

Практическая работа № 2

Создание программ с разветвляющимися алгоритмами

Цель работы: Изучить структуру условного, составного операторов языка Pascal, научиться создавать программы, используя полученные знания.

Содержание работы.

Основные понятия.

1 Разветвляющимся называется такой алгоритм, в котором выбирается один из нескольких возможных вариантов вычислительного процесса. Каждый подобный путь называется ветвью алгоритма.

2 Признаком разветвляющегося алгоритма является наличие операций проверки условия. Различают два вида условий - простые и составные.

3 Простым условием (отношением) называется выражение, составленное из двух арифметических выражений или двух текстовых величин (иначе их еще называют операндами), связанных одним из знаков:

< – меньше, чем...

> – больше, чем...

<= – меньше, чем... или равно

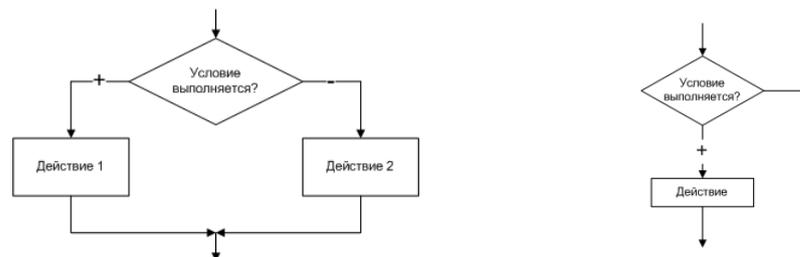
>= – больше, чем... или равно

<> – не равно

= – равно

4 Составное условие - объединение нескольких условий в одну группу.

5 В блок-схемах разветвленные алгоритмы изображаются так, как показано на рисунках



6 В качестве условного оператора используется конструкция **if-else** или ее сокращенный вариант **if**. Также существует оператор выбора **case**, который имеет более специфичное применение.

7 Когда выполнение основной ветки программы доходит до условного оператора **if-else**, то в зависимости от результата логического выражения в его заголовке выполняются разные блоки кода. Если логическое выражение вернуло **true**, то выполняется один блок (в Паскале начинается со слова **then**), если **false** – то другой (начинается со слова **else**). После выполнения одного из вложенных блоков кода, ход программы возвращается в основную ветку. Другой вложенный блок не выполняется.

8 Непосредственно после **then** может стоять только один оператор. При необходимости выполнения нескольких операторов они должны быть заключены в операторные скобки **begin-end**. Пример программы, которая меняет значения переменных местами, только если эти значения различны. Блок **if** содержит четыре выражения, поэтому заключен в **begin-end**.

9 Допустимо вложение одного оператора **if** (или **if-else**) в другой. При этом следует соблюдать осторожность, т.е. бывает трудно определить какому **if** (внешнему или внутреннему) принадлежит ветка **else**. Рекомендуют использовать вложенную конструкцию **if**, только в ветке **else**. К тому же в языке Паскаль действует следующее правило: каждому **then** соответствует ближайшее **else**, не задействованное при установлении соответствия с другим **then**.

Задание

- 1 Составить и записать алгоритм решения задачи в графическом и словесно-формульном виде
- 2 Написать программу на языке Паскаль
- 3 Ввести программу и запустить её на исполнение.
- 4 Протестировать алгоритм с различными данными.
- 5 Отладить программу (найти и исправить возможные ошибки).
- 6 Проанализировать программу и сделать вывод.

Пример задачи.

Исходные данные:

Дано четырехзначное число. Проверить истинность высказывания: «Данное число читается одинаково слева направо и справа налево».

Решение:

Для того, чтобы число читалось одинаково слева направо и наоборот, должны совпадать первая и четвертая цифры, а также вторая и третья. Необходимо выделить цифры из числа для сравнения, например

$$2592 = 2000 + 500 + 90 + 2 = 2 \cdot 1000 + 5 \cdot 100 + 9 \cdot 10 + 2$$

В общем виде исходное число $tcde = t \cdot 1000 + c \cdot 100 + d \cdot 10 + e$, тогда

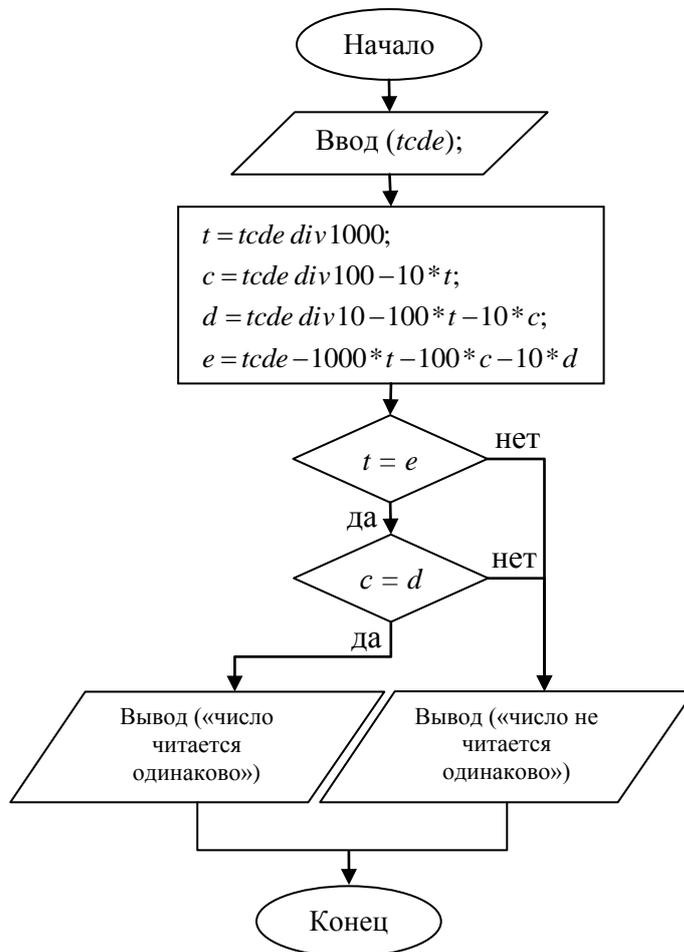
$$t = tcde \operatorname{div} 1000$$

$$d = tcde \operatorname{div} 10 - 100 * t - 10 * c$$

$$c = tcde \operatorname{div} 100 - 10 * t$$

$$e = tcde - 1000 * t - 100 * c - 10 * d$$

1 Алгоритм в графическом и словесном виде:



1. Начало;
2. Ввод ($tcde$);
3. $t = tcde \operatorname{div} 1000$;
4. $c = tcde \operatorname{div} 100 - 10 * t$;
5. $d = tcde \operatorname{div} 10 - 100 * t - 10 * c$;
6. $e = tcde - 1000 * t - 100 * c - 10 * d$
7. Если $t = e$ то
 - { Если $c = d$ то
 - вывод («число читается одинаково»);}
 - Иначе вывод («число не читается одинаково»);
8. Конец.

2 Программа на языке Паскаль:

```

program vibor;
var Tcde, T, C, D, E:integer;
begin
  write('Введите четырехзначное число ');
  readln(Tcde);
  T := Tcde div 1000;
  C := Tcde div 100 - T * 10;
  D := Tcde div 10 - T * 100 - C * 10;
  E := Tcde - T * 1000 - C * 100 - D * 10;
  if T=E then
  begin
    if C=D then

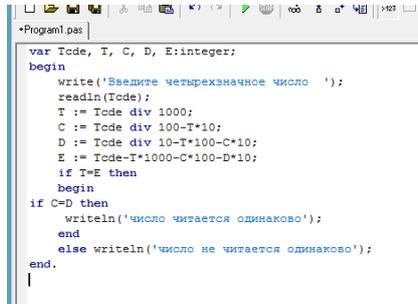
```

```

        writeln('число читается одинаково');
    end
    else writeln('число не читается одинаково');
end.

```

3 Программный код для запуска

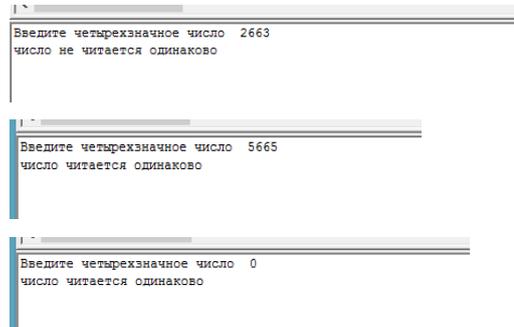


```

Program1.pas
var Tcde, T, C, D, E:integer;
begin
    write('Введите четырехзначное число ');
    readln(Tcde);
    T := Tcde div 1000;
    C := Tcde div 100-T*10;
    D := Tcde div 10-T*100-C*10;
    E := Tcde-T*1000-C*100-D*10;
    if T=E then
        begin
            if C=D then
                writeln('число читается одинаково');
            end
        else writeln('число не читается одинаково');
    end.

```

4 Тестирование программы с различными данными:



```

Введите четырехзначное число 2663
число не читается одинаково

Введите четырехзначное число 5665
число читается одинаково

Введите четырехзначное число 0
число читается одинаково

```

5 Программа выдает неверный результат при неправильном вводе данных, новый код:

```

program vibor;
label m1;
var Tcde, T, C, D, E:integer;
begin
    m1:
        write('Введите четырехзначное число ');
        readln(Tcde);
        if Tcde div 1000=0 then
            begin
                write('Число задано неверно ');
                goto m1;
            end
        T := Tcde div 1000;
        C := Tcde div 100-T*10;
        D := Tcde div 10-T*100-C*10;
        E := Tcde-T*1000-C*100-D*10;
        if T=E then
            begin
                if C=D then
                    writeln('число читается одинаково');
                end
            end
end.

```

```
else writeln('число не читается одинаково');
```

```
end.
```

```
Program1.pas
program vibor;
label m1;
var Tode, T, C, D, E:integer;
begin
  m1:
  write('Введите четырехзначное число ');
  readln(Tode);
  if Tode div 1000=0 then
  begin
    write('Число задано неверно ');
    goto m1;
  end;
  T := Tode div 1000;
  C := Tode div 100-T*10;
  D := Tode div 10-T*100-C*10;
  E := Tode-T*1000-C*100-D*10;
  if T=E then
  begin
    if C=D then
      writeln('число читается одинаково');
    end
  else writeln('число не читается одинаково');
  end.
```

```
<
Введите четырехзначное число 2
Число задано неверно Введите четырехзначное число 1248
число не читается одинаково

Введите четырехзначное число 352
Число задано неверно Введите четырехзначное число 1221
число читается одинаково

Введите четырехзначное число 0
число читается одинаково
```

6 Вывод: Программа, осуществляющая алгоритм, работает правильно, но если пользователь неправильно ввел четырехзначное число, ответ выдает неверный, необходимо добавить проверку исходного числа.

Задания к практической работе.

1 Даны числа x , y . Проверить истинность высказывания: «Точка с координатами (x, y) лежит во второй координатной четверти».

2 Даны числа x , y . Проверить истинность высказывания: «Точка с координатами (x, y) лежит в четвертой координатной четверти».

3 Даны числа x , y . Проверить истинность высказывания: «Точка с координатами (x, y) лежит во второй или третьей координатной четверти».

4 Даны числа x , y . Проверить истинность высказывания: «Точка с координатами (x, y) лежит в первой или третьей координатной четверти».

5 Даны два целых числа: A , B . Проверить истинность высказывания: «Справедливы неравенства $A > 2$ и $B < 3$ ».

6 Даны два целых числа: A , B . Проверить истинность высказывания: «Справедливы неравенства $A > 0$ или $B < -2$ ».

7 Даны три целых числа: A , B , C . Проверить истинность высказывания: «Справедливо двойное неравенство $A < B < C$ ».

8 Даны три целых числа: A , B , C . Проверить истинность высказывания: «Число B находится между числами A и C ».

9 Даны три целых числа: A , B , C . Проверить истинность высказывания: «Каждое из чисел A , B , C положительное».

10 Даны три целых числа: A , B , C . Проверить истинность высказывания: «Хотя бы одно из чисел A , B , C положительное».

11 Даны три целых числа: A , B , C . Проверить истинность высказывания: «Ровно одно из чисел A , B , C положительное».

12 Даны три целых числа: A , B , C . Проверить истинность высказывания: «Ровно два из чисел A , B , C являются положительными».

13 Проверить истинность высказывания: «Среди трех данных целых чисел есть хотя бы одна пара совпадающих».

14 Проверить истинность высказывания: «Среди трех данных целых чисел есть хотя бы одна пара взаимно противоположных».

15 Дано трехзначное число. Проверить истинность высказывания: «Все цифры данного числа различны».

- 16 Дано трехзначное число. Проверить истинность высказывания: «Цифры данного числа образуют возрастающую последовательность».
- 17 Дано трехзначное число. Проверить истинность высказывания: «Цифры данного числа образуют убывающую последовательность».
- 18 Дано целое число. Если оно является положительным, то прибавить к нему 1; если отрицательным, то вычесть из него 2; если нулевым, то заменить его на 10. Вывести полученное число.
- 19 Даны три целых числа. Найти количество положительных чисел в исходном наборе.
- 20 Даны три целых числа. Найти количество отрицательных чисел в исходном наборе.
- 21 Даны два числа. Если их значения не равны, то присвоить каждому сумму этих значений, а если равны, то присвоить числам нулевые значения. Вывести новые значения A и B .
- 22 Даны два числа. Если их значения не равны, то присвоить каждому большее из этих значений, а если равны, то присвоить числам нулевые значения. Вывести новые значения A и B .
- 23 Даны три числа. Найти наименьшее из них.
- 24 Даны три числа. Найти среднее из них (то есть число, расположенное между наименьшим и наибольшим).
- 25 Даны три числа. Вывести вначале наименьшее, а затем наибольшее из данных чисел.
- 26 Даны два целых числа: A , B . Проверить истинность высказывания: «Справедливы неравенства $A > 10$ и $B < 0$ ».
- 27 Даны два целых числа: A , B . Проверить истинность высказывания: «Справедливы неравенства $A > 0$ и $B < -3$ ».
- 28 Даны три числа. Найти наибольшее из них.
- 29 Даны два целых числа: A , B . Проверить истинность высказывания: «Справедливы неравенства $A > 5$ и $B < -5$ ».
- 30 Даны два целых числа: A , B . Проверить истинность высказывания: «Справедливы неравенства $A > 1$ и $B < -3$ ».

ИНСТРУКЦИОННАЯ КАРТА

для проведения практической работы № 2

Тема занятия: *создание программ с разветвляющимися алгоритмами*

Цель выполнения задания: *изучить структуру условного, составного операторов языка Pascal, научиться создавать программы, используя полученные знания*

Необходимо знать: *основные свойства алгоритмов, правила написания блок-схем и словесно-формульного способа представления алгоритма, операторы проверки условия на языке Паскаль*

Необходимо уметь: *применять основные свойства алгоритмов, правила написания блок-схем и словесно-формульного способа представления алгоритма, операторы проверки условия на языке Паскаль*

Оборудование (приборы, материалы, дидактическое обеспечение): *методические рекомендации к выполнению работы; задание и инструкционная карта для проведения практического занятия*

Компьютерные программы: *программы Borland Pascal, Pascal ABC, Ms Word*

Теория: *Для выполнения заданий по данной теме необходимо предварительно изучить теоретические материалы, а также методические рекомендации к выполнению работы*

Порядок выполнения задания, методические указания: *- ознакомиться с теоретическими положениями по данной теме; - изучить схему решения задач; - выполнить задания практической работы; - сформулировать вывод*

Дополнительные задания: *Могут быть сформулированы по ходу занятия*

Содержание отчета: *отчет по практической работе должен содержать: основные определения, рассуждения по решению задач, ответ; вывод по работе*

Контрольные вопросы: 1 Способы представления алгоритма 2 Что такое разветвляющийся алгоритм? 3 Какие бывают условия? 4 Что такое простое условие? 5 Что такое составное условие? 6 Как в блок-схемах изображаются ветвления? 7 Какие операторы в языке Паскаль используются для проверки условия? 8 Как работает

оператор условия? 9 Как выполняется проверка условия в неполном условном операторе?

10 Как выполняется проверка условия во вложенных условных операторах?

Литература:

1 Федотова Е.Л. Информационные технологии в профессиональной деятельности, Москва: ИД «ФОРУМ» – ИНФРА-М, 2011

2 О. В. Горбатова. Информатика: Учебник для техникумов и колледжей ж.-д. транспорта. - М: ГОУ «Учебно-методический центр по образованию на железнодорожном транспорте», 2008. - 242 с.

3 Н. Угринович Информатика и информационные технологии –М. Бином 2003 г.

4 В.Ю. Микрюков Информация. Информатика. Компьютер. Информационные системы. Сети Ростов-на-Дону. Феникс. 2007 г.

5 <http://linux-mgd.3dn.ru/>

6 [http:// pas1.ru/](http://pas1.ru/)

7 <http://life-prog.ru/>