

Лабораторная работа №5

Тема: Тип данных – запись

Задание: разработать приложение, позволяющее вносить данные в виде, определенном в задании. Данные представляются в памяти в виде динамического массива записей. Должны поддерживаться следующие операции: добавление элемента, удаление элемента, подсчет количества элементов. Также должны поддерживаться функции, определенные в задании. Результат вывести в виде текста.

Запись — это простейший способ логического объединения данных в единую сущность, предоставляемый языком программирования. С точки зрения машинного кода запись не отличается от своих полей-переменных, идущих друг за другом. Запись позволяет объединять разнородные поля в единую структуру данных. На практике, записи часто используются в составе массивов. Массив записей — это сложная структура, содержащая набор элементов данных с различной информацией о сущности. По сути, массив записей — это простая, но более-менее полная реализация таблицы базы данных. С отдельной записью можно работать как с единой переменной, при этом произойдет полное копирование значения:

type

TStudent = record

Shifr: Integer;

FIO: string;

DenRozdeniya: TDateTime;

end;

var

AStudent, BStudent: TStudent;

begin

…

AStudent.Shifr := 123456;

AStudent.FIO := ‘Иванов Иван Иванович’;

DenRozdeniya := EncodeDate(1, 1, 1980);

BStudent := AStudent;

…

end;

Однако, непосредственно сравнивать значения типа запись нельзя — компилятор ничего не знает о составе данных внутри типа запись, и не может составить правильную операцию сравнения.

В случае работы с набором одинаковых структур данных, можно использовать динамический массив записей:

type

TPerson = record

F: string;

I: string;

O: string;

end;

TPersons = array of TPerson;

var

Persons: TPersons;

Для ввода данных следует использовать набор полей ввода (это делает возможным внесение всей структуры данных за один прием). Для отображения же целесообразно представить данные в виде таблицы, где строки — единичные записи, столбцы — поля структуры данных, а ячейки — конкретные значения полей. Существует два стандартных компонента, пригодных для отображения таблиц — это TStringGrid (таблица строк) и TListView с расширенным отображением элементов и произвольной отрисовкой. Ознакомимся с ним подробнее.

TListView — это надстройка над компонентом операционной системы. В операционной системе данный компонент используется для отображения списка файлов в окнах проводника и для рабочего стола. Он имеет 4 режима отображения — список, маленькие иконки, большие иконки и отчет. Нас интересует последнее — для этого установим свойство ViewStyle в vsReport. Колонки в таком представлении редактируются с использованием интерактивного редактора свойства Columns.

Каждый элемент списка в TListView — это структура (объект) типа TListItem. У него есть множество свойств:

Index — номер элемента в списке

Data — указатель на пользовательские данные, на самом деле — произвольное 4х байтное число (компонент сам никак не обрабатывает это поле)

Selected — логическое свойство, выделен элемент или нет. Для выделения множества элементов следует установить свойство MultiSelect компонента в true.

Caption — текст элемента. Именно это текст будет показан в ПЕРВОЙ колонке.

SubItems — набор дочерних элементов — TStrings, хранящий текст ячеек, начиная со ВТОРОЙ колонки.

Пример использования:

procedure TForm1.Button1Click(Sender: TObject);

var

P: TPerson;

Len: Integer;

NewItem: TListItem;

begin

P.F := Edit1.Text;

P.I := Edit2.Text;

P.O := Edit3.Text;

// здесь логично проверить ввод на корректность

Len := Length(Persons);

SetLength(Persons, Len+1);

Persons[Len] := P;

NewItem := ListView1.Items.Add;

NewItem.Caption := P.F;

NewItem.SubItems.Add(P.I);

NewItem.SubItems.Add(P.O);

end;

Вариант №1

Данные — сведения о студенте, включают: шифр, фамилию, имя, отчество, группу, факультет, направление подготовки.

1. Найти студентов с введенной фамилией.
2. Найти студентов на введенном факультете
3. Найти студентов в группе
4. Найти студентов по направлению подготовки, обучающихся НЕ на введенном факультете.

Вариант №2

Данные — сведения о сотруднике, включают: личный номер, фамилию, имя, отчество, год рождения, специальность, стаж

1. Найти сотрудников со стажем, больше введенного.
2. Найти сотрудников с определенной специальностью
3. Найти сотрудников с одинаковым стажем и возрастом
4. Найти сотрудников с одинаковой специальностью и стажем

Вариант №3

Данные — сведения о налогоплательщике, включают: ИНН, название (компании или фио), тип (физ. лицо, юр. лицо), форма налогообложения (ООО, ЗАО, ОАО, физ.лицо, ПБОЮЛ, некоммерческая организация), дата подачи последней декларации

1. Найти налогоплательщиков, с даты подачи декларации которых прошло более года.
2. Найти налогоплательщиков с определенной формой налогообложения
3. Найти налогоплательщиков с некорректной формой налогооблажения (ОАО и физ. лицо, юр. лицо и ПБОЮЛ)
4. Найти налогоплательщиков с датой подачи в определенный месяц

Вариант №4

Данные — бюро пропусков, включают: номер пропуска, фамилию, имя, отчество, дата выдачи, срок действия, уровень допуска, дата последней аттестации.

1. Найти все просроченные пропуска
2. Найти все пропуска, у которых с последней аттестации прошло более полугода
3. Найти все пропуска, которые следует перевыпустить (с последней аттестации менее полугода, срок заканчивается в ближайшие 63 дня)
4. Найти все пропуска, выданные на конкретное лицо