

Методические указания



Форма
Ф СО ПГУ 7.18.2/05

Министерство образования и науки Республики Казахстан
Павлодарский государственный университет им. С. Торайгырова
Кафедра информатики и информационных систем

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ ЗАНЯТИЯМ

по дисциплине Программирование

для студентов специальности 050601 – Математика

Павлодар

Лист утверждения к
методическим указаниям



Форма
Ф СО ПГУ 7.18.1/05

УТВЕРЖДАЮ
Декан ФФМиИТ
_____ С.К. Тлеуенов
«__» _____ 200_ г.

Составитель: профессор, к.п.н., доцент Криворучко В. А.
(должность, уч. степень, звание, подпись)

Кафедра Информатика и информационные системы

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ ЗАНЯТИЯМ

по дисциплине Программирование

для студентов специальности 050601 – Математика

Рекомендована на заседании кафедры от «__» _____ 200_ г.

Протокол № _____

Заведующий кафедрой _____ Нурбекова Ж.К.
(подпись)

Одобрена методическим советом факультета ФФМиИТ

«__» _____ 200_ г. Протокол № _____

Председатель МС _____ Даутова А.З.
(подпись)

Лабораторная работа №1 «Освоение работы в системе программирования Turbo Pascal»

Цель: ознакомиться со структурой и принципами работы системы программирования Turbo Pascal

Теоретические сведения

Система программирования Turbo Pascal представляет собой единство двух частей: компилятора с языка программирования Pascal и некоторую программную оболочку - среду Turbo Pascal, представляющую разнообразные сервисные услуги.

Файл turbo.exe содержит среду программирования – текстовый редактор, компилятор, компоновщик. В систему программирования входит основная библиотека, располагающаяся в файле turbo.tpl, справочная служба – файл turbo.hlp, отладчики программы реализующие способы поиска ошибок.

В библиотеку могут быть включены файлы для работы в графическом режиме: graph.tpu – модуль с графическими процедурами и функциями, файлы с расширением .bgi, драйверы разных типов видеосистем, файлы с расширением .chr, содержащие шрифты.

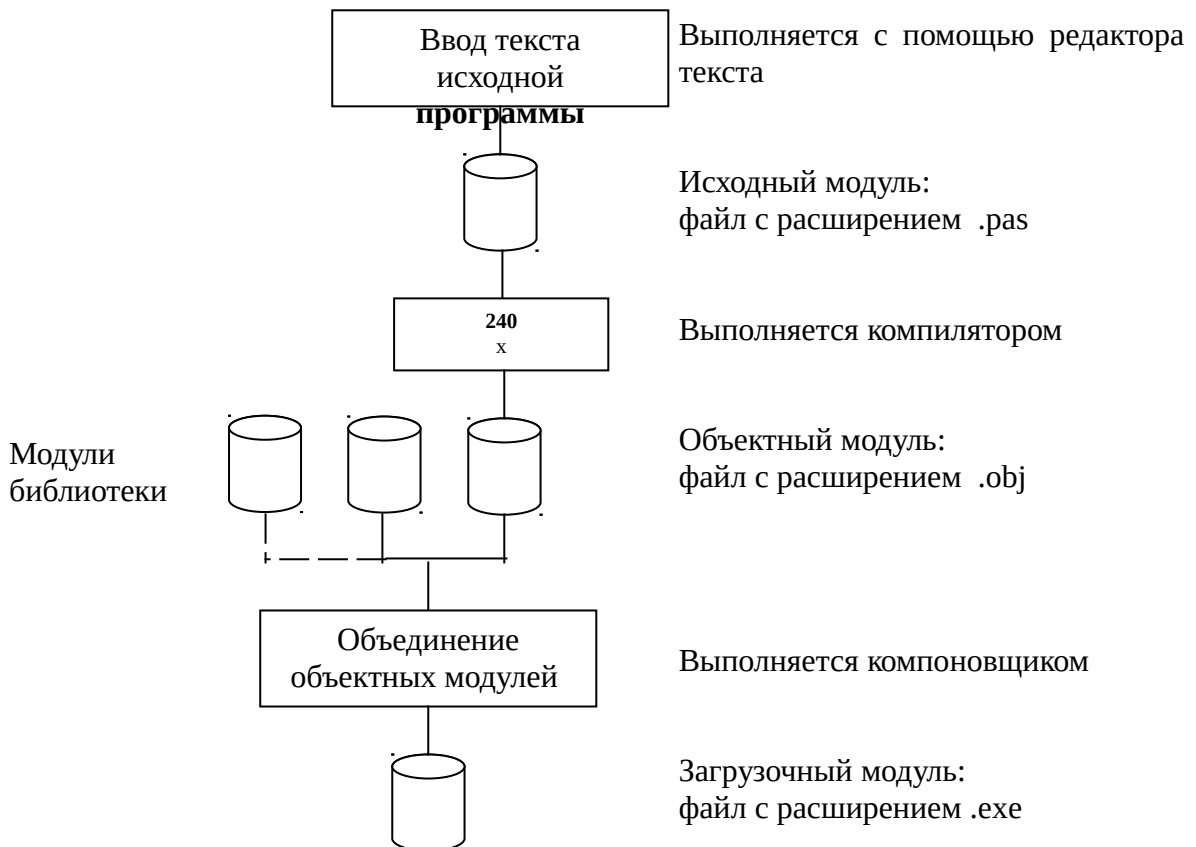


Рисунок 1 - Схема подготовки программы в среде Turbo Pascal

Подготовка программы начинается с формирования файла с исходным модулем (например: In.pas) и выполняется с использованием редактора текста. Затем происходит трансляция исходного модуля с помощью компилятора. В результате трансляции получается файл с объектным модулем (In.obj), который вместе с другими модулями (например, модулями библиотеки) объединяются в один загрузочный модуль (In.exe) с соответствии с рисунком 1.

Задание 1. Ознакомьтесь с запуском и структурой окон системы программирования Turbo Pascal.

Для исполнения написанной вами программы на компьютере необходимо загрузить в оперативную память компьютера систему программирования Turbo Pascal. Для этого выполните следующую команду:

C:\TP7\BIN\turbo.exe

Процесс исполнения программы состоит из двух частей:

1. Ввода текста программы в память машины посредством текстового редактора Turbo Pascal;
2. Запуска программы для исполнения.

После загрузки интегрированной среды разработки (**IDE** – Integrated Design Environment) системы Turbo Pascal на экране компьютера может возникнуть картинка в соответствии с рисунком 2). Это случай, когда в системе нет ни одного окна, поэтому поверхность экрана (*Desktop*) будет пуста (в конкретных случаях начальное изображение экрана может незначительно отличаться от приведенного):

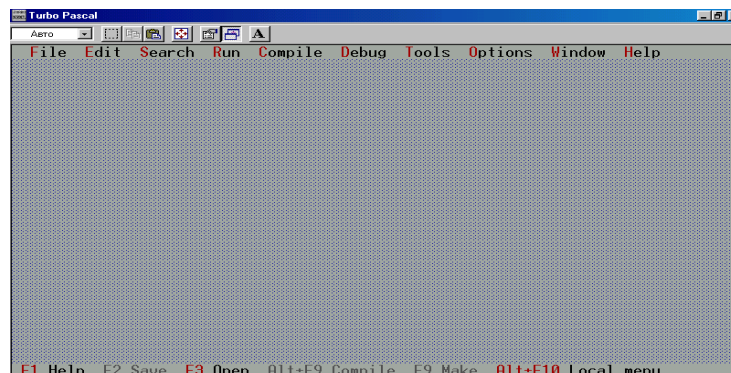


Рисунок 2 - Экран системы Turbo Pascal

Вверху экрана находится строка главного меню. Переход к главному меню осуществляется нажатием клавиши **F10** или щелчком мышки на

каком-либо слове меню. После выполнения команды: **File** □ **New** появится основное рабочее поле системы Turbo Pascal в соответствии с рисунком 3, состоящее из главного меню в верхней строке экрана, нижняя строка экрана содержит оперативную подсказку по так называемым «горячим клавишам» и окно редактора. Набор текста в нем осуществляется, как и в любом текстовом редакторе.

Среда программирования Turbo Pascal позволяет создавать тексты программ, редактировать, компилировать их (переводить на язык понятный компьютеру), находить и исправлять ошибки, выполнять отлаженную программу.

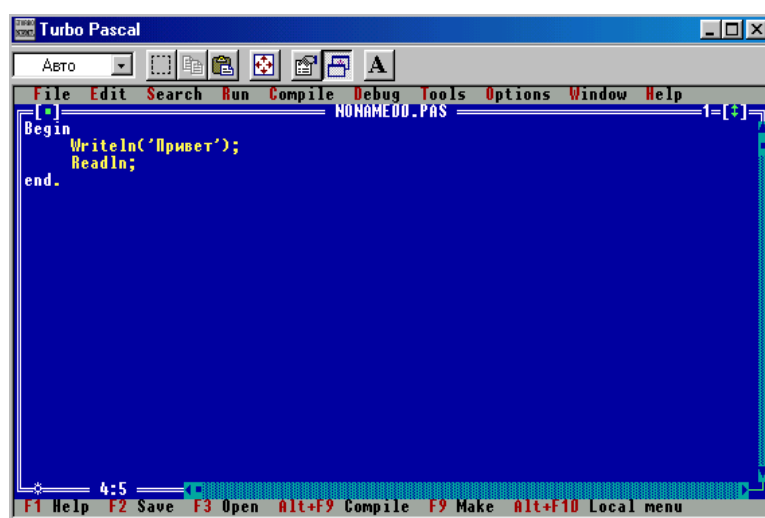


Рисунок 3 - Рабочий экран системы Turbo Pascal

Таблица 1 - Основные позиции главного меню

File (Файл)	Работа с файлами
Edit (Редактирование)	Редактирование текстов
Search (Поиск)	Поиск информации в тексте
Run (Запуск)	Выполнение программы
Debug (Отладка)Формирование программы	Режимы и средства отладки
Compile (Компилирование)	
Tools (Инструменты)	Инструменты отладки
Options (Параметры)	Установка параметров и режимов
Window (Окно)	Работа с окнами

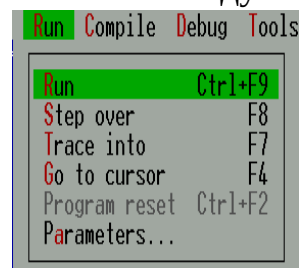
Help (Справка)	Справочная подсистема

Выбор вариантов в этом меню производится с помощью клавиши **F10** и клавиш-стрелок вправо и влево и фиксируется нажатием клавиши **ENTER**. Результатом будет появление на экране соответствующих подменю. Например, чтобы запустить или перезапустить программу с самого начала, надо выполнить команду:

F10 □ **Run** □ **Run**

или

нажать комбинацию клавиш **Ctrl + F9**.



Выход из подменю и выход из главного меню произойдет после нажатия клавиши **ESC**. Полное описание всех этих подменю можно получить на экране компьютера, обратившись к электронному справочнику через подменю **Help (Справка)**.

Задание 2. Освойте работу со справочной системой Turbo Pascal.

Существенным достоинством системы Turbo Pascal является наличие в ней встроенного электронного справочника на русском языке, позволяющего оперативно получить как информацию о правилах работы с системой, так и детальные сведения о языке Pascal.

Обратиться к электронному справочнику системы Turbo Pascal можно двумя способами: нажав клавишу **F1**, либо выбрав в правом верхнем углу главного меню подменю - **Help (Справка)**. Результатом этих действий будет появление на экране основного справочника по системе Turbo Pascal в соответствии с рисунком 4.

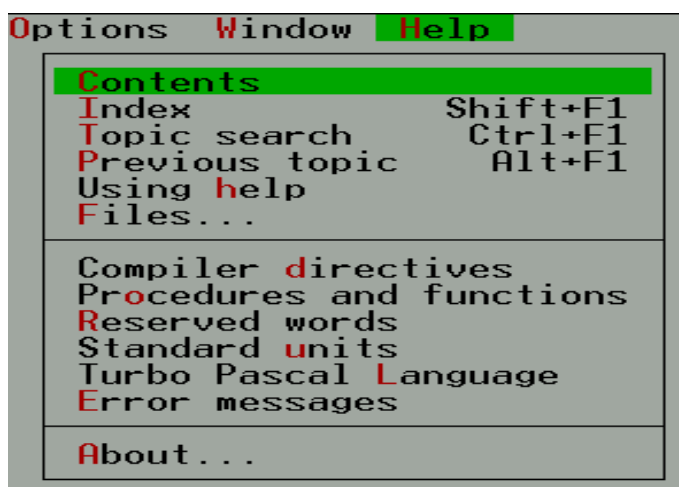


Рисунок 4 – Содержание справочника по системе Turbo Pascal

Далее можно перейти в окно справки в соответствии с рисунком 5.

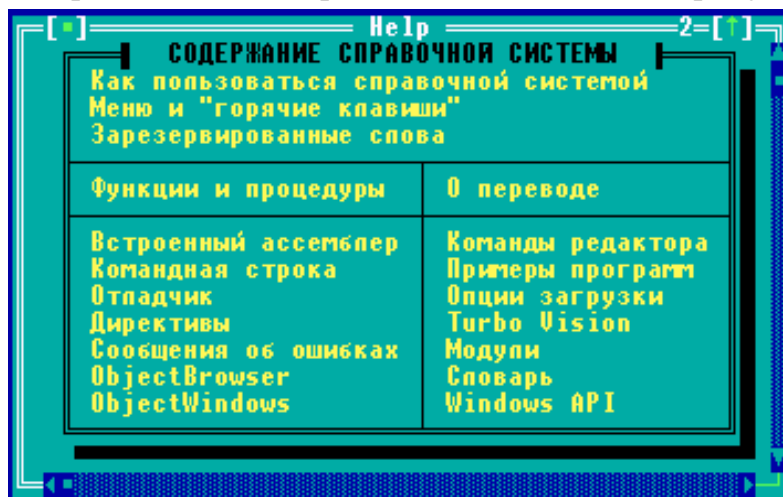


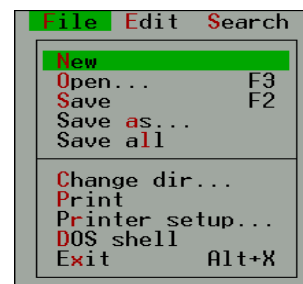
Рисунок 5 - Окно справки системы Turbo Pascal

Выбор необходимой справки производится с помощью мыши или нажатием на клавиатуре клавиши **TAB**, а затем клавиши **ENTER**.

Задание 3. Освойте работу с файлами в системе Turbo Pascal

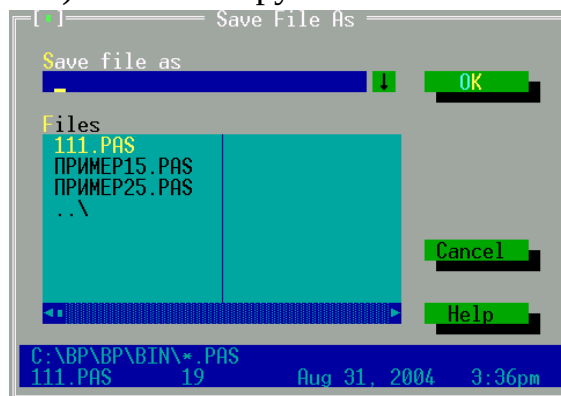
Для работы с файлами на магнитных дисках необходимо перейти к главному меню, нажав клавишу **F10**, и выбрать в нем подменю **File (Файл)**. Подменю имеет вид:

С помощью этого подменю производится поиск программ на дисках и загрузка их в оперативную память, запись программ на диски, перезапись программ на диски под новыми именами, вывод текстов программ на печать, временный выход в DOS и выход из среды Turbo Pascal.



Выбор пункта подменю **New (Новый)** приводит к открытию нового окна редактирования с заданным по умолчанию именем файла **NONAMExx.PAS** (xx – номер от 00 до 99) и активизирует его.

Для записи вновь введенной программы необходимо в подменю **File (Файл)** выбрать пункт **Save (Сохранить)** и нажать клавишу **ENTER**. В окне сохранения файла нужно ввести имя файла, под которым будет записан



(перезаписан) текст программы в соответствии с рисунком 6.

Рисунок 6 - Окно сохранения файла

Для поиска программ, предварительно записанных на магнитные диски, в подменю **File (Файл)** необходимо выбрать пункт **Open (Открыть)** и нажать **ENTER**. В результате на экране появится перечень имен программ на Turbo Pascal, хранящихся в текущем оглавлении в соответствии с рисунком 7.

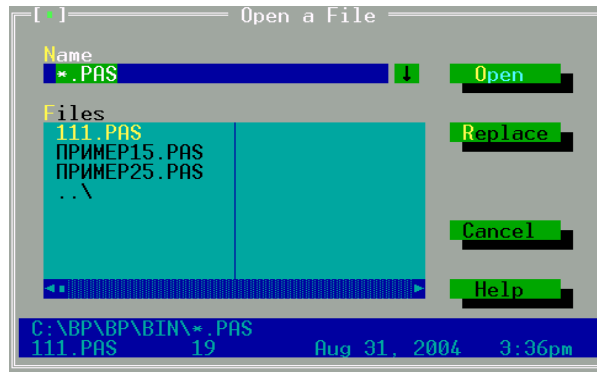


Рисунок 7. Окно открытия файла

Перебор этих имен проводится нажатием клавиш стрелки. Загрузка текста программы (файла) произойдет после нажатия клавиши **ENTER**. Отказ от загрузки текста - нажатие клавиши **ESC**. Клавиша **TAB** служит для перемещения между окнами экрана.

Вывод на печать текста при наличии принтера, подключенного к компьютеру, произойдет при выборе в подменю **File (Файл)** пункта **Print (Печать)**.

Последний пункт в подменю **File (Файл)** - **Exit (Выход)**. Выбор этого пункта приведет к завершению работы системы Turbo Pascal.

Задание 4. Самостоятельно выполните следующие действия:

- запустите систему программирования Turbo Pascal;
- откройте заготовку файла `goba1.pas`;
- запустите программу на выполнение;
- найдите с помощью справки как перейти в окно результата;
- сохраните программу под именем `goba2.pas`;
- запустите файл `primer1.pas`, хранящийся на диске A:\;
- закройте систему программирования Turbo Pascal.

Контрольные вопросы

1. Назовите компоненты, входящие в систему программирования Turbo Pascal.

2. Как запустить систему программирования Turbo Pascal?
3. Назначение основных позиций главного меню.
4. Объясните, как вызывать и пользоваться справочной системой Turbo Pascal?
5. Какие операции можно производить с файлами в системе Turbo Pascal?
6. Как запустить программу на выполнение?
7. Как сохранить программу на диске?
8. Как открыть программу, хранящуюся на диске?
9. Как осуществить выход из системы программирования Turbo Pascal?

Литература:

1[с.197-202], 2[с.164-208]

Лабораторная работа №2

«Программирование линейных алгоритмов и программ»

Цель: сформировать представления о структуре программы в Pascal и навыки по переводу алгоритмов на язык программирования Pascal

Теоретические сведения

При переходе от алгоритмического языка к языку программирования важно сохранить преемственность, т.е. каждой конструкции алгоритмического языка ставится в соответствие инструкция (оператор) языка Pascal. Для решения этой задачи необходимо ввести список команд (операторов Pascal) и выяснить назначение, форматы и применение введенных операторов.

Таблица 2 – Операторы языка Turbo Pascal

Оператор	Форма строки	Назначение, применение
{ }	{Пояснение}	Оператор комментариев
read(читать)	read(список имен);	Оператор ввода данных с клавиатуры. Например: read (a, b); - требует ввода двух чисел
readln	readln(список имен); readln;	После ввода данных курсор переводится на новую строку. Оператор пустого ввода -

		осуществляет перевод курсора на новую строку.
--	--	---

Окончание таблицы 2

имя переменной := выражение	имя переменной := выражение;	Оператор присваивания. Знак «:=» читается «присвоить». Например: a := a+1;
write (писать)	write(список вывода); или write ('подсказка', список вывода);	Оператор вывода данных на экран дисплея. Элементами списка могут быть переменные, числа, выражения и тексты, заключенные в апострофы (одинарные кавычки). Например: а) write ('Результат', 'x=', x); б) write ('x=', x : 6 : 2); - вывод числа в форме с фиксированной точкой.
writeln	writeln; writeln(список вывода);	Оператор пустого вывода - осуществляет перевод курсора на новую строку. Перевод курсора на новую строку после печати

Общий вид структуры программы в Turbo Pascal. Программа состоит из трех частей: заголовка, раздела описаний и раздела операторов.

program имя программы;

label - раздел меток;

const - раздел констант;

type - раздел типов;

var - раздел переменных;

procedure, function - раздел процедур и функций;

begin

раздел операторов

end.

Заголовок содержит служебное слово `program` и имя программы, начинающееся с латинской буквы, и может содержать цифры и знаки подчеркивания.

Раздел описаний предназначен для объявления всех встречающихся в программе данных и их характеристик. Этот раздел содержит в себе другие разделы: описание меток, констант, типов, переменных, процедур и функций. В программе необязательно наличие сразу всех разделов. В простой программе может вообще отсутствовать раздел описаний.

Основным является раздел переменных `var`. В нем указываются имена переменных, используемых в программе, и их тип. Например, в программе используются две целочисленные переменные и одна вещественная. Раздел переменных может иметь вид:

```
var a, b: integer; x: real;
```

Имена переменных одного типа перечисляются через запятую, затем после двоеточия указывается их тип. Описание каждого типа заканчивается точкой с запятой.

Раздел операторов заключается в операторные скобки `begin` и `end`, при этом после `end` ставится точка. В разделе операторов записывается последовательность операторов (команд). Заголовок, разделы описаний, операторы заканчиваются точкой с запятой. После слова `begin` точка с запятой не ставится.

Пример 1. Составить алгоритм и программу для вычисления площади треугольника по формуле Герона, если заданы длины сторон треугольника.

Решение:

Аргументы алгоритма (исходные данные) – длины сторон A , B , C .

Результат – площадь треугольника S .

$$S = \sqrt{P \cdot (P - A) \cdot (P - B) \cdot (P - C)}.$$

Промежуточная величина – полупериметр P определяется:

$$P = \frac{A + B + C}{2}.$$

Рассмотрим соответствие записи алгоритма и программы. Это представляется очень важным, так как в дальнейшем программа должна переводиться с алгоритмического языка. В этом и заключается смысл введения алгоритмического языка, как инструмента

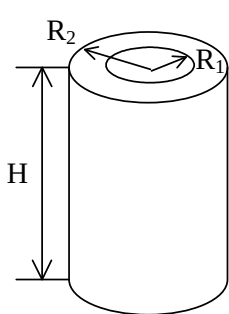
проектирования алгоритмов, для последующего переноса их на компьютер, посредством программы.

Переведем на язык Pascal алгоритм вычисления площади и сопоставим эти два идентичных по смыслу представления алгоритма.

алг вычисление площади	----->	program _7; {Вычисление площади}
(<u>арг</u> <u>вещ</u> A, B, C, <u>рез</u> <u>вещ</u> S)	-->	var a,b,c,p,s : real ;
нач <u>вещ</u> P		begin
<u>ввод</u> A, B, C	----->	readln (a, b, c);
$P := \frac{A+B+C}{2}$	----->	
p:=(a+b+c)/2;		
$S := \sqrt{P \cdot (P - A) \cdot (P - B) \cdot (P - C)}$	-->	s:=sqrt(p*(p-a)*(p-b)*(p-c));
<u>вывод</u> S	----->	writeln ('s=' ,s: 6: 2);
кон	----->	end.

Сравнение показывает нам правила перевода алгоритма с алгоритмического языка на язык Pascal.

Задание 1. Составить программу определения объема цилиндрической втулки (см. рисунок), используя алгоритм:



Для решения воспользуемся формулами:

$$V = V_2 - V_1,$$

$$\text{где } V_1 = \pi R_1^2 \cdot H, \quad V_2 = \pi R_2^2 \cdot H,$$

$$\text{тогда } V = \pi H (R_2^2 - R_1^2),$$

где H – высота, R₁ – внутренний радиус, R₂ – внешний радиус основания втулки.

алг объем втулки (арг вещ R1, R2, H, рез вещ V)

нач вещ л

$$\pi := 3,14$$

ввод R1, R2, H

$$V := \pi H (R_2^2 - R_1^2)$$

ВЫВОД V

КОН

Наберите программу в среде редактора Pascal и исполните ее при $R_1=3,2$; $R_2=5,4$; $H=4,1$, запишите ответ.

Задание 2. Составить программу вычисления функции:

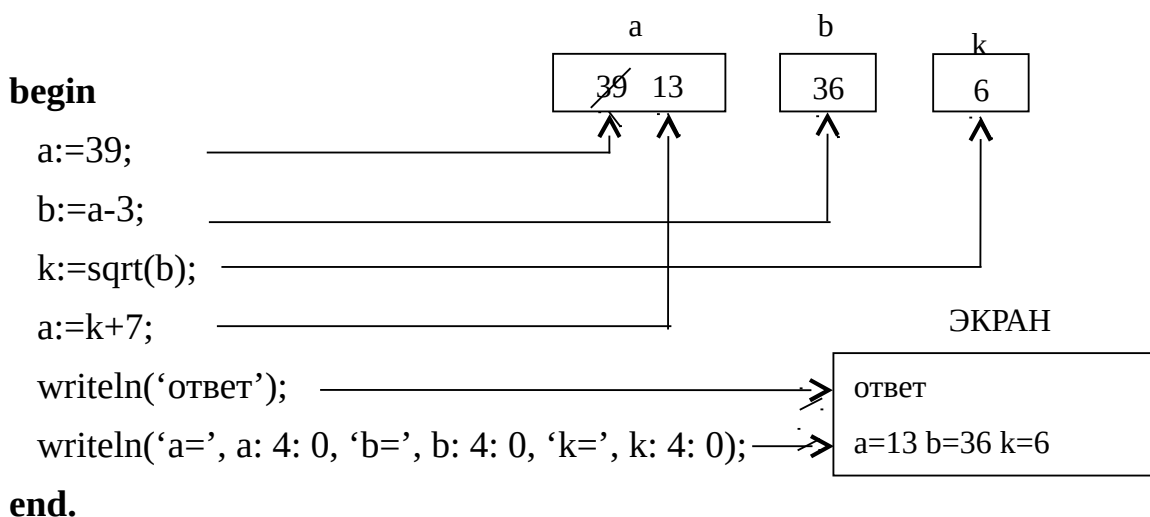
$$Y=X + \sin 50^\circ, \quad \text{где } X=A^2+B^2$$

Исполните программу на компьютере при $A=2,1$; $B=1,5$. Вывести форматированный ответ (два знака после запятой).

Задание 3. Исполнить программу “вручную”.

Рассмотрите последовательность выполнения фрагмента следующей программы.

Память компьютера



end.

Прочитать строку, пояснить, что она делает.

Задание 4. Какой результат отобразится на экране после выполнения фрагмента программы (без компьютера)?

```
a:=81;
b:=sqr(a);
c:= 'ответ:.';
a:=b/3*2;
b:=a+b;
writeln(a/3; b/3);
writeln('a=', a, 'b=', b);
```

Решить задачи самостоятельно по своему варианту:

1. В трех сосудах содержится вода. В первом сосуде V_1 л воды температуры t_1 , во втором – V_2 л температуры t_2 , в третьем – V_3 л температуры t_3 . Воду слили в один сосуд. Составить программу для определения объема V и температуры T воды в этом сосуде (расчет температуры можно вести по упрощенной формуле: $T=(t_1+t_2+t_3)/3$).

2. Вычислить радиус R_B вписанной и радиус R_O описанной около треугольника окружности, если заданы стороны треугольника A, B, C .

$$R_B = \frac{S}{P}; \quad R_O = \frac{A \cdot B \cdot C}{4 \cdot S},$$

где S - площадь, P - полупериметр треугольника.

3. Какой результат отобразится на экране после выполнения фрагмента программы на Pascal?

```
begin
  k:=4;
  writeln ('данные');
  a:=k+10;
  k:=k+a;
  writeln (a, k, k*k, a*a);
  writeln ('результат:', 'a=', a, 'k=', k);
end.
```

4. Восстановить текст программы по имеющейся информации:

память компьютера	экран						
<table style="margin: auto; border-collapse: collapse;"> <thead> <tr> <th style="border: none; padding: 5px;">D</th> <th style="border: none; padding: 5px;">F</th> <th style="border: none; padding: 5px;">S</th> </tr> </thead> <tbody> <tr> <td style="border: 1px solid black; text-align: center; padding: 5px;">3</td> <td style="border: 1px solid black; text-align: center; padding: 5px;">22</td> <td style="border: 1px solid black; text-align: center; padding: 5px;">100</td> </tr> </tbody> </table>	D	F	S	3	22	100	$S - F = 78$ $F - D = 19$
D	F	S					
3	22	100					

5. Написать программу для определения радиуса R_B вписанной и радиуса R_O описанной окружности, площади треугольника, заданного координатами своих вершин $X_1, Y_1, X_2, Y_2, X_3, Y_3$. Стороны треугольника A, B, C определяются формулами:

$$A = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}; \quad B = \sqrt{(X_1 - X_3)^2 + (Y_1 - Y_3)^2};$$

$$C = \sqrt{(X_2 - X_3)^2 + (Y_2 - Y_3)^2}$$

$$H = \frac{2 \cdot \sqrt{P \cdot (P - A) \cdot (P - B) \cdot (P - C)}}{A}; \quad M = \frac{\sqrt{2 \cdot B^2 + 2 \cdot C^2 - A^2}}{2}, \quad \text{где } P = \frac{A + B + C}{2}$$

$$RO = \frac{A \cdot B \cdot C}{4 \cdot \sqrt{P \cdot (P - A) \cdot (P - B) \cdot (P - C)}}; \quad RB = \sqrt{\frac{(P - A) \cdot (P - B) \cdot (P - C)}{P}}$$

6. Составить программу для отгадывания с помощью компьютера задуманного числа. Компьютер предлагает исполнителю произвести следующие действия и ввести результат:

- а) умножить задуманное число на 5;
- б) прибавить 8;
- в) сумму умножить на 2.

По введенному результату компьютер определяет число и печатает его на экране.

7. Автомобиль движется равномерно со скоростью 60 км/ч. Определить за какое время он проедет расстояние, равное 300 м.

8. Найти площадь кольца, если его толщина равна К см, а диаметр внутреннего круга – D см.

9. За один час первый рабочий собирает N пар обуви, второй рабочий собирает L пар, третий рабочий собирает H пар. Сколько пар обуви соберут они вместе за T часов?

10. Определите размер сдачи после покупки в магазине товаров: перчаток - A тенге, портфеля - B тенге, галстука - C тенге. Исходная сумма, выделенная на покупку - D тенге.

11. Написать программу вычисления стоимости поездки на автомобиле на отдых к озеру (туда и обратно). Исходные данные: расстояние до озера (км); количество бензина, которое потребляет автомобиль на 100 км пробега; цена 1 литра бензина.

12. Написать программу пересчета расстояния из верст в километры (1 верста = 1066,8 м).

13. Написать программу, запрашивающую год вашего рождения. Пусть компьютер напечатает год вашего выхода на пенсию. (У девочек и мальчиков программы должны отличаться).

14. Реактивный аэробус летит с пассажирами на борту из Лондона в Нью-Йорк. Три четверти пассажиров имеют билеты второго класса стоимостью X фунтов каждый. Остальные пассажиры имеют билеты первого класса, которые стоят вдвое дороже билетов второго класса. Напишите программу, которая выведет сумму денег, получаемую авиакомпанией от продажи билетов на этот рейс.

15. Для туристической поездки в Болгарию понадобится 26 левов в сутки на гостиницу, 43 лева в сутки на питание и т.п. Предлагается пробыть 5 суток и купить сувениров на 100 левов. Составить программу, определяющую денежную сумму в левах и в рублях, необходимую для поездки (1 лев = 70 коп.).

16. Было когда-то на свете 25 оловянных солдатиков, которых сделали из старой оловянной ложки массой 123 г. 24 солдатика были одинаковыми: друг от друга не отличались. Но двадцать пятый солдатик был не такой, как все. Он оказался одноногим. Его отливали последним, и олова немного не хватило. Какова масса последнего солдатика?

17. Малыш может съесть 600 г варенья за 6 мин, а Карлсон - в два раза быстрее. За какое время они съедят это варенье вместе?

18. Дано трехзначное число. Переставить в нем первую и последнюю цифры.

19. Из трехзначного числа X вычли его последнюю цифру. Когда результат разделили на 10, а к частному слева приписали последнюю цифру числа X , то получилось 237. Найти число X .

20. Дано четырехзначное число. Получить число, образуемое при перестановке его цифр в обратном порядке.

Контрольные вопросы

1. Запишите общий вид алгоритма.
2. Какие типы величин используются в школьном алгоритмическом языке? В языке программирования Pascal?
3. Чем характеризуются переменные величины?
4. Какие имена переменных можно использовать в Pascal?
5. Для чего необходимо описывать данные в программах?
6. Какой оператор используется для ввода данных?
7. Как перевести курсор на новую строку после ввода данных?
8. Как вывести результаты работы программы на экран?
9. Каким образом можно вывести результаты с новой строки?
10. Как удерживать результаты вывода на экране дисплея после срабатывания оператора write?
11. Какое значение будет присвоено переменным M и N после серии команд:
 - а) $M:=0$; $M:=M+2$; $M:=3*M$; $M:=M-2$; $M:=M*M$;
 - б) $N:=1$; $N:=N+1$; $N:=N*N$; $N:=N^2$.

Литература:

2[с.7-25], 9[с.3-7], 10[с.13-24]

Лабораторная работа №3

«Разветвляющиеся алгоритмы»

Цель: Освоить структуру команд ветвления и выбора в языке программирования Pascal; применить конструкции этих команд к решению конкретных задач.

Теоретические сведения

В данной теме рассматриваются две команды: команда ветвления, реализуемая с помощью условного оператора IF и команда выбора (варианта), реализуемая с помощью оператора Case, позволяющие описывать разветвляющиеся алгоритмы.

Условный оператор позволяет выполнять или пропускать операторы программы в зависимости от некоторого условия. Он имеет две формы: полную и сокращенную. В некоторых случаях после слов THEN и ELSE надо выполнить не один оператор, а несколько. Тогда эти операторы заключаются в так называемые операторные скобки, где BEGIN - открывающая скобка, END - закрывающая скобка. Все операторы находящиеся внутри операторных скобок называются составным оператором.

Перед словом ELSE точка с запятой никогда не ставится.

Рассмотреть формы реализации команды ветвления по таблице 3. Далее разобрать пример 1.

Пример 1. Составите программу для вычисления:

$$c = \sqrt{x - 16a}, \text{ если } x = \frac{50 + y}{4}.$$

На этапе анализа задачи необходимо обратить внимание на порядок вычислений с учетом промежуточной величины X.

```
program P1; {Вычисление C}
```

```
var y,a,x,c: real;
```

```
begin
```

```
  readln(y, a);
```

```
  x:=(50+y)/4;
```

```
  if x-16*a>=0 then begin
```

```
    writeln('есть решение');
```

```
    c:=sqrt(x-16*a);
```

```
    writeln('c=',c: 6: 2);
```

```
  end
```

```
  else writeln('решения нет');
```

```
end.
```

Таблица 3 - Формы реализации ветвления

Блок-схема	Алгоритмический язык	Pascal
	<p>Команда ветвления:</p> <p>а) полная форма: <u>если</u> условие <u>то</u> серия 1 <u>иначе</u> серия 2 <u>все</u></p> <p>б) сокращенная форма: <u>если</u> условие <u>то</u> серия <u>все</u></p>	<p>Условный оператор:</p> <p>if условие then begin серия 1; end else begin серия 2; end;</p> <p>В сокращенной форме отсутствует служебное слово else и соответствующая серия команд. Если серия состоит из одной команды, то операторные скобки begin...end; можно опустить.</p>

Демонстрация (слайд 1)

Задача 1. Написать программу, которая вычисляет частное от деления двух чисел. Программа, должна проверять правильность введенных данных, и если делитель равен нулю, то выдавать сообщение об ошибке.

```

program A1;
var x,y,c:real;
begin
writeln ('введите делимое и делитель: X,Y ');
readln(x,y);
if y<>0 then
begin
c:=x/y;
writeln('частное от деления ',x:6:2,' на ',y:6:2,' равно ',c:6:2);
end
else writeln('Ошибка, делитель не может быть равен 0')
end.

```

Безусловный оператор (практически не используется, поскольку Паскаль содержит конструкции, позволяющие полностью обойтись в программе без него):

Осуществляет переход к оператору, перед которым указана метка, объявленная в разделе *label*. (**goto метка**;). Метка (целое число от 0-9999 или набор буквенных символов) располагается перед оператором и отделяется от него двоеточием.

Демонстрация (слайд 2)

```
program P2; { Угадайка }
label 1, 2;
var x, y: real;
begin
  randomize;
  y:= random(101);
  1: writeln ('Введите число от 0 до 100'); readln(x);
  if x=y then
    begin
      writeln ('Поздравляю');
      goto 2;
    end;
  if x>y then writeln ('Ваше число больше');
  if x<y then writeln ('Ваше число меньше');
  goto 1;
2: end.
```

Для построения *сложных условий* в Pascal имеются четыре логических операции:

- NOT - отрицание (НЕТ),
- AND - логическое умножение (И),
- OR - логическое сложение (ИЛИ),
- XOR – исключаящее "или".

Операции отношений имеют более низкий приоритет, чем логические операции, поэтому их следует заключать в скобки при использовании с логическими операциями.

В качестве выполняемого в условном операторе действия может быть другой условный оператор. При форме записи со сдвигом вправо для каждого внутреннего действия, легко понять, к какому из двух слов IF относится слово ELSE.

Задача 2. Даны действительные числа a,b,c. Проверить выполняется ли неравенство $a < b < c$.

```
program A2;
var
a,b,c:real;
begin
writeln('введите значения
a,b,c');
readln(a,b,c);
if (a<b) and (b<c) then
writeln ('условие выполнено')
else
writeln ('условие не
выполнено')
end.
```

```
program A2;
var
a,b,c:real;
begin
writeln('введите значения a,b,c');
readln(a,b,c);
if a<b then
    if b<c then
        writeln ('условие выполнено')
    else
        writeln ('2-е условие не
выполнено')
else writeln ('1-е условие не
выполнено');
end.
```

В программировании существует целый класс задач, где действия производятся только с целыми числами. При решении подобных задач обычно используются операции над целыми MOD и DIV.

Задача 3. Дано двухзначное число. Определить, какая из его цифр больше.

```
program A3;
var n,x1,x2,m:integer;
begin
Write('введите двузначное число');
readln(n);
X1:=n mod 10; {выделяем из числа единицы}
X2:=n div 10; {получает число десятков в числе}
if X1= X2 then writeln('равны')
    else
        if X1> X2 then writeln('единицы больше')
            else writeln('десятки больше')
end.
```

Если результат объявлен как целочисленный, то нельзя использовать наклонную черту деления. Результатом операции деления всегда будет вещественное число. Даже если целое число делится нацело на другое число. *Нельзя использовать операцию div, mod для вещественных чисел*

Очень часто необходимо чтобы программа определила, из каких цифр состоит число, или определила разряд заданной цифры или наоборот цифру в заданном разряде. Для решения этих задач надо вспомнить, что собой представляет любое число десятичной системы счисления (т.е. той, к которой мы с вами привыкли). В позиционной системе счисления (к которой относится десятичная система) величина, обозначаемая цифрой в записи числа, зависит от ее позиции. Например, в числе 326 первая цифра обозначает три сотни, вторая - два десятка, третья - шесть единиц. Любое число можно записать в виде: $333=3*100+2*10+6$

Команда выбора (Case). Если в команде ветвления необходимо указать более двух вариантов действий, то ее запись становится неудобной. Для реализации этих ситуаций в средствах представления алгоритмов имеется команда выбора, которая позволяет представить алгоритм более просто и наглядно, чем при использовании команды ветвления.

Если условный оператор напоминает дорожную развилку, то оператор выбора - это разделение пути на множество дорог, по одной из которых пойдет выполнение программы.

Если при выборе альтернативы необходимо выполнение нескольких операторов, то нужно заключить их в операторные скобки `begin . . . end`; т.е. использовать составной оператор.

Приступая к решению задач этого раздела, следует помнить что:

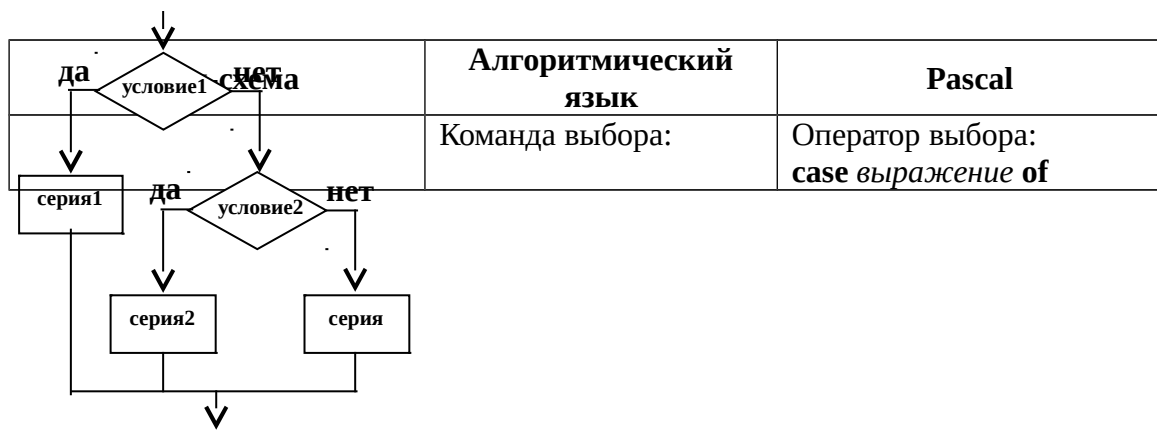
- Инструкция `Case` используется для выбора одного из нескольких направлений дальнейшего хода программы.

- Выбор последовательности инструкций осуществляется во время выполнения программы в зависимости от равенства значения переменной-селектора константе, указанной перед группой инструкций.

- В качестве переменной-селектора можно использовать переменную целого (`integer`) или символьного (`char`) типа.

Рассмотреть формы реализации команды выбора по таблице 4.

Таблица 4 - Формы реализации выбора



	<p>а) полная форма: <u>выбор</u> <u>при условии 1:</u> серия 1 <u>при условии 2:</u> серия 2 <u>иначе серия</u> <u>все</u></p> <p>б) сокращенная форма: <u>выбор</u> <u>при условии 1:</u> серия 1 <u>при условии 2:</u> серия 2 <u>все</u></p>	<p>константа1: begin серия 1; end;</p> <p>константа2: begin серия 2; end;</p> <p>...</p> <p>константаN: begin серия N; end</p> <p>else begin серия; end;</p> <p>end;</p>
--	--	---

Константа должна быть целочисленного, булевского или символьного типа. Условие можно записать:

- 1) через запятую;
- 2) как диапазон значений (через две точки: 1..5).

В сокращенной форме отсутствует служебное слово **else** и соответствующая серия команд.

Исполнитель последовательно проверяет все условия команды выбора, пока не обнаружит первое из них, которое соблюдается. Найдя такое условие, исполнитель выполняет стоящую за ним серию команд, и на этом выполнение команды выбора заканчивается. Если, например, соблюдается и условие 1 и условие 2, то исполнитель все равно выполнит только серию 1.

Когда ни одно из условий не соблюдается, исполняется команда, записанная после слова **иначе**. В сокращенной форме в этом случае не предусмотрено никаких действий.

Задача 4. Определить по номеру месяца соответствующее время года.

```

Program A4; {Время года}
var n: byte;
begin
    readln (n);

```

```

case n of
    1,2,12: writeln ('зима');
    3..5: writeln ('весна');
    6..8: writeln ('лето');
    9..11: writeln ('осень');
end;
end.

```

Задания для тренировки:

1. Написать программу, вычисляющую стоимость междугороднего телефонного разговора. Исходными данными для программы являются код города и продолжительность разговора.
2. Составить программу которая выдает сообщения: «доброе утро», «добрый день», «добрый вечер», «спокойной ночи» в зависимости от введенного времени суток.
3. Составьте программу, определения гласных и согласных букв алфавита.
4. Написать программу, которая после введенного с клавиатуры числа в диапазоне от 1 до 99, обозначающего денежную единицу, дописывает слово «копейка» в правильной форме.
5. Составить программу, которая читает натуральное число N в десятичном представлении ($N \leq 1000$), а на выходе выдает это же число в десятичном представлении и на естественном языке. Например: 7 - семь, 204 - двести четыре, 519 – пятьсот девятнадцать, 322 – триста двадцать два.
6. Вычислить значения функции для указанных промежутков X:

$$Y = \begin{cases} X^2, & \text{если } X = 5,6,7 \\ 2X - 5, & \text{если } X = 8,9 \\ \sqrt{X}, & \text{если } X = 10 \end{cases}$$

Решить задачи самостоятельно по своему варианту:

1. Команда ветвления

1. Составьте программу, которая определяет, будет ли отрицательным произведение двух чисел.

2. Мальчику необходимо пробежать расстояние S км, со скоростью V км/ч. Составьте программу, определяющую уложится ли мальчик в 3 часа и время бега?

3. Известны года рождения двух девочек a_1 и a_2 ($a_1 \neq a_2$). Определите старшую из них.

4. Составить программу к алгоритму:

Модуль действительного числа (МОД):
алг МОД (арг вещ A, рез вещ B)
нач
 ввод A
 если A>0
 то B:=A
 иначе B:=-A
 все
 вывод B
кон

5. Составить программу к алгоритму:

Квадратное уравнение (КВУР):
алг КВУР (арг вещ A, B, C, рез вещ X1, X2, лит Y)
нач
 ввод A,B,C
 D:= $B^2 - 4 \cdot A \cdot C$
 если D<0
 то Y:="нет решения"; вывод Y
 иначе Y:="есть решение"; $X1 := \frac{-B + \sqrt{D}}{2 \cdot A}$;
 $X2 := \frac{-B - \sqrt{D}}{2 \cdot A}$
 вывод Y, X1, X2
 все
кон

6. Составьте блок-схему и программу вычисления значения функции:

$$Y = \begin{cases} 2X^3 + 1, & \text{если } X < 4 \\ X - 5, & \\ 3X^2 - 2, & \text{если } X \geq 4 \end{cases}$$

7. Даны 3 числа. Возвести в квадрат те из них, значения которых отрицательны.

8. Даны 3 числа A, B, C. Если выполняется неравенство $A < B < C$, то напечатать значение суммы, а иначе - произведение чисел.

9. Даны два числа X, Y ($X \neq Y$). Меньшее из них заменить полусуммой, большее – удвоенным произведением этих чисел. Составить программу.

10. Составить программу вычисления значения функции $Y = \frac{4X^2 + 7}{3X - 6}$.

11. Определить является ли введенное в ЭВМ число четным. Составить программу.

12. Одинаковые ли значения примет C после выполнения фрагментов программ?

а) $a := -1; b := 5; c := 16$

б) $a := 9; b := 7; c := 10$

if (a > 1) or (b = 0) then c := a + b;

if (a > 1) and (b = 7) then c := a + b;

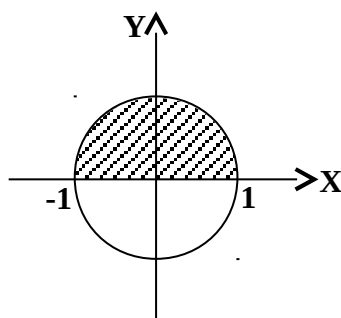
13. Дано четырехзначное число. Определить, сколько раз в нем встречается цифра, равная старшей.

14. Дано трехзначное число. Является ли произведение его цифр трехзначным числом.

15. Дано четырехзначное число. Определите, все ли цифры в числе различны. Выдать соответствующее сообщение, если какие-либо цифры равны.

16. Написать программу по алгоритму, определяющую принадлежит ли точка A(X, Y) заштрихованной фигуре.

Указание: Уравнение окружности: $X^2 + Y^2 = R^2$.



алг принадлежность (арг вещ X, Y)

нач

ввод X, Y

если $X^2 + Y^2 \leq 1$ и $Y \geq 0$

то вывод “точка A принад фигуре”

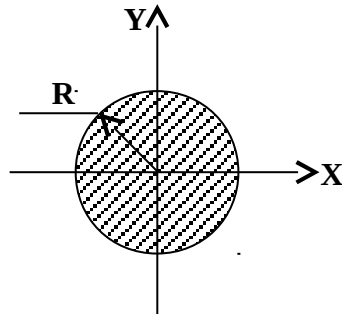
иначе вывод “точка A не принад фигуре”

все

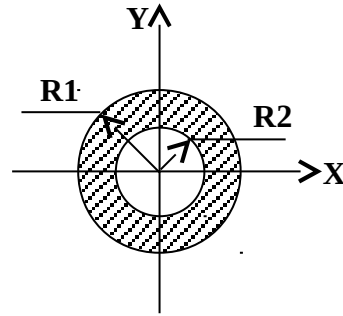
кон

17. Написать программу, определяющую принадлежит ли точка A(X, Y) заштрихованной фигуре.

a)



в)



18. Составить программу для вычисления Y:

$$Y = \begin{cases} X^2, & \text{при } X < 2 \\ \sqrt{X}, & \text{при } 100 > X \geq 2 \end{cases}$$

19. Написать программу определения стоимости разговора по телефону с учетом скидки 20%, предоставляемой по субботам и воскресеньям.

20. Дано целое число. Определить: является ли оно четным, начинается ли оно с цифры 5, делится ли оно на 13, в противном случае выдать сообщение, какое из условий не выполнено (1 или 2 или 3).

2. Команда выбора

Таблица 5 – Задачи-задания

№ задания	Содержание	Исходные данные
1.	$z = \begin{cases} a+b*x+c*X^2, & \text{при } k=1,2,3 \\ d+e*x+f*X^2, & \text{при } k=4,5 \\ a*b+f*x+c*X^2, & \text{при } k=8 \end{cases}$	A=2 B=1.5 C=1 D=3, F=0.5
2.	$z = \begin{cases} 1+c*\sqrt{d}, & \text{при } n=0 \\ x-a, & \text{при } n=1,6 \\ 2/3*X^2-1/2*d, & \text{при } n=2,3,4 \end{cases}$	C=-2 A=1.5 D=2, X=3

3.	$z = \begin{cases} y^2 + 0.3 \cdot a, & \text{при } x = 3 \\ a + e^{y \cdot b}, & \text{при } x = 5, 2 \\ y^2 + y - 3, & \text{при } x = 6, 7, 8 \end{cases}$	A=5 B=1.2 Y=0.3
----	---	-----------------------

Продолжение таблицы 5

4.	$z = \begin{cases} 1 + d \cdot \sin d / a, & \text{при } j = 1 \\ (i-1) \cdot i + a^2, & \text{при } j = 2, 4, 8 \\ i + 2/3 \cdot x, & \text{при } j = 5, 6, 7 \end{cases}$	D=2 A=3.5 X=3, I=2.3
5.	$z = \begin{cases} y/3 + a^2 \cdot x, & \text{при } c = 3 \\ y - \ln b, & \text{при } c = 8, 9, 10 \\ b^2 + y \cdot x, & \text{при } c = 2, 5, 7 \end{cases}$	Y=1 X=2.5 A=4, B=0.4
6.	$z = \begin{cases} a \cdot b + \arctg \chi^2, & \text{при } n = 0, 1, 6 \\ a \cdot h/2, & \text{при } n = 7 \\ \pi \cdot R^2, & \text{при } n = 2, 3, 4 \end{cases}$	A=0.5 B=2 H=4, R=1.4
7.	$z = \begin{cases} p \cdot l, & \text{при } k = 3, 4, 5 \\ p \cdot h/2 + \cos p, & \text{при } k = 1, 2, 3 \\ \pi \cdot R \cdot l, & \text{при } k = 9 \end{cases}$	P=0.6 L=2 H=5, R=4
8.	$z = \begin{cases} \sin x + c \cdot d, & \text{при } b = 1, 2 \\ x/a + \sqrt{2 \cdot d \cdot a}, & \text{при } b = 3, 4, 5 \\ a + d \cdot \cos x, & \text{при } b = 8 \end{cases}$	X=0.65 C=1.5 D=2, A=0.37

9.	$z = \begin{cases} y + (x - a)/(x + a), & \text{при } c = 0, 1, 2 \\ y - x, & \text{при } c = 4, 5 \\ y^2 + e^{a \cdot x}, & \text{при } c = 3 \end{cases}$	$Y = 2.6$ $X = 1.6$ $A = 0.4$
10.	$z = \begin{cases} a + 2/b + 4 \cdot y, & \text{при } y = 4, 5, 6 \\ (a + b)^2, & \text{при } y = 2, 3, 7 \\ a - x \cdot y, & \text{при } y = 8 \end{cases}$	$A = 1.5$ $B = 2$ $X = 0.5$
11.	$z = \begin{cases} 1.5 \cdot x + 9 \cdot x^2 - 1.25, & \text{при } n = 2, 3, 4 \\ d - a \cdot \sin x, & \text{при } n = 5, 6, 8 \\ a - x + y/d, & \text{при } n = 1 \end{cases}$	$X = 0.5$ $A = 4.3$ $Y = 2.6, D = 0.3$
12.	$z = \begin{cases} 0.5 \cdot a + \cos y/a, & \text{при } m = 0, 1, 2 \\ y^2 - i, & \text{при } m = 3, 5, 7 \\ x + x^2/2 - 1/3, & \text{при } m = 4 \end{cases}$	$A = 3$ $Y = 2.7$ $I = 2$ $X = 1$

Продолжение таблицы 5

13.	$z = \begin{cases} \sin x + e^x, & \text{при } m = 5, 9 \\ (x + y)/(1 - x \cdot y), & \text{при } m = 1, 2, 3 \\ x + t^2, & \text{при } m = 0 \end{cases}$	$X = 0.73$ $Y = 0.4$ $T = 2.6$
14.	$z = \begin{cases} e^{a \cdot \sin x} + c^2, & \text{при } d = 2 \\ \sqrt{a + b \cdot x}, & \text{при } d = 3, 4, 5 \\ \pi \cdot R^2, & \text{при } d = 6, 8 \end{cases}$	$A = 2$ $X = 0.54$ $C = 2.3$ $B = 1.8, R = 3$

15.	$z = \begin{cases} 1 - \sin x, & \text{при } t = 8 \\ 1/2 * (1 + \cos a), & \text{при } t = 0, 1, 2, 3 \\ \sqrt{x + c}, & \text{при } t = 4, 6, 7 \end{cases}$	X=0.4 A=0.88 C=3.6
16.	$z = \begin{cases} (a * \sin i * x)^2, & \text{при } i = 3, 4, 5, 6 \\ \sqrt{ a + b * x }, & \text{при } i = 7, 9 \\ i * a * \ln c , & \text{при } i = 0 \end{cases}$	A=-1 B=0.8 X=1 C=-0.7
17.	$z = \begin{cases} -x * y, & \text{при } p = 3, 4, 5 \\ a * b / x * y, & \text{при } p = 6 \\ (a + b) / a^2, & \text{при } p = 8, 10 \end{cases}$	X=2 Y=3.5 A=0.1, B=4
18.	$z = \begin{cases} \arctg(x + y) / (-x * y), & \text{при } k = 5, 6 \\ e^x, & \text{при } k = 0, 1, 2 \\ a + b * x + t, & \text{при } k = 3 \end{cases}$	X=0.8 Y=0.2 A=4, B=5; T=1
19.	$z = \begin{cases} y^2 - 0.3 + a, & \text{при } c = 0, 3 \\ 0, & \text{при } c = 5, 6, 7, 8 \\ (y + x^2) / 2 * a * b, & \text{при } c = 1 \end{cases}$	Y=2 A=0.5 X=1.8 B=0.6
20.	$z = \begin{cases} ((x + y) / t)^2, & \text{при } n = 2, 3, 4, 5 \\ 1 / a^2 * (R / 10), & \text{при } n = 6, 8 \\ \sin(a * b), & \text{при } n = 0, 7 \end{cases}$	X=1.8 Y=2 T=4 A=1.5 R=4.6, B=0.3

Контрольные вопросы

1. Изобразите блок-схему команды ветвления.
2. Объясните механизм работы команды ветвления.
3. Чем отличается полная форма команды от сокращенной?
4. Каково назначение оператора безусловного перехода?
5. Какое условие называется простым?
6. Какое условие называется составным?
7. Каким образом исполняются логические операции и, или?
8. Определите результат выполнения фрагментов программ:

a) `begin`
 `x:= -5;`
 `if x>0 then y:= sqr(x) else y:= x*20;`
 `writeln ('y=', y:6:2);`
`end.`

b) `begin`
 `x:=2;`
 `2: if x>0 then goto 5 else`
 `begin`
 `y:= x*100;`
 `goto 6;`
 `end;`
 `5: x:= -5; goto 2;`
 `6: writeln ('ответ =', y:6:2);`
`end.`

9. Как записывается и выполняется команда выбора (полная и сокращенная форма)?
10. С помощью, каких операторов реализуется алгоритм с командой выбора на языке Pascal?
11. Какие возможны варианты записи условий в операторе выбора на языке Pascal?
12. Определить значение K после выполнения фрагмента программы (без компьютера):

a) **begin**
 `x := -4;`
 case x of
 `-10..-1: k:= 5;`
 `0: k:= 11`

б) **begin**
 `x := 0;`
 case x of
 `-10..-1: k:= 3;`
 `0..10: k:= 7;`

```
    else k:=9  
end;  
end.
```

```
    if x>0 then k:=6 else k:=2;  
end;  
end.
```

Литература:

1[с.35-42], 9[с.8-15, с.108-110], 10[с.24-31]

Лабораторная работа №4

«Циклические алгоритмы»

Цель: Освоить структуру цикла с неизвестным числом повторений в языке программирования Pascal; применить конструкции этих команд к решению конкретных задач.

Теоретические сведения

Если вычислительный процесс содержит многократные вычисления по одним и тем же математическим зависимостям, но для различных значений, входящих в них величин (переменных), то его называют циклическим. Многократно повторяемые участки вычислений называют циклами, а переменные, изменяющиеся в цикле, - переменными цикла. Алгоритм циклической структуры в общем виде должен содержать:

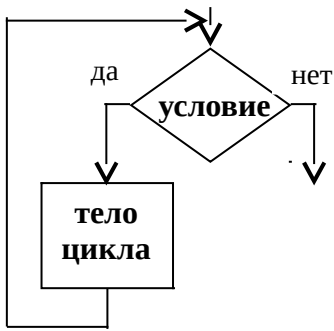
- 1) подготовку цикла: задание начальных значений переменным цикла перед первым его выполнением;
- 2) тело цикла: действия, повторяемые в цикле для различных значений переменных цикла;
- 3) модификацию (изменение) значений переменных цикла перед каждым новым его повторением;
- 4) управление циклом: проверку условия продолжения (или окончания) цикла и переход на начало тела цикла, если выполняется условие продолжения цикла (или выход из цикла по его окончании).

Различают циклы с заданным и заранее неизвестным числом повторений.

Циклы с неизвестным числом повторений

Иногда заранее неизвестно, сколько раз надо повторить цикл, но известно, что он должен выполняться, пока справедливо некоторое условие.

Таблица 6 - Формы реализации циклов с неизвестным числом повторений

Блок-схема	Алгоритмический язык	Pascal
	<p>Команда повторения:</p> <p><u>пока</u> <i>условие</i></p> <p><u>нц</u></p> <p style="padding-left: 40px;"><i>тело цикла</i></p> <p><u>кц</u></p>	<p>Оператор цикла с предусловием:</p> <p>while <i>условие</i> do</p> <p>begin</p> <p style="padding-left: 40px;"><i>тело цикла (серия);</i></p> <p>end;</p> <p>Если серия состоит из одной команды, то операторные скобки можно опустить.</p> <p>Оператор цикла с постусловием:</p> <p>repeat</p> <p style="padding-left: 40px;"><i>тело цикла (серия);</i></p> <p>until <i>условие;</i></p> <p>Цикл выполняется до тех пор, пока условие истинно (while) или пока ложно (repeat ... until).</p>

Вначале рассмотрим команду повторения и форму записи этой команды по таблице 6. Для объяснения принципа выполнения команды повторения удобно воспользоваться ее графическим представлением. Блок-схема помогает учащимся понять и тот факт, что, если условие команды повторения не соблюдается с самого начала, то ни одна из входящих в нее команд не выполнится ни разу.

Далее целесообразно решить задачу:

Задача 1. Вычислить первые 10 членов арифметической прогрессии, заданной формулой N -го члена: $A_N = N^2 - 5$.

Для вычисления необходимо найти значения следующих арифметических выражений:

$$A_1 = 1^2 - 5$$

$$A_2 = 2^2 - 5$$

...

$$A_{10} = 10^2 - 5$$

Для удобства и сокращения записи алгоритма, повторяющиеся действия (определение значения очередного члена арифметической прогрессии и его вывод на печать) поместим в цикл:

$A:=N^2-5$

Вывод A

Определим переменную N, фиксирующую число шагов команды повторения. Тогда цикл начнется со значения N=1, а закончится со значением N=10, т.е. условие окончания работы цикла может быть записано как N>10, следовательно, при N≤10 цикл будет выполняться. Осталось определить формулу для изменения значения переменной N (N=N+1). Программа решения задачи будет иметь следующий вид:

program A1;{Арифметическая прогрессия}

var a: real;

n: byte;

begin

n:=1;

while n<=10 do

begin

a:=sqr(n)-5;

writeln ('a[',n,']=',a:6:2);

n:=n+1;

end;

end.

Таким образом, при каждом выполнении цикла будет сначала проверяться условие(N≤10), затем вычисляться значение очередного члена арифметической прогрессии и печататься на экране.

Задача 2. Вычислить сумму: $\frac{1}{3^2} + \frac{1}{5^2} + \frac{1}{7^2} + \dots + \frac{1}{11^2}$

Для решения воспользуемся оператором цикла с неизвестным числом повторений – оператор **repeat ... until**. Прежде чем составить алгоритм решения, предлагаем выполнить анализ условия задачи. В предложенном примере слагаемые, являющиеся обыкновенными дробями, отличаются друг от друга знаменателями. В свою очередь знаменатели, представленные в виде степени, отличаются друг от друга значениями основания степени и имеют одинаковое значение показателя степени равное 2.

Следовательно, если ввести переменную I (параметр цикла) можно определить каждое слагаемое по формуле $\frac{1}{I^2}$, где I изменяется от 3 до 11 с шагом 2. Таким образом, получим: $S = S + \frac{1}{I^2}$, где Iнач=3, Iкон=11, Iшаг=2.

Запишем программу с использованием нового оператора:

```

program A2; {Сумма}
var s: real;
    i: byte;
begin
    i:=3; s:=0;
    repeat
        s:=s+1/sqr(i);
        i:=i+2;
    until i > 11;
    writeln ('s=',s:6:2);
end.

```

Задания для тренировки:

1. Определить результат выполнения программы (без компьютера):

```

begin
    s:=0 ; x:=10;
    repeat
        s:=s + x;
        x:=x+10;
    until x>12;
    writeln ('s=', s);
    writeln ('x=', x);
end.

```

2. Напечатать таблицу кубов чисел: -10, -8, -6, ..., 10.
3. Вычислить сумму чисел:

$$S = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{20}} + \frac{1}{\sqrt{200}} + \dots + \frac{1}{\sqrt{200000}}$$

4. Составьте программу вычисления среднего арифметического целых чисел из отрезка [-6; 33].

Решить задачи самостоятельно по своему варианту:

1. Вычислить $Y = X^3$ при $x=1,3,5,7,9$
2. Написать программу табулирования функции $y=x^3-2,5$ на отрезке от -3 до 3 с шагом $0,5$.
3. Дан произвольный текст. Признаком конца текста считать нажатие клавиши Enter. Подсчитать общее количество введенных знаков. Условием окончания цикла является проверка конца строки. Пока не обнаружен конец строки (**NOT EOLN**), цикл продолжает выполняться.

4. На срочный вклад в сберегательный банк было внесено A рублей. В конце каждого года вклад увеличивается на P процентов от вклада за предыдущий год. Через сколько лет количество процентов за год превысит сумму первоначального вклада?
5. На срочный вклад в сберегательный банк было внесено A рублей. В конце каждого года вклад увеличивается на P процентов от вклада за предыдущий год. Через сколько лет вклад увеличится в N раз от первоначального вклада?
6. Составить программу вычисления количества первых четных чисел, в сумме дающих 56.
7. Составить программу вычисления количества первых чисел, в сумме дающих 100.
8. Найти все двузначные числа, которые делятся на N .
9. Одноклеточная амеба каждые 3 часа делится на 2 клетки. Определить, через сколько часов будет X амёб.
10. Население города ежегодно увеличивается на $1/40$ наличного состава жителей. Через сколько лет население города утроится?

Примечание. Обратите внимание, что синтаксис **while**-цикла допускает в теле цикла единичный или составной оператор, в то время как **repeat**-цикл может содержать последовательность операторов (для множества операторов не надо использовать операторные скобки).

Контрольные вопросы

1. Какие алгоритмы называются циклическими?
2. Каким образом исполняется команда повторения?
3. Какие характерные ошибки могут возникать при реализации циклических алгоритмов на языке Pascal?
4. Назовите варианты реализации команды повторения в языке Pascal?
5. Чем отличаются цикл с предусловием от цикла с постусловием?

Литература:

1[с.37-40], 9[с.15-20, с.110-112], 10[с.41-46]

Лабораторная работа №5

«Циклические алгоритмы»

Цель: Освоить структуру цикла с известным числом повторений в языке программирования Pascal; механизм работы вложенных циклов и применить эти конструкции к решению конкретных задач.

Теоретические сведения

Циклы с известным числом повторений или его называют еще как команда повторения с параметром позволяет более наглядно записывать алгоритмы циклической структуры.

Таблица 7 - Формы реализации цикла с параметром

Блок-схема	Алгоритмический язык	Pascal
	<p>Команда повторения с параметром:</p> <p><u>для</u> X <u>от</u> X_{нач} <u>до</u> X_{кон} <u>шаг</u> X_{шаг} <u>нц</u> <i>тело цикла</i> <u>кц</u></p> <p>Если значение шага равно 1, служебное слово <u>шаг</u> можно опустить.</p>	<p><i>Вариант 1</i> (с увеличением счетчика, X_{шаг}=1) for x:=X_{нач} to X_{кон} do begin тело цикла (серия); end;</p> <p>Если x_{нач}>x_{кон}, то тело цикла не выполнится ни разу.</p> <p><i>Вариант 2</i> (с уменьшением счетчика, X_{шаг} = -1) for x:=X_{нач} downto X_{кон} do begin тело цикла (серия); end;</p> <p>Если x_{нач}<x_{кон}, то тело цикла не выполнится ни разу.</p> <p>Если серия цикла состоит из одной команды, то операторные скобки можно опустить.</p>

Параметром цикла является переменная, которая последовательно принимает значения в соответствии с указанным шагом X_{шаг}. Механизм выполнения команды повторения с параметром аналогичен команде повторения (цикл «пока»).

Команда повторения с параметром (цикл «для») используется в случаях, когда число повторений заранее известно. Параметры цикла X, X_{нач}, X_{кон} должны быть целочисленного или символьного типа.

Пример 1. Составить программу для вычисления и вывода значений функции $y = \frac{x^2 - 2x + 2}{x - 1}$ при изменении x от -4 до 4 с шагом $0,2$.

Чтобы воспользоваться циклом с параметром сначала подсчитаем число повторений по формуле: $k = \frac{4 - (-4)}{0.2} + 1 = 41$

```

program fun_y;
var x, y: real;
    k: integer;
begin
    x:=-4;
    for k:= 1 to 41 do
    begin {начало цикла}
        y:=(x*x-2*x+2)/(x-1);
        write (x, y);
        x:= x+0,2
    end {конец цикла}
end.

```

Вложенные циклы. Все рассмотренные раньше примеры циклических программ содержали так называемые простые циклы. **Вложенным** называется цикл, содержащий несколько вложенных один в другой простых циклов.

```

for i:=...
begin
    for j:=...
    begin
        серия;
    end;
end;

```

При организации вложенных циклов необходимо учитывать следующее:

1. Имена параметров для циклов, вложенных один в другой, должны быть разными.
2. Внутренний цикл должен полностью входить во внешний.

Задача 1. Напечатать на экране следующую фигуру:

```

*****
*****
*****
*****

```

Решение:

```

program Z1; {Звездочка}
var i, j: byte;
begin
    for i:=1 to 4 do

```

Вначале внешний цикл устанавливает значение номера строки $i=1$.

Внутренний цикл с параметром j

<pre> begin for j:=1 to 6 do write('*'); writeln; end; end.</pre>	<p>печатает шесть «звездочек» в строке. Затем оператор writeln переводит курсор в следующую строку. Принимается следующее значение i=2 и процесс печати повторяется до тех пор, пока $i \leq 4$.</p>
---	---

Задания для тренировки:

1. Составьте программу вычисления первых десяти членов последовательности, заданной формулой n -го члена: $X_n = n^2 + 5$.

2. Определить сумму чисел:

$$S = \frac{1}{X} + \frac{1}{2 \cdot X} + \frac{1}{3 \cdot X} + \dots + \frac{1}{25 \cdot X}$$

3. Составить программу, печатающую на экране следующую фигуру:

```

*****
*****
****
***
**
*
```

4. Определить результат выполнения фрагментов программ (без компьютера):

а) begin
 s:=0;
 for i:=1 to 5 do
 begin
 f:=1;
 for j:= 1 to i do
 f:=f*j;
 s:=s + f;
 end;
 writeln('s=', s);
end.

б) begin
 for i:=1 to 9 do
 begin
 for j:= 1 to i do
 write(i*j);
 writeln;
 end;
end.

Решить задачи самостоятельно по своему варианту:

1. Найти сумму кубов чисел от 1 до 10 с шагом 3.

2. Составить программу вычисления суммы квадратов нечетных чисел (для проверки четности использовать ODD – если число x - четное, то ODD(x)=false) от 7 до 37
3. Каждая бактерия делится на 2 каждую минуту. Сколько их будет через N минут?
4. Вычислить значение функции $Y=x*x$ при $x=2,4,6,8,10,12$. Помните, что переменная счетчика может изменяться только на +1 или -1. Обойдите это ограничение.
5. Вычислить сумму $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{50}$
6. Вычислить сумму $S = \frac{1}{2} + \frac{1}{4} + \frac{1}{6} + \dots + \frac{1}{2N}$
7. Составить программу, которая из первых 10 натуральных чисел все нечетные числа перемножает.
8. Вычислить $S = \frac{1+2^2+3^2+\dots+n^2}{n}$
9. Составить программу вычисления и печати значений функции $y=10\sin X$ для X от 0 до 1 с шагом 0,1. Помните, что переменная счетчика может изменяться только на +1 или -1. Обойдите это ограничение.
10. Составить программу вычисления суммы кубов чисел от 25 до 75.

Контрольные вопросы

1. Поясните механизм работы команды повторения с параметром?
2. Определить результат выполнения фрагмента программы (без компьютера):


```
begin
    s:=0;
    for i:= - 4 to 1 do
      s:=s + i;
    writeln (s);
end.
```
3. Поясните механизм работы вложенных циклов

Литература:

1[с.37-40], 9[с.15-20, с.110-112], 10[с.31-40]

Список литературы

1. Фаронов В.В. Турбо Паскаль 7.0: Практика программирования. Учеб. пособие.-7-е изд., перераб.-М.: Нолидж, 2000.
2. Зуев Е.А. Система программирования Turbo Pascal. – М.: Радио и связь, 1991.
3. Аляев Ю.А., Козлов О.А. Алгоритмизация и языки программирования Pascal, C++, Visual Basic: Учебно-справочное пособие. - М.: Финансы и статистика, 2002.
4. Климова Л. Pascal 7.0 практическое программирование. Решение типовых задач. Кудиц-образ, 2000.
5. Васильев П.П. Турбо Паскаль в примерах и задачах: Освой самостоятельно: Учеб. пособие.-М.: Финансы и статистика, 2002.
6. Глушаков С.В. и др. Turbo Pascal 7.0 для высших и средних учебных заведений / С. В. Глушаков, В.Н. Зорянский, С.Н.Хоменко.- Харьков: Фолио,2002.
7. Голицына О.Л., Попов И.И. Основы алгоритмизации и программирования: Учеб. пособие. - М.: Форум: Инфра-М, 2002.
8. Зеленьяк О.П. Практикум программирования на TURBO PASCAL: Задачи, алгоритмы и решения.-2-е изд., испр. и доп. - СПб.: ДиаСофтЮП, 2002.
9. Криворучко В.А. и др. Сборник задач по информатике (с решениями и ответами): 9кл. / В.А. Криворучко, Л.Н. Кафтункина, Л.Г. Язенко. - Павлодар: ЭКО, 2003.
10. Культин Н.Б. Turbo Pascal в задачах и примерах. - СПб.: БХВ-Петербург, 2000.
11. Попов В.Б. Turbo Pascal для школьников: Учеб. пособие.-3-е изд., доп.- М. : Финансы и статистика, 2002.
12. Программирование на языке Паскаль: Учеб. Пособие / Под ред. О.Ф. Усковой.-СПб.: Питер,2002.- (Учебники для вузов)
13. Ставровский А.Б. Турбо Паскаль 7.0:Учеб. -Киев:ВНУ: Ирина, 2000.- (Б-ка студента)
14. Фаронов В.В. Турбо Паскаль 7.0: Начальный курс: Учебное пособие / В.В.Фаронов.-7-е изд., перераб.-М.: Нолидж, 2001.
Юркин А.Г. Задачник по программированию.- СПб.: Питер, 2002.- (Учебник для вузов)
15. Дузельбаева А.С., Дузельбаев С.Т., Ворожцова С.В. и др. Языки и основы программирования: Учеб. пособие для колледжей и вузов/ Дузельбаева А. С., Дузельбаев С.Т., Ворожцова С.В. и др.- Павлодар: ПГУ им. С.Торайгырова. Ч.1.-2002.

16. Глинский Я.Н. и др. TURBO PASCAL 7.0 и DELPHI: Учебное пособие / Глинский Я.Н., Анохин В.Е., Ряжская В.А. – СПб.: ДиаСофтЮП, 2001.

17. Зайцева Н.М. Программирование на языке Turbo Pascal. Сборник лабораторных работ. – Павлодар, 2003.