МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Белорусский государственный университет информатики и радиоэлектроники

Кафедра программного обеспечения

информационных технологий

**КУРСОВАЯ РАБОТА**

По курсу: “**Основы алгоритмизации и программирования**"

На тему: **“Среда разработки Турбо Паскаль 7.0.**

**Базы данных**"

Исполнитель: студент

Проверил

МИНСК 2002

## Аннотация

Данная курсовая работа посвящена реализации базы данных на примере описания стационарного объекта “Вокзал”. В ходе реализации программы организован удобный пользовательский интерфейс. Этому способствовали методы, используемые средой разработки Turbo Pascal 7.0.

В качестве расписания вокзала в работе используется файл записей. Записи имеют несколько полей, в которых содержится информация по определенному поезду на соответствующую дату. Благодаря использованию переменных типа записи мы получаем доступ сразу ко всей информации, а также можем изменять ее соответствующим образом.

Содержание

Аннотация

Введение

1. Постановка задачи

2. Описание программы

3. Тестирование программы

Заключение

Список использованной литературы

## Введение

Целью написания данной курсовой работы является разработка программы, создающей и управляющей базой данных.

Программа построения базы данных реализована на языке Turbo Pascal.

Этот язык был разработан Н. Виртом первоначально для целей обучения программированию вообще. С этой точки зрения Паскаль имеет некоторое преимущество перед однотипными языками - такими, как, например, язык Си.

По своей идеологии Паскаль близок к современной методике и технологии программирования. Этот язык весьма полно отражает идеи структурного программирования, что отчетливо проявляется в основных управляющих структурах, предусмотренных в Паскале.

Паскаль хорошо приспособлен для применения общепризнанной в настоящее время технологии разработки программ методом нисходящего проектирования (пошаговой детализации). Это проявляется в том, что Паскаль может успешно использоваться для записи программы на разных уровнях ее детализации, не прибегая к помощи блок-схем или специального языка проектирования программ.

В задачи данной курсовой работы входит:

организация алгоритма программы;

организация вывода информации;

возможность добавления информации в базу данных;

поиск информации в базе данных по заданному значению;

информативность и удобство в использовании.

## 1. Постановка задачи

Для реализации программы основная задача - это создание надёжной системы хранения данных. Создание удобного доступа к данным можно осуществить с использованием переменных типа - записи. Пользовательский интерфейс должен быть образован приемлемым образом, для того чтобы обеспечить удобство использования и наилучшее восприятие пользователем выводимой информации.

**МЕТОДЫ (АЛГОРИТМЫ) РЕШЕНИЯ ЗАДАЧИ.**

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

Структура объявления записи такова:

<имя типа> = **RECORD** <список полей> **END**

здеcь

<имя типа> - правильный идентификатор;

RECORD, END - зарезервированные слова;

<список полей> - 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 то друга запятыми. За

идентификатором (идентификаторами) ставится двоеточие и описание типа поля (полей), например:

**type**

birthday = record

day, month: Byte;

year: Word

**end;**

**var**

a,b: birthday;

…….

В этом примере тип BIRTHDAY (день рождения) есть запись с полями DAY, MONTH и YEAR (день, месяц и год рождения); переменные А и В содержат записи типа BIRTHDAY.

Как и в массиве, значения переменных типа записи можно присваивать другим переменным того же типа, например:

a: =b

К каждому из компонентов записи можно получить доступ, если использовать составное имя, т.е. указать имя переменной, затем

точку и имя поля.

a. day: =27;

b. year: =1939;

Для вложенных полей приходится продолжать уточнения:

**if** c. bd. year = 1939 **then …**

**end.**

Чтобы упростить доступ к полям запис, используется оператор присоединения **WITH**

**With** <переменная> **do** <оператор>

Здесь **with, do** - ключевые слова (с, делать);

<переменная> - имя переменной типа запись, за которым, возможно, следует список вложенных полей;

<оператор> - любой оператор Турбо Паскаля.

Например:

**With** c. bd **do** mont: =9

это эквивалентно:

c. bd. month: =9;

Турбо Паскаль разрешает использовать записи с так называемыми вариантными полями, например:

**Type**

Forma = record

Name: string;

**Case** Byte **of**

0: (BirthPlace: string [40]);

1: (Countri: string [30] ;

ExitDate:

1. .31)

**End;**

В этом примере тип FORMA определяет запись с одним фиксированным полем NAМЕ и вариантной часть, которая задается предложением **Case…. of.** Вариантная часть состоит из нескольких вариантов. Каждый из вариантов определяется константой выбора, за которой следует двоеточие и список полей, заключенный в круглые скобки. В любой записи может быть только одна вариантная часть, и, если она есть, располагаться за всеми фиксированными частями.

Замечательной особенностью вариантной части является то обстоятельство, что все заданные в ней варианты накладываются друг на друга, т.е. каждому из них выделяется одна и та же область памяти. Это открывает дополнительны возможности преобразования типов. Ключ выбора фактически игнорируется компилятором: единственное требование, предъявляемое к вам Турбо Паскалем, состоит в том, чтобы ключ определял некоторый стандартный или предварительно объявленный тип. Причем сам этот тип никак не влияет ни на количество следующих за ним вариантных полей, ни даже на характер констант выбора. В стандартном Паскале в качестве ключа всегда необходимо указывать в качестве ключа выбора некоторую переменную порядкового типа, причем в исполняемой части программы можно присваивать некоторое значение этой переменной и тем самым влиять на выбор полей. В Турбо Паскале также можно в поле ключа выбора указывать переменную порядкового типа и даже присваивать ей в программе какое-то значение, что однако не влияет на выбор поля: значения констант выбора в Турбо Паскале могут быть произвольными, в том числе повторяющимися.

Имена полей должны быть уникальными в пределах той записи, где они объявлены, однако, если записи содержат поля-записи, т.е. вложены одна в другую, имена могут повторяться на разных уровнях вложенности, например: **c. bd. f. bd. c**

## 2. Описание программы

Общие сведения.

Для хранения информации обо всей информации в базе данных используется динамическое дерево. Для чего описывается новый тип данных - запись (raspis). В записи raspis описываются следующие поля:

numer - номер поезда в расписании;

datav- дата отправления;

kpunkt - конечный пункт следования;

vremyaot - время отправления;

kmest - количество свободных купейных мест;

pmest-количество свободных плацкартных мест.

В программе так же используются переменные:

work - перменная для обработки данных

BookFile - переменная, в которой хранится имя файла расписания

Все остальные переменные вспомогательные

В программе используются следующие процедуры:

NameFile - задает переменную BookkFile.

*Dobawlenie1 -* запись в файл значения переменной work.

Bronir - бронирование билетов

*Sozdanie* - создание нового файла расписания.

Prosmotr - просмотр файла расписания.

Dobawlenie - добавление информации в базу.

*Poisk -* поиск по конечному пункту

*Udalenie -* удаление файла расписания.

Функциональное назначение

Программу можно использовать для организации базы данных. В программе можно производить поиск и добавлять новую информацию в конец базы данных. Программу также можно усовершенствовать, добавив обход базы данных рекурсивным способом.

В блоке инициализации происходит подготовка экрана и данных. После инициализации выполняется бесконечны цикл проверяющий состояние клавиатуры.

При нажатии на клавиши происходят следующие события:

“1” - Просмотр расписания;

“2” - Добавление информации;

“3” - Удаление информации;

“4” - Поиск по конечному пункту;

“5” - Создание файла;

“6” - Бронирование мест;

“7” - Выход из программы.

Граф-схема программы

В программе используются следующие процедуры:

ramka - перерисовывает экран.

В ней используются функции библиотеки Crt, такие как:

textcolor (color: Byte) - задаёт цвет символов;

clrscr - очищает весь экран;

gotoxy (X,Y) - переводит курсор в место с координатами X,Y;

*Insert\_punkt*- Осуществляет заполнения дерева.

Tree\_rec - основная процедура (обходы дерева).

При старте процедуры инициализируются переменные и подготавливается экран процедурой Ramka. Эта процедура создаёт фон и выводит рамку меню в середине экрана.

Дальше следует бесконечный цикл обработки сообщений от клавиатуры. Для этого используются функции keypressed и readrey. Состояние клавиатуры определяется при помощи функции keypressed, если клавиша нажата, то оператором выбора проверяем, какая именно функцией readkey. Если нажаты такие клавиши как “1", “2”, “3”,”4”,”5”,”6”,”7” то в буфере клавиатуры будет храниться один символ.

Управление организовано при использовании семи клавиш:

“1” - Вывод бинарного дерева;

“2” - Создаёт бинарное дерево

“3” - Удаляет элемент из дерева;

“4” - Удаляет все дерево.

“5” - Запрос о поиске.

“6” - Выводит обходы бинарного дерева.

3.5 Технические средства

Для запуска и надёжной работы программы подойдут любые используемые сейчас компьютеры с операционными системами Windows или Dos. Программа требует немного ресурсов процессора и памяти.

Вызов и загрузка программы

При запуске программы производится вывод меню, с которой можно сразу же начинать работать.

## 3. Тестирование программы

Тестирование производилось на современных компьютерах с процессорами Intel Celeron, Intel Pentium с оперативной памятью 64Mb. На всех компьютерах программа вела себя одинаково. В процессе работы было создано дерево, осуществлено добавление новых элементов и произведён поиск в созданном дереве.

## Заключение

Итак, в ходе выполнения курсовой работы мне удалось убедиться, что язык Паскаль предоставляет весьма гибкие возможности в отношении используемых структур данных. Как известно, простота алгоритмов, а значит, трудоемкость их разработки и их надежность существенно зависят от того, насколько удачно будут выбраны структуры данных, используемые при решении задачи.

Алгоритмический язык Паскаль создавался для обучения. Поэтому он хорошо продуман с точки зрения эффективности (реализация самого языка) и с точки зрения получаемых в результате трансляции машинных команд.

Большое внимание в Паскале уделено также вопросу повышения надежности программ. Средства языка позволяют осуществлять достаточно полный контроль правильности использования данных различных типов и программных объектов, как на этапе трансляции программы, так и на этапе ее выполнения.

Благодаря этим своим особенностям Паскаль находит все более широкое применение не только в области обучения, но и в практической работе. Из всего вышеперечисленного становится понятным, почему Паскаль вызывает повышенный интерес и почему он все чаще выбирается в качестве базового языка при обучении программированию.

На языке Паскаль в работе разработан алгоритм построения генеалогического дерева. Был использован массив динамических записей, которые заполнены информацией о каждом человеке, находящемся на каком-либо уровне генеалогического дерева.

В результате была реализована программа построения бинарного дерева с применением динамических переменных и удобного пользовательского интерфейса.

## Список использованной литературы

1. Климова Л.М. PASCAL 7.0 Практическое программирование. Решение типовых задач. - М.: КУДИЦ-ОБРАЗ, 2000.
2. Абрамов С.А., Зима Е.В. Начала программирования на языке Паскаль. -М.: Наука, 1987.
3. Аладьев В.З., Тупало В.Г. Turbo-Pascal для всех. - Киев: Технiка, 1993.
4. Белецкий, Ян. Турбо Паскаль с графикой для персональных компьютеров. - М.: Машиностроение, 1991.
5. Джонс Ж., Харроу К. Решение задач с системой Турбо Паскаль. -М.: Финансы и статистика, 1991.
6. Епанешников А.М. Программирование в среде Turbo Pascal 7.0. -М.: Диалог-МИФИ, 1996.
7. Климов Ю.С. Программирование в среде Turbo-Pascal 6.0. - Мн.: выш. шк., 1992.
8. Сергиевский М.В. Язык, среда программирования. - М.: Машиностроение, 1994.
9. Турбо Паскаль 7.0. -Киев: торгово-издательское бюро BHV, 1995.
10. Фаронов В.В. Турбо Паскаль 7.0. начальный курс. - М.: "Нолидж", 1997.
11. Фаронов В.В. Турбо Паскаль 7.0. Практика программирования. - М.: "Нолидж", 1997.