# Реферат

Информационно-справочная система

"Устройство персонального компьютера"

Пояснительная записка:

Данный программный продукт, выданный как тема для курсового проекта, представляет собой реализацию информационной системы, предназначенной для ознакомления пользователей основным свойствам тригонометрических функций. В программном продукте реализованы следующие возможности:

1. предоставление справки о разработчике системы;
2. справка об простейших тригонометрических функций;
3. предоставление справки по работе с программой.

Эта программа удобна в использовании, так как имеет простой и интуитивно-понятный графический интерфейс.

Перспективами дальнейшей разработки программного продукта являются: оптимизация графических возможностей рабочей области программы, увеличение объема информации об тригонометрических функциях.

Содержание

Введение

1. Постановка задачи

1.1 Цель разработки

1.2 Функциональные требования

1.3 Информационные потоки программы

2. Описание и обоснования метода решения

3 Алгоритм решения задачи

3.1 Словесное описание алгоритма программы

3.2 Алгоритм программы

3.3 Обоснование выбора языка программирования

4. Проектирование интерфейса

4.1 Обоснования выбора вида интерфейса

4.2 Особенности организации выбранного графического режима

5. Описание программной реализации

5.1Функционально-логическая схема

5.2 Описание структур и форматов

5.3 Основные процедуры

5.4 Схема взаимодействия подпрограмм

6 Комплект поставки и порядок установки

6.1 Комплекс поставки

6.6 Порядок установки

6.7 Запуск программного продукта

7. Анализ эффективности

Выводы

Перечень использованных источников

Приложение А. Техническое задание

Приложение Б. Руководство пользователя

Приложение В. Листинг программы

Приложение Г. Экранные формы

Введение

На сегодняшний день, разнообразные информационные системы являются, наряду с компьютерными играми, пожалуй, одним из самых популярных и востребованных видов программного обеспечения. Их назначение – помощь в процессе приобретения и накопления пользователем знаний и навыков в узких предметных областях. Спектр применения информационных систем достаточно широк, и это обусловливает высокий потребительский спрос на программные продукты подобного типа, что в свою очередь способствует дальнейшему их развитию, и появлению все новых и новых видов обучающих систем. Практически на каждом персональном компьютере можно встретить хотя бы одну такую программу. Их надежность и высокие показатели эффективности привели к тому, что информационные системы стали широко применяться во многих школах и вузах. А использование ими новейших достижений в области мультимедиа позволяет использовать такие системы даже в процессе так называемого дистанционного обучения.

Стоит отметить, что, некоторые обучающие системы позволяют так же моделировать на экране ситуации, близкие к реальным. И это обусловливает еще одну возможность применения информационно-обучающих систем, а именно, их использование в качестве тренажеров – симуляторов при подготовке и переподготовке специалистов самых различных уровней в самых различных областях знаний.

Данная информационная система позволяет эффективно хранить необходимую пользователю информацию и осуществлять быструю навигацию по накопленному в базе материалу.

Таким образом, простота обращения с информационно-обучающими системами, их доступность и понятность широкому кругу пользователей, полнота и структурированность изложенной в них информации, возможность выбора наиболее подходящего для каждого конкретного пользователя метода обучения способствуют все большему и большему росту их популярности. Компьютер, снабженный информационно-обучающей системой становиться не просто машиной для проведения сложных расчетов, но и активным помощником в процессе приобретения и накопления новых знаний.

**1. Постановка задачи**

**1.1 Цель разработки**

Целью разработки данного программного продукта является создание узкоспециализированной информационной системы, предназначенной для ознакомления и обучения широкого круга пользователей, которых интересует тригонометрические функции.

**1.2 Функциональные требования**

Поскольку основной задачей данного программного продукта является обучение, то необходимой является его ориентация именно на пользователей. Необходимо создать условия, при которых эффект от процесса обучения был бы максимальным. То есть необходимо создание простого, понятного, функционального и привлекательного пользовательского интерфейса, а так же удобных инструментов, которые позволили бы использовать возможности информационной системы в полной мере.

С этой точки зрения оптимальным является графический интерфейс. Он удобен и практичен в использовании, а так же является интуитивно-понятным, что в нашем случае является не менее важным.

Управление основными возможностями системы должно быть организовано с посредством системы раскрывающихся меню, доступ к которым должен обеспечиваться непосредственным выбором.

Справочная система должна содержать информацию о способе навигации в системе, функциональных клавишах, информацию о разработчике программного продукта и основных возможностях которыми располагает данная система.

Система обучения должна обеспечивать простоту доступа пользователя к необходимой ему информации, а это предоставляет простой графический интерфейс, и кнопка выбора определенного пункта.

**1.3 Информационные потоки программы**

Данный подраздел содержит схему информационных потоков, которые фигурируют в программе.

Данная схема предоставляет основные информационные потоки данного программного продукта. Цилиндр показывает данные, хранящиеся на жестком диске компьютера. Прямоугольник изображает данные, которые выводятся на экран.

Стрелками указаны движение информации в блоках.

Блок информационных материалов

Файлы с текстовой информацией

Вывод информационных материалов

Файлы с текстовой информацией

Блок справочной информации

Справочная информация

Блок заставки

Графический лист информации выполнения (титульный лист)

Рисунок 1.1 - Схема информационных потоков

Данный продукт разбит на три блока, за которыми закреплена определенная информация:

- за блоком заставки закреплена определенная информация, выводимая на экран монитора с помощью программного кода продукта;

- за блоками информационных материалов и блока справочной информации закреплена информация в виде файлов, которые находятся на жестком диске.

**2. Описание и обоснования метода решения**

Главными задачами, поставленные перед программой, были: создание информационной базы материала, вывод из этой базы необходимого файла, выбранного пользователем. Программа представляет из себя комплекс процедур и функций, предназначенных для решения вышеуказанных задач. При запуске программы, происходит начало диалога пользователя с программой, который реализован при помощи графических пользовательских меню. Эти меню являются чем-то вроде посредника между пользователем и информационно-справочной системой. При необходимости пользователь может воспользоваться необходимым пунктом в меню и тем самым запустить какой-то процесс.

Каждый пункт в меню реализован в виде отдельной процедурой. Это позволяет легко ориентироваться в написанном коде и при надобности вносить необходимые корректировки и изменения.

**3 Алгоритм решения задачи**

**3.1 Словесное описание алгоритма программы**

Структура программы представляет собой комплекс процедур и функций. Этот комплекс подпрограмм составляет общую функциональность всей программы в целом.

После запуска программы на экране происходит очистка его, а затем, выполнение процедуры, которая отображает главное меню. Затем выполняется другая процедура, которая выводит специально разработанный интерфейс программы, затем выполняется процедура, отвечающая за работу главного меню. Далее выполняется процедура, которая отслеживает нажатие клавиш, предусмотренных в теле данной подпрограммы. При нажатии соответствующей клавиши эта процедура производит запуск соответствующей процедуры. Далее передвигаясь по экранному меню, пользователь останавливается на определенном пункте, для которых заранее предусмотрены процедуры, отвечающие за каждый пункт. Экранное меню, разработанное в графическом режиме, содержит четыре пункта: информация, справка, разработчик, выход.

В случае выбора процедуры «разработчик» на экран выводится графический интерфейс, предоставляющий пользователю информацию о разработчике.

В случае выбора пункта «справка» на экран выводится справка по данной информационной базе.

В случае выбора пункта «информация» пользователь получает доступ еще к одному меню, который содержит восемь пунктов: история, Sin X, Cos X, tg X, ctg X, arctg X, arcctg X, помощью. Каждому пункту меню соответствует свой текстовый файл, хранящийся на жестком диске. Пункт «помощь» на экран выводит справку по данной информационной базе.

В случае выбора пункта «Выход» пользователь автоматически завершает работу с программой.

Для вывода имеющегося материала в информационной базе запускается отдельная процедура, которая инициализирует и читает текстовый файл с информацией о имеющихся информационных файлах и путь к ним на диске.

**3.2 Алгоритм программы**

При создании данного программного продукта использовался принцип нисходящего программирования, т.е. задача разбивалась на некоторые подзадачи, каждая из которых в свою очередь снова детализировалась, пока не достигались, простые задачи. В первую очередь были выделены и решены следующие четыре основные подзадачи:

- организация пользовательского интерфейса;

- организация меню информации тригонометрических функций;

- организация изображения титульного листа.

Следует отметить, что система была реализована таким образом, что в каждый конкретный момент времени ее состояние могло измениться только в результате некоторых действий пользователя. Поэтому система использует только так называемый последовательный процесс, при котором все действия выполняются строго последовательно. На рисунке 2.1 представлен алгоритм работы основной программы. Заметим, что структурной основной единицей процесса работы программы является процесс обработки команды пользователя. Ему принадлежит основной приоритет.

Инициализация программы

2) Прорисовка рабочего меню

3) Обработка команды пользователя

4) Прорисовка меню информации тригонометрических функций.

7) Прорисовка меню справки

8) Прорисовка титульного листа

6) Завершающие действия (закрытие файлов и т.д.)

9) Обработка команды пользователя

5) Обработка команды пользователя

Рисунок 3.1 – Алгоритм работы программы

Рисунок 3.1 Показывает алгоритм данного программного продукта. Его действие обусловлено следующими шагами, которые с легкостью управляются пользователем:

1. На начальном этапе работы программы необходима ее инициализация;

2) После этого прорисовывается рабочее меню, состоящее из четырех пунктов, которым соответствует определенные процедуры. Двигаясь по меню, пользователь делает выбор;

3) После этого происходит обработка команды пользователя, которое завершается или 4), или 7), или 8), или 10) уровнем;

4) Прорисовывает меню информации тригонометрических функций. После нажатия управляющей клавиши, 4)-ой уровень переходит в 5);

5) Уровень обработки команды пользователя, завершающийся 6) уровнем;

6) Уровень закрытия информационных файлов, в случае пункта выхода уровень переходит на уровень 4);

7) Уровень прорисовки меню справки, который с помощью управления пользователя переходит в 9) уровень;

8) Прорисовка титульного листа, который с помощью управления пользователя переходит в 9) уровень;

9) Уровень обработки команды пользователя завершается 2) уровнем;

10) Уровень выхода из программы.

**3.3 Обоснования выбора языка программирования**

Данный курсовой проект написан на языке Turbo Pascal версии 7.0

Конечно, стоит отдать должное Паскалю как языку программирования за его наглядность при работе с такими общими для всех языков программирования темами, как структуры данных, рекурсия, способы и механизмы передачи параметров. А вот что главное в Паскале, так это его простота: именно поэтому он в большинстве случаев используется как обучающий язык на начальном уровне подготовки будущих программистов. В нем доступна для понимания работа с необходимыми процедурами и функциями, общей синтаксической конструкцией языка. Однако, Паскаль давно уже морально устарел и опытными программистами не может применяться, постольку существует ряд других, более гибких и практичных в использовании языков.

**4. Проектирование интерфейса**

**4.1 Обоснования выбранного вида интерфейса**

В данном программном продукте интерфейс - это работа с графическим режимом. Он очень удобен в использовании, приятно восприимчив для глаз. В нем сможет разобраться даже совершенно неопытный пользователь. В нижней части экрана имеется строка подсказок, которые еще больше упрощают программу для пользователя. Поэтому я считаю, что выбранный мною вид интерфейса является актуальным в плане информационно-справочном программном продукте.

**4.2 Особенности организации выбранного графического режима**

Данный интерфейс состоит из нескольких пунктов и единиц: основное меню имеет вид централизованного списка, по которому легко передвигается строка выбора, информационное меню имеет также централизованный список пунктов. В данной работе статусная строка имеет вид: утвердить, движение вверх, движение вниз, выход из программы или меню.

Рисунок 4.1 предоставляет схему организации данного интерфейса, который состоит из пяти активных окон, легко вызывающиеся в любой момент. Стрелками указано взаимодействие окон между собой, а также цикл выхода из программы. Статусная строка поможет без труда передвигаться между окнами.

Первое активное окно (окно основного меню)

Второе активное окно (окно информационного меню)

Третье активное окно (окно вывода файлов)

Четвертое активное окно (окно вывода справки)

Пятое активное окно (окно разработчика)

Рисунок 4.1-Схема организации интерфейса

**5 Описание программной реализации**

**5.1 Функционально-логическая схема**

Данная схема предоставляет функционально-логическую схему.

Обрисовка главного меню программы

Ввод пункта меню Р

Р=F1

Р=F3

Р=F4

Р=F5

Р=F2

Вывод окна информации о тригонометрических функциях

Вывод информации о тригонометрических функциях

Вывод информации справки

Вызов информации о разработчике

Выход из программы

Рисунок 5.1-Функционально-логическая схема

Рисунок 5.1 представляет функционально-логическую схему, которая реализует логическую функциональность блоков программы. Каждый пункт программы выполняет определенное действие, поэтому от каждого пункта изображена линия, которая показывает действие выбранного пункта.

**5.2 Описание структур и форматов**

В данном разделе описаны выходные данные и разобраны их структура

**5.1.1 Выходные данные**

Выходными данными являются данные в памяти файлов, которые хранятся на жестком диске компьютера. Эти файлы используются для вывода информации о выбранных устройствах персонального компьютера, а также для вывода информации о руководстве пользователя. Они имеют расширение «\*.txt».

**5.3 Основные процедуры**

Ниже перечислены процедуры, которые встречаются в программе.

Procedure initgr ();

Процедура для инициации графического режима

Procedure main;

Процедура для реализации рабочего главного меню.

Procedure greator();

Отображает титульный лист с информацией об авторе программы.

Procedure colo();

Процедура для вывода экрана приветствия

Procedure ReadF (name:string, f1:boolean);

Процедура для инициализации файла.

Procedure select;

Процедура прорисовки окна выбора справки;

Procedure WorSpece;

Процедура построения рабочей области экрана;

Procedure move;

Процедура выбора меню;

Procedure help;

Процедура прорисовки справочного окна программы.

**6. Комплекс поставки и порядок установки**

В данном разделе будет рассмотрен комплекс поставки программного продукта, а также порядок установки его.

**6.1 Комплект поставки**

Программный продукт поставляется на дискете 3,5". В корневом каталоге дискеты находится каталог Saniokkurs, содержащий следующие файлы:

mykurs.pas – главный исполняемый файл;

Egavga.bgi – драйвер видеоадаптера;

Keyrus.com – драйвер клавиатуры;

Также в папке mykurs находится еще один каталог, содержащий файлы

fill.txt – файл содержащий информацию про историю тригонометрии;

sin.txt – файл, содержащий информацию о функции sin X;

cos.txt – файл, содержащий информацию о функции cos X;

tg.txt – файл, содержащий информацию о функции tg X;

ctg.txt – файл, содержащий информацию о функции ctg X;

arctg.txt – файл, содержащий информацию о функции arctg X;

arcctg.txt – файл, содержащий информацию о функции arcctg X;

help.txt – файл, содержащий информацию помощи;

**6.2 Порядок установки**

Процедура инсталляции состоит в следующем:

- средствами операционной системы создаётся каталог для программы (возможно также использование уже существующего каталога, но создание для программы отдельного каталога является предпочтительным);

- средствами операционной системы файлы, входящие в комплект поставки, переносятся из каталога Saniokkurs дискеты в подготовленный каталог.

**6.3 Запуск программного продукта**

Для запуска программного продукта для начала необходимо выполнить файл Keyrus.com, а после запустить файл mykurs.exe. В случае отсутствия некоторых файлов, система выдаст соответствующее предупреждение, и если эти файлы были критически важными, прекратит работу.

**7. Анализ эффективности**

Данная программа обладает довольно качественным и понятным интерфейсом. Пользователь без труда может сориентироваться при запуске приложения при помощи подсказок, находящихся в нижней части экрана. Также одним из достоинств программы является то, что диалог пользователя с программой осуществляется при помощи понятных и довольно удобных диалоговых окон. Следует также заметить то, что в целом, программа выполнена в классическом варианте для приложений, разработанных для операционной MS-DOS. То есть, это удобная, не надоедливая цветовая гамма, которая не раздражает после долгой работы органы зрения пользователя.

В программе реализована удобная навигация по информационной базе, насыщенная информационная база.

**Выводы**

После завершения работы над программой, её можно протестировать и выявить как достоинства, так и недостатки. Мною были выполнены поставленные задачи, а именно:

- в программе был реализован удобный графический пользовательский интерфейс.

Также следует дополнительные достоинства: это возможность пополнения базы новыми информационными материалами.

В будущем, возможна модернизация программы в следующих направлениях:

- сделать программу более удобной при чтении материала из информационной базы;

- сделать программу под более современную операционную систему.

**Список использованной литературы**

1. «Программирование в среде Turbo Pascal 7.0», А. И. Марченко; 1998.
2. Конспект лекций по предмету «ОП и АЯ».
3. Энциклопедия современного персонального компьютера Москва 2005 г.
4. Ресурс в сети Интернет:

http://garbo.umasa.fi/pc/turboobj.html

http://garbo.umasa.fi/pc/turbopas.html

http://infobase.hypermart.net/pascal/programs.html

<http://infocity.kiev.ua/>

5. Самоучитель Turbo Pascal О.А. Меженный Киев 2004.

**Приложение А**

## **Техническое задание на проектируемый программный продукт**

А.1 Общие сведения

Полное название программного продукта: “Разработка информационно системы «Простейшие тригонометрические функции»”. Ее условное обозначение «ПТФ». Работа выполнена студенткой 1-го курса Донецкого государственного института искусственного интеллекта (ДонГИИИ),факультета современных компьютерных информационных технологий, группы СУА-05, Булавиным Александром Николаевичам.

Основанием для разработки «ПТФ» является задание, выданное кафедрой Программного обеспечения интеллектуальных систем (ПОИС).

Плановый срок начала работы: 17.02.06.

Дата защиты работы: 22.05.06.

Курсовая работа выполняется согласно графику, приведенному в таблице А.1.

##### А.2 Назначения создания программы ведения

Программа предназначена для обучения пользователя простейшим тригонометрическим функциям.

А.3 Требования к программе

А.3.1 Требования к программе в целом

В целом к программе предъявляются следующие требования:

1. удобный и простой для пользования графический интерфейс;
2. удобный способ ввода-вывода данных;
3. контроль действий пользователя, связанные с работой программы;
4. хранение информации во внешнем источнике данных;

- наличие справочной информации

А.3.2 Требования к задачам и функциям программного продукта

Программа должна реализовать следующие функции:

1. удобный доступ к информационному материалу;
2. вывод легко доступной для восприятия справочной информации;

А.3.3 Требования к программному обеспечению

Программа будет реализована на языке Turbo Pascal 7.0 и для ее работы необходимо наличие операционной системы MS-DOS 5.0 и выше.

А.3.4 Требования к техническому обеспечению

Для корректной работы программы необходимы:

1. IBM – совместимый ПК с процессором не ниже 80386;
2. VGA монитор.

А.4 Стадии разработки программы:

Таблица А.1 – Стадии разработки программы

|  |  |
| --- | --- |
| Этапы работы | Срок выполнения (недели) |
| Постановка задачи: определение требований к программному комплексу. Формулировка постановки задачи: исходные данные, ограничения, результат, связь. | 1-2 |
| Составление технического задания | 2-3 |
| Техническое проектирование.  Разработка метода решения задачи. Модульный анализ: определение структуры программы, выделение модулей, организация и их взаимосвязи. | 3-4 |
| Разработка основного алгоритма функционирования программы. Составление спецификаций модулей. | 5 |
| Рабочее проектирование:  Определение структур входных и выходных данных, алгоритмов работы модулей. Оценки структур данных и алгоритмов. Описание входных и выходных данных. | 6-7 |
| Написания программы. | 8-10 |
| Отладка программы на верхнем уровне. | 11-12 |
| Комплексная отладка и тестирование. | 12-13 |
| Написание пояснительной записки. | 11-14 |
| Защита курсового проекта. | 22.05.06 |

**Приложение Б**

## **Руководство пользователя**

Помощь при работе с информационной системой

Информация.

Отображает список доступных для рассмотрения тем. Выбрав этот пункт меню, появится окно с названием простейших тригонометрических функциями. Используя клавиши "вверх", "вниз" и "enter" вы можете выбрать функцию, которое вас интересует.

Справка:

Отображает справочное меню.

Разработчик:

Отображает информацию о разработчике.

Выход:

Используя этот пункт, вы можете выйти из программы.

**Приложение В**

**Листинг программы**

Program my\_kursuch;

Uses Graph,crt;

Var i,j,y:integer;

com:char;

txt:array[1..750]of ^string;

procedure worspece; forward;

{|----------------------------------------------|}

{| Процедура вывода текстового файла на экран |}

{|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|}

Procedure ReadF(name:string;fl:boolean);

var

com:char;

str:string;

f:text; pos:integer;

n:integer;

i,j:integer;

begin

name:=concat('\TXT',name);

cleardevice;

worspece;

assign(f,name);

{$I-}

reset(f);

{$I+}

if IOresult<>0 then

begin

cleardevice;

textcolor(red);

outtext('Не найден файл, проверьте наличие файла на диске.');

readkey;

end

else

begin

n:=1;

while not eof(f) do {Read from Fille in Massiv}

begin

readln(f,str);

txt[n]^:=str;

n:=n+1;

end;

n:=n-1;

if n<=25 then

begin

pos:=40;

for j:=1 to n do

begin

outtextXY(40,20+pos,txt[j]^);

pos:=pos+10;

end;

repeat

until readkey=#27;

end

else

begin

pos:=40;

for j:=1 to 25 do

begin

outtextXY(40,20+pos,txt[j]^);

pos:=pos+10;

end;

i:=25;

while com<>#27 do

begin

com:=readkey;

if com=#72 then {///Up}

if i>=26 then

begin

worspece;

delay(60);

pos:=40;

for j:=i-26 to i-1 do

begin

outtextXY(40,20+pos,txt[j]^);

pos:=pos+10;

end;

i:=i-1;

end;

if com=#73 then {/// PageUp}

if i>=50 then

begin

pos:=40;

worspece;

delay(60);

for j:=i-49 to i-25 do

begin

outtextXY(40,20+pos,txt[j]^);

pos:=pos+10;

end;

i:=i-25;

end

else

if i<>25 then

begin

worspece;

delay(60);

pos:=40;

for j:=1 to 25 do

begin

outtextXY(40,20+pos,txt[j]^);

pos:=pos+10;

end;

i:=25;

end;

if com=#80 then {///Down}

if i<=n-1 then

begin

worspece;

delay(60);

pos:=40;

for j:=i-24 to i+1 do

begin

outtextXY(40,20+pos,txt[j]^);

pos:=pos+10;

end;

i:=i+1;

end;

if com=#81 then {/// PageDown}

if i<=n-25 then

begin

worspece;

delay(60);

pos:=40;

for j:=i to i+25 do

begin

outtextXY(40,20+pos,txt[j]^);

pos:=pos+10;

end;

i:=i+25;

end

else

if i<>n then

begin

worspece;

delay(60);

pos:=40;

for j:=n-25 to n do

begin

outtextXY(40,20+pos,txt[j]^);

pos:=pos+10;

end;

i:=n;

end;

end;

end;

end;

com:=#1;

end;

{----------------------КОНЕЦ-------------------------------}

{|----------------------------------------------|}

{| Процедура прорисовки окна выбора справки |}

{|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|}

procedure select;

begin

setfillstyle(1,blue);

bar(0,0,640,240);

setfillstyle(1,yellow);

bar(0,210,640,480);

setcolor(blue);

bar3d(30,215,150,235,0,topoff);

outtextXY(40,220,'История');

setcolor(red);

outtextXY(50,250,'SIN X');

outtextXY(60,280,'COS X');

outtextXY(70,310,'tg X');

outtextXY(80,340,'ctg X');

outtextXY(90,370,'arctg X');

outtextXY(100,400,'arcctg X');

outtextXY(110,430,'Помощь');

com:=#1;

i:=1;

j:=1;

repeat

com:=readkey;

if((i=1) and (com=#72)) then

begin

j:=i;

i:=i+7;

end

else

if((i=8) and (com=#80)) then

begin

j:=i;

i:=i-7;

end

else

f((y<>1) and (com=#72)) then

begin

j:=i;

i:=i-1;

end

else

f((y<>1) and (com=#80)) then

begin

j:=i;

i:=i+1;

end;

setcolor(blue);

case i of

1:begin bar3d(30,215,150,235,0,topoff); outtextXY(40,220,'История');end;

2:begin bar3d(40,245,160,265,0,topoff); outtextXY(50,250,'SIN X');end;

3:begin bar3d(50,275,170,295,0,topoff); outtextXY(60,280,'COS X');end;

4:begin bar3d(60,305,180,325,0,topoff); outtextXY(70,310,'tg X');end;

5:begin bar3d(70,335,190,355,0,topoff); outtextXY(80,340,'ctg X');end;

6:begin bar3d(80,365,200,385,0,topoff); outtextXY(90,370,'arctg X');end;

7:begin bar3d(90,395,210,415,0,topoff); outtextXY(100,400,'arcctg X');end;

8:begin bar3d(100,425,220,445,0,topoff); outtextXY(110,430,'Помощь');end;

end;

setcolor(red);

if ((com=#72) or (com=#80)) then

case j of

1:begin

setcolor(yellow);

bar3d(30,215,150,235,0,topoff);

setcolor(7);

outtextXY(40,220,'История');

end;

2:begin

setcolor(yellow);

bar3d(40,245,160,265,0,topoff);

setcolor(8);

outtextXY(50,250,'SIN X');

end;

3:begin

setcolor(yellow);

bar3d(50,275,170,295,0,topoff);

setcolor(9);

outtextXY(60,280,'COS X');

end;

4:begin

setcolor(yellow);

bar3d(60,305,180,325,0,topoff);

setcolor(10);

outtextXY(70,310,'tg X');

end;

5:begin

setcolor(yellow);

bar3d(70,335,190,355,0,topoff);

setcolor(11);

outtextXY(80,340,'ctg X');

end;

6:begin

setcolor(yellow);

bar3d(80,365,200,385,0,topoff);

setcolor(12);

outtextXY(90,370,'arctg X');

end;

7:begin

setcolor(yellow);

bar3d(90,395,210,415,0,topoff);

setcolor(13);

outtextXY(100,400,'arcctg X');

end;

8:begin

etcolor(yellow);

bar3d(100,425,220,445,0,topoff);

setcolor(17);

outtextXY(110,430,'Помощь');

end;

end;

until((com=#27) or (com=#13));

end;

{---------------КОНЕЦ----------------------------}

{|----------------------------------------------|}

{| Процедура построения рабочей области экрана |}

{|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|}

procedure WorSpece;

begin

setcolor(15);

setfillstyle(1,9);

bar3d(0,0,getmaxX,getmaxY,0,true);

setfillstyle(1,7);

setcolor(0);

bar3d(15,15,getmaxX-15,getmaxY-15,0,true);

setcolor(4);

setfillStyle(1,8);

bar3d(16,16,getmaxX-16,30,0,true);

setcolor(0);

line(17,32,17,getmaxY-17);

line(17,32,getmaxX-17,32);

line(getMaxX-17,32,getmaxX-17,getmaxY-17);

line(17,getmaxY-17,getmaxX-17,getmaxY-17);

bar3d(17,getmaxY-120,getmaxX-17,getmaxY-17,0,true);

setcolor(15);

outtextXY(90,getmaxY-100,'Page Up-страница вверх, Page Down-страница вниз');

outtextXY(105,getmaxY-80,'Стрелки вверх и вниз-перемещение по строкам');

setcolor(4);

end;

{----------------------КОНЕЦ----------------------------------------}

{|------------------------------------------------------------}

{| Процедура вывода реквизитов разработчика на экран |}

{|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|}

procedure greator;

var i:byte;

begin

cleardevice;

begin

i:=60;

{---------------------------рисуем мышь--------------------------------}

setfillstyle (1,4);

fillellipse (320,240,80,140);

setfillstyle (1,1);

fillellipse (320,240,60,20);

setcolor (15);

settextstyle (4,0,30);

outtextXY (295,235,'S V E N');

line (250,180,390,180);

line (320,180,320,100);

{-----------------------------------------------------------------------}

SETFILLSTYLE (1,15);

setcolor (black);

settextstyle(9,0,15);

bar3D (0,460,300,477,0,FALSE);

outtextxy (40,465,'Любая клавиша-возврат в меню');

bar3D (320,460,638,477,0,FALSE);

outtextxy (400,465,'Любая клавиша-возврат в меню');

setcolor (10);

outtextxy (120,i,'ДОНЕЦКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА');

delay (200);

outtextxy (140,i+40,'КАФЕДРА ПРОГРАММНОГ ОБЕСПЕЧЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ');

delay (200);

outtextxy (230,i+80, ' КУРСОВАЯ РАБОТА');

delay (200);

outtextxy (170,210,' ПО ДИСЦИПЛИНЕ:"АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ И ПРОГРАММИРОВАНИЕ"');

delay (200);

outtextxy (120,260,'ТЕМА:"РАЗРОБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ"');

delay (200);

outtextxy (160,280,'"ПРОСТЕЙШИЕ ТРИГОНОМЕТРИЧЕСКИЕ ФУНКЦИИ"');

delay (200);

outtextxy (480,300,'ВЫПОЛНИЛА:');

delay (200);

outtextxy (460,320,'СТ.ГР.:');

delay (200);

outtextxy(460,340,'');

end;

end;

{--------------------КОНЕЦ ПРОЦЕДУРЫ---------------------------}

{|------------------------------------------------------------}

{| Процедура выбора меню |}

{|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|}

procedure move;

begin

com:=#1;

i:=1;

j:=1;

repeat

com:=readkey;

if((i=1) and (com=#72)) then

begin

j:=i;

i:=i+3;

end

else

if((i=4) and (com=#80)) then

begin

j:=i;

i:=i-3;

end

else

if((y<>1) and (com=#72)) then

begin

j:=i;

i:=i-1;

end

else

if((y<>1) and (com=#80)) then

begin

j:=i;

i:=i+1;

end;

setcolor(blue);

case i of

1:begin bar3d(30,215,150,235,0,topoff); outtextXY(40,220,'√ Информация');end;

2:begin bar3d(40,245,160,265,0,topoff); outtextXY(50,250,'√ Справка');end;

3:begin bar3d(50,275,170,295,0,topoff); outtextXY(60,280,'√ Разработчик');end;

4:begin bar3d(60,305,180,325,0,topoff); outtextXY(70,310,'√ Выход');end;

end;

setcolor(red);

if ((com=#72) or (com=#80)) then

case j of

1:begin

setcolor(yellow);

ar3d(30,215,150,235,0,topoff);

setcolor(red);

outtextXY(40,220,'√ Информация');

end;

2:begin

setcolor(yellow);

bar3d(40,245,160,265,0,topoff);

setcolor(red);

outtextXY(50,250,'√ Справка');

end;

3:begin

setcolor(yellow);

bar3d(50,275,170,295,0,topoff);

setcolor(red);

outtextXY(60,280,'√ Разработчик');

end;

4:begin

setcolor(yellow);

bar3d(60,305,180,325,0,topoff);

setcolor(red);

outtextXY(70,310,'√ Выход');

end;

end;

until(((com=#13) and (i=4))or (com=#13) or(com=#27));

end;

{--------------------КОНЕЦ ПРОЦЕДУРЫ--------------------------}

{|------------------------------------------------------------}

{| Процедура инициализации графики |}

{|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|}

procedure initgr;

var GD,GM:integer;

begin

Gd:=Detect;

InitGraph(Gd, Gm, ' '); { Путь к BGI драйверам }

If GraphResult <> grOk Then

begin

writeln('Ошибка при подключении графики...');

halt(1);

end;

end;

{--------------------КОНЕЦ ПРОЦЕДУРЫ---------------------------}

{|-----------------------------------------------------------|}

{| Процедура прорисовки справочного окна программы |}

{|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|}

procedure help;

begin

cleardevice;

bar3d(30,215,150,235,0,topoff);

outtextXY(40,220,'История');

outtextXY(50,250,'SIN X');

outtextXY(60,280,'COS X');

outtextXY(70,310,'tg X');

outtextXY(80,340,'ctg X');

outtextXY(90,370,'arctg X');

outtextXY(100,400,'arcctg X');

outtextXY(110,430,'Помощь');

setcolor(12);

outtextXY(50,460,'Вверх,Вниз-Движение Enter-Выбор');

com:=#125;

select;

if (com=#13)and(i=1) then

begin

readf('file.txt',false);

help;

end;

if (com=#13)and(i=2) then

begin

readf('edit.txt',false);

select;

end;

if (com=#13)and(i=3) then

begin

readf('search.txt',false);

select;

end;

if (com=#13)and(i=4) then

begin

readf('run.txt',false);

select;

end;

if (com=#13)and(i=5) then

begin

readf('debug.txt',false);

select;

end;

if (com=#13)and(i=6) then

begin

readf('compile.txt',false);

select;

end;

if (com=#13)and(i=7) then

begin

readf('options.txt',false);

select;

end;

if (com=#13)and(i=8) then

begin

readf('help1.txt',false);

select;

end;

end;

{--------------------КОНЕЦ ПРОЦЕДУРЫ---------------------------}

{|------------------------------------------------------------}

{| Процедура прорисовки главного окна программы |}

{|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|}

procedure main;

begin

cleardevice;

{-----ФЛАГ---------}

setfillstyle(1,4);

bar(0,0,640,240);

setfillstyle(1,yellow);

bar(0,210,640,480);

{---КОНЕЦ ФЛАГА----}

{-----------------прорисовка дискеты-----------------------------}

setfillstyle(1,9);

BAR (100,100,200,200);

setfillstyle(1,8);

BAR (105,105,110,110);

BAR (190,105,195,110);

setfillstyle(1,8);

BAR (130,100,170,140);

setfillstyle(1,15);

BAR (120,150,180,180);

setcolor (red);

outtextxy (120,160,'COS+SIN=');

outtextxy (120,190,'SUMSUNG');

{------------------------------------------------------------------}

setcolor (black);

outtextxy (200,120,'---->A R C T G X');

outtextxy (200,140,'---->A R C C T G X');

outtextxy (200,160,'---->A R C S I N X');

outtextxy (200,180,'---->A R C C O S X');

setfillstyle(1,4);

setfillstyle(1,yellow);

setcolor(blue);

bar3d(30,215,150,235,0,topoff);

outtextXY(40,220,'√ Информация');

setcolor(red);

outtextXY(50,250,'√ Справка');

outtextXY(60,280,'√ Разработчик');

outtextXY(70,310,'√ Выход');

SETFILLSTYLE (1,15);

setcolor (black);

settextstyle(9,0,15);

bar3D (0,460,300,477,0,FALSE);

outtextxy (40,465,'Вверх,Вниз-Движение');

bar3D (320,460,638,477,0,FALSE);

outtextxy (400,465,'Enter-Выбор');

setfillstyle(1,yellow);

setcolor(12);

com:=#125;

move;

if (com=#13)and(i=3) then

begin

greator;

eadkey;

main;

end;

if (com=#13)and(i=1) then

begin

help;

main;

end;

if (com=#13)and(i=2) then

begin

readf('help.txt',false);

main;

end;

end;

{--------------------КОНЕЦ ПРОЦЕДУРЫ---------------------------}

begin

initgr;

for y:=1 to 750 do

new(txt[y]);

main;

for y:=1 to 750 do

dispose(txt[y]);

End.

**Приложение Г**

**Экранные формы**

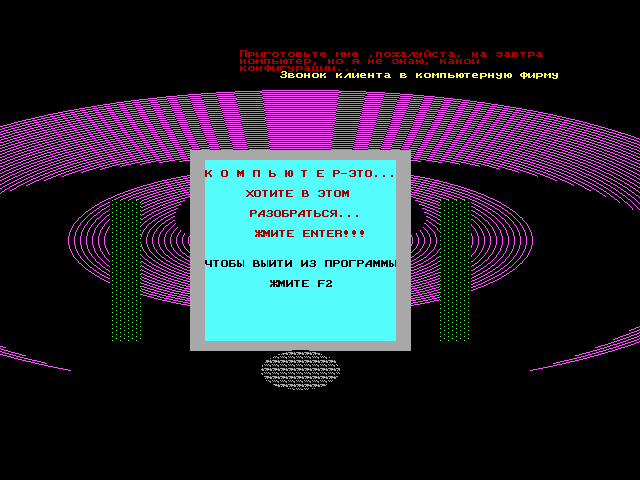


Рисунок Б.1 Экран приветствия



Рисунок Б.2 Вид главного меню

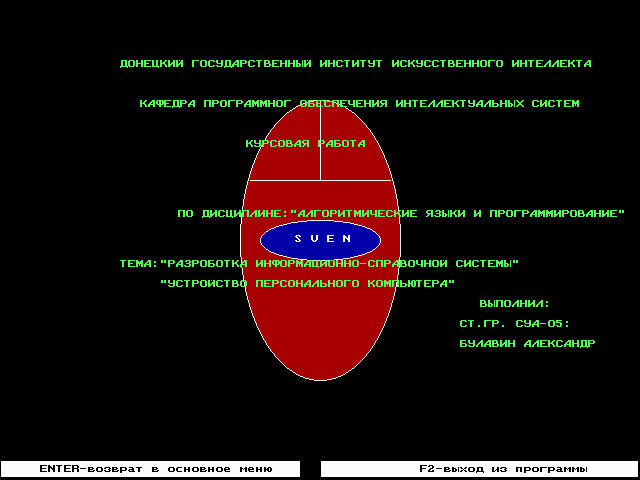


Рисунок Б.3 Вид окна о программе

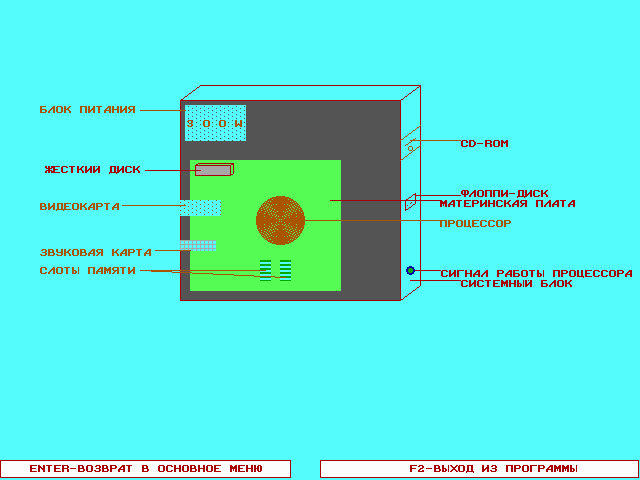


Рисунок Б.4 Вид окна наглядного представления персонального компьютера



Рисунок Б.5 Вид окна информационно-справочного материала