и статистика, 1995.
5. С. Прага. Язык программирования Си. Киев, ДиаСофт, 2001.
6. С. Прага. Язык программирования С++. Киев, ДиаСофт, 2001.
7. Уэйт У., Мартин Дж., Прага Л. Язык Си для начинающих. М., Мир, 1988.
8. Шень А.Х. Алгоритмика (учебник для математических классов). Части 1 и 2. М., МЦНМО, 1995.

Консультативная и методическая литература (для учителя)

1. Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию. М., Наука, 1988.
2. Дагене В.А. и др. 100 задач по программированию. М., Просвещение, 1993.
3. Зенкин А.А.. Когнитивная компьютерная графика. М.: Наука, 1991.
4. Керниган Б., Ритчи Д.. Язык программирования С++. М., Финансы и статистика, 1980.
5. Котов Ю.В. Как рисует машина. М., Наука, 1988.
6. Паронджанов В.Д. Как улучшить работу ума. (Новые средства для образного представления знаний, развития интеллекта и взаимопонимания) – М.: Радио и связь, 1998.
7. Тихомирова Л.Ф., Басов А.В.. Развитие логического мышления детей. М., 1995.
8. Шикин Е.В., Боресков А.В., Зайцев А.А. – Начала компьютерной графики. М., “ДИАЛОГ-МИФИ”, 1993.

Приложение. Материалы к экзамену по информатике

1. Вопросы к экзамену

1. Общая схема рассмотрения алгоритмической задачи. Примеры применения схемы рассмотрения задач.
2. Понятие об исполнителях. Свойства исполнителя. Понятие о системе команд. Ошибки при работе с исполнителем.
3. Способы записи алгоритмов. Примеры и сравнительная характеристика способов записи.
4. Свойства и классификация алгоритмов. Понятие об эффективности алгоритмов (по времени).
5. Ошибки в программировании и их виды. Классификация ошибок, методы их устранения.

6. Общая структура программы. Модульный принцип построения программ. Ошибки, связанные с нарушением модульного принципа.
7. Понятие о стиле программирования. Ошибки, связанные с нарушением стиля программирования.
8. Понятие переменной. Имя, значение и тип переменной. Ошибки при работе с переменными, примеры возникновения и устранения.
9. Понятие подпрограммы. Виды подпрограмм. Параметры и переменные подпрограммы.
10. Характерные ошибки при работе с подпрограммами. Примеры возникновения и устранения.
11. Локальные переменные, их назначение и свойства. Понятие об области видимости и времени жизни переменной.
12. Циклические алгоритмы и программы. Схема циклического алгоритма. Виды циклов в программе.
13. Характерные ошибки при работе с циклическими алгоритмами. Примеры возникновения и устранения.
14. Вложенные циклы и работа с ними. Задача о графике неравенства. Построение алгоритма графического решения неравенства.
15. Условные алгоритмы и программы. Условный оператор и его виды. Характерные ошибки при работе с условными алгоритмами.
16. Понятие массива. Структура и свойства массива. Основные приемы работы с массивами.
17. Характерные ошибки при работе с массивами. Понятие о доказательном программировании.
18. Понятие сортировки массива. Виды сортировки. Понятие об эффективности алгоритма сортировки.
19. Алгоритм сортировки массива методом выбора. Анализ эффективности в сравнении с другими алгоритмами.
20. Алгоритм сортировки массива методом пузырька. Анализ эффективности в сравнении с другими алгоритмами.
21. Сравнение эффективности некоторых методов сортировки массивов. Применение различных методов сортировки.
22. Алгоритмы поиска данных. Виды алгоритмов поиска. Последовательный поиск. Анализ эффективности работы последовательного поиска.
23. Алгоритм двоичного поиска в массиве. Свойства алгоритма, анализ эффективности.
24. Понятие строки. Основные приемы работы со строками. Ошибки при работе со строками. Применение схем алгоритма и отладчика для устранения

ошибок.

25. Сравнение свойств строк различной структуры. Анализ применимости различных видов строк.
26. Алгоритмы перевода чисел в различные системы счисления. Анализ и примеры реализации.

2. Задачи к экзамену

Вставки/удаления

1. Дана текстовая строка. Каждую гласную, находящуюся в ней, размножить в 3 раза.
2. Дана текстовая строка, возможно, содержащая ошибки вида «жы» и «шы». Заменить их на правильные варианты и добавить после каждой ошибки восклицательный знак.
3. Дана текстовая строка, возможно, содержащая запятые. Вставить после каждой запятой или точки слово «однако».
4. Дана текстовая строка, содержащая предложение. Поменять в строке местами первые два слова и в конце поставить вопросительный знак.

Копирование с условием

5. Дана текстовая строка, возможно, содержащая пробелы. Удалить их из строки полностью.
6. Дана строка, содержащая текст, части которого могут быть заключены в скобки. Во вторую строку переписать текст из первой, за исключением находящегося в скобках.
7. Дана текстовая строка. Записать во вторую строку все буквы, встречающиеся в первой строке (каждая буква — только 1 раз).
8. Дана текстовая строка. Напечатать, какие буквы в ней встречаются и сколько раз.

Сортировка

9. Дана текстовая строка. Переставить в ней буквы так, чтобы в начале оказались большие буквы, затем маленькие.
10. Дан целочисленный массив. Расположить в нем числа следующим образом: вначале положительные, затем отрицательные, затем нули.
11. Дан целочисленный массив. Расположить в нем числа следующим образом: вначале ненулевые четные, затем нули, затем нечетные.
12. Дан целочисленный массив. Отсортировать в нем числа по возрастанию их модуля.

Поиск

13. Для заданного символа за минимальное время определить, является ли он гласной или согласной буквой русского языка.
14. Даны две текстовые строки. Определить длину участка первой строки, состоящую только из символов, присутствующих во второй строке.
15. Даны две текстовые строки. Определить длину участка первой строки, состоящую только из символов, отсутствующих во второй строке.
16. Даны две текстовые строки. Определить, все ли буквы из первой строки содержатся во второй строке.
17. Дан целочисленный массив. Найти в нем первое встреченное простое число.
18. Дан массив, содержащий последовательность целых чисел. Оставить в нем только простые числа (остальные обнулить).

Системы счисления

19. Дана текстовая строка, содержащая двоичное число. Перевести его в восьмеричную систему счисления и записать во вторую строку.
20. Дана текстовая строка, содержащая двоичное число. Перевести его в шестнадцатеричную систему счисления и записать во вторую строку.

Расстановки

21. Дана текстовая строка. Определить, является ли она палиндромом (справа налево и слева направо читается одинаково).
22. Дан целочисленный массив. Распечатать его в следующем порядке: сначала нулевой элемент, затем последний, затем первый, затем предпоследний и т.д.
23. Дана текстовая строка. Определить максимальную длину ее участка, образованного одной повторяющейся буквой.
24. Дан целочисленный массив. Найти в нем длину максимальной возрастающей последовательности чисел.
25. Дан целочисленный массив. Найти в нем длину максимальной убывающей последовательности чисел.

Программа курса
«Технология объектно-ориентированного
программирования»
для 9 мат. классов

(4 часа в неделю, всего 128 часов)

Пояснительная записка

Курс рассчитан на учащихся 9 классов, выбравших физико-математический профиль и основывается на курсе «Алгоритмика» 8 класса. Цель курса — развить и закрепить навыки структурного программирования, ввести объектно-ориентированную методологию программирования. Важным аспектом курса является отработка организации мышления ученика — развитие системного подхода, методологии анализа и синтеза решений, планирования, реализации.

Предлагаемый курс, как и серия других спецкурсов профиля «Системное программирование», представляет собой систематическое изложение методологии и технологии структурного и объектно-ориентированного программирования, языков Си и Си++, представленное в виде работы над серией проектов. Нарботки, полученные учащимися (программы и библиотеки), активно используются затем в дальнейшем обучении, что подчеркивает взаимосвязь материала и его актуальность. Одна из задач курса — разработка среды визуализации алгоритмов для практикума по курсу алгоритмов и структур данных, а также подготовка к курсу «Программирование в ОС Windows» 10-11 класса. По этой причине для обучения используются следующие принципы:

1. Во главу угла ставится задача, ее решение и, главное, путь от задачи к решению. Во всякой задаче подчеркивается разделение на идею решения и технологию реализации.
2. Самостоятельность решения является ключевым условием, которое необходимо доказать при сдаче работы.
3. Понимание учащимся тех средств, с помощью которых он решил задачу, ставится выше уровня самих средств решения.
4. Аккуратность и надежность решения ставятся выше «программистских трюков», иногда позволяющих в отдельных случаях добиться несколько лучших результатов.
5. Задачи ставятся в нескольких вариантах различной сложности (от базового до творческого), при сдаче работы засчитывается решение на любом уровне.

не (но удовлетворяющее принципам 2-4). Однако уровень сложности фиксируется и используется как дополнительная информация к оценке, полученной учеником. В частности, для выяснения уровня профессионализма ученика (и его повышения).

Курс излагается с опорой на развитие самостоятельного мышления учащихся, самообразование. Для выполнения учащимися выбираются задания в виде проектов с большой возможностью развития темы. Формы занятий — лекция, семинар и практическая работа. Формы контроля — сдача практических и проектных работ (основная), письменных контрольных работ, проверка конспектов.

В качестве инструментария используются среды разработки Dev.-C++, Code::Blocks, Microsoft Visual Studio.

Результатом прохождения курса учащимися должно стать практическое овладение технологией объектно-ориентированного процедурного программирования.

Тематическое планирование

1. Повторение основных принципов структурного программирования. Структура программы в Си. Переменные и их типы. Организация потока управления (повторение). Проект «модель идеального газа» (постановка и разбор задачи). Каркасный код проекта. Простейшая реализация проекта.
2. Проект «модель идеального газа». Постановка задачи. Классы. Определение класса. Данные и методы класса. Конструктор и деструктор. Защита данных. Ключевые моменты в проектировании класса.
3. Наследование. Приведение к базовому классу. Виртуальные функции. Виртуальный деструктор. Понятие полиморфизма. Функциональный подход к построению базового класса.
4. Внутренний механизм работы виртуальных функций и способы его реализации. Альтернативные механизмы виртуальности функций.
5. Управление совокупностью объектов. Классы-менеджеры. Задача о простейшем физическом движении.
6. Взаимодействие объектов в простейшем физическом движении.
7. Задача о «химическом движении». Моделирование химических реакций. Динамическое приведение типов. Разработка программы моделирования реакций.
8. Переопределение операторов в C++. Класс «вектор» и реализация операторов векторной алгебры. Реализация моделей освещения тела по Ламберту и по Фонгу.
9. Проект «Строковая библиотека на Си++». Класс «Массив». Переопределение операторов в Си++. Ссылки и работа с ними. Понятие о Lvalue. Понятие о шаблонах.

10. Особенности работы с ресурсами в Си++. Конструктор копирования и оператор присваивания.
11. Матричная библиотека на Си++. Реализация основных действий над матрицами.
12. Проект «Многооконная библиотека». Понятие о событийной модели программирования. Постановка задачи.
13. Понятие о характеристиках и контексте окна. Отсечение графических примитивов. Реализация функций вывода в окно.
14. Реализация оконного менеджера. Проектирование иерархии классов оконной библиотеки.
15. Понятие о фокусе ввода. Ввод с клавиатуры. Реализация работы с клавиатурой. Ввод текстовой строки.
16. Проектирование элементов интерфейса. Окно приложения, окно сообщения, окно ввода, таблица, текстовый редактор.
17. Реакция системы на команды. Реализация командных кнопок. Указатели на членов класса. Статические методы класса.
18. Захват управления, модальные окна. Иерархия оконных менеджеров. Реализация диалогового окна.
19. Реализация меню. Одноуровневые меню. Реакция на «горячие клавиши». Представление об организации иерархических меню.
20. Понятие о многозадачности. Невытесняющая многозадачность. Библиотека задач. Множественное наследование и проблемы, связанные с ним. Виртуальные базовые классы, интерфейсы
21. Обработка ошибок. Старые методы. Понятие об исключениях. Механизм обработки исключений в Си++.
22. Проблемы перерисовки и прямоугольной области отсечения. Алгоритм отсечения по прямоугольной области.
23. Алгоритмы обработки растровых изображений. Реализация простейших фильтры: негатив, размытие, контурная резкость, барельеф, контур, изогелия, соляризация. Алгоритм эквализации яркостей.

Список рекомендованной учебной и методической литературы

Учебная литература (для учеников)

1. Березин Б.И., Березин С.Б. Начальный курс С и С++. М., ДИАЛОГ-МИФИ, 1999.
2. Брудно А.Л., Каплан Л.И. Московские олимпиады по программированию. М., Наука, 1990.
3. Керниган Б., Ритчи Д. Язык программирования Си: Пер. с англ. – 2-е изд., перераб. и доп. М., Финансы и статистика, 1992.
4. Котов Ю.В., Павлова А.А. Основы машинной графики. М., Просвещение, 1993.
5. Першиков В.И., Савинков В.М. Толковый словарь по информатике. М., Финансы и

статистика, 1995.

6. Страуструп Б. Язык программирования Си++. М., 1999.
7. С. Прага. Язык программирования Си. Киев, ДиаСофт, 2001.
8. С. Прага. Язык программирования С++. Киев, ДиаСофт, 2001.
9. Уэйт У., Мартин Дж, Прага Л. Язык Си для начинающих. М., Мир, 1988.
10. Шень А.Х. Алгоритмика (учебник для математических классов). Части 1 и 2. М., МЦНМО, 1995.

Консультативная и методическая литература (для учителя)

11. Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию. М., Наука, 1988.
12. Аммерал Л. Машинная графика на персональных компьютерах. Пер. с англ. М., “Сол Систем”, 1992.
13. Аммерал Л. Принципы программирования в машинной графике: Пер. с англ. М., “Сол Систем”, 1992.
14. Аммерал Л. Программирование графики на Турбо Си. Пер. с англ. М., “Сол Систем”, 1992.
15. Бейбер Р.Л. Программное обеспечение без ошибок. Пер. с англ. М., Джон Уайли энд Санз, Радио и связь, 1996.
16. Боресков А.В., Шикин Е.В., Шикина Г.Е. Компьютерная графика: первое знакомство. М., Финансы и статистика, 1996.
17. Бочков С.О., Субботин Д.М. Язык программирования Си для персонального компьютера. М., Радио и связь, 1990.
18. Брой М. Информатика. В 4-х частях. Пер. с нем. М., “ДИАЛОГ-МИФИ”, 1996.
19. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. 2-е изд. Пер. с англ. М., “Издательство Бинум”, СПб.: “Невский диалект”, 1999.
20. Вайнер Р., Пинсон Л. С++ изнутри. Пер. с англ. – К.: НПФИФ “ДиаСофт”, 1993.
21. Дагене В.А. и др. 100 задач по программированию. М., Просвещение, 1993.
22. Джордейн Р. Справочник программиста персональных компьютеров типа IBM PC, XT и AT. Пер. с англ. М., Финансы и статистика, 1992.
23. Котов Ю.В. Как рисует машина. М., Наука, 1988.
24. Липаев В.В. Надежность программных средств. Серия “Информатизация России на пороге XXI века”. М., СИНТЕГ, 1998.
25. Лукас П. С++ под рукой. Пер. с англ. – К.: НПФИФ “ДиаСофт”, 1993.
26. Мика Б. и др. Практическое руководство по программированию: Пер. с англ. М., Радио и связь, 1986.
27. Намиот Д.Е. Основные особенности языка программирования С++. Реализация Turbo С++. М., МП “Память”, 1991.
28. Проектирование пользовательского интерфейса на персональных компьютерах. Стандарт фирмы IBM. – Вильнюс: DBS Ltd., 1992.
29. Рассохин Д. От С к С++. М., “ЭДЕЛЬ”, 1993.
30. Советов Б.Я. Информационная технология: Учеб. для вузов. М., Высшая школа, 1994.
31. Уилтон Р. Видеосистемы персональных компьютеров IBM PC и PS/2. Руководство по программированию. Пер. с англ. М., Радио связь, 1994.
32. Фролов А.В., Фролов Г.В. MS-DOS для программиста. Серия “Библиотека системного программиста”. Тома 18-19, части 1-2. М., ДИАЛОГ-МИФИ, 1995.
33. Фролов А.В., Фролов Г.В. Операционная система MS-DOS. Серия “Библиотека системного программиста”. Том 1, книги 1, 2, 3. М., ДИАЛОГ-МИФИ, 1991.
34. Фролов А.В., Фролов Г.В. Программирование видеоадаптеров CGA, EGA и VGA. Серия “Библиотека системного программиста”. Том 3. М., ДИАЛОГ-МИФИ, 1992.

35. Фролов А.В., Фролов Г.В. Программирование видеоадаптеров. Серия “Библиотека системного программиста”. Том 21. М., ДИАЛОГ-МИФИ, 1995.
36. Шикин Е.В., Боресков А.В., Зайцев А.А. – Начала компьютерной графики. М., “ДИАЛОГ-МИФИ”, 1993.
37. Шикин Е.В., Плис А.И. Кривые и поверхности на экране компьютера. Руководство по сплайнам для пользователей. М., “ДИАЛОГ-МИФИ”, 1996.
38. Шилдт Г. Теория и практика С++. Пер. с англ. – СПб.: ВНУ, 1996.

Программа факультативного курса

«Алгоритмические задачи»

для 9 мат. классов

(3 часа в неделю, всего 96 часов)

Пояснительная записка

Курс рассчитан на учащихся 9 классов физико-математического профиля и основывается на курсах «Алгоритмика» и «Технология ООП» 8 и 9 класса. Цель курса — развитие алгоритмической культуры учеников, получение навыков решения различных алгоритмических задач.

Курс состоит из двух разделов: «Элементарные вычислительные методы» и «Нечисленные и полунисленные алгоритмы». В первом изучаются основы компьютерных вычислений и методы работы с численными данными. Материал раздела может (и должен) использоваться затем в изучении смежных предметов, например, расчетов результатов лабораторных работ по физике. Во второй части рассматриваются элементы теории множеств и отношений, последовательности, переборные задачи и простейшие алгоритмы на графах. Эта часть курса является подготовительной к центральному курсу профиля «Алгоритмы и структуры данных II», изучаемым в 10 классе.

Материал курса активно используется при выполнении курсовой работы, в которую включаются также результаты практических работ курса «Технология ООП».

Изложение ведется с опорой на развитие самостоятельного мышления учащихся, самообразование. Для выполнения учащимися выбираются задания, допускающие глубокое развитие темы. Формы занятий — лекция, семинар и практическая работа. Формы контроля — сдача практических работ (основная), письменных контрольных работ, проверка конспектов.

В качестве инструментария используется язык программирования C++ и среды разработки Dev-Cpp, Code::Blocks и Microsoft Visual Studio.

Тематическое планирование

Часть 1. Элементарные вычислительные методы

1. Технология программирования. Разбор этапов создания программы Идеология алгоритмизации и программирования (повторение). Математические ме-

тоды решения задач. Понятие о вычислительных методах. Сравнение аналитических и вычислительных методов. Особенности вычислительных методов. Понятие о точности вычислений.

2. Задача о поиске корней уравнений непрерывных функций. Формальная постановка задачи. Особенности постановки задачи для численного расчета. Программа нахождения корней уравнений простым методом итераций. Программа нахождения корней уравнений простым методом итераций (практическая работа).
3. Особенности вычислительных методов (обсуждение). Ошибки, связанные с поиском численных решений уравнений и способы их устранения. Особенности работы с действительными числами. Понятие о точности решения. Программа нахождения корней уравнений простым методом итераций и методом смены знака с графической иллюстрацией (практическая работа).
4. Численное решение уравнений методом деления отрезка пополам для непрерывных монотонных функций. Сходимость метода. Зависимость точности метода от времени расчета. Алгоритм и программа поиска решений. Программа нахождения корней уравнений методом деления отрезка пополам с графической иллюстрацией (практическая работа).
5. Способы задания функций. Построение графиков функций по формулам (повторение). Понятие об интерполяции функций. Задача интерполяции. Понятие об интерполяционном полиноме. Схема программы интерполяции функций. Построение графиков функций с помощью кусочно-линейной интерполяции (практическая работа).
6. Интерполяционные формулы порядка более 1. Квадратичная интерполяция. Интерполяционные многочлены Лагранжа и Ньютона. Кубическая интерполяция сплайнами (полиномы Безье). Сравнение различных методов интерполяции. Построение графика функций при помощи интерполяции полиномами Ньютона и Безье (практическая работа).
7. Понятие о сигнале и шуме. Задача отделения сигнала от шума. Сглаживание функций и данных. Использование сглаживания для анализа экспериментальных данных в естественных науках. Сглаживание функций методом усреднения полиномиальными окнами (практическая работа).
8. Восстановление функциональных зависимостей (фиттинг). Построение естественного критерия для оценки качества фиттинга. Построение линейной зависимости при помощи метода наименьших квадратов. Простейшие вероятностные методы. Алгоритм оптимизации методом Монте-Карло. Нахождение параметров линейной и нелинейной функций методом Монте-Карло (практическая работа).
9. Метод градиентного спуска. Дельта-поверхность и путь оптимизации на дельта-поверхности. Понятие производной и частной производной, алгоритм вычисления. Геометрическая интерпретация метода. Алгоритм оптимизации методом градиентного спуска. Комбинация с методом Монте-Карло. Нахож-

дение параметров линейной и нелинейной функций методом градиентного спуска (практическая работа).

10. Понятие о моделировании. Виды моделей. Математическое моделирование физического эксперимента. Этапы моделирования, модельный эксперимент, цикл жизни модели. Моделирование движения тел под действием сил тяжести. Понятие о дискретизации (дифференцировании). Понятие об интегрировании. Практическая работа над модельной задачей.

Часть 2. Нечисленные и получисленные алгоритмы

1. Элементы дискретной математики. Множества и их свойства. Множества и последовательности. Алгоритмы генерации множеств. Разложение числа на слагаемые и на множители. Множество простых чисел. Реализация операций над множествами.
2. Отношения и их свойства. Замыкание отношения, построение транзитивного замыкания отношения, алгоритм Уоршалла. Понятие о классах эквивалентности.
3. Элементы комбинаторики. Сочетания и перестановки. Порождение комбинаторных объектов. Коды Грея и аналогичные задачи. Подсчет мощности множеств комбинаторных объектов.
4. Элементы булевой алгебры. СДНФ булевой функции. Алгоритм построения таблиц истинности для СДНФ.
5. Поиск с возвратами. Задача о восьми ферзях. Задача о выходе из лабиринта. Рекурсивные алгоритмы.
6. Алгоритмы на последовательностях (сравнение последовательностей, поиск, нахождение общей подпоследовательности).
7. Основные понятия теории графов. Представления графов. Элементарные алгоритмы на графах. Поиск в глубину и в ширину. Поиск кратчайших путей.
8. Элементы растровой графики. Построение отрезка. Отсечение отрезков. Заполнение многоугольников. Заполнение произвольных областей.

Список рекомендованной учебной и методической литературы

Учебная литература (для учеников)

1. Брудно А.Л., Каплан Л.И. Московские олимпиады по программированию. М., Наука, 1990.
2. Новиков Ф.А. Дискретная математика для программистов. Учебник. СПб, Питер, 2001.
3. Першиков В.И., Савинков В.М. Толковый словарь по информатике. М., Финансы и статистика, 1995.
4. С. Прата. Язык программирования Си. Киев, ДиаСофт, 2001.

5. С. Прага. Язык программирования С++. Киев, ДиаСофт, 2001.
6. Уэйт У., Мартин Дж, Прага Л. Язык Си для начинающих. М., Мир, 1988.
7. Шень А.Х. Алгоритмика (учебник для математических классов). Части 1 и 2. М., МЦНМО, 1995.
8. Шикин Е.В., Плис А.И. Кривые и поверхности на экране компьютера. Руководство по сплайнам для пользователей. М., “ДИАЛОГ-МИФИ”, 1996.

Консультативная и методическая литература (для учителя)

9. Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию. М., Наука, 1988.
10. Бейбер Р.Л. Программное обеспечение без ошибок. Пер. с англ. М., Джон Уайли энд Санз, Радио и связь, 1996.
11. Брой М. Информатика. В 4-х частях. Ч. 1. Пер. с нем. М., “ДИАЛОГ-МИФИ”, 1996.
12. Дагене В.А. и др. 100 задач по программированию. М., Просвещение, 1993.
13. Керниган Б., Ритчи Д. Язык программирования С++. М., Финансы и статистика, 1980.
14. Кнут Д. Искусство программирования для ЭВМ. Том 1-3. Пер. с англ. М., Мир, 1978.
15. Котов Ю.В., Павлова А.А. Основы машинной графики. М., Просвещение, 1993.
16. Ласло М. Вычислительная геометрия и компьютерная графика на С++: Пер. с англ. М., “БИНОМ”, 1997.
17. Липаев В.В. Надежность программных средств. Серия “Информатизация России на пороге XXI века”. М., СИНТЕГ, 1998.
18. Мика Б. и др. Практическое руководство по программированию: Пер. с англ. М., Радио и связь, 1986.
19. Паронджанов В.Д. Как улучшить работу ума. (Новые средства для образного представления знаний, развития интеллекта и взаимопонимания) – М.: Радио и связь, 1998.
20. Ракитин В.И., Первушин В.Е. Практическое руководство по методам вычислений с приложением программ для персональных компьютеров: Учеб. пособие. М., Высшая школа, 1998.
21. Советов Б.Я. Информационная технология: Учеб. для вузов. М., Высшая школа, 1994.
22. Страуструп Б. Язык программирования Си++. М., 1999.
23. Шикин Е.В., Боресков А.В., Зайцев А.А. – Начала компьютерной графики. М., “ДИАЛОГ-МИФИ”, 1993.

Программа курса
«Алгоритмы и структуры данных»
для 10 мат. классов
(4 часа в неделю, всего 128 часов)

Пояснительная записка

Курс рассчитан на учащихся 10 классов физико-математического профиля и основывается на курсах «Алгоритмика» и «Технология ООП» 8 и 9 классов.

Данный курс является центральным в обучении по профилю «Системное программирование». В нем суммируются знания, полученные на предыдущих курсах, и создается основной фундамент для изучения последующего материала. Курс интенсивно развивает абстрактное мышление учащихся, умение формализовать задачу и выбрать наиболее эффективный способ ее реализации.

Курс состоит из трех частей. В первой части систематически излагаются свойства основных структур данных и алгоритмов над ними. После каждого раздела выполняются практические работы. Задания для них подобраны так, чтобы потом легко составить основу для курса по трансляторам (3 часть). Во второй части излагаются основы функционального и логического программирования (языки Лисп и Пролог), как некоторая альтернатива процедурному программированию. Изложение ведется с опорой на первую часть курса. В третьей части рассматриваются вопросы трансляции с машинных языков и эмуляции вычислительных систем, фундаментальные для системного программирования. Итогом работы по курсу трансляторов является курсовая работа, которая сдается в конце года и может являться частью итогового проекта в 11 классе.

Курс излагается с опорой на развитие самостоятельного мышления учащихся, самообразование. Для выполнения учащимися выбираются задания, допускающие при желании значительную степень углубления. Формы занятий — лекция, семинар и практическая работа. Формы контроля — сдача практических работ (основная), письменных контрольных работ, проверка конспектов.

В качестве инструментария используется язык программирования C++ и среды разработки Dev-Cpp, Code::Blocks и Microsoft Visual Studio.

Результатом прохождения курса учащимися должны стать понимание принципов работы со структурами данных, алгоритмами над ними, овладение технологией реализации широкого спектра алгоритмов.

Тематическое планирование

Часть 1. Структуры данных и алгоритмы над ними

1. Понятие об алгоритме, его свойствах и реализации, эффективности (повторение). Виды эффективности. Понятие об оптимальном алгоритме. Понятие о структуре данных. Использование различных структур данных для повышения эффективности алгоритмов. Основные виды структур данных и их особенности.
2. Примеры использования различных алгоритмов и структур данных на примере задачи о поиске в словаре. Идея алгоритмов линейного поиска, бинарного поиска, поиска в хеш-таблице и дереве. Эффективность алгоритмов поиска. Компромисс между эффективностью по времени и эффективностью по памяти. Сравнительная эффективность алгоритмов.
3. Понятие об абстрактной структуре данных и контейнерном объекте. Классификация методов контейнерного объекта. Методы, общие для всех объектов вообще. Назначение и классификация. Методы, общие для всех контейнерных объектов. Понятие о методах, специфичных для каждого данного контейнерного объекта. Понятие обобщенных методов.
4. Хранение данных в контейнерных объектах. Реализация «хранилища» и доступа к нему. Доступ к данным по индексам. Исключительные ситуации, связанные с использованием структур данных и их обработка. Использование нулевого индекса как индикатора исключительной ситуации. Примеры использования индексного доступа и исключительных ситуаций.
5. Понятие согласованности данных в объекте. Ошибки, связанные с нарушением согласованности. Методы проверки согласованности. Контрольная распечатка объекта. Итераторы контейнерных объектов. Виды итераторов и их использование в программе.
6. Объектно-ориентированный подход в программировании. Понятие объекта, его данных и методов. Использование объектов для выражения понятий. Объект как тип языка программирования. Описание объекта, создание его экземпляров и работа с ними. Общность и индивидуальность экземпляров объекта.
7. Понятие метода объекта. Описание, определение и использование методов. Контекст использования метода объекта. Особенности работы с переменной, являющейся экземпляром объекта. Доступ к данным объекта и вызов методов. Стратегия работы с данными объекта. Ошибки, связанные с применением объектно-ориентированного подхода в программировании.
8. Объект «Стек». Основные характеристики как абстрактной структуры данных и набор методов объекта. Классификация методов. Реализация методов объекта «Стек». Особенности реализации специфичных методов. Сравнительная характеристика эффективности различных способов реализации методов добавления и удаления элементов стека.

9. Примеры использования стека в различных задачах. Программа стекового калькулятора. Ошибки, связанные с реализацией стека и его использованием. Примеры возникновения и устранения.
10. Принципы построения и эмуляции вычислительных систем. Проектирование и реализация безадресного интерпретатора. Обработка ошибок различных видов.
11. Простейший безадресный компилятор. Понятие байт-кода. Эмуляция безадресного процессора. Отладчик для эмулятора. Транслятор с языка ассемблера для безадресной машины.
12. Практическая работа «Объект Стек». Программа стекового калькулятора.
13. Практическая работа «Реализация условных и безусловных переходов и функций в стековой машине».
14. Практическая работа «Реализация транслятора (ассемблера) для стекового калькулятора». Особенности трансляции адресов, обработка меток.
15. Дизассемблер и отладчик для стекового калькулятора, их реализация и применение.
16. Объект «Очередь». Виды очередей. Очередь как обобщение стека. Основные характеристики как абстрактной структуры данных и набор методов объекта. Классификация методов. Реализация методов объекта «Очередь». Особенности реализации специфичных методов. Сравнительная характеристика эффективности различных способов реализации методов добавления и удаления элементов очереди.
17. Реализация очереди при помощи кольцевого буфера. Итераторы очередей. Понятие нормализации индекса. Примеры использования итераторов очередей. Примеры использования очереди в различных задачах. Возможности очереди с двумя концами. Понятие о приоритете. Ошибки, связанные с реализацией очереди и его использованием. Примеры возникновения и устранения.
18. Практическая работа «Объект Очередь». Программа «Телеграф Морзе».
19. Объект «Список». Виды списков. Основные характеристики как абстрактной структуры данных и набор методов объекта. Классификация методов. Односвязный список. Схема реализации и эффективность методов вставки и удаления. Двусвязный список. Схема реализации и эффективность методов вставки и удаления.
20. Реализация методов объекта «Список». Особенности реализации специфичных методов. Понятие о списке свободных элементов и методы его реализации. Эффективность использования списка свободных элементов. Примеры использования списков в различных задачах. Сравнение эффективности алгоритмов с использованием списков и без их использования.
21. Практическая работа «Объект Двухнаправленный список». Программа «Спи-

сковый процессор».

- 22.Обобщающий урок по линейным структурам данных. Стратегия реализации контейнерных объектов. Сравнение наборов методов различных контейнерных объектов. Реализация обобщенных методов добавления и удаления элементов.
- 23.Нелинейные структуры данных. Особенности строения и реализации. Реализация нелинейных структур данных через линейные.
- 24.Хеш-таблица, ее свойства. Хеш-таблица как контейнерный объект. Идея хеш-поиска. Эффективность хеш-поиска. Хеш-функции и их характеристики. Требования, предъявляемые к хеш-функциям. Распределение значений хеш-функций. Стохастический характер работы хеш-таблицы.
- 25.Реализация хеш-поиска. Реализация хеш-таблицы при помощи массива списков. Набор методов объекта «хеш-таблица». Классификация методов. Особенности реализации специфических методов. Практическая работа «Объект Хеш-таблица».
- 26.Использование хеш-таблиц для представления множеств. Словари и таблицы имен как примеры множеств. Реализация программы русско-английского и англо-русского словаря с помощью хеш-таблицы. Множественность хеш-таблиц для словарей. Таблицы имен идентификаторов. Пример использования таблицы имен в программе калькулятора.
- 27.Практические работы «Англо-русский и русско-английский словарь» и «Стековый калькулятор с таблицей имен».
- 28.Понятие дерева. Дерево как вид общего графа. Элементы графов, виды графов, понятия пути, достижимости, связности, корня, цикла. Дерево как связный ациклический граф. Понятия узла, арности узла и дерева. Понятие уровня узла и дерева.
- 29.Хранение данных и поиск на деревьях. Применение деревьев для задачи поиска в словаре. Особенности словарного дерева. Вырожденные случаи деревьев, сбалансированные и несбалансированные деревья. Время поиска с использованием деревьев. Вставка и удаление элементов дерева.
- 30.Бинарные деревья. Особенности бинарных деревьев. Поиск на бинарном дереве. Построение бинарных деревьев. Перечисление элементов дерева. Прохождение деревьев и его виды. Рекурсивный характер алгоритмов прохождения деревьев. Вставка и удаление элементов бинарного дерева. Перегруппировка поддеревьев при вставке и удалении элементов.
- 31.Балансировка деревьев. Необходимость балансировки для поддержания оптимальности поиска. Алгоритм балансировки бинарного дерева. Особенности алгоритма балансировки. Деревья с быстрой балансировкой (2-3, AVL- и красно-черные деревья).
- 32.Практическая работа «Объект двоичное дерево (экспертная система-вопросник на двоичных деревьях)».

33. Использование двоичных деревьев для кодирования информации. Код Хаффмана. Понятие об избыточности информации и ее сжатии. Характеристики сжатой информации и алгоритма сжатия. Схема работы архиватора. Алгоритм работы архиватора, работающего при помощи метода Хаффмана. Статический и динамический варианты метода.
34. Пирамиды. Пирамиды как бинарные полностью сбалансированные деревья. Работа с пирамидой. Сортировка при помощи пирамиды. Другие методы сортировки. Расчет эффективности методов сортировки. Практическая работа «Объект Пирамида».
35. «Классические» алгоритмы сортировки и их особенности. Сравнение с методами сортировки на деревьях. Практическая работа «Сравнение и анализ различных видов сортировки».
36. Общие графы. Представление графов. Понятие конечного автомата (КА). Детерминированные и недетерминированные КА. Преобразование недетерминированного КА в детерминированный. КА и регулярные выражения. Задача об удалении комментариев. Лексический анализ, КА для основных типов лексем. Практическая работа «Лексический анализатор».
37. Классические алгоритмы на графах (обзор задач и основных алгоритмов).

Часть 2. Функциональное и логическое программирование

38. Концепция функционального программирования. Язык Лисп. Основные конструкции языка. Префиксная нотация. Операции со списками.
39. Определение функции в Лиспе. Аргументы функции. Вычисление значений функции.
40. Программирование повторений в Лиспе. Рекурсия. Примеры программирования на Лиспе.
41. Концепция логического программирования. Система предикатов первого порядка. Метод резолюций Робинсона. Язык Пролог.
42. Структура программы в языке Пролог. Факты и правила. Использование переменных. Примеры простейших программ. Рекуррентный характер программ на Прологе.
43. Работа со списками в Прологе. Примеры программ. Динамические предикаты. Задача о поиске пути в графе.
44. Работа с нелинейными структурами данных в Прологе. Обход дерева. Задача о восьми ферзях.
45. Применение Пролога для создания баз данных и знаний. Понятие о задачах искусственного интеллекта.

Часть 3. Элементы машинной трансляции

46. Структурные различия языков высокого и низкого уровней. Задача синтаксического разбора. Понятие языка и грамматики. Контекстно-свободные (КС) и контекстно-зависимые грамматики. Общий алгоритм разбора КС-языка.
47. Операторные грамматики. Метод простого предшествования. Реализация метода простого предшествования. Метод приоритетов. Трансляция арифметических выражений.
48. Метод рекурсивного спуска. Схема алгоритма. Трансляция арифметических выражений методом рекурсивного спуска. Обработка ошибок в ходе трансляции. Реализация алгоритма рекурсивного спуска.
49. Средства автоматической генерации синтаксических анализаторов. Программы LEXX и YACC (Bison). Синтаксис входных файлов и порождаемый исходный текст. Объектно-ориентированная оболочка для исходного текста, порождаемого генераторами анализаторов.
50. Практическая работа «Проектирование и реализация транслятора с языка высокого уровня для безадресной машины».
51. Обсуждение систем с одно-, двух- и трехадресным кодом. Регистровые вычислительные системы. Проблема распределения регистров. Обсуждение проблем реализации и оптимизации программ.
52. Принципы дизассемблирования программ. Простейший дизассемблер для процессоров рассмотренных типов.
53. Задача о символьном дифференцировании. Преобразование деревьев выражений при символьном дифференцировании.
54. Практическая работа «Символьное дифференцирование выражений».
55. Задача об упрощении арифметического выражения. Простейшие (тривиальные) способы упрощения выражений.
56. Практическая работа «Тривиальное упрощение выражений».
57. Обсуждение проблем эмуляции систем функционального и логического программирования (Лиспа и Пролога). Обсуждение возможности реализации подмножеств интерпретаторов функционального и логического языков.

Список рекомендованной учебной и методической литературы

Учебная литература (для учеников)

1. Брудно А.Л., Каплан Л.И. Московские олимпиады по программированию. М., Наука, 1990.
2. Керниган Б., Ритчи Д. Язык программирования C++. М., Финансы и статистика, 1980.
3. Малпас Дж. Реляционный язык Пролог и его применение. М., Мир. 1982.

4. Першиков В.И., Савинков В.М. Толковый словарь по информатике. М., Финансы и статистика, 1995.
5. С. Прага. Язык программирования Си. Киев, ДиаСофт, 2001.
6. С. Прага. Язык программирования С++. Киев, ДиаСофт, 2001.
7. Уэйт, Мартин, Прага. Язык Си для начинающих. М., Мир, 1988.
8. Страуструп Б. Язык программирования Си++. М., 2000.
9. Шень А.Х. Алгоритмика (учебник для математических классов). Части 1 и 2. М., МЦНМО, 1995.

Консультативная и методическая литература (для учителя)

10. Feldman, Michael B. Data structures with Ada. – PRENTICE-HALL, INC., Englewood Cliffs, New Jersey 07632.
11. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. М., Наука, 1972.
12. Бар Р. Язык Ада в проектировании систем. Пер. с англ. М., Мир, 1988.
13. Бейбер Р.Л. Программное обеспечение без ошибок. Пер. с англ. М., Джон Уайли энд Санз, Радио и связь, 1996.
14. Братко И. Программирование на языке Пролог для искусственного интеллекта. Пер. с англ. М., Мир, 1990.
15. Брой М. Информатика. Вычислительные структуры и машинно-ориентированное программирование. В 4-х ч. Ч. 2. Пер. с нем. М., “ДИАЛОГ-МИФИ”, 1998.
16. Брой М. Информатика. Структуры систем и системное программирование. В 4-х ч. Ч. 3. Пер. с нем. М., “ДИАЛОГ-МИФИ”, 1996.
17. Брой М. Информатика. Теоретическая информатика, алгоритмы и структуры данных, логическое программирование, объектная ориентация: В 4-х ч. Ч. 4. Пер. с нем. М., “ДИАЛОГ-МИФИ”, 1998.
18. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. 2-е изд. Пер. с англ. М., “Издательство Бином”, СПб.: “Невский диалект”, 1999.
19. Грис Д. Конструирование компиляторов для цифровых вычислительных машин. М., Радио и связь, 1982.
20. Кнут Д. Искусство программирования для ЭВМ. Тома 1-3. Пер. с англ. М., Мир, 1978.
21. Костин А.Е., Шаньгин В.Ф. Организация и обработка структур данных в вычислительных системах: Учеб. пособ. для вузов. М., Высшая школа, 1987.
22. Липаев В.В. Надежность программных средств. Серия “Информатизация России на пороге XXI века”. М., СИНТЕГ, 1998.
23. Лэнгсам Й., Огенстайн М., Тененбаум А. Структуры данных для персональных ЭВМ. Пер. с англ. М., Мир, 1989.
24. Мика Б. и др. Практическое руководство по программированию: Пер. с англ. М., Радио и связь, 1986.
25. Нильсон Н. Искусственный интеллект. Методы поиска решений: Пер. с англ. М., “Мир”, 1973.
26. Паронджанов В.Д. Как улучшить работу ума (Новые средства для образного представления знаний, развития интеллекта и взаимопонимания) – М.: Радио и связь, 1998.
27. Руднев В.Е., Володин В.В., Лучанский К.М., Петров В.Б. Формирование технических объектов на основе системного анализа. М., Машиностроение, 1991.
28. Советов Б.Я. Информационная технология: Учеб. для вузов. М., Высшая школа, 1994.
29. Советов Б.Я., Яковлев С.А. Моделирование систем: Учеб. для вузов. М., Высшая школа, 1998.
30. Хантер Дж. Трансляция с языков высокого уровня. М., Радио и связь, 1982.
31. Холстед М.Х. Начала науки о программах. Пер. с англ. М., Финансы и статистика, 1981.
32. Шилдт Г. Теория и практика С++. Пер. с англ. – СПб.: ВHV, 1996.

Примерный список задач-проектов к теме «Элементы машинной трансляции»

Словообразование

1. *[Комбинаторные слова.]* Дано множество A_+ (массив неповторяющихся) букв и длина слова (от...до). Получите все возможные слова как последовательности букв заданной длины.
2. *[Расширенные комбинаторные слова.]* Задача 1, но допускаются также любые другие буквы (буквы из A_+ должны быть обязательно).
3. *[Список слов.]* Дан длинный текст в виде файла, уберите в нем все, кроме слов. Слова запишите во второй файл, каждое слово на своей строке.
4. *[Получить словарь.]* Отсортируйте файл из задачи 3 и выкиньте дублирующиеся слова.
5. *[С контролем правильности.]* Задача 2, но необходимо искать образующиеся слова в словаре из задачи 4.
5. *[Поиск заданных.]* Найдите в словаре все слова, содержащие все буквы из A_+ .
6. *[С количеством слогов.]* Задача 5 + составьте словарь слогов и подсчитывайте количество слогов в образующемся слове.
7. *[Встречаемость букв, или модель порядка 0.]* Определите частоты (вероятности) F_0 появления букв (пробел считается буквой) в тексте из задачи 3 как
$$F_0("A") = (\text{количество букв "A" в тексте}) / (\text{длина текста}).$$
8. *[Слова модели порядка 0.]* Задача 2, но буквы выбираются согласно частотам, измеренным в задаче 7: сначала более вероятные, затем менее. Подсчитывается средняя вероятность (для слова длиной n — корень n -й степени из произведения всех частот).
9. *[Модель порядка 1.]* Анализом текста из задачи 3 определите частоты появления двухбуквенных сочетаний, например:
$$F_1("AB") = (\text{количество сочетаний "AB" в тексте}) / (\text{количество букв "A" в тексте}).$$
Это характеризует, насколько велика вероятность появления "Б" при условии, что перед ней уже стоит "А".
8. *[Слова модели порядка 1.]* Задача 2, но буквы выбираются согласно частотам, измеренным в задаче 9, при этом для подбора следующей буквы используются частоты, связанные с буквой, поставленной на предыдущем шаге.

9. [*Модели порядков 2-5.*] Задача 8, но длины последовательностей — от 3 до 6 букв. Здесь будут большие массивы данных, надо думать, как хранить.
10. [*Слова порядков 2-5.*] Задача 8, но вероятности взяты из задачи 9.
11. [*Лексические ассоциации.*] Задача 10, и дан словарь слов и фрагментов слов с определенными значениями (признаками). Нужно искать в этом словаре и оставлять слова только если в словаре найдены необходимые признаки.
12. [*Фонетика слова.*] Дано слово (с ударением), надо получить его фонетическую транскрипцию (первая часть фонетического разбора).
13. [*Фонетика текста.*] Задача 12, но дан текст.
14. [*Запрещенная фонетика.*] Дан словарь запрещенных слов, составьте по нему фонетический словарь.
15. [*Контроль фонетики.*] Задача 13 + проверьте фонетику по словарю из задачи 14 без учета разделения на слова (все слова слитно).
16. [*Фонетические ассоциации.*] Задача 11, но словарь фонетический.

Эмуляторы одноадресных процессоров

1. [*Безадресный интерпретатор.*] Реализуйте эмулятор процессора с системой команд: PUSH <число> (затолкнуть в стек), ADD, SUB, MUL, DIV (сложить, вычесть, умножить, разделить 2 последних числа в стеке), SQRT (квадратный корень из числа на вершине стека), POP (вытолкнуть из стека), DUP (продублировать число на вершине стека), IN (ввести число), OUT (вывести) и STOP. Вычисления ведите в интерактивном режиме: ввел строку с командой—разобрал—выполнил. Должны быть команды отладки (распечатать весь стек и т.п.). Для проверки вычислите корень какого-нибудь квадратного уравнения.
2. [*Безадресный код.*] Задача 1, но имеется программа с кодами операций (соответствующим командам), хранящаяся в массиве. Подумайте, как быть с PUSH (у нее есть аргумент). Должно быть чтение программы из файла с определенной структурой (подписи и т.п.).
3. [*Безадресный отладчик.*] Задача 2, но сделайте отладчик (просмотр стека, пошаговое выполнение).
4. [*Безадресный транслятор.*] Напишите транслятор с ассемблера для процессора из задачи 2.
5. [*С константами.*] Задача 3, добавляются команды IN <текст> (ввести число с приглашением) и OUT <текст> (вывести с пояснением). Подумайте, как хранить строки в программе. Соответственно модифицируйте транслятор.
6. [*С переходами.*] Задача 5, добавляются команды JMP <адрес> (переход на команду с определенным адресом) и JB (переход, если число на вершине стека меньше нуля). Модифицируйте транслятор, введите понятие метки,

задаваемой идентификатором. Используйте двухпроходную схему: 1 проход — коды операций, 2 проход — корректировка адресов переходов.

7. [*С подпрограммами.*] Задача 6, добавляются команды CALL <адрес> (вызов процедуры с определенным адресом) и RET (возврат). Реализуйте стек вызовов (отдельно от стека вычислений. Подумайте, чем выгодно и невыгодно смешение стеков вызовов и вычислений).
8. [*С памятью.*] Задача 7, но вводится память из N ячеек и команды PUSH <адрес> и POP <адрес> (операции с ячейками памяти). В трансляторе пока не делайте описания данных (пусть они пока задаются через PUSH и POP).
9. [*С данными в программе.*] Задача 8, но в трансляторе сделайте возможность задания переменных с именами и начальными значениями.
10. [*С полем r/m/i и косвенной адресацией.*] Задача 8 или 9, но сделайте так, чтобы аргументы команд PUSH, POP, переходов и вызовов можно было задавать в 3 формах: immediate (аргумент указан в команде, как в задаче 1), memory (аргумент в памяти — задача 7) и register (аргумент на вершине стека). Введите байт формата аргумента и байт длины команды (более правильно упаковать это в 1 байт). Учтите, что режимы могут комбинироваться!
11. [*С графикой.*] Задача 9 или 10 + добавьте графические команды.
12. [*PostScript.*] Реализуйте подмножество языка описания графических изображений Adobe PostScript. (Только не надо пытаться что-либо понять, смотря в PS-файлы! Они обычно генерируются графическими редакторами, там черт ногу сломит. Возьмите лучше файл с книжкой по PostScript, она простая.)

Эмуляторы регистровых процессоров

13. [*Регистровый интерпретатор.*] Реализуйте эмулятор процессора с N регистрами (N от 4 до 8, но можно и принципиально большое число) и системой команд пересылки данных MOV <регистр> <регистр или число> (пересылает справа налево) и арифметики ADD, SUB, MUL, DIV (сложить, вычесть, умножить, разделить два регистра и результат записать в первый из них; действия будут затирать один из операндов), SQRT (квадратный корень из регистра), IN (ввести число), OUT (вывести) и STOP. См. замечания к задаче 1. Для проверки вычислите корень какого-нибудь квадратного уравнения. Вы помучаетесь, но многие говорят, что регистровые процессоры быстрее стековых.
14. [*Все круче и круче.*] Выполняются задачи 2-11 с учетом задачи 13. Когда будете писать программы для своего процессора, заметьте, как тяжело бывает иногда с регистрами и как часто хочется побольше дополнительных переменных.
15. [*Оптимизация.*] Подумайте, как можно оптимизировать работу с регистрами при вычислении выражений и выполнении типовых операций (циклы,

передача параметров в процедуры и т.п.).

15. *[Трехадресный код.]* Реализуйте такие возможности: пусть в командах можно будет указывать до трех операндов (где находятся данные и куда положить результат). Операнды могут быть регистрами, константами или адресами памяти.
16. *[Трехадресный код с косвенной адресацией.]* Добавьте к задаче 15 косвенную адресацию.
17. *[Без памяти!]* Замените все ячейки памяти большим количеством регистров (эту задачу можно делать вместо переделки задачи 8).
18. *[Минимальная система команд.]* Подумайте, какова могла бы быть минимальная система команд для Вашего процессора.
19. *[Высокий уровень.]* Вам не надоело еще писать на ассемблере? Подумайте, как можно транслировать язык высокого уровня (какое-нибудь его сильно ограниченное подмножество) для Вашего процессора.

Эмуляторы всяческих процессоров

20. *[Двадцать лет спустя.]* ...было такое семейство вычислительных машин PDP. Реализуйте эмулятор для процессора одной из машин семейства. (Набор команд не очень велик, вполне реальная задача.)
21. *[Не так давно.]* ...некая фирма Integrated Electronics (теперь ее сокращают по 2-3 буквам из каждого слова) выпустила первый процессор под номером 8086. Ну, вы поняли. Только не все команды, а подмножество, иначе замучаетесь. Постарайтесь использовать оригинальные коды команд и форматы байтов r/m.
23. *[Классика.]* В классическом (но не законченном) труде Дональда Кнута «Искусство программирования для ЭВМ» (его английское издание, на котором будут клясться все системные программисты, лежит в кабинете 204. Там же иногда лежит и пряник) приведено описание машины MIX, которую Кнут использует для иллюстрации алгоритмов. Вы не будете первый, кто сделает ее эмулятор, но приобщитесь к классике.
25. *[Для тех, кто собирается на ВМК или Физтех.]* Обучение на факультете ВМК МГУ и Физтехе начинается с программирования на машине Тьюринга. Напишите ее эмулятор и Вы будете более готовы к первому коллоквиуму на первом курсе, чем Ваши однокурсники.
26. *[Любителям рекурсий.]* Есть такой язык LISP (от «LISt Processing», иногда расшифровывают как «a Lot of Idiotic Silly Parentheses», будете писать — поймете) для обработки списков. Считался (и отчасти сейчас считается) классическим языком искусственного интеллекта. Реализуйте эмулятор LISP-процессора и вы забудете цикл «for» навсегда! (Может быть, этот процессор легче сделать, чем написать некоторые программы для него.)

Программа курса
«Программирование в ОС Windows»
для 11 мат. классов
(2 часа в неделю, всего 64 часа)

Пояснительная записка

Курс рассчитан на учащихся 10 классов физико-математического профиля и основывается на курсе «Технология ООП», изучаемым в 9 классе.

Курс посвящен событийному программированию для ОС Windows с помощью Win32 API и библиотеки Microsoft Foundation Classes (MFC). Также в курсе рассматриваются современные технологии программирования (COM, OLE, ActiveX, автоматизация и др.).

Данный курс тесно связан с курсами «Технология ООП», «Введение в язык ассемблера», «Технология проектирования», «Алгоритмы и численные методы».

Формы занятий — лекция, семинар и практическая работа. Формы контроля — сдача практических работ (основная), контрольные работы.

В качестве инструментария используется среда Microsoft Visual C++.

Закрепление материала обеспечивается использованием полученных знаний и навыков при выполнении учащимися 11 классов выпускных работ (проектов).

Тематическое планирование

1. Программирование Win32 API. Событийное программирование. Окно и функция. Обработка сообщений. Примеры.
2. Win32 API. Пользовательский интерфейс. Окна.
3. Регистрация класса окна. Создание окна. Стили окна. Ресурсы приложения. Меню. Элементы управления. Диалоги. Примеры.
4. Программирование Win32 API. Управление графическим выводом. Контекст устройства. Графические примитивы. Цвет и цветовые палитры. Битовые изображения. Шрифты. Примеры.
5. Клавиатура. Манипулятор «мышь». Таймер. Примеры.
6. Многооконный интерфейс. Справочная система. Примеры.
7. Работа с памятью в Win32. Примеры.

8. Межпроцессное взаимодействие. Примеры.
9. Microsoft Foundation Classes (MFC). Общая структура каркаса приложений. Однооконное и многооконное приложение, базовые классы MFC, обработка сообщений. Модель документ-отображение (документ-вид), понятие “документ” и шаблон документа, однооконный и многооконный интерфейс. Меню. Примеры.
10. MFC. Элементы управления (кнопки, статический орган управления, полоса просмотра, редактор текста, списки), обработка командных сообщений. Диалоговые панели. Стандартные диалоговые панели. Сериализация данных. Примеры.
11. MFC. Работа с графикой, контекст устройства, графические примитивы. Обработка исключений. Межпроцессное взаимодействие. Примеры.
12. Избранные главы программирования под Windows. Библиотеки динамической компоновки. Локализация приложений. Консольные приложения. Особенности работы в Windows NT. Модель COM. ActiveX. OLE. Автоматизация. Примеры.

Список рекомендованной учебной и методической литературы

Учебная литература (для учеников)

1. Баженова И.Ю. Visual C++ 6.0 (Visual Studio 98). Уроки программирования. М., “ДИАЛОГ-МИФИ”, 1999.
2. Фролов А.В., Фролов Г.В. Графический интерфейс GDI в Microsoft Windows. Серия “Библиотека системного программиста”. Том 14. М., ДИАЛОГ-МИФИ, 1994.
3. Фролов А.В., Фролов Г.В. Операционная система Microsoft Windows 3.1 для программиста. Серия “Библиотека системного программиста”. Тома 11-12-13, части 1-2-3. М., ДИАЛОГ-МИФИ, 1994.
4. Фролов А.В., Фролов Г.В. Операционная система Windows 95 для программиста. Серия “Библиотека системного программиста”. Том 22. М., ДИАЛОГ-МИФИ, 1996.
5. Фролов А.В., Фролов Г.В. Программирование для Windows NT. Серия “Библиотека системного программиста”. Тома 26-27, части 1-2. М., ДИАЛОГ-МИФИ, 1997.
6. Холзнер С. Visual C++ 6. Учебный курс. СПб, Питер 2001.

Консультативная и методическая литература (для учителя)

7. Барфилд Э., Уолтерс Б. Программирование “клиент-сервер” в локальных сетях. Пер. с англ. М., Информационно-издательский дом “Филинь”, 1997.
8. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 2-е изд. Пер. с англ. М., “Издательство Бином”, СПб.: “Невский диалект”, 1999.
9. Крокет Ф. MFC. Мастерская разработчика. Пер. с англ. М., Издательский отдел “Русская редакция” ТОО “Channel Trading Ltd.”, 1998.
10. Круглински Д. Основы Visual C++, версия 4. Пер. с англ. М., Издательский отдел “Русская редакция” ТОО “Channel Trading Ltd.”, 1997.
11. Мюррей У., Паппас К. Создание переносимых приложений для Windows. Пер. с англ. – СПб.: ВHV – Санкт-Петербург, 1997.

12. Проектирование пользовательского интерфейса на персональных компьютерах. Стандарт фирмы IBM. – Вильнюс: DBS Ltd., 1992.
13. Рихтер Дж. Windows для профессионалов. Программирование в Win32 API для Windows NT 3.51 и Windows 95. Пер. с англ. М., “Русская редакция”, 1995.
14. Филлипс Д., Гарсиа-Диас А. Методы анализа сетей. Пер. с англ. М., Мир, 1984.

Программа курса

«Технология обучения и проектирования»

для 8-9 мат. классов

(8 класс – 1 час в неделю, всего 32 часа,
9 класс – 2 часа в неделю, всего 64 часа)

Пояснительная записка

Курс рассчитан на учащихся 9 классов физико-математического профиля и основывается на курсе «Алгоритмика» 8 класса. Основная цель курса — изучение и отработка технологии проектной работы, организация мышления ученика, развитие системного подхода, методологии анализа и синтеза решений, планирования, реализации.

Курс имеет ярко выраженную практическую направленность, в связи с тем, что преимущественно теоретическое изучение принципов проектирования носит достаточно абстрактный характер и плохо воспринимается учащимися 9 классов. Обучение ведется в тесной связи с профильными курсами «Технология ООП» и «Алгоритмы и структуры данных I» на конкретных примерах и проектах, их сопоставлении, в ходе которого анализируются находки и ошибки. Общие принципы выводятся как некоторое следствие из этих частных результатов. Для выполнения учащимися выбираются задания из других профильных курсов, изучаемых параллельно (Технология ООП, Алгоритмы и структуры данных I).

К сожалению, учащиеся зачастую не владеют приемами системного мышления (или лишь знают об их существовании), что существенно затрудняет любую серьезную проектную деятельность. В курсе предусмотрен практикум по развитию психологических и когнитивных навыков системного мышления, без которых разговор о проектной деятельности будет пустым звуком.

В первой части курса отрабатываются отдельные элементы технологии: путь от задачи к решению, технологии постановки и формализации задачи, синтеза и анализа решений, идеологии реализации; технология планирования деятельности; технологии обсуждения и презентации проекта; технологии групповой работы. Во второй части эти элементы закрепляются в виде выполнения развернутого проекта на одну из тем курса «Алгоритмы и структуры данных I». В течение всего курса проходят рабочие семинары по проектной деятельности учащихся, на которых они демонстрируют практическое применение изученных технологий для работы над выбранным проектом. В конце года предусмотрен

ряд итоговых семинаров для проведения защит курсовых проектов.

В материале курса не предусмотрено каких-либо самостоятельных задач, кроме некоторых учебных тренингов. Вместо этого рассматриваются примеры из профильных и смежных непрофильных дисциплин, что повышает актуальность курса и мотивацию учащихся.

Курс излагается с опорой на развитие самостоятельного мышления учащихся, самообразование. Формы занятий — лекция, семинар и практическая работа. Формы контроля — сдача практических работ (основная), письменных контрольных работ, проверка конспектов.

В качестве инструментария используются MS Word, MS Excel, MS PowerPoint, MS FrontPage.

Результатом прохождения курса учащимися должно стать практическое овладение принципами технологии проектной деятельности.

Тематическое планирование

1. Задачи курса. Понятие проекта. Виды программных проектов, их назначение. Участники процесса проектирования. Критерии оценки проекта. Этапы работы над проектом. Анализ проекта «Арконоид» из курса технологии ООП.
2. Путь от задачи к решению. Примеры технологичного и нетехнологичного решения задач. Анализ ошибок в проектах. Универсальность принципов технологии решения. Технологии системного подхода, мыслительные операции (МО).
3. Отработка элементов системного мышления. Объекты и их свойства. Объекты вообще и объектно-ориентированная технология программирования (обсуждение проекта «Arcanoid++»).
4. Технология построения определений. Мыслительные операции (МО) «представления», «размножения», «описания» и «узнавания». Формализация сходства и различия объектов. Связь с ООП (по материалам курса ООП). Замещение одного объекта другим (тренировка гибкости мышления). Задания на развитие воображения.
5. Понятие классификации. Построение классификаций. МО «разделения» и «объединения». Критерии основания для классификации. Множественность классификаций. Связь классификаций и концепций ООП (по материалам курса «Технология ООП»).
6. Формализация взаимодействия объектов. Связи между объектами. Понятие системы и системного подхода, контекста. МО «осмотрись» и «всмотрись». Обсуждение менеджера объектов в проекте «Arcanoid++».
7. Путь от задачи к решению как система. Технологии постановки и формализации задачи. Анализ условий задач. Корректность постановки задачи. Устойчивость условий. Работа с экспертом (заказчиком).

8. Синтез и анализ решений. Методы синтеза решений. Использование мыслительных операций на этапе синтеза. Анализ и отбор решений, технология «фантазер-критик-реалист-технолог». Понятие об алгоритме решения изобретательских задач.
9. Реализация идей. Противоречивость реализации. Компромиссный характер реализации. Идеологии реализации. Обсуждение реализации строковой библиотеки (по материалам курса «Технология ООП»).
10. Технология планирования деятельности. Постановка проблемы. Анализ фаз деятельности. Стадии развития профессионального проекта. Понятие сетевого графика.
11. Поисковая стадия проекта. Восприятие информации и ее виды. Информационные поля и потоки. Структура информационного поля. Ключевые слова и их иерархия. Понятие семантической сети. МО «рассредоточения» и «сосредоточения». Семантическая сеть курса «Алгоритмы и структуры данных I».
12. Технология поиска информации. Тренировка ситуаций с необходимостью быстро усвоить и воспроизвести информацию. Практическая работа «Ответ с ходу». Обработка больших объемов информации. Технология написания рефератов.
13. Стадия эскизного проекта. Понятие модели проекта. Роль моделирования в создании технологий. Масштабируемость эскизного проекта. Обсуждение проектов по курсу «Алгоритмы и структуры данных I».
14. Стадия рабочего проекта. Требования к рабочему проекту. Технология тестирования программ. Работа с версиями. Обсуждение проектов по курсу «Алгоритмы и структуры данных I».
15. Особенности программных проектов. Назначение программного продукта. Понятие о «жанре» программы и ее «экологической нише». Взаимодействие с пользователем. Характеристики групп пользователей.
16. Технологии обсуждения и презентации проекта. Технология построения тематического сообщения. Отработка связного рассказа по теме.
17. Графическое сопровождение доклада. Принципы читабельности. Синхронизация графического сопровождения. Дизайн графического сопровождения.
18. Жанры доклада. Понятие о целевой аудитории. Опорный конспект доклада. Презентация доклада. Фазы доклада. Работа с оппонентами. Виды оппонирования.
19. Технологии групповой работы. Взаимодействие в группе. Разделение обязанностей. Синхронизация индивидуальных графиков. Устойчивость группового графика.
20. Накопление опыта и профессиональный рост. Виды опыта. Способы получения опыта (изучение, анализ, экспериментирование). Взаимоотношение

способов. Понятие о метатехнологии.

21. Итоговые семинары по защите курсовых проектов.

Список рекомендованной учебной и методической литературы

Учебная литература (для учеников)

1. Зенкин А.А. Когнитивная компьютерная графика. М.: Наука, 1991.
2. Першиков В.И., Савинков В.М. Толковый словарь по информатике. М., Финансы и статистика, 1995.

Консультативная и методическая литература (для учителя)

3. Альтшуллер Г. С. Творчество как точная наука. М., 1990.
4. Альтшуллер Г. С. Алгоритм изобретения. М., 1974.
5. Альтшуллер Г. С., Верткин И.М. Жизненная стратегия творческой личности. Минск, 1994.
6. Паронджанов В.Д. Как улучшить работу ума. (Новые средства для образного представления знаний, развития интеллекта и взаимопонимания) – М.: Радио и связь, 1998.
7. Тихомирова Л.Ф., Басов А.В.. Развитие логического мышления детей. М., 1995.

Программа факультативного курса

«Введение в язык ассемблера»

для 9–10 мат. классов

(2 часа в неделю, всего 64 часа)

Пояснительная записка

Курс рассчитан на учащихся 9 классов физико-математического профиля и основывается на курсах «Алгоритмика» и «Технология ООП» 8 и 9 класса. Цель курса — изучение технологии оптимизации программ с использованием языка ассемблера, получение навыков низкоуровневого программирования.

Поскольку курс читается после курса «Технология ООП» и вторая часть его проходит параллельно разделу «Нечисленные алгоритмы» курса «Алгоритмы и структуры данных I», особое внимание учащихся обращается на модульность программ и их взаимодействие с программами, разрабатываемыми с применением языка Си++.

Данный курс, в сочетании со специальными курсами специальности «Технологии программирования», позволит учащимся составить и закрепить на практике представление о современном состоянии технологии программирования.

Основная форма занятий — совмещенная лекционно-практическая, подведение итогов проводится на основании оценок за практические работы. В качестве инструментария используется Turbo Assembler 4.5 в среде Borland IDE.

Материал курса используется в последующих спецкурсах 10 класса: «Алгоритмы и структуры данных» и «Программирование в ОС Windows». Закрепление технологических приемов происходит при подготовке курсовой и выпускной работ.

Тематическое планирование

1. Первоначальные сведения об ассемблере. Понятие о языках высокого и низкого уровней. Особенности языков низкого уровня. Область применения ассемблера. Использование ресурсов компьютера с максимальной эффективностью.
2. Строение процессора. Работа процессора. Исполнение программы процессором. Понятие о системе команд процессора.

3. Понятие оперативной памяти и адресации к ней. Особенности адресации. Единицы измерения оперативной памяти.
4. Пример создания простейшей программы на ассемблере. Общие сведения о процессорах x86: адресация; область портов ввода/вывода; работа с памятью; адресная шина и шина данных; регистры процессора; флаговый регистр.
5. Регистры процессора. Типы команд: команды передачи данных; арифметические команды; побитовые команды. Команды ввода-вывода. Использование логических операций для установки и сброса битов. Команды косвенной адресации. Работа с сегментом данных BIOS и видеопамятью. Циклы с участием регистра CX. Команды безусловного перехода.
6. Виды адресов. Понятие о сегменте и смещении. Линейный адрес. Переменные в языке ассемблера. Команды пересылки данных. Виды адресации: регистровая; прямая; непосредственная; косвенная; базовая; индексная; базово-индексная.
7. Разбор примеров программирования на ассемблере. Работа с отладчиком.
8. Концепции программирования на ассемблере: требования языка; ассемблирование и выполнение программы; определение данных. Модели памяти; процедуры; макросы. Форматы команд.
9. Регистр SP и понятие стека. Сохранение и восстановление данных в стеке. Регистр BP. Интерфейс с языками высокого уровня.
10. Процедуры в ассемблере и работа с ними. Работа стека при использовании процедур. Программные прерывания.
11. Регистр флагов. Команды сравнения и условного перехода. Особенности использования регистра флагов.
12. Регистры SI и DI и строковые команды. Использование строковых команд для пересылки и сравнения данных.
13. Прерывания: обслуживание прерываний: механизм прерываний; таблица векторов прерываний; изменение таблицы векторов прерываний; особенности обработки аппаратных прерываний. Прерывания BIOS: клавиатура; манипулятор “мышь”; часы реального времени; системный таймер. Выполнение практических заданий.
14. Прерывания DOS: работа с файлами; управление программами; управление памятью; другие функции. Обработка ошибок: обработчик критических ошибок. Выполнение практических заданий.
15. Резидентные программы. Особенности взаимодействия с DOS и BIOS.
16. Заключительное занятие.

Список рекомендованной учебной и методической литературы

Учебная литература (для учеников)

1. Абель П. Язык ассемблера для IBM PC и программирования. Пер. с англ. М., Высшая школа, 1992.
2. Джордейн Р. Справочник программиста персональных компьютеров типа IBM PC, XT и AT. Пер. с англ. М., Финансы и статистика, 1992.
3. Скэнлон Л. Персональные ЭВМ IBM PC и XT. Программирование на языке ассемблера. Пер. с англ. М., Радио и связь, 1991.

Консультативная и методическая литература (для учителя)

4. Аммерал Л. Машинная графика на персональных компьютерах. Пер. с англ. М., “Сол Систем”, 1992.
5. Аммерал Л. Принципы программирования в машинной графике: Пер. с англ. М., “Сол Систем”, 1992.
6. Использование Turbo Assembler при разработке программ. – К.: “Диалектика”, 1994.
7. Мика Б. и др. Практическое руководство по программированию: Пер. с англ. М., Радио и связь, 1986.
8. Фролов А.В., Фролов Г.В. MS-DOS для программиста. Серия “Библиотека системного программиста”. Том 18-19, части 1-2. М., ДИАЛОГ-МИФИ, 1995.
9. Фролов А.В., Фролов Г.В. Аппаратное обеспечение IBM PC. Серия “Библиотека системного программиста”. Тома 1-2, книги 1-2-3. М., ДИАЛОГ-МИФИ, 1992.
10. Штернберг Л.Ф. Разработка и отладка программ. М., Радио и связь, 1984.

Приложение 1. Информатика — Q&A (некоторые мнения)

Q: Что такое информатика и что под этим обычно понимают в школе?

А: Информатика в научном смысле — наука о наиболее общих законах, связанных с информацией, фактически, раздел математики. В школе под ней понимают обычно три вещи: (1) Информационные технологии, или пользовательский курс — умение работать с готовыми программами: сверстать и распечатать объявление, открытку, реферат, сборник песен, книгу, составить базу данных аудиозаписей, своих оценок, вести электронную переписку, настроить соединение в Интернете и т.д. и т.п. Это становится необходимым, как пользование телефоном или метро. (2) Некие общетеоретические сведения из компьютерно-ориентированных разделов математики, такие как системы счисления и логика. (3) Основы программирования на каком-либо распространенном языке, Си, Паскале или Бейсике.

Q: Есть ли какая-либо единая программа по информатике?

А: В отличие от уже сложившихся дисциплин (математика, физика, русский язык и т.д.), в школьной информатике сегодня существуют несколько подходов к образованию. Для многих тем и целых разделов существует множество способов их изучать. Вряд ли сейчас можно сказать, какой из них истинно наилучший. Существует Московский региональный стандарт (перечень тем, знаний и умений), который обязателен для каждого курса и ученика, его изучающего, он опубликован и доступен для каждого желающего с ним ознакомиться. Он определяет перечень обязательных разделов курса, но при этом допускает значительную гибкость в определении уровня их изучения.

Q: Нужен ли систематический курс информатики в школе?

А: Ответ исходит из необходимости и возможности. В современном мире обходиться без компьютерных умений так же сложно, как 100 лет назад без умения читать или 50 лет назад без телевизора (то есть пока можно, но очень неудобно, и чем дальше — тем больше). От проклятых компьютеров в прямом смысле слова некуда деваться. С другой стороны, в школе наиболее удобно изучать это дело, так как информатика — чрезвычайно творческий предмет. "Не-творчески" изучать его довольно противно. А творческая активность человека с возрастом, как известно, падает. Такие дела.

Q: Что в первую очередь должно быть заложено в курсе информатики?

А: Самое существенное — самостоятельность и творчество, поиск своих решений, активная позиция, саморазвитие, следование не стереотипам, а своему

воображению, своей цели. Бездумное копирование чужих действий, знаний и приемов в информатике бессмысленно и попросту невозможно. Ситуации всегда разные. Поэтому самое главное — это

Знать, где найти информацию ("Знание — когда знаешь, где искать")

Знать, зачем может пригодиться то, что учишь

Знать, как применить то, что умеешь

Знать, что реально сделать, а что нереально

Знать, как сделать что-то лучше (или быстрее, или и то, и другое)

Знать, как можно развить ту тему, которая тебе интересна

Уметь самостоятельно изучить нужную программу

Уметь выбрать для себя лучшую программу из многих

Уметь экономить свои время и силы, зная возможности программы

Уметь сделать самостоятельно то, без чего нельзя, но чего в программе нет

Уметь пользоваться результатами чужого труда

Уметь делать свой труд доступным для других

Уметь довести свой проект до конца, а не бросать, не доведя до «точки с запятой»

Уметь при необходимости сопровождать и развивать проект

По компоновке курса:

Общий пользовательский курс — знать и уметь то, что превращает компьютер в универсальный оргтехнический прибор (печатная машинка/печатный станок/рисовалка/чертежная доска/почтовое отделение/записная книжка и т.д.)

Специальный пользовательский курс — то, что помогает в работе по конкретному профилю (написание сценария/раскадровка/монтаж фильма/создание мультфильма/титры/.../обработка интервью/верстка печатного издания/.../математическое моделирование систем/численный анализ и т.д.)

Специальный курс программирования — то, что позволяет выполнить задачу общего или специального пользовательского курса, но значительно быстрее, чем вручную (в 10 — 100000... раз), с помощью сценариев для компьютера (программ), часто очень и очень простых. Некоторые примеры: типовая обработка длинных текстов (50-1000 стр), упрощение анимации, 2-3 рисунка вместо 200-300 (Macromedia Flash), автоматизация нудных рисовательных (Corel Script) и чертежных (AutoLisp) задач, математических расчетов (Math-Lab) и т.д. Цель курса — переложить наиболее нудную часть работы на компьютер и поднять КПД (коэффициент полезного действия) человека с 1% до 90%.

Общий курс программирования — то, без чего нельзя понять специальный курс программирования.

Общетеоретическая база — набор сведений, без которых нельзя понять мно-

гие другие разделы курса.

Q: Говорят, что информатика — это математика. Так ли это?

А: Для кого как. Есть разделы, которые понять надо и которые есть математика. Есть разделы, где она практически не нужна. Однако без общей логики, без понимания принципов (которые во многом — математика) в компьютерном деле работать можно, но сложновато. Логика (не математическая, а «строе-ние мозгов») позволяет не заучивать наизусть правила работы, а разбираться самостоятельно. Иначе человек становится рабом компьютера, хотя изна-чально было задумано наоборот.

Q: Говорят, что информатика — это программирование. Так ли это?

А: Не обязательно, хотя программирование повышает возможности человека в 10-10000... раз и во столько же раз сокращает время выполнения работы. По-этому часто говорят (и так часто бывает), что трудолюбивый будет 1000 раз нажимать на кнопку, а ленивый придумает, как это сделать с помощью программирования, и потом пойдет пить чай.

Q: Говорят, что информатика — это Интернет. Так ли это?

А: Нет, хотя чем дальше, тем больше между этими словами ставят знак равенст-ва. Это все равно что говорить, что кино — это телевидение. В любом случае технологии интернет — часть информатики, может быть, наиболее модная, но далеко не самая полезная. И всего лишь часть.

Q: Говорят, что информатика — это обязательно компьютер. Так ли это?

А: В практическом смысле (см. выше о пользовательских курсах) — да. В науч-ном смысле — нет, т.к. принципы, по которым в информатике все строится, доступны для изучения и разработки на бумаге, без компьютера. Однако в этом случае мы получаем только математическую сторону, в основном без явного практического применения. В общем, в реальности процесс должен выглядеть так: вначале планы и наброски в голове и на бумаге (обязательно без компьютера, т.к. на этой стадии он мешает), потом, когда все более-менее ясно, выполнение задачи на машине. Кто делает наоборот — теряет много времени, приблизительно 2-3 раза переделывает всю работу заново и получа-ет меньшее конечное качество.

Q: Я не понимаю, зачем проходить те или иные темы в курсе информатики.

А: В этом случае лучше исходить из того, что в школьный курс вряд ли будут пихать то, что ну совсем не нужно. В информатике и так времени на обуче-ние не хватает. Можно спросить об этом учителя (по конкретной теме), разу-меется не в форме "на фига это мне надо?", а в более приличной, и, скорее всего, непосредственная причина обнаружится. Что точно нельзя — это ко-

пить такие вопросы, т.к. из-за этого нарастает нудность и бесполезность обучения.

Q: Я не знаю, зачем мне изучать компьютерные технологии.

А: Абсолютно все то, что "умеет" компьютер, можно сделать без него. Но в большинстве своем затратив в 10-10000... раз больше времени и сил и потеряв в 2-20... раза в качестве. Компьютер при правильном применении — это быстрый, внимательный и неутомимый придаток медленного, забывчивого и быстро утомляющегося человека (по сравнению с машинами люди именно такие). У человека есть только одна черта, которую нельзя скопировать в машину — это творчество, и это в идеале это единственное, что требуется от человека. На самом деле приходится еще и знать, как эти творческие замыслы реализовывать. Это и есть то, что называют технологиями.

Q: Я уже умею пользоваться компьютером, мне скучно изучать то, что уже знаю.

А: Вы в этом уверены?

Скорее всего, есть потенциально интересные вам области, про которые вы даже не подозреваете, поэтому о них не возникает даже вопросов. Подумайте, как обратить зло во благо и использовать то, что вы изучаете, для своей пользы. Если вы творческий человек, то вы найдете выход. Если нет — постарайтесь постепенно становиться им, так жить интереснее. Информатика при правильном подходе этому весьма способствует.

Q: Я не знаю, зачем мне изучать программирование.

А: Чтобы не тратить время и силы на однообразную работу, выполняя униженную роль компьютерного придатка, предназначенного для нажатия кнопочек и двигания мышкой. Все операции на машине можно автоматизировать, т.е. написать программу, запустить и пойти пить чай. Некоторые вещи и вправду требуют большого программистского труда, но никто не будет от вас требовать программу в 5000 строк в школьном курсе. Если вам интересно, то вы напишете и 50000, но по собственной инициативе.

Q: Как быть тем, кто не очень хочет изучать информатику?

А: Надо попытаться понять, почему не хочется. Может быть, вы просто не знаете, зачем вам все это. Попробуйте подумать, что вам больше всего нравится и как вы можете для этого применить компьютер. Например, в совершенстве изучить текстовые и графические редакторы можно, просто задавшись целью сделать собственные обложки для CD-дисков или кассет в вашей коллекции. И т.д.

Q: Я слишком ленив, чтобы изучать что-то про компьютеры.

А: Помните, что информатика придумана для ленивых!

Не ленивый и сам делает все, что надо, без всякого компьютера.

Например, перепечатает в 5 экземплярах на пишущей машинке (или перепишет от руки каллиграфическим почерком) 5 сборников любимых песен в 100 страниц каждый, абсолютно внимательно проверит все на ошибки, и собственноручно проиллюстрирует их оригинальными рисунками! Не понимаю, зачем вообще такому человеку компьютер.

Q: Ваше мнение о компьютерных играх?

A: Если честно, то я считаю, что это замечательный способ заставлять людей платить деньги тем, кто эти игры продает. Сейчас уже совершенно точно известно, что компьютерные игры вызывают психическую зависимость. Конечно, некоторую разрядку и видимость отдыха они иногда дают. Но, как говорится, при правильном применении. Вы вряд ли едите еду совсем без соли, но и не кладете в тарелку пару ее килограммов. Поэтому вопрос не в принципе, а в количестве и качестве. Есть полезные игрушки, типа The Incredible Machine, логические игры, но и с ними, если заиграться, можно потерять реальность. Единственно что нужно точно знать — что, если вы много играете, из этого НЕ следует, что вы умеете обращаться с компьютером. Скорее всего вы умеете его включать и запускать игру, и далее все только в ее рамках. "Тренировка владения клавиатурой и мышью" — тоже чушь, т.к. эти умения, требующиеся в игре и в остальной деятельности, сильно отличаются. (Точнее, создатели игр не утруждают себя тренировкой ваших умений — им этого не надо, им надо, чтобы вы покупали и платили.)

Q: Вы нудный и неинтересный человек, мы хотим играть, а вы собираетесь нам это запрещать.

A: Задача учителя — обратить ваше внимание не только на игры, а на массу других вещей, которые могут оказаться, и в большинстве случаев оказываются, интересными и полезными. Играть вы и сами научитесь. Если же идет только игра — дело совершенно точно стоит. Время идет, мозги киснут, см. ответ на 3-й вопрос.

Приложение 2. Журналы результатов

Обозначения на диаграммах														
Оценки и уровни	5	4	3	2		-т	-с	-б	-н					
Примеры	5тр	5сс		4тс	4ср	4бр		3тр	3сс	3бр	3нр		2бр	2нр
<p>Обозначения: Оценка-Уровень-Язык, например: "4ср" - оценка 4, уровень средний, язык pascal "?" - не готов к уроку (нет домашнего задания), "!" - не сдано При выставлении оценок и уровней язык не учитывается и на диаграммах не отображается.</p> <p>На диаграмме в примерах выделены часто встречающиеся комбинации.</p>														
Оценки и уровни														
Примеры														

