

Муниципальное бюджетное образовательное учреждение дополнительного
образования «Центр технического творчества»
городского округа Тольятти

ПРИНЯТО
Педагогическим советом

Протокол № 1
от 29.08.2019 г.

УТВЕРЖДАЮ
Директор МБОУ ДО ЦТТ



С.В. Ситникова

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ
«КЛАССИЧЕСКОЕ СТРУКТУРНОЕ ПРОГРАММИРОВАНИЕ»
(Модульная)**

Возраст обучающихся: 12 – 15 лет
Срок реализации программы: 2 года

Автор-составитель: Батижевская
Светлана Леонидовна

Методист: Леоновец
Ольга Корнеевна

2019 – 2020 уч. год

1. Пояснительная записка

Человечество всем стадиям своего развития дало меткие эпитеты: каменный век; бронзовый век; индустриальный век; космический век.

Наступивший XXI век уже получил свое название – компьютерный, информационный век. Информация, информированность, информатизация, информатика... Все эти термины прочно вошли в словарный запас наших современников.

В наш век скоростей и бешеных ритмов все большее значение приобретают возможность мгновенной связи на больших расстояниях, скоростная обработка получаемой информации, качественный анализ и быстрое принятие решений по насущным вопросам. Все это и многое другое помогают осуществлять компьютерные системы.

Однако растущие потребности человечества требуют постоянного совершенствования технических средств и программного обеспечения. В свою очередь – возрастание сложности и расширение сфер деятельности вычислительной техники требуют, если уж не профессиональной подготовки, то хотя бы достаточной компьютерной грамотности от людей самых разных профессий: бухгалтер и экономист, юрист и секретарь, референт-переводчик и инженер, продавец и товаровед, кладовщик и коммерсант, банковский служащий и библиотекарь – и многих, многих других.

Современная деловая жизнь предъявляет строгие требования к молодому поколению, пополняющему ряды работников всех сфер деятельности. Начинающий работник должен обладать достаточным багажом знаний и умений, чтобы не быть невостребованным на рынке труда, чтобы быть социально защищенным в нашем скоростном, информационно насыщенном мире.

Актуальность

Все вышеуказанное диктует необходимость разработки новых образовательных технологий, среди которых особое место занимает технология программированного обучения.

А. Берг писал: «Научимся программировать – научимся обучать». Это **актуально** как для учителя, ищущего пути оптимизации учебного процесса, так и для ученика, осознавшего важность самообучения, самообразования.

Компьютерные технологии, как считают современные педагоги (Г.К.Селевко, Н.В.Апатова, В.П.Беспалько), развивают идеи программированного обучения и открывают совершенно **новые**, еще не исследованные технологические варианты обучения, связанные с

уникальными возможностями компьютеров и телекоммуникаций.

Широкое освоение этих технологий в практике образования возможно лишь при безусловной **компьютерной грамотности**, которую можно рассматривать как особую часть содержания компьютерной технологии. Таким образом, разработка методики преподавания информатики – одна из **актуальных проблем** современной педагогики (и в общедидактическом и частнометодическом планах).

Педагогическая целесообразность преподавания курса «Классическое структурное программирование»

В настоящее время практически ни одна сфера деятельности человека не мыслима без компьютера. Компьютеры позволяют собирать, хранить и обрабатывать большое количество информации намного легче и с большей скоростью, чем это было возможно ранее. Владение современными аппаратными и программными средствами открывают широкие перспективы перед квалифицированным пользователем компьютера. Трудно перечислить все сферы применения вычислительной техники и области применения информационных технологий.

Отвечая требованиям времени, в настоящий момент особую значимость приобретают все курсы дополнительного образования, которые связаны с дисциплиной «Информатика».

Одним из них является курс «Классическое структурное программирование».

Новизна и актуальность

Принципиальная **новизна** предлагаемой программы заключается в использовании метода проектов с первого урока и до последнего. Каждое занятие посвящено какой-либо теме из классического набора, но цель использования того или иного понятия у каждого обучающегося своя собственная. Таким образом, переплетаются методы проектов с индивидуальным обучением.

Новизна предложенной программы состоит еще и в том, что она сориентирована уже начиная с первого занятия не только на развитие стандартных пользовательских навыков, но и на техническое творчество учащихся. Базой для такого подхода с первого года обучения становятся профессиональные программные продукты (Microsoft Office, Borland Pascal, Adem и т.д.), которые органично переплетаются с темами по программированию.

Изучение курса «Информатика» в школе преследует **две цели**:

- общеобразовательную;

- прагматическую.

Общеобразовательная цель заключается в освоении учащимися фундаментальных понятий современной информатики. **Прагматическая** – в получении практических навыков работы с аппаратными и программными средствами современных персональных компьютеров.

В разработанной программе все темы (теоретические и практические) слиты воедино, изучение материала представляет собой непрерывный процесс, по завершении которого к учащемуся предъявляются следующие требования:

- Свободное владение вычислительной техникой;
 - знание основных устройств ВТ;
 - знание классификации программных средств;
 - умение работать с сервисными программами (оболочками, архиваторами и т.д.)
 - умение работать в редакторах (текстовых и графических), электронных таблицах, интегрированных средах и т.д.;
- Знание основ алгоритмизации и программирования:
 - знание основных типов переменных;
 - знание основных конструкций алгоритмического языка (линейное следование, ветвление, цикл);
- Знание элементов математической логики;
- Умения и навыки:
 - умение анализировать поставленную задачу;
 - умение выбирать методы и способы решения поставленной задачи;
 - умение анализировать полученные результаты и делать выводы.

Кроме наличия всех вышеперечисленных знаний, умений и навыков выпускник должен обладать стройным представлением целостной картины информационного мира, высокой компетентностью в области прикладной информатики.

Разработанная программа позволяет динамично реагировать на изменение требований, предъявляемых к аппаратному и программному обеспечению. Данная программа не привязана ни к какому конкретному дидактическому наполнению курса. Выбор конкретных задач зависит от ориентации каждой конкретной группы обучающихся. Кроме тематической ориентации каждая предлагаемая учащимся задача предполагает различную степень детализации в зависимости от уровня развития способностей каждого учащегося, что позволяет осуществлять дифференцированный и индивидуальный подход.

В программе отведено достаточное время для проектной работы, позволяющей раскрыть все творческие способности учащегося.

Но с увеличением обучения, одновременно увеличивается и без того не малая нагрузка (физическая, эмоциональная и интеллектуальная) на ученика, на неокрепший организм ребенка.

Необходимо, не сокращая объема получаемых знаний и умений, уменьшить загруженность учащихся (особенно учащихся младшего и среднего звена общеобразовательной школы).

Для решения поставленной задачи первоначально автором были подготовлены тексты – "шпаргалки".

В настоящее время огромное значение приобретает влияние загруженности учебного процесса на здоровье учащихся. Все чаще рождаются дети с ослабленным здоровьем, врожденными заболеваниями. Поэтому столь пристальное внимание в настоящее время уделяется профилактике заболеваний, соблюдению санитарно-гигиенических норм, физическому развитию подрастающего поколения, здоровью сберегающим технологиям обучения.

Рассказывать о санитарных нормах применяемых в настоящее время нет смысла – есть соответствующие нормативные акты и предписания, обязательные для выполнения и соблюдения. Хотя почти все ныне занимаемые компьютерными классами помещения не соответствуют каким-либо указанным нормам:

- недостаточные площади;
- неправильное освещение;
- неправильная ориентация (на солнечную сторону);
- отсутствие кондиционеров;
- мониторы с неизменяемым (или мало изменяемым) наклоном;
- мебель без возможности подбора высоты;

и т.д. и т.п.

Как правило, чаще всего говорится о вреде, наносимом здоровью ребенка длительной работой за компьютером или слишком ранним знакомством с ним. Вследствие чего появляется много противников введения компьютеризированных уроков информатики (ставятся задачи о разработке программ обучения по безмашинному варианту, вводятся все более жесткие временные ограничения для компьютеризированных занятий).

В данной программе предлагается иной взгляд на рассматриваемую проблему, который сводится к тому, что вводить машинный вариант изучения предметов связанных с информатикой (ОИиВТ, информационные технологии, информационная культура, логика, предметы на ЭВМ и т.д.) необходимо как можно раньше. Аргументы, как не странно, те же, что и у сторонников запрещения: забота о сохранении здоровья ребенка.

Чем раньше ребенок научится правильно общаться с вычислительной

техникой, тем больше гарантий, что, став взрослым, он не погубит своего здоровья по неосведомленности. К величайшему сожалению в правилах техники безопасности при работе с вычислительной техникой, как правило, не упоминаются многие **исключительно важные аспекты корректного поведения пользователя.**

Корректное поведение – это значит:

- сидеть за машиной, сохраняя правильную осанку (исключаются приобретенные сколиоз, остеохондроз, как следствие головные боли, боли в шейно-плечевой области);
- правильно держать руки: либо до локтя на рабочем столе, либо полностью на весу (предупреждается профессиональное, хроническое заболевание – ишемия мягких тканей (сдавливание сосудов и нервных стволов, нарушение кровоснабжения), приводящая к невритам, миозитам и т.д.);
- располагать монитор так, чтобы исключить бликование экрана и добиться прямого угла падения взгляда на поверхность экрана;
- уметь отрегулировать яркость и контрастность изображения так, чтобы создать максимально комфортные условия глазам (позволит избежать быстрой утомляемости глаз, как следствие – падения остроты зрения);
- привыкнуть во время работы выполнять физические упражнения (физ. паузы) для снятия чрезмерного напряжения мышц;
- научиться контролировать процесс моргания (как бы фантастично это не звучало)
при нормальном режиме работы человек моргает до 10 – 15 раз в минуту, при работе с монитором нетренированный пользователь сокращает количество смаргиваний до 3 – 5, результатом становится высыхание склер и как следствие – хронические склеротиты, конъюнктивиты.

Все вышеперечисленные навыки ребенок должен получить в детстве, когда закладываются основные привычки (вырабатываются условные рефлексы). Для молодого поколения правильное поведение за компьютером должно быть нормой, привычкой. Только тогда можно надеяться, что и позже приступив к профессиональной деятельности, человек сохранит свое природное здоровье.

На долю педагога выпадают задачи:

- терпеливо научить детей не только составлять алгоритмы и писать программы, но и правильно общаться с вычислительной техникой;
- построить свой урок так, чтобы дать максимум знаний и при этом соблюсти (по возможности) санитарно-гигиенические нормы.

Все эти задачи трудные, но выполнимые.

Но заботиться следует не только о физическом здоровье учащихся. Не меньше внимания следует уделять и интеллектуальному развитию, развитию творческих способностей.

Творчество - создание новых по замыслу культурных или материальных ценностей. (С.И.Ожегов и Н.Ю.Шведова. Толковый словарь русского языка)

Что такое "творческие способности"? Что именно развивать в ребенке? Чем измерить эти самые способности? Какие пути выбрать для развития этого сложно определяемого свойства?

Наверное, чтобы развивать в ребенке что-либо, нужно, прежде всего, самому обладать этим свойством, качеством. Нельзя научить другого человека плавать, если не умеешь сам. Научить "создавать новое по замыслу" может только тот, кто сам умеет это делать. Более того, мало уметь самому необходимо еще и уметь обучать.

Следовательно, прежде всего, стоит выяснить обладаешь ли сам теми качествами, которые собираешься развивать в другом.

Было бы очень нескромно со стороны автора заявить: "Да, я творческая личность! Да, я знаю, что и как развивать в своих учениках!"

Автор всего лишь пытается не стоять на месте, искать решения новых задач, сформировать в своих учениках уверенность в своих силах, перебороть в них стандартность мышления, заставить посмотреть на известный факт с другой точки зрения, увлечь учащихся поиском неизвестного в знакомом.

В руках педагога великолепный инструмент для всех вышеперечисленных задач – **компьютер**.

В каникулярное время авторами используются нестандартные формы занятий:

- тематические викторины и конкурсы;
- занятия – «Мозговая атака»;
- ролевые игры;
- тематические экскурсии;
- исследовательская работа;

Работа в каникулярное время позволяет использовать дополнительные факторы, влияющие на интеллектуальное, эстетическое и нравственное развитие учащихся:

- межвозрастное общение;
- творческое взаимодействие;
- изменение ролевого поведения;

Опыт работы педагогов показывает, что организация и проведение

подобных мероприятий позволяет эффективно соединить отдых и творческую работу, помогает ребятам открыть в себе новые таланты, поверить в собственные силы, найти направление для приложения своих творческих способностей. Результаты проводимой работы традиционно очень высоки. Познавательная активность учащихся возрастает в несколько раз.

Критерии оценки ожидаемых результатов

Результаты внедрения предлагаемой программы достаточно разнообразны:

- Оценка качества знаний учащихся в динамике является интересным показателем для анализа качественного изменения информационной подготовленности учащихся, психологических особенностей этапов педагогического процесса;
- Результативность проектного метода осуществляемого в рамках данного проекта является хорошим показателем подготовленности учащихся к самостоятельной интеллектуальной, научно-исследовательской деятельности;
- Результативность учащихся принимающих участие в конкурсных мероприятиях районного, городского и областного уровней – яркий показатель успешности внедрения предлагаемой системы;
- Выбор выпускниками школы специальностей информационной направленности является достаточным показателем высокой допрофессиональной подготовки учащихся и хорошей профориентационной результативности предлагаемой концепции.

Преподавание информатики интегрировано со следующими дисциплинами:

№	Наименование дисциплины	Класс
1	Логика	5 – 11
2	Предметные дисциплины (практические занятия)	5 – 9
3	Предметные дисциплины (проекты)	8 – 11

На практических занятиях используются задания требующие использующие знания и умения учащихся в следующих дисциплинах:

5 класс	математика, экономика
6 класс	математика, физика, география
7 класс	математика, физика, химия
8 класс	математика, физика, химия, литература, ИЗО, МХК

В программе отведено достаточное время для проектной работы, позволяющей раскрыть все творческие способности учащегося.

Проект представляет собой самостоятельную работу учащегося, рассчитанную на продолжительное время. Учащийся самостоятельно выбирает предметную область и тему своего проекта, формулирует цель работы, выдвигает предположение, выбирает средства и методы решения поставленной перед собой задачи, реализует решение, анализирует полученные результаты и делает выводы о достижении поставленной цели.

Продолжительность создания проекта:

8 класс	10 часов	III – IV четверть
9 класс	14 часов	III – IV четверть
10 класс	18 часов	III – IV четверть
11 класс	24 часа	II – IV четверть

При изучении курса «Информатика» предусмотрены следующие формы контроля знаний: терминологический диктант, самостоятельная работа, контрольная работа, зачет по теме, защита проекта.

2. Учебно-тематический план программы

2.1. Первый год обучения (Модуль 1)

№ п/п	Тема	Количество часов			
		Теория	Практика	Контроль	Всего
Раздел 1. Программирование: история и современность					
1.	Тема 1.1. Введение в курс «Классическое структурное программирование»	1	-	-	1
2.	Тема 1.2 Операции, операнды, операторы	2	3	-	5
3.	Тема 1.3 Уровни программного обеспечения	1	-	-	2
Раздел 2. Общепользовательские сведения					
4.	Тема 2.1. Оболочка к операционной системе	2	2	-	4
5.	Тема 2.2. Текстовый редактор	2	4	-	6
Раздел 3. Основы математической логики					
6.	Тема 3.1. Логические переменные и логические операции	2	4	-	6
Раздел 4. Основы структурного программирования					
7.	Тема 4.1. Переменные, операторы	2	4	-	6
8.	Тема 4.2. Алгоритмы	2	4	-	6
9.	Тема 4.3. Блочное программирование	2	3	-	5
10.	Тема 4.4. Промежуточный контроль	-	-	1	1
11.	Тема 4.5. Линейные операторы	2	4	-	6
12.	Тема 4.6. Операторы условия	2	4	-	6
13.	Тема 4.7. Операторы цикла повторения	1	2	-	3
14.	Тема 4.8. Подпрограммы	2	4	-	6
15.	Тема 4.9. Создание программного обеспечения	2	4	-	6
16.	Тема 4.10. Файлы	2	4	-	6
17.	Тема 4.11. Создание прикладных программных продуктов	3	5	-	9
Раздел 5. Ученик - пользователь компьютера					
18.	Тема 5.1. Интегрированные системы	2	4	-	6

19.	Тема 5.2. Пакеты прикладных программ (ППП)	2	2	-	4
20.	Тема 5.3. Подведение итогов изучения курса «Классическое структурное программирование» 1-го года обучения	-	1	-	1
21.	Тема 5.4. Итоговый контроль	-	-	1	1
ИТОГО		96			

2.2. Второй год обучения (Модуль 2)

№ п/п	Тема	Количество часов			
		Теория	Практика	Контроль	Всего
Раздел 1. Программирование: история и современность					
1.	Тема 1.1 Введение в курс «Классическое структурное программирование» 2-го года обучения	1	-	-	1
2.	Тема 1.2 Операции, операнды, операторы. Целочисленная арифметика	2	6	-	8
3.	Тема 1.3 Уровни программного обеспечения	2	-	-	2
Раздел 2. Общепользовательские сведения					
4.	Тема 2.1. Оболочка к операционной системе: повторение	2	5	-	7
5.	Тема 2.2. Промежуточный контроль	-	-	1	1
6.	Тема 2.3. Текстовый редактор. Дополнительные возможности	3	8	-	11
Раздел 3. Основы математической логики					
7.	Тема 3.1. Логические переменные и логические операции. Построение сложных условий	5	7	-	12
Раздел 4. Основы структурного программирования					
8.	Тема 4.1. Переменные, операторы. Основные назначения в программе	2	10	-	12
9.	Тема 4.2. Алгоритмы. Основные алгоритмические структуры	3	6	-	9

10.	Тема 4.3. Блочное программирование. Виды	2	7	-	9
11.	Тема 4.4. Линейные операторы. Повторение	-	-	1	1
12.	Тема 4.5. Операторы условия. Полное/неполное условие. Способы задания и описания	2	3	-	5
13.	Тема 4.6. Операторы цикла: решение задач повышенного уровня с различными видами циклов	2	10	-	12
14.	Тема 4.7. Подпрограммы: решение задач повышенного уровня	1	5	-	6
15.	Тема 4.8. Создание прикладных программных продуктов	1	9	-	10
16.	Тема 4.9. Подведение итогов изучения курса «Классическое структурное программирование» 2-го года обучения	-	1	-	1
17.	Тема 4.10. Итоговый контроль	-	-	1	1
ИТОГО					108

2.2. Третий год обучения (Модуль 3)

№ п/п	Тема	Количество часов			
		Теория	Практика	Контроль	Всего
Раздел 1. Программирование: история и современность					
18.	Тема 1.1 Введение в курс «Классическое структурное программирование» 3-го года обучения	1	-	-	1
19.	Тема 1.2 Операции, операнды, операторы. Целочисленная арифметика	2	6	-	8
20.	Тема 1.3 Уровни программного обеспечения	2	-	-	2
Раздел 2. Общепользовательские сведения					
21.	Тема 2.1. Оболочка к операционной системе: повторение	2	5	-	7
22.	Тема 2.2. Промежуточный	-	-	1	1

	контроль				
23.	Тема 2.3. Текстовый редактор. Дополнительные возможности	3	8	-	11
Раздел 3. Основы математической логики					
24.	Тема 3.1. Логические переменные и логические операции. Построение сложных условий	5	7	-	12
Раздел 4. Основы структурного программирования					
25.	Тема 4.1. Переменные, операторы. Основные назначения в программе	2	10	-	12
26.	Тема 4.2. Алгоритмы. Основные алгоритмические структуры. Повторение	3	6	-	9
27.	Тема 4.3. Арифметика многоразрядных целых чисел	2	7	-	9
28.	Тема 4.4. Комбинаторные алгоритмы	-	-	1	1
29.	Тема 4.5. Перебор и методы его сокращения	2	3	-	5
30.	Тема 4.6. Алгоритмы на графах	2	10	-	12
31.	Тема 4.7. Алгоритмы вычислительной геометрии	1	5	-	6
32.	Тема 4.8. Создание прикладных программных продуктов	1	9	-	10
33.	Тема 4.9. Подведение итогов изучения курса «Классическое структурное программирование»	-	1	-	1
34.	Тема 4.10. Итоговый контроль	-	-	1	1
ИТОГО		108			

3. Содержание программы

I. Введение.

1. История и современность.

Понятия: информация, информатика, программирование, программное обеспечение. Системы счисления: "10", "12", "16", "2" и др. - их историческое происхождение, следы этих систем в языке и байте. Двоичная система счисления, как система кодирования в ЭВМ.

2. Операции, операнды, операторы.

Понятия: константы, переменные, выражения, операции, операнды, операторы, функции, команды, программы (программное обеспечение).

3. Уровни программного обеспечения:

- операционные системы;
- оболочки к операционным системам;
- прикладное программное обеспечение.

Иерархическая (древесная) структура программного обеспечения.

II. Общепользовательские сведения

1. Оболочка к операционной системе

Понятия: директория (каталог), дерево, файл (текстовый, командный, выполняемый), имя файла, расширение имени файла, создание, изменение, сохранение, копирование, удаление файлов и директорий.

2. Текстовый редактор

Понятия: Создание и редактирование текста, курсор, вставка, замена, копирование, форматирование абзаца, вывод текста на печать, запись текстового файла в директорию ученика.

III. Основы математической логики.

1. Логические переменные и логические операции.

Понятия: логические переменные, логические операции, логические выражения. Логические операции: AND, OR, NOT. Наборы значений логических переменных. Таблицы истинности.

IV. Основы структурного программирования.

1. Переменные, операторы.

Типы переменных: простые (оригинальные) и сложные. Размерность типов. Области применения различных типов переменных (Boolean, Byte, Integer, Real, Char, String).

Типы операторов: линейные (ввода/вывода, чтения/записи, присваивания и т.д.), условные операторы (выбора), операторы цикла (повторения).

2. Алгоритмы.

Понятия: постановка задачи, алгоритм решения поставленной задачи, результат, оценка правильности алгоритма, единственность алгоритма, реализация алгоритма, виды алгоритмов.

3. Блочное программирование.

Понятие структурный блок. Виды структурных блоков. Пять принципов структурного программирования.

4. Линейные операторы.

Линейные операторы: ввода/вывода, чтения/записи, присваивания.

5. Операторы условия.

Понятия: IF - форматы записи, организация блоков BEGIN..END.

6. Операторы цикла (повторения).

Операторы цикла (повторения) с постусловием (REPEAT) - синтаксис, область применения.

7. Создание программного обеспечения.

Разработка алгоритма, создание блок-схемы, комментарии к программному обеспечению.

Написание программы - основы.

Отладка программного обеспечения. Принципы отладочных печати и подсветки.

Тестирование готового программного продукта.

1. Сложные типы переменных массив (ARRAY), запись (RECORD).

Размерность типов. Области применения различных типов переменных.

Типы операторов: линейные (ввода/вывода, чтения/записи, присваивания и т.д.), условные операторы (выбора), операторы цикла (повторения).

2. Алгоритмы. Вспомогательные алгоритмы, метод пошаговой детализации.

Понятия: постановка задачи, алгоритм решения поставленной задачи, результат, оценка правильности алгоритма, единственность алгоритма,

реализация алгоритма, вспомогательный алгоритм, детализация, укрупнение.

3. Блочное программирование.

Инвариантность к языкам принципов структурного программирования.

Анализ реализации в языках программирования различных структурных единиц. Способы организации структурных единиц в случае их отсутствия в языке. Сложные многоуровневые блок-схемы. Принцип вложенности структурных блоков.

4. Операторы условия (выбора).

Операторы условия (выбора): IF, CASE - сравнительная характеристика реализации операторов условия (выбора) в различных языках программирования.

5. Операторы цикла (повторения).

Операторы цикла (повторения) с заведомо известным количеством повторений (FOR) - синтаксис, область применения, ограничения в различных языках программирования. Оператор цикла FOR для работы с матрицами.

Операторы цикла (повторения) с предусловием (WHILE) - синтаксис, область применения, ограничения в различных языках программирования. Оператор цикла WHILE для работы с массивами типа RECORD.

Операторы цикла (повторения) с постусловием (REPEAT) - синтаксис, область применения, ограничения в различных языках программирования. Оператор цикла REPEAT для работы с массивами типа RECORD.

Прочие применения операторов цикла.

5. Подпрограммы.

Процедуры и функции. Формальные и фактические параметры. Библиотеки стандартных процедур и функций.

7. Создание программного обеспечения.

Этапы создания программного продукта. Работа с экраном. Различные стили оформления экрана. Контроль корректного ввода/вывода. Оформление результатов выполнения программы: сообщения, таблицы, диаграммы, графики. Тестирование готового программного продукта. Защита и реклама.

1. Файлы.

Работа с файлами. Типизованные и не типизованные файлы. Файловые переменные. Чтение с диска и запись на диск. Процедуры и функции для обработки файлов.

2. Создание прикладных программных продуктов имеющих сложное структурное строение с использованием всех известных конструкций, отладка, тестирование, защита программного обеспечения,

Библиотеки стандартных процедур и функций, подключаемые модули, создание собственных библиотек.

Создание прикладных программных продуктов имеющих сложное структурное строение с использованием всех известных конструкций, отладка, тестирование, защита программного обеспечения.

Арифметика многоразрядных целых чисел. Основные арифметические операции. Комбинаторные алгоритмы. Перестановки. Размещения. Сочетания. Скобочные последовательности. Представление графа в памяти компьютера. Поиск в графе. Алгоритм Дейкстры. Алгоритм Флойда. Алгоритмы вычислительной геометрии. Базовые процедуры. Задачи о прямоугольниках. Тестирование программы при решении задач.

IV. Ученик - пользователь компьютера.

1. Интегрированные системы.

Исследовательская деятельность. Работа в неизвестных средах (русифицированных и англоязычных). Самостоятельное изучение второго языка программирования. Сравнительный анализ программных продуктов одного класса.

2. Пакеты прикладных программ (ППП).

Многообразие ППП для различных областей деятельности. Принципы построения ППП. Интерфейс, "хелпирование", основные функции. Внедрение и сопровождение прикладных пакетов программ.

Литература

1. Delphi справочное руководство. Borland International, Inc. 1997.
 2. InterBase справочное руководство. Borland International, Inc. 1996.
 3. А.Жуков. Изучаем Delphi. Санкт-Петербург, Питер 2000.
 4. Добудько Т.В., Пугач В.И. Открытые программные комплексы «Роботландия». Самара, 1996.
 5. Добудько Т.В., Пугач В.И. Программируем на Pascal. Самара, 1996.
 6. Документация в электронном виде к Borland Pascal 7.0
 7. Документация в электронном виде к CAD/CAM системе Adem for Windows 3.03.
 8. Документация в электронном виде к Delphi 3.0
 9. Документация в электронном виде к музыкальному конструктору Dance Machine.
 10. Документация в электронном виде к музыкальному редактору «Шарманщик».
 11. Документация в электронном виде к музыкальному редактору Cakewalk Professional.
 12. Документация в электронном виде к музыкальному редактору Sound Forge.
 13. Документация в электронном виде к языку программирования Basic.
 14. Инженерная графика. Методические указания к лабораторным работам для студентов Тольяттинского университета им. Татищева. Тольятти, 1996.
 15. Н.Вирт. Алгоритмы и структуры данных. Москва «Мир» 1989.
 16. Н.Рубенкинг. Турбо Паскаль для Windows том 1. Москва «Мир», СК Ферлаг Интернешнл, 1994.
 17. Рекламный буклет. Тольяттинский политехнический институт. Учебно-методический центр CAD/CAM.
 18. Стефанюк В.Л. Компьютер обретает разум. Москва, «Мир», 1990.
 19. Учебное пособие под редакцией Чемпинского Л.А. Компьютерные чертежно-графические системы для разработки конструкторской и технологической документации в машиностроении. Москва АСАДЕМА 2002.
 20. Чемпинский Л.А. Выполнение чертежей на ПЭВМ. Методические указания к практическим занятиям. Самара 1994.
- Чемпинский Л.А. Инженерная графика на ПЭВМ. Самара, 19

21. Окулов С.М. Программирование в алгоритмах. М.: БИНОМ. Лаборатория знаний, 2002