# Обработка экономической информации средствами языка Pascal

Лист задания

**Аннотация**

Пояснительная записка курсовой работы "Обработка экономической информации с помощью файлов" содержит в себе постановку задачи, разработку математической модели, пояснение к алгоритму, пояснение к программе, описание входных, выходных данных. При изучении дисциплины "Основы информатики" для написания курсовой работы использовались различные литературные источники, которые перечислены в соответствующем документе.

В данной курсовой работе приведена программа, которая применяется для вычисления остатка продукции на конец года. Здесь был использован метод структурного программирования для облегчения написания и отладки программы, а также повышения ее наглядности и читаемости. Целью написания данной работы было получение и закрепление практических навыков разработки алгоритмов различными методами .

Представленная программа реализована на языке программирования Borland Pascal версии 7.0.

Язык Borland Pascal 7.0 - это один из самых мощных языков для ЭВМ типа IBM PC/AT, работающий в среде операционной системы DOS, сочетающий в себе как мощность низкоуровневых языков программирования, так и структурированность с надежностью, присущих языкам высокого уровня.

Для успешной работы с представленной программой необходим компьютер IBM 80286 или любой совместимый с ним. Для реализации данного алгоритма был выбран язык программирования - TURBO PASCAL, для ЭВМ типа IВМ РС/АТ/ХТ, работающих в среде операционной системы МS DOS.

Файл с текстом программы (kurs.pas) имеет размер, равный 12,5Kb;

Исполняемый файл (kurs.exe) имеет размер 33,3Kb;

**Содержание**

|  |  |
| --- | --- |
| 1. Введение  2. Постановка задачи  3. Математической модели  4. Описание алгоритма  5. Описание программы  5.1. Общие сведения языка программирования  5.2. Операторы Turbo Pascal  5.3. Входные и выходные данные  5.4. Описание переменных  5.5. Описание программы  6. Список литературы  Приложение А  Приложение Б | 5  7  8  9  11  12  22  23  25  44  45  46 |

**1 Введение**

Информатика - научная дисциплина, изучающая структуру и общие свойства информации, а также закономерности всех процессов обмена ин­формацией при непосредственном устном и письменном общении специали­стов до формальных процессов обмена посредством различных носителей информации. Значительную часть этих процессов составляет научно-инфор­мационная деятельность по сбору, переработке, хранению, поиска и распро­странению информации. Объектом изучения информатики не является со­держание конкретной научно-информационной деятельности, которой должны заниматься специалисты в соответствующих отраслях науки и тех­ники. Она изучает внутренние механизмы реферирования документов на ес­тественных языках, разрабатывает общие методы такого реферирования.

Информатику рассматривают как один из разделов кибернетики, счита­ется, что в последнюю входят проблемы автоматизации информационной службы, перевода и реферирования научно-технической литературы, по­строение информационно-поисковых систем и ряд других задач.

Как было сказано выше информатика входит в состав более общей науки кибернетики, изучающей общую теорию управления и передачи информа­ции. Основное свойство кибернетики заключается в том, что она пригодна для исследования любой системы, которая может записывать, накапливать, обрабатывать информацию, благодаря чему ее можно использовать в целях управления.

Кибернетика - наука об общих законах получения, хранения, передачи и переработки информации в сложных системах. При этом под сложными сис­темами понимаются технические, биологические и социальные системы, поэтому кибернетика нуждалась в мощном инструменте, и этим инструмен­том стали компьютеры.

Информация - это совокупность сведений, определяющих меру наших знаний о тех или иных событиях, явлениях или факторах. Информация явля­ется абстрактной категорией и связана с процессом познания человеком ок­ружающего мира.

Информатика - это наука, которая изучает структуру и особенность ин­формации, а также вопросы, связанные со сбором, хранением и переработкой информации.

Как правило, все числа внутри компьютера представляются с помо­щью нулей и единиц, а не десяти цифр, как это привычно для людей. Иными словами, компьютеры обычно работают в двоичной системе счисления, по­скольку при этом их устройство получается значительно более простым. Ввод чисел в компьютер и вывод их для чтения человеком может осуществляться в привычной для людей десятичной форме.

Для обработки на компьютере текстовой информации обычно при вводе в компьютер каждая буква кодируется определенным числом, а при выводе на внешние устройства ( экран или печать ) для восприятия человеком по этим числам строятся соответствующие изображения букв. Соответствие между набором букв и числами называется кодировкой символов.

**2. Постановка задачи**

Обработка экономической информации, представленной в виде таблицы (см. приложение №1).

Основные данные, входящие в данную таблицу хранить в файле на внешнем носителе информации.

Предусмотреть в данном курсовом проекте возможность вывода таблицы на экран, создание новых записей, добавления новых записей в файл на внешнем носителе, удаление записей из файла, возможность редактирования записи в таблице и графическую заставку для данного курсового проекта с использованием модуля Graph.

Представить вместе с исходным текстом программы, написанной на языке Pascal, пояснительную записку.

**3 Построение математической модели**

Даны исходные данные:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Переходящий остаток продукции на начало года (А1) | Приход продукции,  шт (А2) | Расход, шт  (А3) | Цена за  1 шт  (А4) | Остаток на конец года  (А) |
| 1 | 45 | 32 | 21 | 6.90 | (A′) |
| 2 | 234 | 3454 | 32 | 5.67 | (A′′) |
| … | … | … | … | … | … |
| Итого: As=A′+A′′+…An | | | | | |

Для вычисления окончательных данных в таблице используются следующие формулы:

А=((А1+А2)-А3)\*А4

As=A′+A′′+…An

**4 Описание алгоритма**

Шаг 1. Запрос о запуске графической заставки –(y/n)

Шаг 1.1. При нажатие клавиши выполняется вывод графической заставки на экран

Шаг 1.2. При нажатие любой клавиши (кроме ) выполняется переход к главному меню программы

Шаг 2. Выбор пунктов главного меню

Шаг 3. Выполнение подалгоритма вывода основной таблицы при выборе пункта «1» главного меню.

Шаг 3.1. Печать на экран «шапки» таблицы.

Шаг 3.2. Печать основных данных таблицы.

Шаг 3.3. Возврат к шагу 2 или выход из программы.

Шаг 4. Выполнение подалгоритма создания новых записей при выборе пункта «2» главного меню.

Шаг 4.1. Подтверждение создания новой таблицы.

Шаг 4.2. Ввод количества записей.

Шаг 4.3. Выполнение алгоритма создания новых записей.

Шаг 4..4. Возврат к шагу 2.

Шаг 5. Выполнение подалгоритма добавления записи при выборе пункта «3» главного меню.

Шаг 5.1. Подтверждение добавления записи.

Шаг 5.2. Ввод количества добавляемых записей.

Шаг 5.3. Выполнение алгоритма добавления записей.

Шаг 5.4. Возврат к шагу 2.

Шаг 6. Выполнение подалгоритма удаления записи при выборе пункта «4» главного меню.

Шаг 6.1. Подтверждение операции удаления записи

Шаг 6.2. Выбор удаляемой записи с использованием предварительного просмотра таблицы.

Шаг 6.3. Выполнение алгоритма удаления записи.

Шаг 6.4. При нажатие клавиши - возврат к шагу 6.1, или при нажатие любой клавиши, кроме - к шагу 2

Шаг 7. Выполнения подалгоритма редактирования записи при выборе пункта «5» главного меню.

Шаг 7.1. Подтверждение операции редактирования записи

Шаг 7.2. Выбор редактируемой записи с использованием предварительного просмотра таблицы.

Шаг 7.3. Выполнение алгоритма редактирования записи.

Шаг 7.4. Возврат к шагу 2.

Шаг 8. Выполнение просмотра заставки при выборе пункта «6» главного меню.

Шаг 8.1. Подтверждения запуска заставки

Шаг 8.2. Алгоритм вывода графической заставки на экран

Шаг 9. Выполнение подалгоритма выхода из программы при выборе пункта «7» главного меню.

**5 Описание программы**

***5.1 Общие сведения языка программирования***

Система программирования Турбо Паскаль, разработанная американ­ской корпорацией Borland, остается одной из самых популярных систем программирования в мире, которой по плечу любые задачи от создания простых программ, предназначенных для решения не­сложных вычислительных задач, до разработки сложнейших реляцион­ных систем управления базами данных.

К основным достоинствам языка следует отнести гибкость и надежность, простоту и ясность конструкций, легкость реализации на ЭВМ, возможность полного контроля правильности программ во время компиляции и выполнения, наличие набора структурных типов данных (массивов, файлов и записей), возможность построения новых типов данных.

Версия языка TURBO PASCAL V 7.0. включает в себя возможности:

-использование отдельно компилируемых модулей;

-интегрированный отладчик;

-интегрированная среда разработчика;

-объектно-ориентированную оболочку для создания прикладных программ.

Турбо Паскаль 7.0 оперирует объектами, присущими типичной ЭВМ: символами, целыми числами и числами с плавающей запятой. Программисту разрешено управлять размещением объектов и использовать в программах знания размера объектов и способов их взаимного расположения.

***5.2 Операторы Turbo Pascal.***

В своей простейшей форме программа Borland Pascal состоит из заголовка программы, который именует программу, и основного программного блока, выполняющего назначение программы. В основном программном блоке находится секция кода, заключенная между ключевыми словами begin и end. Простейшая программа, иллюстрирующую эти принципы:

program Privet;

begin

Writeln('Добро пожаловать в Borland Pascal');

end.

Первая строка - это заголовок программы, который именует данную программу. Остальная часть программы - это исходный код, который начинается ключевым словом begin и заканчивается end. Хотя данная конкретная программа содержит только одну строку, их может быть много. В любой программе Borland Pascal все действия выполняются между begin и end.

Исходный код между begin и end содержит операторы, которые описывают выполняемые программой действия. Это называются операторной частью программы.

Приведем примеры операторов:

A := B + C; { присвоить значение }

Calculate(Length, Height); { активизировать процедуру }

if X < 2 then { оператор условия }

Answer := X \* Y;

begin { составной оператор }

X := 3;

Y := 4;

Z := 5;

end;

while not EOF(InFile) do { оператор цикла }

begin

ReadLn(InFile, Line);

Process(Line);

end;

*Условный оператор:*

Позволяет проверить некоторое условие и в зависимости от результатов проверки выполнить то или иное действие. Структура условного оператора имеет следующий вид:

IF THEN elese ;

Вначале вычисляется условное выражение . Если результат есть TRUE, то выполняется , а пропускается; если результат есть FALSE, наоборот, пропускается, а выполняется .

*Счетный оператор цикла FOR:*

FOR := to do

- параметр цикла

- начальное значение

- конечное значение

- произвольный оператор Турбо Паскаля.

При выполнении оператора FOR вначале вычисляется выражение и осуществляется присваивание := .

После этого цикл повторяется:

* Проверка условия <= , если условие не выполнено оператор FOR завершает свою работу;
* Выполнение оператора ;
* Наращивание переменной на единицу.

*Оператор цикла WHILE* с предпроверкой условия:

WHILE DO .

- выражение логического типа;

- произвольный оператор Турбо Паскаля

Если выражение имеет значение TRUE, то выполняется оператор, после чего вычисление выражения и его проверка повторяются. Если имеет значение FALSE, оператор WHILE прекращает работу.

*Оператор цикла REPEAT…UNTIL* с постпроверкой условия:

REPEAT UNTIL .

- произвольная последовательность операторов Турбо Паскаля.

- выражение логического типа.

Операторы выполняются хотя бы один раз, после чего вычисляется выражение : если его значение FALSE, операторы повторяются, в противном случае оператор REPEAT…UNTIL завершает свою работу.

## Оператор выбора

CASE OF [else ] END

- ключ выбора;

- одна или более конструкций вида;

- произвольные операторы ТП.

Вначале вычисляется значение выражения , а затем в последовательности операторов отыскивается такой, которому предшествует константа, равная вычисленному значению. Найденный оператор выполняется, после чего оператор выбора завершает свою работу. Если в списке выбора н будет найдена константа, соответствующая вычисленному значению ключа выбора, управление передается операторам, стоящим за словом else. Часть else можно опускать. Тогда при отсутствии в списке выбора нужной константы ничего не произойдет и оператор выбора просто завершит свою работу.

*Метки и операторы перехода.*

GOTO .

- метка

Метка – это произвольный идентификатор, позволяющий именовать некоторый оператор программы и таким образом ссылаться на него. Описание меток состоит из зарезервированного слова LABEL, за которым следует список меток.

Важнейшей особенностью языка Turbo Pascal является иерархическая структура программ, которая возможна благодаря наличию процедур и функций. Процедуры и функции позволяют разделить логику программы на более мелкие и управляемые фрагменты и аналогичны подпрограммам в других языках. Как и в основном блоке программы, все действия в процедурах и функциях заключаются в begin и end.

Процедуры и функции - главная особенность языка Турбо Паскаль

7.0, обеспечивающая возможность практической реализации принципов

структурного программирования.

*Процедура:*

Procedure (список формальных параметров);

Begin End;

*Функция:*

Function (список формальных параметров>:тип функции;

Begin end;

При написание данного курсового проекта необходимо было воспользоваться структурированные типами данных (массивы, записи, множества, файлы).

*Запись* – это структура данных, состоящая из фиксированного числа компонентов, называемых полями записи. В отличии от массива, компоненты (поля) записи могут быть различного типа. Чтобы можно было ссылаться на тот или иной компонент записи, поля именуются.

Структура объявления типа записи:

= RECORD END

- идентификатор;

- список полей; представляет собой последовательность разделов записи, между которыми ставится точка с запятой. Каждый раздел записи состоит из одного или нескольких идентификаторов полей, отделяемых друг от друга запятыми. За идентификатором (идентификаторами) ставится двоеточие и описание типа поля (полей).

## Файлы. Под файлом понимается либо именованная область внешней памяти ПК либо логического устройства. Любой файл имеет три характерные особенности. Во-первых, у него есть имя, что дает возможность программе работать одновременно с несколькими файлами. Во-вторых, он содержит компоненты одного типа. В-третьих, длина вновь создаваемого файла никак не оговаривается при его объявлении.

# = FILE OF ;

# = TEXT;

# = FILE;

# - имя файлового типа;

TEXT – имя стандартного типа текстовых файлов;

- любой тип ТП, кроме файлов.

В зависимости от способа объявления можно выделить три вида файлов:

* Типизированные файлы (file of…);
* Текстовые файлы (определяются типом TEXT);
* Нетипизированные файлы (определяются типом FILE).

Файловая переменная связывается с именем файла в результате обращения к стандартной процедуре ASSIGN:

ASSIGN (, );

- файловая переменная;

- текстовое выражение, содержащие имя файла или логическое устройство.

В ТП можно открыть файл для чтения, для записи информации, а также для чтения и записи одновременно.

Для чтения файл инициируется с помощью стандартной процедуры RESET:

RESET ();

В ТП разрешается обращаться к типизированным файлам, открытым процедурой RESET с помощью процедуры WRITE (т.е. для записи информации). Такая возможность позволяет легко обновлять ранее созданные типизированные файлы и при необходимости расширять их. Для текстовых файлов, открытых процедурой RESET, нельзя использовать процедуру WRITE.

Стандартная процедура REWRITE () инициирует запись информации в файл или в лог. устройство, связанное ранее с файловой переменной . Но этой процедурой нельзя инициировать запись информации в ранее существовавший дисковый файл: при выполнении этой процедуры старый файл уничтожиться.

Процедура READ.

Обеспечивает чтение очередных компонентов типизированного файла. Формат обращения: READ (, )

- список ввода, содержащий одну или более переменных такого же типа, что и компоненты файла.

Процедура SEEK.

Смещает указатель файла к требуемому компоненту. Формат обращения:

SEEK (, )

- выражение типа Longint.

Процедуру нельзя применять к текстовым файлам.

Функция FileSize.

Возвращает значение, которое содержит количество компонентов файла.

FileSize()

Функция EOF ():Boolean.

Логическая функция, тестирующая конец файла.

##### Модули.

Модуль – это автономно компилируемая программная единица, включающая в себя различные компоненты раздела описаний (типы, константы, переменные, процедуры, функции) и, возможно, некоторые исполняемые операторы инициирующей части.

Связь устанавливается спец. объявлением:

USES - список связываемых модулей.

При написание данного курсового проекта, использовались два дополнительных модулей: GRAPH и CRT.

*Модуль GRAPH*

GRAPH - представляет собой библиотеку подпрограмм, обеспечивающую полное управление графическими режимами для различных адаптеров дисплеев: CGA, EGA, VGA... Содержит 78 процедур и функций (как базовых - рисование точек, линий, окружностей, прямоугольников), так и расширяющих возможности базовых (многоугольники, сектора, дуги, закрашивание фигур, вывода текста и т.д.).

Для запуска программы с использованием модуля GRAPH необходимо,

чтобы в рабочем каталоге находились графические драйверы с расширением \*.BGI, также должен быть доступен компилятору модуль GRAPH.TPU.

В начале программы модуль GRAPH должен быть подключен

USES GRAPH;

InitGraph(driver,mode,’путь к BGI’);

driver – определяет тип графического драйвера,

mode – переменная, задающая режим работы графического адаптера;

*Процедуры и функции модуля GRAPH*

ClearDevice - Очищает экран и помещает курсор в точку (0,0);

SetViewPort(x1,y1,x2,y2:integer; Clip: boolean) - Устанавливает текущее окно для графического вывода. X1,y1 – координата верхнего левого угла; x2,y2 - координата нижнего правого угла. Если Clip=true, то все изображения отсекаются на границах вывода;

Closegraph – закрывает графический режим

ClearViewPort - Очищает текущее окно;

GetMaxX: integer - возвращает максимальную горизонтальную координату графического экрана;

GetMaxY :integer - возвращает максимальную вертикальную координату графического экрана;

GetX: integer - возвращает координату Х текущего указателя в окне;

GetY: integer - возвращает координату Y курсора в окне;

SetLineStyle(Line, Pattern, Thickness: word) - Устанавливает стиль (0..4), шаблон штриховки (0..12) и толщину (1-ноpм, 3-утpоенная);

SetFillStyle(Pattern, Color: word) - Устанавливает образец штриховки и цвет (0..15 и 128-меpцание);

SetGraphMode(Mode: integer) - Устанавливает новый графический режим и очищает экран;

SetColor(Color: word) - Устанавливает основной цвет, которым выполняется рисование (0..15);

SetBkColor(Color: word) - Установка цвета фона.

PutPixel(X,Y: integer; Color: word) - Выводит точку цветом Color с координатой X,Y;

LineTO(X, Y: integer) - Рисует линию от текущего указателя к точке с координатой Х,У;

LineRel(DX, DY: integer) - Рисует линию от текущего указателя к точке, заданной приращением координат;

Line(X1,Y1,X2,Y2:integer) - Рисует линию от точки (X1,Y1) к точке с координатой Х2,У2;

MoveTO(X, Y: integer) - Смещает текущий указатель к точке с координатой Х,У;

MoveRel(DX, DY: integer) - Смещает текущий указатель к точке, заданной приращением координат;

Rectangle(X1,Y1,X2,Y2:integer) - Рисует прямоугольник, используя текущий цвет и тип линии по верхней левой и нижней правой точкам;

Bar(X1,Y1,X2,Y2:integer) - Рисует закрашенный прямоугольник, используя установку SetFillStyle;

Bar3D(X1,Y1,X2,Y2:integer;Depth:word;Top:Bolean) - Рисует закрашенный паpаллелипипед. Depth - глубина в Pixel (1/4 ширины). Если Тор=True, то рисуется верхняя грань пеpеллелипипеда;

Circle(X,Y:integer;R:word) - Рисует окружность радиуса R, используя X,Y как координаты центра;

Fillellipse(X,Y: integer; XR,YR: word) - Рисует защтpихованный эллипс, используя X,Y как центр и XR,YR как горизонтальный и вертикальный радиусы.

RestorCRTMode - Восстанавливает текстовый режим работы экрана;

OutText(Text: string) - Выводит текстовую строку на экран.

OutTextXY(X,Y: integer; Text: string) - Выводит текст в заданное место экрана.

*Модуль CRT*

Включает процедуры и функции, которые управляют текстовым режимом работы дисплея. С помощью подпрограмм модуля можно перемещать курсор по экрану дисплея, менять цвет выводимых символов и фона, создавать окна на экране, управлять звуком, работать с клавиатурой.

*Функции управления клавиатурой:*

KeyPressed: Boolean; - возвращает True, если в текущий момент на клавиатуре была нажата клавиша, иначе False. Не приостанавливает выполнение программы. Используется при организации работы циклов.

ReadKey: Char; - читает и возвращает в программу символ с клавиатуры (без отображения на экране). Приостанавливает исполнение программы до нажатия на любую клавишу символа. Используется в операторах IF для проверки символов.

WhereX: Byte; - возвращает горизонтальную координату текущей позиции курсора относительно окна.

WhereY: Byte; - возвращает вертикальную координату текущей позиции курсора относительно окна.

*Процедуры управления дисплеем*

ClrEol; - удаляет все символы справа от курсора до конца строки без перемещения курсора.

ClrScr; - очищает экран (окно на экране) и помещает курсор в верхний левый угол.

Delay(D: word); - приостанавливает работу программы на D миллисекунд.

GotoXY(X,Y: byte); - перемещает курсор в нужное место экрана (окна). Левый верхний угол экрана (окна) имеет координаты (1,1). Отсчет координат идет слева направо и сверху вниз. Количество символов в строке и самих строк зависит от типа дисплея (обычно 25 строк и 80 символов).

InsLine; - вставляет пустую строку в позицию курсора.

HighVideo; - устанавливает высокую яркость символов.

LowVideo; - устанавливает низкую яркость символов.

NormVideo; - устанавливает нормальную яркость символов.

Window(X1,Y1,X2,Y2:byte); - определяет размеры окна на экране для вывода текста. X1,Y1 - координаты левого верхнего угла, X2,Y2 - правого нижнего угла.

TextBackGround(Color: byte); - устанавливает цвет фона экрана (окна) от 0 (черный) до 15 (белый).

TextColor(Color: byte); - устанавливает цвет символа.

TextMode(Mode: word); - устанавливает нужный текстовый режим:0,1,2...

*Управление звуковым генератором*

Sound(F: word); - включает звук генератора, F - частота звука в герцах.

Delay(T: word); - устанавливает продолжительность звучания в миллисекундах.

NoSound; - выключает звук генератора.

***5.3 Входные и выходные данные***

Входными данными являются записи: номер продукции (n), переходящий остаток продукции на начало года (e1), приход продукции (e2), расход (e3), цена за 1 шт (e4). При завершении ввода данных и выборе соответствующего пункта меню программы, программа выдаст пользователю итоговые данные в виде таблицы.

Выходными данными, т.е. результатом программы является таблица которая содержит исходные записи и следующие переменные:

* Остаток на конец года (st);
* Итого (itog) ;

Файл с текстом программы (kurs.pas) имеет размер, равный 12,5 Kb;

Исполняемый файл (kurs.exe) имеет размер 33,3 Kb;

***5.4 Описание переменных***

В программе используются следующие переменные:

**Bas** – запись.

type

bas=record

***n***:integer;

***e1,e2,e3***:integer;

***e4***:real;

end;

Где:

n - номер продукции;

e1 - переходящий остаток продукции на начало года ;

e2 - приход продукции;

e3 – расход;

e4 - цена за 1 шт .

**menu2** – метка, используется для работы с главным меню программы;

**C**:char, используется для работы оператора readkay (читает символ с клавиатуры без эхоповтора на экране);

**O**:real используется для редактирования записей.

**ww, l, nn, q, i, gd, gm**:integer;

ww-используется для вывода шапки таблицы (координата по Y).

l-используется для работы с меню и для удаления записи

nn, q- используется в программе для создание новых записей

i-счетчик

gd, gm - для графики:

gd – определяет тип графического драйвера,

gm – переменная, задающая режим работы графического адаптера;

**pp,qq,n,mm**:integer; - переменные для добавления новой записи;

**f,f2**:file of bas; - файлы

**a,b**:bas;

a,b - переменные, содержащие записи типа Bas.

Переменная b используется для удаления записей.

**st**:real;

st – для вычисления суммы в таблице

*Процедуры:*

procedure **view**(w:integer);

- процедура вывода таблицы, где w–координата по Х;

var

***itog:***real; -для вычисления итогового значения сумма в таблице;

Procedure **menu**;

- процедура вывода меню;

Procedure **zast**;

- процедура вывода графической заставки

var

***xm,ym***:integer;

xm,ym-координаты для вывода граф. объектов

***size***:integer; ***saucer***:pointer;

***k,kk***:integer;

size,saucer,k,kk-для движения

Procedure **error**;

- процедура вывода сообщения об ошибке

Procedure **exitt**;

- процедура выхода из программы

***5.5 Описание программы***

При запуски программы идет подключение дополнительных модулей Crt и Graph:

uses crt, graph;

Описание переменных (см. пункт 5.4)

Описание процедур:

*Процедура вывода таблицы на экран* - **view**:

Установка текстового окна на экране для вывода шапки с координатами левого верхнего угла -1,1 и правого нижнего – 80,6

window(1,1,80,6);

Установка цвета фона и символов

textbackground(5);textcolor(15);

Печать названия таблицы

gotoxy(w,1);write('Реализация продукции за год');

textcolor(9);gotoxy(w-16,2);textbackground(0);

write('--------------------------------------------------------------');

Установка цвета символов и печать шапки таблицы

textcolor(15);

gotoxy(w-15,4);write('N');

gotoxy(w-13,3);write('Переходящий остаток');

gotoxy(w-10,4);write('продукции на ');

gotoxy(w-10,5);write('начало года');

gotoxy(w+9,3);write('Приход');

gotoxy(w+8,4);write('продукции');

gotoxy(w+11,5);write('шт');

gotoxy(w+18,4);write('Расход');

gotoxy(W+20,5);write('шт');

gotoxy(w+26,4);write('Цена за');

gotoxy(w+27,5);write('1 шт');

gotoxy(w+35,4);write('Остаток на');

gotoxy(w+35,5);writeln('конец года');

textcolor(9); gotoxy(w-16,6);

write('--------------------------------------------------------------');

Установка цвета символов и текстового экрана для вывода данных таблицы

textcolor(15);

window(1,7,80,25);

Подготовка файла к чтению

reset(f);

Установка координаты y, необходимой для вывода данных таблицы и обнуление итогового значения таблицы.

ww:=1; itog:=0;

Вывод основных данных таблицы

repeat

writeln; read(f,a);

st:=((a.e1+a.e2)-a.e3)\*a.e4;

itog:=itog+st;

textcolor(13);

gotoxy(w-15,ww);write(a.n);

textcolor(15);

gotoxy(w-5,ww);write(a.e1);

gotoxy(w+11,ww);write(a.e2);

gotoxy(w+20,ww);write(a.e3);

gotoxy(w+29,ww);write(a.e4:4:2);

gotoxy(w+37,ww);write(st:4:2);

ww:=ww+1;

Проверка для вывода очередной строки на экран. Необходимо в том случае, когда число записей превышает число строк на мониторе.

if ww>=19 then

begin

if (ww=19) then

begin

writeln;

write('Для просмотра используйте клавишу ');

write(' Для продолжения -');

c:=readkey;

delline;

if c=#32 then ww:=ww-1 else break;

end;

end;

Вывод данных в таблицу пока не конец файла

until eof(f);

Установка цвета символа и вывод линии на экран

textcolor(9); gotoxy(w-16,ww);

write('--------------------------------------------------------------');

Вывод итогового значения

textcolor(9);gotoxy(w+7,ww);

write('¦'); textbackground(5);

textcolor(10);write('Итого:',itog:4:2);

textbackground(0); textcolor(9);

write('+'); writeln;

Завершение процедуры вывода таблицы

end;

*Процедура вывода основного меню* – **menu**:

Очистка экрана и установка стандартного текстового режима

clrscr; textmode(Co80);

Установка текстового окна на экране с координатами левого верхнего угла -13,3 и правого нижнего – 70,15

window(13,3,70,15);

Установка цвета фона и символов

textbackground(14);textcolor(15);

Перемещение указателя и печать сообщения

gotoxy(1,3); write('Для дальнейшей работы с программой выберите номерок:');

Установка цвета фона

textbackground(0);

Печать пунктов меню, устанавливая цвет для цифр меню и свой цвет для названия пункта меню

gotoxy(16,5); textcolor(10); write('1'); textcolor(7); write(' - Просмотр');

gotoxy(16,6); textcolor(10); write('2'); textcolor(7); write(' - Cоздать');

gotoxy(16,7); textcolor(10); write('3'); textcolor(7); write(' - Добавить');

gotoxy(16,8); textcolor(10); write('4'); textcolor(7); write(' - Удалить');

gotoxy(16,9); textcolor(10); write('5'); textcolor(7); write(' -Редактировать');

gotoxy(16,10);textcolor(10); write('6'); textcolor(7); write(' - Заставка');

gotoxy(16,11);textcolor(10); write('7'); textcolor(7); write(' - Выход');

gotoxy(16,13); write('Введите номер:');

Установка цвета символа и его мерцание

textcolor(10+128);

Ввод переменной l, необходимой для дальнейшей работы программы и установка стандартного текстового режима

readln(l); textmode(Co80);

Конец процедура вывода главного меню

end;

*Процедура вывода заставки* – **zast**:

Очистка экрана

clrscr;

Установка белого цвета символов

textcolor(white);

Вывод сообщения: «Показать заставку?»

Write(‘Показать заставку? (y/n)’);

Считывание нажатия клавиши.

c:=readkey

При нажатие клавиши - выполняется дальнейший ход процедуры:

Инициализация графики

initgraph(gd,gm,'bgi')

Установка координат xm, ym для вывода графических объектов

xm:=getmaxx div 4; ym:=getmaxy div 4;

Установка текущего шрифта, стиля и размера текста

settextstyle(2,0,5)

Вывод текста на экран, используя координаты x=xm+100,y=10

outtextxy(xm+100,10,'## Sergei Kovbasyuk ##');

Вычесление числа байт, необходимого для сохранения прямоугольной области экрана

size:=imagesize (xm+100,10,xm+275,10+20);

Создание динамической переменной

getmem(saucer,size);

Сохранение в переменной битовый образ указанной части экрана.

getimage(xm+100,10,xm+275,10+20,saucer^);

Вывод битового образа на экран

putimage(xm+100,10,saucer^,xorput);

Обнуление переменных, требуемых для вывода на экран битового изображения, используя при этом оператор цикла REPEAT…UNTIL и цикл FOR.

k:=0; kk:=0; i:=0; n:=0;

Цикла REPEAT…UNTIL

repeat

Установка текущего шрифта, стиля и размера текста

settextstyle(10,0,5);

Установка текущего цвета шрифта

setcolor(i+1);

Вывод текста на экран

outtextxy((getmaxx div 5),(getmaxy div 4),'"Sorge pro"');

Установка текущего шрифта, стиля, размера текста.

settextstyle(5,0,5); setcolor(15);

Вывод текста на экран

outtextxy((getmaxx-200),(getmaxy-250),'present');

Установка текущего шрифта, стиля, размера текста.

settextstyle(3,0,1);

Вывод текста на экран

outtextxy(40,(getmaxy-50),'http://sorgepro.agava.ru');

outtextxy(290,(getmaxy-50),'e-mail: office@sorgepro.agava.ru');

Установка текущего шрифта, стиля, размера текста.

settextstyle(2,0,5);

Вывод текста на экран

outtextxy(getmaxx-150,getmaxy-20,'Press ');

outtextxy(20,getmaxy-20,'(c)2000. All rights don''t reserved.');

Наращивание переменных для вывода на экран битового изображения и для смены текущего цвета

k:=k+5;i:=i+1;

Вывод битового образа на экран

putimage(k,10,saucer^,xorput);

Задержка программы

delay(50);

Наращивание переменной для вывода на экран битового изображения

kk:=kk+5;

Вывод битового образа на экран

putimage(kk,10,saucer^,xorput);

Если битовое изображение достигло заданных границ экрана, то идет обнуление координат вывода этого изображения на экран

if k>getmaxx-100 then begin k:=0; kk:=0 end;

Выполнение этого цикла, пока не нажата какая-нибудь клавиша

until keypressed;

Вывод на экран трех окружностей, меняющих свой размер с помощью наращивания переменной *.*

*{-----------krugi------------}*

*for i:=0 to 399 do*

*begin*

*setcolor(red);*

*circle(200,100,i);*

*setcolor(green);*

*circle(400,100,i);*

*setcolor(yellow);*

*circle(300,300,i);*

*delay(10);*

*end;*

*Вывод на экран окружности, уменьшающей свой радиус.*

*for i:=399 downto 0 do*

*begin*

*setcolor(i);*

*сircle(322,250,i);*

*delay(10);*

*end;*

*{----------end\_krugi---------}*

*Закрытие граф. режима*

*closegraph;*

*Завершение составного оператора begin…end*

*end;*

*Завершение процедуры вывода заставки*

*end;*

*Процедура вывода сообщения об ошибке –* ***error****:*

*begin*

*Обнуление счетчика*

*i:=0;*

*Цикл вывода сообщения об ошибке, пока i не равно 5*

*repeat*

*textmode(Co80);*

*clrscr;*

*gotoxy(24,8);textcolor(15);*

*write('ВЫ ВВЕЛИ НЕПРАВИЛЬНЫЙ НОМЕР');*

*звук, при выводе сообщения*

*sound(185);delay(100);nosound;*

*clrscr;*

*gotoxy(24,8);textcolor(7);*

*write('ВЫ ВВЕЛИ НЕПРАВИЛЬНЫЙ НОМЕР');*

*sound(174);delay(100);nosound;*

*наращивание счетчика*

*i:=i+1;*

*until i=5;*

*Отключение звука*

*nosound;*

*Возврат в главное меню*

*menu;*

*Завершение процедуры вывода сообщения об ошибке*

*end;*

*Процедура выхода из программы –* ***exitt****:*

*Очитстка экрана*

*clrscr;*

*Инициализация графики*

*initgraph(gd,gm,'bgi');*

*Вывод текста на экран пока не нажата какая-нибудь клавиша*

*repeat*

*settextstyle(7,0,4);setcolor(15);*

*outtextxy(90,getmaxy div 4, 'Technical support by e-mail:');*

*settextstyle(1,0,4);setcolor(10);*

*outtextxy(120,(getmaxy div 4)+50, 'support@sorgepro.agava.ru');*

*settextstyle(7,0,4);setcolor(15);*

*outtextxy(140,(getmaxy div 4)+100, 'or visited our web site:');*

*settextstyle(1,0,4);setcolor(10);*

*outtextxy(120,(getmaxy div 4)+150, 'http://sorgepro.agava.ru');*

*settextstyle(2,0,5);setcolor(15);*

*outtextxy(getmaxx div 4,getmaxy-30,'(c)2000."Sorge pro".All rights don''t reserved.');*

*until keypressed;*

*Закрытие граф. режима*

*closegraph;*

*Выход из программы*

*exit;*

*Завершение процедуры*

*end;*

*{==========================Proga===========================}*

*Идет выполнение процедуры Zast:*

*при нажатие какой-либо другой клавиши – устанавливается связь с файлом*

*assign (f,’'путь к файлу ’)*

*Установка стандартного текстового режима*

*textmode(Co80);*

*Печать главного меню, используя процедуру menu*

*Установка метки*

*menu2:*

*Если l=1, то выполняется печать таблицы*

*if l=1 then*

*begin*

*Печать таблицы с помощью процедуры view, где w=27;*

*Вывод сообщения для пользователя после вывода таблицы*

*textcolor(15);*

*gotoxy(7, ww+1);*

*writeln('Для выхода нажмите или -возврат в главное меню. ');*

*Считывание нажатия клавиши с клавиатуры*

*c:=readkey;*

*Установка стандартного текстового режима*

*textmode(Co80);*

*Если нажата клавиша происходит возврат в главное меню: выполнение процедуры menu и переход к метки menu2*

*if c=#27 then begin menu; goto menu2 end*

*Иначе выполняется процедура выхода – exitt*

*else begin exitt;*

*Завершение составного оператора begin…end*

*end*

*Если l не равно 1, а равно 2 то выполняется алгоритм создания записей заново*

*else*

*if l=2 then*

*begin*

*Очистка экрана*

*clrscr;*

*Установка цвета фона и символов и вывод на экран сообщения*

*textbackground(14);textcolor(15);*

*writeln('Вы действительно хотите создать новые записи?');*

*writeln('!!!ПРЕДЫДУЩИЕ ЗАПИСИ БУДУТ УТЕРЯНЫ!!!');*

*Установка цвета фона и символов и вывод на экран сообщения*

*textbackground(0);*

*gotoxy(1,4);*

*writeln('Для продолжения нажмите или -возврат в главное меню.');*

*Считывание клавиши с клавиатуры*

*c:=readkey;*

*Если нажата клавиша , то выполняется возврат к главному меню и переход к метки menu2*

*if c=#27 then begin menu; goto menu2 end*

*Иначе идет ввод переменной q – количество создаваемых записей*

*else*

*begin*

*writeln('Введите количество записей:');*

*read(q);*

*Очистка экрана*

*clrscr;*

*Подготовка файла к записи*

*rewrite(f);*

*Цикл ввода новых записей. Выход из цикла осуществиться в том случае, когда a.n будет равно q*

*repeat*

*writeln('введи текущий номер продукции...'); readln(a.n);*

*writeln('введи переходящий остаток продукции на начало года...');*

*readln(a.e1);*

*writeln('введи приход продукции...'); readln(a.e2);*

*writeln('введи расход...');readln(a.e3);*

*writeln('введи цену за 1 шт...');readln(a.e4);*

*write(f,a);*

*until (a.n=q);*

*Очистка экрана и вывод сообщения*

*clrscr;*

*writeln('ВСЁ OK.');*

*Задержка программы. Возврат в главное меню и переход к метки menu2*

*delay(1000);*

*menu; goto menu2;*

*Завершение составного оператора begin…end*

*end;*

*Завершение составного оператора begin…end*

*end*

*Если переменная l не равна 1 и 2, а равна 3, то выполняется алгоритм добавления записи*

*else*

*if l=3 then*

*begin*

*Очистка экрана, установка цвета фона и символов и вывод сообщения*

*clrscr;*

*textbackground(0);*

*textcolor(15);*

*writeln('Продолжить? (y/n)');*

*Считывание клавиши с клавиатуры*

*c:=readkey;*

*Если нажата любая клавиша, кроме , то выполняется возврат к главному меню и переход к метки menu2*

*if c<>#121 then begin menu;goto menu2; end*

*Иначе выполняется алгоритм добавления новой записи*

*else*

*begin*

*очистка экрана*

*clrscr;*

*Ввод количества добовляемых записей -mm*

*writeln('Введите количество добовляемых записей:');*

*read(mm);*

*очистка экрана*

*clrscr;*

*Подготовка файла к чтению, определения конца файла и установка указателя в конец файла*

*reset(f); n:=filesize(f); seek(f,n);*

*установка номера добовляемой записи*

*pp:=n+1;*

*Цикл добавления записи. Завершится тогда, когда номер последней добавляемой записи qq будет равен mm*

*repeat*

*writeln('введи текущий номер продукции...');readln(qq); a.n:=pp;*

*writeln('введи переходящий остаток продукции на начало года...');*

*readln(a.e1);*

*writeln('введи приход продукции...'); readln(a.e2);*

*writeln('введи расход...');readln(a.e3);*

*writeln('введи цену за 1 шт...');readln(a.e4);*

*write(f,a);*

*pp:=pp+1;*

*until (qq=mm);*

*Очистка экрана и вывод сообщения*

*clrscr;*

*writeln('ВСЁ OK.');*

*Задержка программы*

*delay(1000);*

*Возврат в главное меню и переход к метки menu2*

*menu; goto menu2;*

*Завершение составного оператора begin…end*

*end;*

*Завершение составного оператора begin…end*

*end*

*else*

*Если l не равно 1, не 2 и не 3, а равно 4, то выполняется алгоритм удаления записей*

*if l=4 then*

*begin*

*Очистка экрана*

*clrscr;*

*Установка текстового экрана для вывода таблицы с помощью процедуры view*

*window(1,1,68,22);*

*view(17);*

*Установка текстового экрана для управления удаления записей*

*window(64,1,80,22);*

*Установка цвета символов и вывод сообщения*

*textcolor(15);*

*writeln('Продолжить? (y/n)');*

*Считывание клавиши с клавиатуры*

*c:=readkey;*

*Если нажата любая клавиша, кроме , то выполняется возврат к стандартному текстовому режиму, к главному меню и переход к метки menu2*

*if c<>#121 then begin textmode(Co80); menu; goto menu2; end*

*Иначе происходит ввод удаляемой записи - nn*

*else*

*begin*

*write('Введи номер:');*

*readln(nn);*

*Если вводимого номера записи не существует – выполняется процедура вывода сообщения об ошибке – error, и переход к метки menu2*

*if nn>filesize(f) then begin error; goto menu2; end;*

*Удаление записи: запись в новый файл:*

*Подготовка файла к чтению*

*reset(f);*

*Установка связь с новым файлом*

*assign (f2,'k2');*

*Подготовка нового файла к записи*

*rewrite(f2);*

*Обнуление переменных, необходимых для удаления записи*

*l:=0;q:=0;*

*Определение количества записей в первоначальном файле*

*n:=filesize(f);*

*Цикл перезаписи записей из первого файла в новый, при этом удаляемая запись игнорируется. Цикл завершается, когда все записи будут переписаны*

*repeat*

*l:=l+1;*

*if l<>nn then*

*begin*

*q:=q+1;*

*seek(f,l-1);*

*read(f,a);*

*b.n:=q; b.e1:=a.e1; b.e2:=a.e2; b.e3:=a.e3; b.e4:=a.e4;*

*write(f2,b);*

*end;*

*until l=n;*

*Подготовка только что созданного файла к чтению.*

*reset(f2);*

*Подготовка первоначального файла к перезаписи*

*rewrite(f);*

*Цикл перезаписи из только что созданного файла записей в первоначальный, но уже без требуемой записи. Перезапись будет производиться, пока не конец второго файла*

*repeat*

*read(f2,b);*

*a.n:=b.n; a.e1:=b.e1; a.e2:=b.e2; a.e3:=b.e3; a.e4:=b.e4;*

*write(f,a);*

*until eof(f2);*

*Вывод сообщения и задержка программы*

*writeln;*

*writeln('Всё Ok!');*

*delay(1000);*

*Вывод сообщения о возможности продолжения удаления новых записей или возврата к главному меню*

*writeln('Reload-');*

*write('Exit-');*

*Считывание клавиши с клавиатуры*

*c:=readkey;*

*Если нажата клавиши , то происходит возврат к главному меню и переход к метки menu2*

*if c=#27 then begin menu; goto menu2; end*

*иначе устанавливается стандартный текстовой режим, l присваивается 4 и осуществляется переход к метке menu2, т.е. происходит опять алгоритм удаления записи*

*else begin textmode(Co80); l:=4; goto menu2; end;*

*Завершение составных операторов begin…end*

*end;*

*end*

*Если l не равно 1, не 2 и не 3, и не 4, а равно 5, то выполняется алгоритм редактирования записей*

*else*

*if l=5 then*

*begin*

*Выполняется процедура вывода таблицы*

*view(27);*

*Установка цвета символов и вывод сообщения*

*textcolor(15);*

*writeln('Продолжить? (y/n)');*

*Считывание клавиши с клавиатуры*

*c:=readkey;*

*Если нажата любая клавиша, кроме , то выполняется возврат к стандартному текстовому режиму, к главному меню и переход к метки menu2*

*if c<>#121 then begin textmode(Co80); menu;goto menu2; end*

*Иначе происходит ввод номера записи для редактирования - nn*

*else*

*begin*

*write('Введите номер для редактирования:'); read(nn);*

*Если вводимого номера записи не существует – выполняется процедура вывода сообщения об ошибке – error, и переход к метки menu2*

*if nn>filesize(f) then begin error; goto menu2; end;*

*Очистка экрана и установка переменной i, для предварительного просмотра записи*

*clrscr;*

*i:=27;*

*Подготовка файла к чтению и записи и установка указателя на нужной редактируемой записи*

*reset(f); seek(f,nn-1);*

*Считывание и вывод на экран редактируемой записи для предварительного просмотра*

*read(f,a);*

*st:=((a.e1+a.e2)-a.e3)\*a.e4;*

*gotoXY(i-15,1);write(a.n);*

*gotoXY(i-5,1);write(a.e1);*

*gotoxy(i+11,1);write(a.e2);*

*gotoxy(i+20,1);write(a.e3);*

*gotoxy(i+29,1);write(a.e4:4:2);*

*gotoxy(i+37,1);write(st:4:2);*

*gotoxy(7,2);*

*Установка указателя на необходимой записи, т.к. после вывода на экран указатель сдвинулся*

*seek(f,nn-1);*

*Вывод сообщений о подтверждении редактирования каждого компонента редактируемой записи.*

*Если при этом будет нажата , то выполняется перезапись компонента записи, путем ввода новой (nn) и присваивание ее –старой (a.e1…a.e4).*

*Если при выводе сообщения будет нажата любая клавиши, кроме , то выполняется переход к редактированию следующего компонента данной записи.*

*writeln('Изменить переходящий остаток продукции на начало года?*

*(y/n)');*

*c:=readkey;*

*if c=#121 then begin write('Вводим новое:');read(nn); a.e1:=nn; end;*

*writeln;*

*writeln('Изменить приход продукции? (y/n)');c:=readkey;*

*if c=#121 then begin write('Вводим новое:'); read(nn);a.e2:=nn; end;*

*writeln;*

*writeln('Изменить расход? (y/n)');c:=readkey;*

*if c=#121 then begin write('Вводим новое:'); read(nn);a.e3:=nn; end;*

*writeln;*

*writeln('Изменить цену за 1 шт? (y/n)');c:=readkey;*

*if c=#121 then begin write('Вводим новое:'); read(o);a.e4:=o; end;*

*write(f,a);*

*Вывод сообщение и задержка программы*

*writeln; write('Все Ok'); delay(1000);*

*Очистка экрана, возврат к главному меню и переход к метке menu2*

*clrscr; menu;*

*goto menu2;*

*Завершение составных операторов begin…end*

*end;*

*end*

*else*

*Если l не равно 1, не 2 и не 3, и не 4, и не 5, а равно 6, то выполняется процедура вывода заставки - zast, после завершения которой происходит переход к главному меню и метке menu2*

*if l=6 then begin zast; menu; goto menu2; end;*

*Если l не равно 1, не 2 и не 3, и не 4, не 5, и даже не 6, а - 7, то выполняется процедура выхода из программы –exitt*

*if l=7 then exitt*

*Если l не равно 1, не 2 и не 3, и не 4, не 5, и даже не 6, и уж точно не 7, то выполняется процедура вывода сообщения об ошибке. После завершения этой процедуры происходит переход к метке menu2*

*else*

*begin error; goto menu2; end;*

*{========================END\_Proga========================}*

*ЗАВЕРШЕНИЕ ПРОГРАММЫ*

*END.*

***6 Список литературы***

*1. Фаронов В.В. Турбо Паскаль 7.0. Начальный курс. Учебное пособие.-М.: «Нолидж», 1998.-616 с.*

*2. Вычислительная техника и программирование. Под редакцией А. В. Петрова М, Высшая школа, 1990*

*3. Марченко А.И., Марченко Л.И. Программирование в среде Turbo-Pascal 7.0-М., Бином Универсал, К.:Юниор,1997*

*4. http://www.rusdoc.ru*

*5. http://www.borland.ru*

*6. http://www.infocity.kiev.ua*

***Приложение А***

***Реализация продукции за год.***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Переходящий остаток продукции на начало года | Приход продукции,  Шт | Расход, шт | Цена за  1 шт | Остаток на конец года |
| 1 | 45 | 32 | 21 | 6.90 | 386.40 |
| 2 | 234 | 3454 | 32 | 5.67 | 20729.52 |
| 3 | 79 | 345 | 34 | 3.99 | 1556.10 |
| 4 | 234 | 4567 | 54 | 9.30 | 44147.10 |
| 5 | 500 | 566 | 865 | 4.98 | 1000.98 |
| 6 | 5867 | 11 | 543 | 4.67 | 24914.45 |
| 7 | 560 | 7789 | 55 | 2.90 | 24052.60 |
| 8 | 129 | 2000 | 234 | 4.89 | 9266.55 |
| 9 | 123 | 567 | 435 | 7.90 | 2014.50 |
| 10 | 875 | 907 | 800 | 8.90 | 8739.80 |
| 11 | 743 | 800 | 600 | 2.67 | 2517.81 |
| 12 | 456 | 800 | 560 | 6.89 | 4795.44 |
| Итого: 144121.25 | | | | | |

***Приложение Б***

*uses crt,graph;*

*type*

*bas=record*

*n:integer;*

*e1,e2,e3:integer;*

*e4:real;*

*end;*

*label menu2; {метка}*

*var*

*c:char; {с - для ридкея}*

*o:real;*

*ww,l,nn,q,i:integer; {w,ww-для шапки,l-для меню и для удал*

*nn,o-редактирование q-для обнавления*

*i-счетчик}*

*gd,gm:integer; {для графики}*

*pp,qq,n,mm:integer; {для добавки}*

*f,f2:file of bas; {файлы}*

*a,b:bas;*

*st:real; {сумма в таблице}*

*{-----------------------------------------вывод таблицы -------------------------------------}*

*procedure view(w:integer);*

*var*

*itog:real; {cумма в таблице, всего}*

*begin*

*window(1,1,80,6); textbackground(5);textcolor(15);*

*gotoxy(w,1);write('Реализация продукции за год');*

*textcolor(9);gotoxy(w-16,2);textbackground(0);*

*write('--------------------------------------------------------------');*

*textcolor(15);*

*gotoxy(w-15,4);write('N');*

*gotoxy(w-13,3);write('Переходящий остаток');*

*gotoxy(w-10,4);write('продукции на ');*

*gotoxy(w-10,5);write('начало года');*

*gotoxy(w+9,3);write('Приход');*

*gotoxy(w+8,4);write('продукции');*

*gotoxy(w+11,5);write('шт');*

*gotoxy(w+18,4);write('Расход');*

*gotoxy(W+20,5);write('шт');*

*gotoxy(w+26,4);write('Цена за');*

*gotoxy(w+27,5);write('1 шт');*

*gotoxy(w+35,4);write('Остаток на');*

*gotoxy(w+35,5);writeln('конец года');*

*textcolor(9); gotoxy(w-16,6);*

*write('--------------------------------------------------------------');*

*textcolor(15);*

*window(1,7,80,25);*

*reset(f); ww:=1; itog:=0;*

*repeat*

*writeln; read(f,a);*

*st:=((a.e1+a.e2)-a.e3)\*a.e4;*

*itog:=itog+st;*

*textcolor(13);*

*gotoxy(w-15,ww);write(a.n);*

*textcolor(15);*

*gotoxy(w-5,ww);write(a.e1);*

*gotoxy(w+11,ww);write(a.e2);*

*gotoxy(w+20,ww);write(a.e3);*

*gotoxy(w+29,ww);write(a.e4:4:2);*

*gotoxy(w+37,ww);write(st:4:2);*

*ww:=ww+1;*

*if ww>=19 then {типа скролинга}*

*begin*

*if (ww=19) then*

*begin*

*writeln;*

*write('Для просмотра используйте клавишу ');*

*write(' Для продолжения -');*

*c:=readkey;*

*delline;*

*if c=#32 then ww:=ww-1 else break;*

*end;*

*end;*

*until eof(f);*

*textcolor(9); gotoxy(w-16,ww);*

*write('--------------------------------------------------------------');*

*textcolor(9);gotoxy(w+7,ww);*

*write('¦'); textbackground(5);*

*textcolor(10);write('Итого:',itog:4:2);*

*textbackground(0); textcolor(9);*

*write('+'); writeln;*

*end;*

*{----------------------------------------основное меню--------------------------------------}*

*procedure menu;*

*begin*

*textmode(Co80); clrscr;*

*window(13,3,70,15); textbackground(14);textcolor(15);*

*gotoxy(1,3); write('Для дальнейшей работы с программой выберите номерок:');*

*textbackground(0);*

*gotoxy(16,5); textcolor(10); write('1'); textcolor(7); write(' - Просмотр');*

*gotoxy(16,6); textcolor(10); write('2'); textcolor(7); write(' - Cоздать');*

*gotoxy(16,7); textcolor(10); write('3'); textcolor(7); write(' - Добавить');*

*gotoxy(16,8); textcolor(10); write('4'); textcolor(7); write(' - Удалить');*

*gotoxy(16,9); textcolor(10); write('5'); textcolor(7); write(' - Редактировать');*

*gotoxy(16,10);textcolor(10); write('6'); textcolor(7); write(' - Заставка');*

*gotoxy(16,11);textcolor(10); write('7'); textcolor(7); write(' - Выход');*

*gotoxy(16,13); write('Введите номер:');*

*textcolor(10+128);readln(l); textmode(Co80);*

*end;*

*{----------------------------------------------заставка-----------------------------------------}*

*procedure zast;*

*var*

*xm,ym:integer; {xm,ym,x,y-координаты }*

*size:integer;*

*saucer:pointer; {size,saucer,k,kk-для движения}*

*k,kk:integer;*

*begin*

*clrscr; textmode(Co80); textcolor(white);*

*write('Показать заставку? (y/n)');*

*c:=readkey;*

*if c=#121 then*

*begin*

*initgraph(gd,gm,'bgi');*

*xm:=getmaxx div 4; ym:=getmaxy div 4;*

*settextstyle(2,0,5);*

*outtextxy(xm+100,10,'## Sergei Kovbasyuk ##');*

*size:=imagesize (xm+100,10,xm+275,10+20);*

*getmem(saucer,size);*

*getimage(xm+100,10,xm+275,10+20,saucer^);*

*putimage(xm+100,10,saucer^,xorput);*

*k:=0;kk:=0;i:=0;n:=0;*

*repeat*

*settextstyle(10,0,5); setcolor(i+1);*

*outtextxy((getmaxx div 5),(getmaxy div 4),'"Sorge pro"');*

*settextstyle(5,0,5); setcolor(15);*

*outtextxy((getmaxx-200),(getmaxy-250),'present');*

*settextstyle(3,0,1);*

*outtextxy(40,(getmaxy-50),'http://sorgepro.agava.ru');*

*outtextxy(290,(getmaxy-50),'e-mail: office@sorgepro.agava.ru');*

*settextstyle(2,0,5);*

*outtextxy(getmaxx-150,getmaxy-20,'Press ');*

*outtextxy(20,getmaxy-20,'(c)2000. All rights don''t reserved.');*

*k:=k+5;i:=i+1;*

*putimage(k,10,saucer^,xorput);*

*delay(50);*

*kk:=kk+5;*

*putimage(kk,10,saucer^,xorput);*

*if k>getmaxx-100 then begin k:=0; kk:=0 end;*

*until keypressed;*

*{-----------krugi------------}*

*for i:=0 to 399 do*

*begin*

*setcolor(red);*

*circle(200,100,i);*

*setcolor(green);*

*circle(400,100,i);*

*setcolor(yellow);*

*circle(300,300,i);*

*delay(10);*

*end;*

*for i:=399 downto 0 do*

*begin*

*setcolor(i);*

*circle(322,250,i);*

*delay(10);*

*end;*

*{----------end\_krugi---------}*

*closegraph;*

*end;*

*end;*

*{------------------------------------Вывод об ошибке---------------------------------------}*

*Procedure error;*

*begin*

*i:=0;*

*repeat*

*textmode(Co80);*

*clrscr;*

*gotoxy(24,8);textcolor(15);*

*write('ВЫ ВВЕЛИ НЕПРАВИЛЬНЫЙ НОМЕР');sound(185);delay(100);nosound;*

*clrscr;*

*gotoxy(24,8);textcolor(7);*

*write('ВЫ ВВЕЛИ НЕПРАВИЛЬНЫЙ НОМЕР');sound(174);delay(100);nosound;*

*i:=i+1;*

*until i=5;*

*nosound;*

*menu;*

*end;*

*{-----------------------------------------------Bыход-----------------------------------------}*

*procedure exitt;*

*begin*

*clrscr;*

*initgraph(gd,gm,'bgi');*

*repeat*

*settextstyle(7,0,4);setcolor(15);*

*outtextxy(90,getmaxy div 4, 'Technical support by e-mail:');*

*settextstyle(1,0,4);setcolor(10);*

*outtextxy(120,(getmaxy div 4)+50, 'support@sorgepro.agava.ru');*

*settextstyle(7,0,4);setcolor(15);*

*outtextxy(140,(getmaxy div 4)+100, 'or visited our web site:');*

*settextstyle(1,0,4);setcolor(10);*

*outtextxy(120,(getmaxy div 4)+150, 'http://sorgepro.agava.ru');*

*settextstyle(2,0,5);setcolor(15);*

*outtextxy(getmaxx div 4,getmaxy-30,'(c)2000."Sorge pro".All rights don''t reserved.');*

*until keypressed;*

*closegraph;*

*exit;*

*end;*

*{==========================Proga===========================}*

*BEGIN*

*zast;*

*assign (f,'k');*

*textmode(Co80);*

*{печать первого меню}*

*menu;*

*menu2:*

*{-------------------------------------------печать тaблицы-----------------------------------}*

*if l=1 then*

*begin*

*view(27);*

*textcolor(15);*

*gotoxy(7, ww+1);*

*writeln('Для выхода нажмите или -возврат в главное меню. ');*

*c:=readkey;*

*textmode(Co80);*

*if c=#27 then begin menu; goto menu2 end*

*else exitt;*

*end*

*else*

*{-----------------------------------------Заново------------------------------------------------}*

*if l=2 then*

*begin*

*clrscr;*

*textbackground(14);textcolor(15);*

*writeln('Вы действительно хотите создать новые записи?');*

*writeln('!!!ПРЕДЫДУЩИЕ ЗАПИСИ БУДУТ УТЕРЯНЫ!!!');*

*textbackground(0);*

*gotoxy(1,4);*

*writeln('Для продолжения нажмите или -возврат в главное меню.');*

*c:=readkey;*

*if c=#27 then begin menu; goto menu2 end*

*else*

*begin*

*writeln('Введите количество записей:');*

*read(q);*

*clrscr;*

*rewrite(f);*

*repeat*

*writeln('введи текущий номер продукции...'); readln(a.n);*

*writeln('введи переходящий остаток продукции на начало года...'); readln(a.e1);*

*writeln('введи приход продукции...'); readln(a.e2);*

*writeln('введи расход...');readln(a.e3);*

*writeln('введи цену за 1 шт...');readln(a.e4);*

*write(f,a);*

*until (a.n=q);*

*clrscr;*

*writeln('ВСЁ OK.');*

*delay(1000);*

*menu; goto menu2;*

*end;*

*end*

*else*

*{----------------------------------Добавление записи---------------------------------------}*

*if l=3 then*

*begin*

*clrscr;*

*textbackground(0);*

*textcolor(15);*

*writeln('Продолжить? (y/n)');*

*c:=readkey;*

*if c<>#121 then begin textmode(Co80); menu;goto menu2; end*

*else*

*begin*

*clrscr;*

*writeln('Введите количество добовляемых записей:');*

*read(mm);*

*clrscr;*

*reset(f); n:=filesize(f); seek(f,n);*

*pp:=n+1;*

*repeat*

*writeln('введи текущий номер продукции...');readln(qq); a.n:=pp;*

*writeln('введи переходящий остаток продукции на начало года...'); readln(a.e1);*

*writeln('введи приход продукции...'); readln(a.e2);*

*writeln('введи расход...');readln(a.e3);*

*writeln('введи цену за 1 шт...');readln(a.e4);*

*write(f,a);*

*pp:=pp+1;*

*until (qq=mm);*

*clrscr;*

*writeln('ВСЁ OK.');*

*delay(1000);*

*menu; goto menu2;*

*end;*

*end*

*else*

*{-----------------------------------------удаление записи------------------------------------}*

*if l=4 then*

*begin*

*clrscr;*

*window(1,1,68,22);*

*view(17);*

*window(64,1,80,22);*

*textcolor(15);*

*writeln('Продолжить? (y/n)');*

*c:=readkey;*

*if c<>#121 then begin textmode(Co80); menu;goto menu2; end*

*else*

*begin*

*write('Введи номер:');*

*readln(nn);*

*if nn>filesize(f) then begin error; goto menu2; end;*

*{удаление записи:запись в новый файл}*

*reset(f);*

*assign (f2,'k2');*

*rewrite(f2);*

*l:=0;q:=0;n:=filesize(f);*

*repeat*

*l:=l+1;*

*if l<>nn then*

*begin*

*q:=q+1;*

*seek(f,l-1);*

*read(f,a);*

*b.n:=q; b.e1:=a.e1; b.e2:=a.e2; b.e3:=a.e3; b.e4:=a.e4;*

*write(f2,b);*

*end;*

*until l=n;*

*{удаление записи: перезапись}*

*reset(f2); rewrite(f);*

*repeat*

*read(f2,b);*

*a.n:=b.n; a.e1:=b.e1; a.e2:=b.e2; a.e3:=b.e3; a.e4:=b.e4;*

*write(f,a);*

*until eof(f2);*

*writeln;*

*writeln('Всё Ok!');*

*delay(1000);*

*writeln('Reload-');*

*write('Exit-');*

*c:=readkey;*

*if c=#27 then begin menu; goto menu2; end*

*else begin textmode(Co80); l:=4; goto menu2; end;*

*end;*

*end*

*else*

*{---------------------------------------редактирование--------------------------------------}*

*if l=5 then*

*begin*

*view(27);*

*textcolor(15);*

*writeln('Продолжить? (y/n)');*

*c:=readkey;*

*if c<>#121 then begin textmode(Co80); menu;goto menu2; end*

*else*

*begin*

*write('Введите номер для редактирования:'); read(nn);*

*if nn>filesize(f) then begin error; goto menu2; end;*

*clrscr;*

*i:=27;*

*reset(f); seek(f,nn-1);*

*read(f,a);*

*st:=((a.e1+a.e2)-a.e3)\*a.e4;*

*gotoXY(i-15,1);write(a.n);*

*gotoXY(i-5,1);write(a.e1);*

*gotoxy(i+11,1);write(a.e2);*

*gotoxy(i+20,1);write(a.e3);*

*gotoxy(i+29,1);write(a.e4:4:2);*

*gotoxy(i+37,1);write(st:4:2);*

*gotoxy(7,2);*

*seek(f,nn-1);*

*writeln('Изменить переходящий остаток продукции на начало года? (y/n)');*

*c:=readkey;*

*if c=#121 then begin write('Вводим новое:');read(nn); a.e1:=nn; end;*

*writeln;*

*writeln('Изменить приход продукции? (y/n)');c:=readkey;*

*if c=#121 then begin write('Вводим новое:'); read(nn);a.e2:=nn; end;*

*writeln;*

*writeln('Изменить расход? (y/n)');c:=readkey;*

*if c=#121 then begin write('Вводим новое:'); read(nn);a.e3:=nn; end;*

*writeln;*

*writeln('Изменить цену за 1 шт? (y/n)');c:=readkey;*

*if c=#121 then begin write('Вводим новое:'); read(o);a.e4:=o; end;*

*write(f,a); writeln; write('Все Ok'); delay(1000);*

*clrscr; menu;*

*goto menu2;*

*end;*

*end*

*else*

*{--------------------------------------вывод заставки------------------------------------}*

*if l=6 then begin zast; menu; goto menu2; end;*

*{---------------------------------------выход----------------------------------------------}*

*if l=7 then exitt*

*else*

*begin error; goto menu2; end;*

*{========================END\_Proga========================}*

*END.*

|  |
| --- |
| Writed by Sergei Kovbasyuk  (c)2000."Sorge pro". Rights don't reserved.  e-mail:office@sorgepro.agava.ru  http://sorgepro.agava.ru |