

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ТУЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт прикладной математики и компьютерных наук
Кафедра информационной безопасности

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторным работам (часть 2)

по дисциплине

ПРОГРАММИРОВАНИЕ

*Уровень профессионального образования: высшее образование –
бакалавриат*

Направление подготовки: *090303 Прикладная информатика*

Профиль подготовки: *Прикладная информатика в экономике*

Квалификация выпускника: *бакалавр*

Тула 2018 г.

ЛАБОРАТОРНАЯ РАБОТА №5

КОМБИНИРОВАННЫЙ ТИП ДАННЫХ (ЗАПИСИ)

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ.

Приобретение практических навыков работы с комбинированным типом данных (записями).

2. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Комбинированный тип данных представляет собой совокупность логически связанных компонент, принадлежащих к разным типам. Компоненты записи, называемые полями, различаются по именам. Записи могут быть описаны в разделе определения типов следующим образом:

```
TYPE <имя>=RECORD  
    <список полей>  
END;
```

где <имя> - имя типа,

<список полей> - имена полей с описанием их типа.

Поля в списке разделяются ";". Например, запись о студенте, состоящая из полей "ФАМИЛИЯ", "СРЕДНИЙ БАЛЛ", "ПРОПУСКИ", будет описана следующим образом:

```
TYPE STUD=RECORD  
    FIO: ARRAY [1..16] OF CHAR;  
    BAL: REAL;  
    PROP: INTEGER  
END;
```

В разделе VAR определяются имена записей, которые будут иметь тип STUD, например:

```
VAR S1, S2: STUD;
```

Записи могут быть определены непосредственно в разделе VAR, без использования вспомогательного имени в разделе TYPE. Каждое поле записи можно рассматривать как обычную переменную, которую можно использовать отдельно. Вместе с тем запись можно обрабатывать как единое целое.

Доступ к полям осуществляется с помощью составных имен, включающих имя записи и имя поля, разделенных точкой, например:

```
S1.FIO; S1.BAL; S1.PROP;
```

Обращение к компонентам записи можно упростить, если воспользоваться оператором присоединения WITH. Оператор позволяет вместо составных имен использовать непосредственно имена полей. При этом имя записи выносится в заголовок оператора WITH. Оператор WITH имеет следующую структуру:

```
WITH <имя> DO <оператор>;
```

где имя - имя записи (можно задавать несколько имен);

оператор - простой или составной оператор, выполняющий действия над полями записи.

Поля записи в частном случае могут иметь одинаковый тип, например:

```
VAR DATA: RECORD  
    DEN, MES, GOD: INTEGER  
END;
```

В Паскале допускается любая вложенность одной записи в другую. Допускается также использовать записи в качестве компонент массива и файла, например:

```
VAR GRUP: ARRAY [1..50] OF STUD;
```

Т.е. в этом случае каждым элементом массива GRUP будет являться запись студента. Чтобы обратиться к некоторому полю определенной записи массива, необходимо указать имя массива, индекс требуемого элемента (записи) и имя поля. Например, оператор WRITE(GRUP[10].BAL); выводит величину среднего балла записи десятого студента.

Пример.

Запись о студенте имеет следующую структуру: фамилия студента, средний балл, количество пропусков занятий. Ввести записи студентов всей группы, определить средний балл группы и количество пропусков по группе, а также выбрать студентов, имеющих средний балл, равный 5.

```
PROGRAM EXSAMEN;  
CONST K=16;  
      KOL=25;  
TYPE STUD=RECORD  
    FIO: ARRAY [1..K] OF CHAR;  
    BAL: REAL;  
    PROP: INTEGER  
END;  
VAR GRUP: ARRAY [1..KOL] OF STUD;  
    N, PR, I, J: INTEGER;  
    SR: REAL;  
BEGIN  
  { ВВОД ДАННЫХ ПО ГРУППЕ }  
  SR:=0;  
  PR:=0;  
  WRITE('УКАЖИТЕ ЧИСЛО СТУДЕНТОВ В ГРУППЕ -');  
  READLN(N);  
  WRITELN;  
  WRITELN('ВВЕДИТЕ ДАННЫЕ О СТУДЕНТАХ');  
  WRITELN(': ФАМИЛИЯ И.О. : СР.БАЛЛ : ПРОПУСКИ :');  
  FOR I:=1 TO N DO  
    BEGIN  
      WRITE(I, '. ');
```

```

FOR J:=1 TO K DO
  READ(GRUP[I].FIO[J]);
  READLN(GRUP[I].BAL,GRUP[I].PROP);
  SR:=SR+GRUP[I].BAL;
  PR:=PR+GRUP[I].PROP;
END;
SR:=SR/N;
WRITELN;
WRITELN('СРЕДНИЙ БАЛЛ ГРУППЫ - ',SR:3:1);
WRITELN('КОЛИЧЕСТВО ПРОПУСКОВ - ',PR:3);
WRITELN;
WRITELN('СПИСОК СТУДЕНТОВ - ОТЛИЧНИКОВ');
FOR I:=1 TO N DO
  IF GRUP[I].BAL=5.0 THEN BEGIN
    FOR J:=1 TO K DO
      WRITE(GRUP[I].FIO[J]);
      WRITELN;
    END;
  END;
END.

```

3. ОБОРУДОВАНИЕ

ПЭВМ IBM PC, SVGA монитор с разрешением не менее 800*600 пикселей; клавиатура; мышь. Среда Free Pascal, Lazarus.

4. ЗАДАНИЕ НА РАБОТУ

Составить программу, предусматривающую создание массивов записей. Все данные задаются студентами самостоятельно.

Варианты задания выбираются по последнему номеру последнему номеру зачетки.

Ввод и вывод данных в лабораторной работе организовать в виде таблиц

0. Ввести данные следующей структуры: наименование товара, страна-производитель, цена товара, количество товара, фирма-поставщик, номер отдела универмага. Организовать просмотр исходных данных и вывести список товаров, производимых в данной стране (по выбору), указанием цены товара, количества и фирмы-поставщика.

1. Ввести данные следующей структуры: регистрационный номер, марка, номер двигателя, год выпуска, дата последнего техосмотра. Организовать просмотр исходных данных и вывести список автомобилей, с указанием регистрационного номера, марки, года выпуска.

2. Ввести данные следующей структуры: тип самолета (пассажирский, грузовой), номер самолета, номер рейса, пункт отправления, пункт назначения, время отправления, время полета. Организовать просмотр исходных данных и вывести список самолетов, с указанием номера рейса, пункта отправления, пункта назначения, времени отправления, времени полета.

3. Ввести данные следующей структуры: код товара, название товара, количество на складе, страна-производитель, дата выработки, срок годности. Организовать просмотр исходных данных и вывести список товаров, с указанием кода товара, названия товара, количества на складе.

4. Ввести данные следующей структуры: ФИО спортсмена, вид спорта, страна, занятое место и количество набранных баллов. Организовать просмотр исходных данных и вывести список спортсменов данной вида спорта (по выбору), с указанием страны, занятого места и количества набранных баллов.

5. Ввести данные следующей структуры: № рейса, направление движения и время отправления автобуса, стоимость билета, ФИО водителя. Организовать просмотр исходных данных и вывести список автобусов, с указанием № рейса, направления движения, времени отправления.

6. Ввести данные следующей структуры: тип поезда (пассажирский, скорый, грузовой), номер поезда, пункт отправления, пункт назначения, время отправления, стоимость билета. Организовать просмотр исходных данных и вывести список поездов, в зависимости от типа поезда (по выбору), с указанием номера поезда, пункта отправления, пункта назначения.

7. Ввести данные следующей структуры: название книги, автор, год издания, количество экземпляров, тематика. Организовать просмотр исходных данных и вывести список книг заданной тематики (по выбору), с указанием названия книги, автора, количества экземпляров.

8. Ввести данные следующей структуры: название книги, автор, год издания, количество экземпляров, тематика. Организовать просмотр исходных данных и вывести список книг, с указанием названия книги, автора, года издания.

9. Ввести данные следующей структуры: номер квитанции, Ф.И.О. клиента, марка автомобиля, № автомобиля, вид неисправности, стоимость ремонта. Организовать просмотр исходных данных и вывести список автомобилей, с указанием номера квитанции, Ф.И.О. клиента, вида неисправности, стоимости ремонта.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить теоретические положения.
2. Разработать схему программы решения задачи.
3. Составить программу.
4. Отладить программу. Результаты работы программы показать преподавателю.
5. Оформить отчет.
6. Защитить лабораторную работу перед преподавателем.

6. СОДЕРЖАНИЕ ОТЧЕТА

1. Номер и название лабораторной работы
2. Цель и задачи
3. Задание на работу. Описание задания в соответствии с вариантом.
4. Схема программы
5. Текст программы
6. Результаты и выводы по лабораторной работе

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как задается комбинированный тип данных?
2. Можно ли определить записи непосредственно в разделе описания переменных?
3. Объясните назначение составного имени при работе с записями?
4. Как используется оператор WITH?
5. Приведите пример вложенности одной записи в другую.

ЛАБОРАТОРНАЯ РАБОТА №6

РАБОТА С ФАЙЛАМИ

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ.

Получить практические навыки работы с файлами последовательного и прямого доступа.

2. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Файловый тип данных

Файловый тип данных (файл) представляет собой упорядоченную совокупность однотипных компонент (количество компонент в файле заранее не оговаривается). Файловый тип данных может быть описан в разделе определения типов следующим образом:

TYPE T=FILE OF T1;

где T - имя файлового типа;

T1 - тип компонент (базовый тип).

Базовый тип может являться стандартным типом (кроме BOOLEAN) или ранее определенным типом. Очень часто в качестве базового используется комбинированный тип данных. Файлы могут быть определены как с использованием вспомогательного имени типа в разделе TYPE, так и непосредственно в разделе описания переменных VAR. Например:

TYPE F= FILE OF REAL;

SAP= RECORD

DEN, MES, GOD: INTEGER

END;

VAR RAB, NAM: F;

DATA: FILE OF SAP;

Вводятся файловый тип F, комбинированный тип SAP и три файла с именами RAB, NAM, DATA. Компонентами файлов RAB и NAM являются вещественные числа, а файла DATA - записи.

Последовательные файлы

В последовательном файле компоненты обрабатываются последовательно с начала файла друг за другом.

СОЗДАНИЕ ФАЙЛА начинается с оператора ASSIGN (<имя файла>, '<имя физического файла>');

При этом происходит связывание файловой переменной <имя файла> с именем физического файла, заданным в ' '.

ПОДГОТОВКА К ЗАПИСИ компонент файла осуществляется оператором REWRITE(<имя файла>);

При этом файл становится открытым для записи и закрытым для чтения.

ЗАПИСЬ компонент В ФАЙЛ производится оператором

WRITE(<имя файла>,<имя переменной>);

При этом после записи очередной компоненты формируется признак конца файла.

ПОДГОТОВКА К ЧТЕНИЮ компонент файла осуществляется оператором RESET(<имя файла>);

При этом файл становится открытым для чтения, но закрытым для записи. ЧТЕНИЕ компонент ФАЙЛА производится оператором READ(<имя файла>,<имя переменной>);

При этом переменной присваивается значение очередной компоненты файла. Для определения конца файла при считывании компонент используется логическая функция EOF(<имя файла>). При достижении конца файла значение функции становится истинным. Функцию EOF можно использовать в качестве условия в операторах IF и WHILE. Например, чтение записей всего файла до его окончания можно представить следующей конструкцией:

```
WHILE NOT EOF(F) DO READ(F,P);
```

Файлы, имена которых определяются при описании файлового типа, являются временными (внутренними). После завершения программы они не сохраняются, так как не поддерживаются операционной системой. Файлы можно сделать постоянными (внешними), если связать внутреннее имя файла с его внешним именем, которое используется операционной системой. Внешним именем определяется некоторый участок на магнитном диске, который отводится под файл. Внутреннее имя связывается с внешним с помощью оператора ASSIGN. Поскольку в программах обычно используются файлы данных, то в качестве расширения имени внешнего (физического) файла указывается тип DAT.

Пример. Создать последовательный файл с именем FIL, состоящий из трех записей целого типа. Выполнить чтение записи файла с выводом их значений на экран.

```
PROGRAM WWOD;  
CONST C1=7;  
      C2=77;  
      C3=177;  
VAR FIL : FILE OF INTEGER;  
    S: INTEGER;  
BEGIN  
  { ФОРМИРОВАНИЕ ФАЙЛА  }  
  ASSIGN(FIL,'PROBA.DAT');  
  REWRITE(FIL);  
  WRITE(FIL,C1);  
  WRITE(FIL,C2);  
  WRITE(FIL,C3);  
  CLOSE (FIL);  
  { ЧТЕНИЕ ЗАПИСЕЙ ФАЙЛА  }  
  RESET(FIL);  
  WHILE NOT EOF(FIL) DO  
    BEGIN  
      READ(FIL,S);
```

```

        WRITELN('ЗАПИСЬ - ',S:3)
    END;
CLOSE (FIL);
END.

```

Для выполнения модификации последовательного файла используется некоторый рабочий файл, в который производится перезапись исходного файла с соответствующим изменением записи или добавлением новых. Затем из рабочего файла все записи копируются в исходный.

Пример. Изменить значение второй записи ранее созданного файла FIL (вместо 77 записать 777).

```

PROGRAM MODIF;
VAR FIL, RAB: FILE OF INTEGER;
    S: INTEGER;
BEGIN
{ МОДИФИКАЦИЯ ФАЙЛА FIL С ЗАПИСЬЮ В ФАЙЛ RAB }
ASSIGN(FIL,'PROBA.DAT');
RESET(FIL);
REWRITE(RAB);
WHILE NOT EOF(FIL) DO
    BEGIN
        READ(FIL,S);
        IF S=77 THEN S:=777;
        WRITE(RAB,S)
    END;
CLOSE(RAB);
CLOSE(FIL);
{ ПЕРЕЗАПИСЬ RAB В FIL }
RESET(RAB);
REWRITE(FIL);
WHILE NOT EOF(RAB) DO
    BEGIN
        READ(RAB,S);
        WRITELN(' ЗАПИСЬ - ',S:3);
        WRITE(FIL,S)
    END;
WRITELN('ФАЙЛ FIL ИЗМЕНЕН');
CLOSE(RAB);
CLOSE(FIL);
END.

```

Файлы прямого доступа

В файлах прямого доступа выполняется непосредственное обращение к любой компоненте файла, т.е. при этом нет необходимости просматривать все предыдущие компоненты. Каждая компонента файла адресуется идентификатором,

называемом номером записи (записи нумеруются последовательно, начиная с единицы). Обращение к необходимой компоненте файла производится с помощью оператора

SEEK(<имя файла>,<номер записи>);

После оператора SEEK обычно следуют операции чтения или записи из/в файл.

Пример. Создать файл прямого доступа, который состоит из 10 записей целого типа. Прочитать пятую запись и вывести ее значение на экран.

```
PROGRAM DOST;  
CONST N=10;  
VAR NAD: FILE OF INTEGER;  
    MAS: ARRAY[1..N] OF INTEGER;  
    S, I: INTEGER;  
BEGIN  
  WRITELN('ВВЕДИТЕ МАССИВ MAS');  
  FOR I:=1 TO N DO  
    READ(MAS[I]);  
  { ФОРМИРОВАНИЕ ФАЙЛА NAD }  
  ASSIGN (NAD,'DOST.DAT/SEEK');  
  REWRITE(NAD);  
  FOR I:=1 TO 10 DO  
    BEGIN  
      SEEK(NAD,I);  
      WRITE(NAD,MAS[I])  
    END;  
  { ЧТЕНИЕ 5-ОЙ ЗАПИСИ }  
  RESET(NAD);  
  SEEK(NAD,5);  
  READ(NAD,S);  
  WRITELN('5 ЗАПИСЬ - ', S:2);  
END.
```

3. ОБОРУДОВАНИЕ

ПЭВМ IBM PC, SVGA монитор с разрешением не менее 800*600 пикселей; клавиатура; мышь. Среда Free Pascal, Lazarus.

4. ЗАДАНИЕ НА РАБОТУ

Изменить программу, разработанную в предыдущей работе (**варианты задания сохраняются те же , что и в лабораторной работе №5**), организовав хранение записей в файле. Файл должен содержать не менее 15 записей.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

7. Изучить теоретические положения.

8. Разработать схему программы решения задачи.
9. Составить программу.
10. Отладить программу. Результаты работы программы показать преподавателю.
11. Оформить отчет.
12. Защитить лабораторную работу перед преподавателем.

6. СОДЕРЖАНИЕ ОТЧЕТА

1. Номер и название лабораторной работы
2. Цель и задачи
3. Задание на работу. Описание задания в соответствии с вариантом.
4. Схема программы
5. Текст программы
6. Результаты и выводы по лабораторной работе

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как задается комбинированный тип данных?
2. Можно ли определить записи непосредственно в разделе описания переменных?
3. Объясните назначение составного имени при работе с записями?
4. Как используется оператор WITH?
5. Приведите пример вложенности одной записи в другую.
6. Каким образом определяется файловый тип данных?
7. К какому типу данных могут принадлежать компоненты файла?
8. Каково назначение функции EOF?
9. Для каких операций применяются процедуры RESET и REWRITE?
10. Что понимается под внешним файлом?
11. Как связать внутреннее имя файла с внешним?
12. Какие ключи управления вводом-выводом задаются при работе с файлами прямого доступа?

ЛАБОРАТОРНАЯ РАБОТА №7

МЕТОДЫ СОРТИРОВКИ

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ.

Получение практических навыков составления программ упорядочивания информации, использующейся в управлении.

2. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

В теории информационного поиска различают внутреннее и внешнее упорядочивание. Если весь процесс упорядочения происходит в оперативной памяти ЭВМ, то говорят о внутреннем упорядочении.

Для оценки методов внутреннего упорядочения можно рекомендовать два основных критерия: число операций сравнения и размер требуемого рабочего поля ОЗУ. Обычно, чем меньше рабочее поле памяти, тем больше число операций сравнения необходимо, и наоборот.

В данной работе рассматривается ряд методов упорядочения, которые можно условно разбить на две группы. К первой группе относятся методы упорядочения, требующие минимального объема рабочего поля, ко второй группе — методы с минимальным числом операций.

Для реализации методов сортировки используют циклы

1. Метод обмена (выборки)

Сущность данного метода заключается в следующем: в исходном массиве ищем максимальный (минимальный) элемент и меняем его местами с элементом, стоящим на первом месте. Далее из оставшихся элементов (исключаем первый элемент), определяем максимальный (минимальный) элемент и меняем его местами с элементом, стоящим на втором месте и т.д. В результате формируется массив, упорядоченный по убыванию (по возрастанию).

2. Метод вставки

Этот метод относится к первой группе. Для заданного массива, начиная со второго элемента, расставляем элементы по местам в порядке возрастания (минимальный признак) или убывания (максимальный признак). Берем элемент, сравниваем его с предыдущими и, если он больше какого-то числа, раздвигаем элементы и вставляем обрабатываемый на нужное место.

Итак, сущность метода заключается в следующем: предполагается, что элементы массива с номерами 1, 2, 3, ..., $k-1$ уже упорядочены, нужно найти и вставить элемент с индексом k . В этом случае следует найти элемент с номером i ($1 \leq i \leq k-1$) такой, что $A_i \leq A_k \leq A_{k-1}$ и вставить k -ую запись после i -той, сдвинув все записи (элементы) с номерами $i+1$, $i+2$, ..., $k-1$ на одну позицию. Процесс вставки начинают с записи $k=2$ и после вставки номер возрастает на единицу ($k=k+1$); при $k=n+1$ упорядочивание заканчивается.

3. Метод пузырька

Упорядочивание методом пузырька близко к упорядочиванию методом вставки. Метод пузырька состоит в организации упорядоченного списка элементов, в которых на соответствующие места добавляются один за другим неотсортированные элементы. Алгоритм сортировки методом пузырька имеет вид:

1. Сравнить A_1 и A_2 : если $A_1 > A_2$, то осуществить перестановку; если $A_1 < A_2$, то перейти к выбору следующего элемента.

2. Сравнить A_2 и A_3 : $A_2 < A_3$, то перейти к выбору следующего элемента; если $A_2 > A_3$, то осуществить перестановку. Сравнить A_1 и A_2 и сделать перестановку, если она необходима.

3. Предположим, что часть элементов списка $1., 2., \dots, i$ уже упорядочена; сравнить A_i и A_{i+1} .

4. Если $A_i < A_{i+1}$, то установить $i = i + 1$, и если $i \leq n - 1$, то перейти к п.3 и продолжить сравнение; если $i > n - 1$, закончить сортировку. Если $A_i > A_{i+1}$, то поменять их местами и убедиться в том, что элемент A_i находится на своем месте в упорядоченном списке.

5. Провести проверку правильности расположения элементов в отсортированном списке, для чего установить $k = i$.

6. Сравнить A_{k-1} и A_k : если A_k окажется меньше, то сделать перестановку, установить $k = k - 1$; если $k > 1$, то продолжить проверку в соответствии с п.6. Если $A_k \geq A_{k-1}$, то продолжить выборку неотсортированных элементов в соответствии с п.4.

4. Метод Шелла.

Этот метод также относится к первой группе. В большинстве случаев методом Шелла массивы обрабатываются быстрее чем методом вставки, так КАК для первоначальной обработки берутся группы записей меньшие всего массива, а при дальнейшей обработке частично упорядочиваются большие группы записей.

Сущность метода заключается в следующем:

1. Делим элементы массива на две группы по k элементов в каждой ($k = n \text{ div } 2$). Присваиваем $i = 1$.

2. Сравниваем A_i и A_{i+k} : если $A_i > A_{i+k}$, то проводим перестановку. Если $A_i < A_{i+k}$, то переходим к следующей паре элементов.

3. Присваиваем $i = i + 1$. Если $i \leq k$, то продолжаем сравнение в соответствии с п.2.

4. Уменьшаем количество элементов в группе $k = k \text{ div } 2$. Если $k = 0$, то сортировка закончена. В противном случае продолжаем сравнение в соответствии с п.2.

5. Метод слияния

Этот метод относится ко второй группе. Он основан на слиянии двух упорядоченных массивов в один, упорядоченный потому же признаку, что и исходные. При этом сначала анализируются первые записи обоих массивов. Та запись, у которой значение ключа меньше, переписывается в новый массив. Оставшаяся запись последовательно сравнивается с записями из другого массива. В новый массив после каждого сравнения попадает запись с меньшим значением ключа. Оставшаяся запись сравнивается с записями из другого массива. Процесс

продолжается до исчерпания записей одного из массивов. Затем остаток другого массива дописывается в новый массив.

Этот метод позволяет отсортировать и один массив, состоящий из n элементов. Для этого массив разбивается на четное число подмассивов и к каждой паре из них применяется метод слияния. Получается вдвое меньше упорядоченных подмассивов. Затем к каждой паре полученных подмассивов снова применяются метод слияния.

6. Древовидная сортировка

Алгоритм древовидной сортировки предполагает последовательное деление исходного файла по выбранному признаку на три подмассива. Первый из них содержит значения признака, меньшие, чем выбранный, второй - равные и большие, третий - только записи со значением выбранного признака. Затем каждый из подмассивов делится по вновь назначаемому признаку на три части. Процесс деления продолжается до тех пор, пока каждый из подмассивов не будет содержать всего одну запись. Следует заметить, что записи в подмассиве просматриваются поочередно то с конца, то с начала; признаки запоминаются для учета при последующем их выборе. Массив после каждого просмотра формируется следующим образом: первый подмассив помещается в начале исходного массива, затем располагается второй массив, и между ними - третий, содержащий всего одну запись.

7. Шейкер-сортировка

Сущность данного метода заключается в следующем: из предложенного массива выбирается минимальный элемент и перемещается в начало массива [1] из оставшихся элементов выбирается максимальный и размещается в конце массива [n].

Из оставшихся элементов ищется минимальный и размещается на [2] позиции.

Из оставшихся элементов, после очередного выбора, ищется максимальный и размещается на [n-1] позицию и т.д.

3. ОБОРУДОВАНИЕ

ПЭВМ IBM PC, SVGA монитор с разрешением не менее 800*600 пикселей; клавиатура; мышь. Среда Free Pascal, Lazarus.

4. ЗАДАНИЕ НА РАБОТУ

0. Ввести и сохранить в файл данные следующей структуры: наименование товара, страна-производитель, цена товара, количество товара, фирма-поставщик, номер отдела универмага. Организовать просмотр исходных данных и вывести список товаров, производимых в данной стране (по выбору), отсортированный по количеству (по убыванию метод пузырька), с указанием цены товара, количества и фирмы-поставщика.

1. Ввести и сохранить в файл данные следующей структуры: регистрационный номер, марка, номер двигателя, год выпуска, дата последнего техосмотра. Организовать просмотр исходных данных и вывести список автомобилей, отсортированный по году выпуска (по возрастанию метод пузырька), с указанием регистрационного номера, марки, года выпуска.

2. Ввести и сохранить в файл данные следующей структуры: тип самолета (пассажирский, грузовой), номер самолета, номер рейса, пункт отправления, пункт назначения, время отправления, время полета. Организовать просмотр исходных данных и вывести список самолетов, отсортированный по номеру рейса (по возрастанию метод Шелла), с указанием номера рейса, пункта отправления, пункта назначения, времени отправления, времени полета.

3. Ввести и сохранить в файл данные следующей структуры: код товара, название товара, количество на складе, страна-производитель, дата выработки, срок годности. Организовать просмотр исходных данных и вывести список товаров, отсортированный по количеству на складе (по возрастанию метод слияния) с указанием кода товара, названия товара, количества на складе.

4. Ввести и сохранить в файл данные следующей структуры: ФИО спортсмена, вид спорта, страна, занятое место и количество набранных баллов. Организовать просмотр исходных данных и вывести список спортсменов данной вида спорта (по выбору), отсортированный по количеству набранных баллов (по убыванию метод вставки), с указанием страны, занятого места и количества набранных баллов.

5. Ввести и сохранить в файл данные следующей структуры: № рейса, направление движения и время отправления автобуса, стоимость билета, ФИО водителя. Организовать просмотр исходных данных и вывести список автобусов, отсортированный по времени отправления автобуса (по возрастанию метод выборки записи с максимальным значением) с указанием № рейса, направления движения, времени отправления.

6. Ввести и сохранить в файл данные следующей структуры: тип поезда (пассажирский, скорый, грузовой), номер поезда, пункт отправления, пункт назначения, время отправления, стоимость билета. Организовать просмотр исходных данных и вывести список поездов, в зависимости от типа поезда (по выбору), отсортированный по номеру поезда (по возрастанию метод обмена), с указанием номера поезда, пункта отправления, пункта назначения.

7. Ввести и сохранить в файл данные следующей структуры: название книги, автор, год издания, количество экземпляров, тематика. Организовать просмотр исходных данных и вывести список книг заданной тематики (по выбору), отсортированный по количеству экземпляров (по убыванию метод вставки), с указанием названия книги, автора, количества экземпляров.

8. Ввести и сохранить в файл данные следующей структуры: название книги, автор, год издания, количество экземпляров, тематика. Организовать просмотр исходных данных и вывести список книг, отсортированный по году издания (по возрастанию метод Шелла), с указанием названия книги, автора, года издания.

9. Ввести и сохранить в файл данные следующей структуры: номер квитанции, Ф.И.О. клиента, марка автомобиля, № автомобиля, вид неисправности, стоимость ремонта. Организовать просмотр исходных данных и вывести список автомобилей, отсортированный по стоимости ремонта (по возрастанию метод обмена), с указанием номера квитанции, Ф.И.О. клиента, вида неисправности, стоимости ремонта.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить теоретические положения.
2. Разработать схему программы решения задачи.
3. Составить программу.
4. Отладить программу. Результаты работы программы показать преподавателю.
5. Оформить отчет.
6. Защитить лабораторную работу перед преподавателем.

6. СОДЕРЖАНИЕ ОТЧЕТА

1. Номер и название лабораторной работы
2. Цель и задачи
3. Задание на работу. Описание задания в соответствии с вариантом.
4. Схема программы
5. Текст программы
6. Результаты и выводы по лабораторной работе

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем заключается особенность метода обмена?
2. В чем заключается суть метода вставки?
3. Чем отличается метод пузырька?
4. В чем суть метода Шелла?
5. В чем особенность метода слияния?
6. Допускается ли применение метода слияния для одного массива?
7. В чем особенность и преимущества метода шейкер-сортировки