## Введение

В ходе выполнения лабораторных работ необходимо составить блок-схему и программу для решения задачи (задач) на языке программирования TURBO PASCAL, отладить программу, получить результат и составить отчет о работе. Защита лабораторной работы проходит в виде собеседования. Для защиты представляется отчет, включающий в себя следующие разделы:

1. Номер лабораторной работы;
2. Тема лабораторной работы;
3. Цель работы;
4. Условие задачи;
5. Блок-схема;
6. Листинг программы;
7. Результат выполнения программы;
8. Вывод.

Отчет оформляется на листах формата А4 в текстовом редакторе WORD шрифт Times New Roman размер шрифта 12, поля: сверху и снизу 1,5 см., слева 2 см., справа 1,5 см. Оформление блок-схемы выполняется с использованием автофигур.

После выполнения цикла лабораторных работ все отчеты подшиваются в единый журнал и сдаются преподавателю. Пример оформления титульного листа журнала отчетов приведен в приложении.

## Лабораторная работа № 1

#### Изучение интегрированной среды программирования Turbo Pascal. Создание алгоритмов линейного типа

Цель работы: изучение структуры окна и системы меню интегрированной среды программирования Turbo Pascal, изучение стандартных процедур ввода-вывода данных, применения оператора присваивания, применения стандартных процедур и функций модуля CRT (очистка экрана – CLRSCR, задержка экрана – ReadKey, Delay) при создании программ.

Интегрированная среда TP выключает в себя экранный редактор, компилятор, компоновщик, отладчик и систему помощи. Она позволяет создавать и редактировать программы, отлаживать и запускать их на выполнение.

*Структура окна среды программирования TP:*

1. В верхней части экрана расположена строка меню (File, Edit, Search и т.д.)
2. Средняя часть экрана – рабочее поле для ввода текста программ. В верхней части рамки рабочего поля расположены: кнопка закрытия файла, имя файла, номер окна, кнопка «свернуть – восстановить». Справа и снизу расположены полосы прокрутки. В левом нижнем углу рамки указаны текущие координаты курсора.
3. В нижней части экрана расположена строка, содержащая информацию о назначении основных функциональных клавиш.

*Функции основных пунктов меню среды программирования Turbo Pascal:*

1. Меню работы с файлами – **File** – вызов имеющихся файлов, создание новых, переименование, запись на диск и т.д.

**New** – создать новый файл

**Open** – открыть существующий файл

**Save** – сохранить файл

**Save as** – сохранить файл под именем

**Save all** – сохранить все измененные файлы

**Change dir** – изменить текущий каталог

**Exit** – выход

1. Меню редактирования - **Edit** – выполнение операций с фрагментами текста – копирования, перемещения, удаления и т.д.

**Undo** – удалить изменение

**Redo** – восстановить изменения

**Gut** – удалить фрагмент (удаляет выделенный блок текста и помещает его в промежуточный буфер)

**Copy** – скопировать фрагмент (Копирует выделенный блок текста в промежуточный буфер без удаления из текущего файла)

**Paste** – вставить фрагмент (Помещает информацию из промежуточного буфера в текущий файл, начиная с позиции курсора)

**Clear** – удалить фрагмент

1. Меню выполнения программы - **Run** – запускает программу, выполняет отдельные части программы во время ее отладки.

**Run** – запустить программу

1. Меню отладки **Debug** – задает параметры, точки останова, вычисляемые параметры и др.

**User screen** – окно пользователя (Открывает окно пользователя. Для выхода из окна следует нажать любую клавишу)

*Горячие клавиши :*

**F1** – справочная система

**F2** – сохранить файл

**F3** – открыть файл

**F5** – увеличить окно отладки на весь экран, повторное нажатие возвращает окно к первоначальному состоянию

**F7** – пошаговое выполнение программы

**Ctrl + F9** – компиляция и выполнение программы

**Alt + F5** – просмотр результатов в окне пользователя

**Alt + X** – выход из среды программирования TP

*Алгоритм и его свойства.*

Алгоритм – точное предписание, определяющее процесс перехода от исходных данных к результату. Предписание считается алгоритмом, если оно обладает тремя свойствами:

* Определенностью;
* Универсальностью;
* Результативностью.

Способы описания алгоритма: словесный, табличный и графический (блок-схема).

Словесный - представление алгоритма с помощью слов и предложений.

Табличный – представление алгоритма в виде таблиц и расчетных формул.

Блок-схема – графическое представление алгоритма, где каждое действие алгоритма записывается соответствующей геометрической фигурой.

Условные обозначения:

Начало и конец алгоритма

Арифметическое действие

Выбор действия в зависимости от истинности или ложности условия

Повторение действий заданное количество раз

Ввод значений переменных с клавиатуры

Вывод значений переменных и текста на экран

Представление алгоритма в виде блок-схемы позволяет наглядно отразить последовательность действий, необходимых для решения поставленной задачи.

После разработки алгоритма решения задачи и представления его в виде блок-схемы, переходят к написанию программы - последовательности инструкций на выбранном языке программирования, соответствующей разработанному алгоритму.

Различают три типа алгоритмов: линейный, разветвляющийся и циклический. Линейный алгоритм не содержит логических условий и имеет одну ветвь вычислений. Разветвляющийся алгоритм содержит одну или несколько ветвей вычислений, выбор которых зависит от результата проверки логического условия. Циклический алгоритм содержит многократно повторяемые части.

#### Пример выполнения задания

Условие задачи

Составить алгоритм и написать программу для нахождения произведения двух чисел. Вывести результат работы программы на экран.

Формула : p= a\* b

*Блок – схема:*

начало

P, a,b

P=a\*b

P

конец

1

2

3

4

5

*Описание блок – схемы:*

Блок №1 – начало программы

Блок №2 – описание переменных и ввод исходных данных

Блок №3 – нахождение произведения двух чисел

Блок №4 – вывод на экран результата

Блок №5 – конец программ

*Листинг программы:*

**Program L\_1;** {заголовок программы}

**Uses CRT;** {раздел описания модулей}

**Var**  {раздел описания переменных}

**P, a : real;**

**B: integer;**

**BEGIN** {начало исполняемой части программы}

**CLRSCR;** {процедура очистки экрана}

**Writeln (‘ введите два числа через пробел ’);**

**Readln (a, b);** {ввод исходных данных}

**P:= a\*b;** {вычисление произведения чисел}

**Writeln (‘ результат:’);**

**Write (‘P= ‘, P:4:2);**{вывод на экран результата}

**Readkey;** {процедура задержки экрана}

**END.** {конец программы}

**Варианты заданий для лабораторной работы:**

1. Составьте алгоритм и программу для вычисления площади равнобедренной трапеции. Длина большего основания a = 18 см., длина меньшего основания b = 10 см., диагонали d = 16 см.

Формулы: 

1. Составьте алгоритм и программу для вычисления объема и полной поверхности конуса с радиусом основания R = 5,2 см. и высотой h = 11 см.

Формулы:  , где L - образующая

1. Составьте алгоритм и программу для вычисления периметра прямоугольного треугольника, если катеты равны: a =15,5 см., d = 23,5 см.

Формулы: 

1. Составьте алгоритм и программу для вычисления гипотенузы и площади прямоугольного треугольника, катеты которого равны: a = 3 см., b = 8 см.

Формулы: 

1. Составьте алгоритм и программу для вычисления площади круга, в который вписан правильный треугольник со стороной a = 17,5 см.

Формулы: 

1. Составьте алгоритм и программу для вычисления длин радиусов вписанной в треугольник и описанной возле треугольника окружностей. Длины сторон треугольника равны a = 13 см., b=12 см., c = 17 см.

Формулы: 

1. Составьте алгоритм и программу для вычисления площади правильного треугольника, вписанного в окружность радиусом r = 18 см.

Формулы: 

1. Составьте алгоритм и программу для вычисления объема прямой призмы с боковым ребром l = 12 см., в основании которой ромб со стороной a = 5 см. и острым углом .

Формулы: 

1. Составьте алгоритм и программу для вычисления площади кольца, внутренний радиус которого r = 20 см., а внешний R = 27 см.

Формулы: 

1. Составьте алгоритм и программу для вычисления периметра прямоугольного треугольника. Катет a = 16 см., гипотенуза с = 20 см.

Формулы: 

1. Составьте алгоритм и программу для вычисления полной поверхности пирамиды, в основании которой лежит правильный треугольник, сторона которого a = 7,5 см., апофема l = 12,5 см.

Формулы: 

1. Составьте алгоритм и программу для вычисления объема пирамиды, в основании которой лежит прямоугольник. Стороны прямоугольника равны: a = 6 см., b = 4 см., высота h = 10 см.

Формулы: 

1. Составьте алгоритм и программу для вычисления площади прямоугольного треугольника, если радиус описанной около него окружности R = 18 см., угол между гипотенузой и катетом 

Формулы: 

1. Составьте алгоритм и программу для вычисления длин высот треугольника, стороны которого равны a =25 см., b = 36 см., угол между ними 

Формулы: 

1. Составьте алгоритм и программу для вычисления площади круга, вписанного в правильный треугольник со стороной a = 36 см.

Формулы: 

## Лабораторная работа № 2

###### Создание алгоритмов разветвленного типа. Изучение оператора условного перехода.

Цель работы: изучение работы оператора IF, вложенных операторов условия и составных операторов.

Ветвление – это такая форма организации действий, при которой в зависимости от выполнения или невыполнения некоторого условия совершается либо одна, либо другая последовательность действий. Каждое отдельное направление вычислений называется ветвью. Выбор ветви осуществляется в результате проверки логического условия.

Условные операторы предназначены для выбора к исполнению одного из двух возможных действий (операторов) в зависимости от некоторого условия. В качестве условия выбора используется значение логического выражения.

Выбор действия в зависимости от выполнения условия может быть реализован при помощи оператора IF.

Оператор IF:

**IF условие THEN оператор\_1**

**ELSE оператор\_2;**

Этот оператор выполняется следующим образом:

* Вычисляется значение условия;
* Если значение выражения истинно, то выполняется оператор1, иначе выполняется оператор 2.

Если какое-либо действие нужно выполнить только при выполнении условия, оператор IF может иметь сокращенный вид:

**IF условие THEN оператор\_1;**

#### Пример выполнения задания

Условие задачи:

Составить алгоритм и программу для выбора наибольшего из двух чисел A и B. На экран выдать соответствующее сообщение.

*Блок – схема:*

##### *Описание блок – схемы*

начало

A, B

A>B

Число A больше, чем B

Число B больше, чем A

конец

2

3

5

4

6

да

нет

Блок №1 – начало программы

Блок №2 – описание и ввод данных

Блок №3 – проверка условия (если условие

##### истинно, выполняется блок №4, иначе

выполняется блок №5)

Блок №4 – вывод сообщения на экран

Блок №5 – вывод сообщения на экран

Блок №6 – конец программы

1

*Листинг программы:*

**Program L\_2;**

**Var A, B: integer;**

**BEGIN**

**Writeln (‘введите два числа’);**

**Readln (A, B);**

**If A>B Then Write (‘ЧислоА больше, чем B’)** {проверка условия}

**Else Write (‘Число B больше, чем A’);**

**END.**

#### Пример выполнения задания с использованием вложенных операторов условия

Условие задачи:

Вычислить значение переменной Y, если



*Блок – схема:*

*Описание блок – схемы:*

начало

Y, X

X=0

X<0

Y=0

Y=-X

Y=X2

Y

конец

да

нет

нет

да

2

3

4

5

7

6

8

9

1

Блок №1 – начало программы

Блок №2 – описание и ввод исходных данных

##### Блок №3 – проверка условия (если условие

истинно, выполняется блок №4, иначе блок

№5)

Блок №4 – вычисление Y

Блок №5 – проверка условия (если условие

истинно, выполняется блок №6, иначе блок

№7)

Блок №6 – вычисление Y

Блок №7 – вычисление Y

Блок №8 – конец программы

*Листинг программы:*

**Program L\_3;**

**Var**

**X, Y: real;**

**BEGIN**

**Writeln(‘введите значение X’);**

**Readln (X);**

**If X = 0 Then Y:= 0** {проверка 1-го условия}

##### **Else If X < 0 Then Y :=X2** {проверка 2-го условия}

**Else Y := -X;**

**Write(‘результат Y = ’, Y:4:2);**

**END.**

**Варианты заданий для лабораторной работы:**

#### Задание 1

1. Ввести два числа. Выдать на печать сумму чисел, если числа больше 10 и меньше 35.
2. Ввести два числа. Большее число возвести в третью степень, а меньшее увеличить на 10. На печать выдать сумму полученных чисел.
3. Ввести два числа. Если оба числа равны 5,5, то выдать их на печать, иначе удвоить число не равное 5,5.
4. Ввести два числа. Если первое число больше второго, ввести еще одно число. Выдать на печать сумму всех чисел, если они больше 150.
5. Ввести два числа. Из большего числа вычесть 100, а меньшее удвоить. На печать выдать произведение полученных чисел.
6. Ввести два числа. Если числа равны, то вывести на печать оба числа, иначе большее из чисел.
7. Ввести два числа. Если оба числа отрицательные, то сменить им знаки. На печать выдать среднее арифметическое полученных чисел.
8. Ввести два числа. Уменьшить первое число в 3 раза, второе увеличить на 5. Если сумма полученных чисел меньше 10,5, то вывести их на печать, иначе вывести на печать разность введенных чисел.
9. Ввести два числа. Если они равны, то выдать на печать их сумму, если не равны, то меньшее из чисел.
10. Ввести два числа. Если среднее арифметическое этих чисел больше 120, то уменьшить их на 10, иначе возвести в квадрат.
11. Ввести три числа. Выдать числа на печать, если их сумма меньше 100, а произведение меньше 150.
12. Ввести два числа. Найти сумму чисел. Если сумма положительна, то вычислить квадратный корень, иначе возвести в квадрат сумму.
13. Ввести два числа. Вывести на печать квадрат числа, остаток которого при делении на 5 наибольший.
14. Ввести число. Проверить, кратно ли оно 4. Если нет, то полученное частное округлить до целого и выдать на печать.
15. Ввести число. Если число четное, то увеличить его в два раза и вывести на печать результат, иначе вывести на печать соответсвующее сообщение.

#### Задание 2

##### Вычислить значение функции

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Функция** | **Условие** | **Исходные данные** |
| **1** |  |  |  |
| **2** |  |  |  |
| **3** |  |  |  |
| **4** |  |  |  |
| **5** |  |  |  |
| **6** |  |  |  |
| **7** |  |  |  |
| **8** |  |  |  |
| **9** |  |  |  |
| **10** |  |  |  |
| **11** |  |  |  |
| **12** |  |  |  |
| **13** |  |  |  |
| **14** |  |  |  |
| **15** |  |  |  |