Задание для студентов гр. ИС 2 на период с 27.04.2020 – 28.04.2020 (4 часа – 2 пары)

Дисциплина «Математика»
Преподаватель Токарская М.С.
Почта для обратной связи: maya_tok@mail.ru
Тел. 89147174421 – WhatsApp – если есть вопросы.

Все задания отправлять на почту!!!!!

Задание:

Выполнить лабораторную работу в среде программирования Турбо Паскаль

Лабораторная работа № 2 «Организация ветвления в среде Турбо Паскаль»

Цель работы: научиться работать в среде программирования Турбо Паскаль, составлять программы ветвления.

План работы:

1. Повторить теоретический материал по составлению блок – схем ветвления и основной структуре программы (см. предыдущие занятие);
2. Ознакомиться с дополнительным материалом и приведенным примерами;
3. Записать лекцию полностью с примерами в тетрадь;
4. Выполнить задание 1.
5. Выполнить задание 2.
6. Письменно ответить на контрольные вопросы.

Краткий материал.

В языке Паскаль используется два оператора для реализации условных переходов - IF и CASE, а также оператор безусловного перехода GOTO. Они позволяют нарушить последовательный порядок выполнения инструкций программы.

Оператор условного перехода

Оператор условного перехода в Турбо Паскаль имеет вид:

if *условие* then *оператор 1* else *оператор 2*;

Условие - это логическое выражение, в зависимости от которого выбирается одна из двух альтернативных ветвей алгоритма.

Если значение условия истинно (TRUE), то будет выполняться *оператор 1*, записанный после ключевого слова then.

В противном случае будет выполнен *оператор 2*, следующий за словом else, при этом *оператор 1* пропускается.

После выполнения указанных операторов программа переходит к выполеннию команды, стоящей непосредственно после оператора if.

Необходимо помнить, что перед ключевым словом else точка с запятой никогда не ставится!

else - часть в операторе if может отсутствовать:

if условие then оператор 1;

Тогда в случае невыполнения логического условия управление сразу передается оператору, стоящему в программе после конструкции if.

Следует помнить, что синтаксис языка допускает запись только одного оператора после ключевых слов then и else, поэтому группу инструкций обязательно надо объединять в составной оператор (окаймлять операторными скобками begin ... end). В противном случае возникает чаще всего логическая ошибка программы, когда компилятор языка ошибок не выдает, но программа тем не менее работает неправильно.

Примеры.

```
1.  if x > 0 then modul := x else modul := -x;


2.  if k > 0 then WriteLn('k - число положительное');


3.  if min > max then begin
    t := min;
    min := max;
    max := t;
  end;
```

Оператор выбора

Часто возникают ситуации, когда приходится осуществлять выбор одного из нескольких альтернативных путей выполнения программы. Несмотря на то, что такой

выбор можно организовать с помощью оператора if .. then, удобнее воспользоваться специальным оператором выбора.

Его формат:

case *выражение* of
** *вариант* : *оператор*;**
** ...**
** *вариант* : *оператор*;**
end;

или

case *выражение* of
** *вариант* : *оператор*;**
** ...**
** *вариант* : *оператор*;**
** else *оператор***
end;

Выражение, которое записывается после ключевого слова case, называется ***селектором***, оно может быть любого перечисляемого типа.

Вариант состоит из одной или большего количества констант или диапазонов, разделенных запятыми. Они должны принадлежать к тому же типу, что и селектор, причем недопустимо более одного упоминания *варианта* в записи инструкции case. Из перечисленного множества *операторов* будет выбран только тот, перед которым записан *вариант*, совпадающий со значением селектора. Если такого *варианта* нет, выполняется *оператор*, следующий за словом else (если он есть).

Пример

```
case ch of
  'A'..'Z', 'a'..'z' : WriteLn('Буква');
  '0'..'9'           : WriteLn('Цифра');
  '+', '-', '*', '/' : WriteLn('Оператор');
  else WriteLn('Специальный символ')
```

```
end;
```

Оператор безусловного перехода

Помимо операторов условного перехода существует также оператор безусловного перехода goto.

Формат:

goto *метка*

Оператор goto переходит при выполнении программы к определенному оператору программы, перед которым находится *метка*.

Метка должна быть описана в разделе описания меток той программы (процедуры или функции), в которой она используется. Нельзя перейти из одной процедуры или функции в другую.

Необходимо, чтобы в программе существовал оператор, отмеченный указанной меткой. Она записывается перед оператором и отделяется от него двоеточием.

Пример

```
label 1;
begin
  ...
  goto 1;
  ...
  1: WriteLn('Переход к метке 1');
end.
```

Учтите! Само понятие структурного программирования и общепринятый стиль программирования на структурных языках **НЕ ПРИВЕТСТВУЕТ** применение меток и операторов перехода в программах. Это затрудняет понимание программы как автором, так и потребителями, кроме того, применение меток отрицательно сказывается на эффективности генерируемого кода.

Задание 1. Перед вами программа ветвления. Необходимо по программе:

1. Описать задачу, которую она решает;
2. Составить блок-схему решения задачи;
3. Составить программу на псевдокоде;

4. Набрать программу, провести компиляцию и запустить программу на исполнение; сделать скрин экрана с программой и результатом.

```
program number19;
uses crt;
    var A:real;
begin
    clrscr;
    write ('Введите число A: ');
    readln (A);
    if A>=0 then
       writeln ('Вы ввели положительное число')
    else
       Writeln('Вы ввели отрицательное число');
    readln
end.
```

Задание 2.

Составить блок-схему и программу для вычисления значения в соответствии со своим вариантом. Запустить программу на исполнение, предварительно проведя компиляцию. Сделать скрин программы и результата.

1. Вычислить z:

$$z=\begin{cases}\sin x & если \quad -\frac{\pi}{2}\le x\le\frac{\pi}{2}\\ tg^2x & если \quad \frac{\pi}{2}\le x\le\pi\\ x & в\_остальных\_случаях\end{cases}$$

2. Вычислить y:

$$y=\sqrt{x}+\frac{1}{x^2-5x+6}$$

3. Вычислить y:

$$y=\begin{cases}x+1 & если \quad x<1\\ 2x & если \quad x>2\\ x & в\_остальных\_случаях\end{cases}$$

4. Вычислить N:

$$N=\begin{cases}17-R, если & R\ge 0\\ 17 \quad если\ в\_остальных\_случаях\end{cases}$$

5. Вычислить T

$$T=\begin{cases}tgx, \_если & x=1\\ e & в\_остальных\_случаях\end{cases}$$

6. Вычислить y :

$$y=\begin{cases}\ln x & , \quad если \quad x >= 1\\ 1 & , \quad если \quad -1 < x < 1\\ e^x & , \quad если \quad x <= -1\end{cases}$$

7. Вычислить k:

$$k=\begin{cases}\sin x & , \quad если \quad x>2\\ 3x & , \quad если \quad -2<x<2\\ x & , \quad если \quad x<=-2\end{cases}$$

8. Вычислить z :

$$z=\begin{cases}8x^2+x & , \quad если\ x>0\\ 2x & , \quad если\ x<4\\ 15 & , \quad в\ остальных\ случаях\end{cases}$$

9. Вычислить f(x):

$$f(x)=\begin{cases}4{,}5x-7 & , \text{ если } x<=2\\ \sin(x-1) & , \text{ если } x>2\end{cases}$$

10. Вычислить y:

$$y=\begin{cases}|x-1| & , \text{ если } x<=3\\ 12{,}5x-3{,}4 & , \text{ если } x>3\end{cases}$$

11. Вычислить к:

$$к=\begin{cases}\sqrt{(5-x)} & , \text{ если } x<=3\\ 2x & , \text{ если } x>3\end{cases}$$

12. Вычислить y:

$$y=\begin{cases}-1 & , \text{ если } x<0\\ 0 & , \text{ если } x=0\\ 1 & , \text{ если } x>0\end{cases}$$

13. Вычислить функцию:

$$F(z)=\begin{cases}Z^2 & , \text{ если } z>=1\\ 1-z & , \text{ если } z<1\end{cases}$$

14. Вычислить функцию:

$$F(x)=\begin{cases}0 & для \text{ целого } x\\ \operatorname{ctg}\pi x & для \text{ нецелого } x\end{cases}$$

15. Вычислить y:

$$y=\begin{cases}10-x & , \text{ если } x<=-5\\ tg(\pi x) & , \text{ если } x>-5\end{cases}$$

16. Вычислить функцию:

$$y=\begin{cases}x^3+1, & x>2\\ 4, & x=2\\ 2x-1, & x<2\end{cases}$$

17. Вычислить значение переменной:

$$V=\begin{cases}\sqrt{R^3}, & R\ge 6\\ R^2-1, & R<6\end{cases}$$

Контрольные вопросы.

1. Как программируется на Паскале полное и неполное ветвление?
2. Что такое составной оператор?
3. В каких случаях составной оператор используется в операторе ветвления?
4. Опишите, для чего нужен оператор GOTO.