



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

УПРАВЛЕНИЕ ЦИФРОВЫХ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Кафедра «Естественные науки»

ИНФОРМАТИКА

Использование языка Turbo Pascal для решения прикладных задач

Методические указания иностранным слушателям
дополнительных общеобразовательных программ

Авторы

Моренко Б.Н.

Бабакова Л.Д.

Николенко О.В.



Ростов-на-Дону, 2019

Аннотация

Предлагаемые методические указания является частью учебно-методического комплекса по информатике. Они предназначен для использования на практических занятиях и для самостоятельной работы иностранных слушателей дополнительных общеобразовательных программ. На минимально достаточном лексико-грамматическом материале и в доступной для понимания форме изложены теория и методика выполнения практических работ по естественнонаучным дисциплинам. Ввод и отработка фонетики, лексики и грамматических моделей соотнесены с программой обучения русскому языку как иностранному.

Рекомендуется для использования на практических занятиях по информатике, а также для самостоятельной работы при подготовке к текущему контролю и итоговой аттестации.

Авторы:

- Моренко Б.Н. – к.т.н., доцент кафедры
«Естественные науки»
- Бабакова Л.Д. – старший преподаватель кафедры
«Русский язык как иностранный»
- Николенко О.В. – к. филол. н., доцент кафедры
«Русский язык как иностранный»



Оглавление

	стр.
Аннотация	2
Введение	4
Работа в среде программирования Turbo Pascal	5
ПРАКТИЧЕСКАЯ РАБОТА №1	
Расчёт электрического сопротивления цепи	7
ПРАКТИЧЕСКАЯ РАБОТА №2	
Расчёт времени свободного падения тела	16
ПРАКТИЧЕСКАЯ РАБОТА №3	
Определение молярной массы вещества	18
ПРАКТИЧЕСКАЯ РАБОТА №4	
Решение квадратного уравнения	20
ПРАКТИЧЕСКАЯ РАБОТА №5	
Расчёт площади треугольника по формуле Герона	25
ПРАКТИЧЕСКАЯ РАБОТА №6	
Расчет заработной платы	26
Вопросы для самоконтроля	30
Литература и Интернет-источники	31

ВВЕДЕНИЕ

Предлагаемые методические указания являются частью учебно-методического комплекса по информатике. Методические указания предназначены для использования на практических занятиях по информатике и для самостоятельной работы иностранных слушателей дополнительных общеобразовательных программ инженерно-технической, технологической, экономической и естественнонаучной направленностей обучения. Основной целью методических указаний является изложение в доступной для понимания форме технологии выполнения практических работ по информатике на минимально достаточном лексико-грамматическом материале. Введение и накопление необходимого объема лексики и грамматических конструкций способствует успешному формированию коммуникативной компетенции иностранных слушателей. Использование лексики, структурно-логических моделей и схем соотнесены с программой обучения русскому языку как иностранному. Перед выполнением практических работ иностранным слушателям рекомендуется повторить темы «Алгоритмы и алгоритмизация» и «Язык программирования Turbo Pascal».

Особое место на занятиях по информатике занимают профильно-ориентированные задания, которые носят деятельностный характер, моделируют практическую ситуацию и строятся на актуальном для иностранных слушателей учебном материале. Эти задания способствуют формированию практических навыков, развивают способность иностранных слушателей действовать, находить и применять на практике полученные знания, умения и навыки при решении учебных и прикладных задач по физике, химии, математике и другим дисциплинам.

РАБОТА В СРЕДЕ ПРОГРАММИРОВАНИЯ TURBO PASCAL

1. Включите компьютер. Дождитесь полной загрузки операционной системы Microsoft Windows.
2. Щёлкните два раза левой клавишей мыши на ярлыке «Turbo Pascal». Ярлык находится на Рабочем столе компьютера. Откроется окно программирования Turbo Pascal.
3. В окне программирования Turbo Pascal напишите текст программы. Для переключения между русским и латинским языками используйте клавиши Ctrl и Shift. Одновременное нажатие левых клавиш «Ctrl+Shift» обеспечивает ввод букв латинского алфавита, нажатие правых клавиш «Ctrl+Shift» обеспечивает ввод букв русского алфавита.
4. После набора текста программы её следует сохранить в файле. Для этого:
 - нажмите на клавишу F2 или выберите в главном меню пункт File, а затем Save;
 - в появившемся диалоговом окне «Save file as...» («Сохраните файл как...») напишите имя файла, в котором будет сохранена ваша программа; имя файла нужно писать в формате «*****_01.pas», где пять символов «*****» – это первые пять букв вашего имени, записанного латинскими буквами;
 - нажмите клавишу «Enter» или кнопку «Сохранить» в диалоговом окне «Save file as...».
5. Проверьте текст программы на наличие ошибок. Для этого нажмите клавишу F9. Компьютер покажет ваши ошибки. Исправьте обнаруженные компьютером ошибки. Если ошибок нет, то в появившемся новом диалоговом окне «Compiling» будет написано: «Compile successful: Press any key» («Компиляция выполнена. Нажмите любую клавишу»). Рекомендуется нажать клавишу «Пробел».

6. Для запуска программы на выполнение расчётов используйте комбинацию клавиш «Ctrl+F9» или соответствующие пункты главного меню: Run → Run. Для выхода в окно с текстом программы нажмите клавишу «Enter».
7. Выполните предложенные преподавателем задания, ответьте на поставленные вопросы.
8. Порядок окончания работы:
 - сохраните все изменения в программе. Для этого нажмите клавишу F2 или выберите в главном меню пункт File, а затем Save;
 - выйдите из среды программирования Turbo Pascal. Для этого нажмите клавиши «Alt+X» или воспользуйтесь пунктами меню: File → Exit.
9. Для открытия файла, который вы создали на предыдущем занятии, используйте клавишу F3. В появившемся диалоговом окне выделите нужный вам файл и нажмите клавишу «Enter» или нажмите левой клавишей мыши кнопку «Open» («Открыть»).
10. В процессе работы для открытия/закрытия окна с текстом написанной вами программы используйте комбинацию клавиш «Alt+F5».
11. После окончания работы выключите компьютер. Для этого:
 - закройте все открытые программы;
 - отключите и вытащите из USB-разъёма компьютера Flash-накопитель;
Внимание! Убедитесь, что в дисковом диске нет компакт-диска. Это может привести к проблемам при последующем включении компьютера!
 - выполните последовательно следующие действия: Пуск → Выключение → Выключить компьютер → Выключение → Ок;
Внимание! Если вы делаете небольшой перерыв в работе, то лучше оставить компьютер включённым!

- когда на экране монитора появится надпись «Нет сигнала» выключите питание монитора и всех включённых периферийных устройств;

Внимание! Компьютер нельзя выключать путём его отключения от электрической сети! Для аварийного (быстрого) выключения компьютера или невозможности его корректного выключения (см. пункт 11) нужно нажать клавишу «Power» и удерживать её в нажатом состоянии 6-8 секунд.

ПРАКТИЧЕСКАЯ РАБОТА №1 РАСЧЕТ СОПРОТИВЛЕНИЯ ЭЛЕКТРИЧЕСКОЙ ЦЕПИ

Цель работы. Приобрести навыки построения блок-схем алгоритмов, разработки программ и расчёта на персональном компьютере сопротивления электрической цепи из нескольких резисторов.

Студенты! Перед выполнением практической работы повторите из курса информатики тему «Алгоритмы»; из курса физики повторите темы «Постоянный электрический ток», «Закон Ома». Для повторения учебного материала используйте конспекты занятий и рекомендуемую литературу.

При самостоятельном изучении материала обратите внимание на следующие понятия: алгоритм, свойства и способы описания алгоритмов, типы алгоритмов, блок-схемы алгоритмов, программа, программирование, язык программирования Turbo Pascal.

Алгоритм - это сформулированная на некотором языке последовательность действий и правил решения задачи.

Обратите внимание! Алгоритмы делятся на три вида: линейные алгоритмы (следования), разветвляющиеся (ветвления) и циклические алгоритмы (Рис. 1).

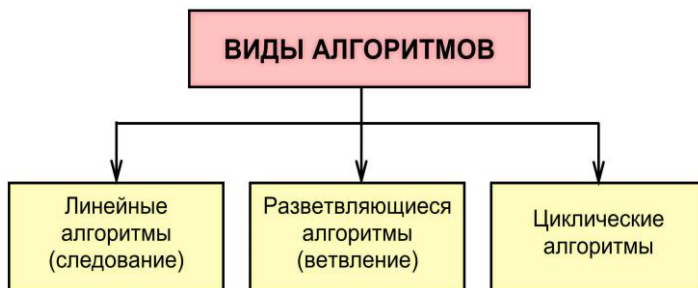


Рис. 1. Виды алгоритмов

Обратите внимание на понятие «программа». *Программа – законченная последовательность команд, которые определяют набор операций и порядок их выполнения при решении задачи.*

Процесс составления алгоритма и решения задачи на алгоритмическом языке (языке программирования) называется программированием.

При повторении физики обратите внимание, что за единицу измерения сопротивления в Международной системе единиц (СИ) принят **Ом** (международное обозначение: Ω – омега).

Сопротивление – это физическая величина, которая характеризует свойства проводника препятствовать прохождению электрического тока

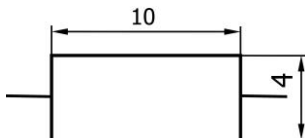
1 Ом равен электрическому сопротивлению проводника, между концами которого возникает напряжение 1 вольт при силе тока 1 ампер.

На практике используют и другие производные единицы измерения сопротивления, например, килоом (кОм, $k\Omega$) и мегаом (МОм, $M\Omega$).

$$1 \text{ килоом (кОм, } k\Omega) = 1\,000 \text{ Ом} = 10^3 \text{ Ом}$$

$$1 \text{ мегаом (МОм, } M\Omega) = 1\,000\,000 \text{ Ом} = 10^6 \text{ Ом}$$

Запомните! Резисторы на электрической схеме изображают в виде прямоугольника размером 10 x 4 мм.



Дополнительные задания.

1. Выведите формулу для расчёта сопротивления трёх последовательно соединённых резисторов.
2. Выведите формулу для расчёта сопротивления трёх параллельно соединённых резисторов.

Задача 1.1. Составить блок-схему алгоритма и написать программу расчёта общего сопротивления электрической цепи, которая состоит из двух последовательно соединённых резисторов (Рис. 2).

Примечание. Общее сопротивление двух последовательно соединённых резисторов определяется по формуле:

$$R_{\text{общ}} = R_1 + R_2$$

где $R_{\text{общ}}$ – общее сопротивление электрической цепи,
 R_1 и R_2 – сопротивления резисторов.

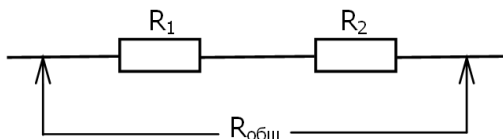
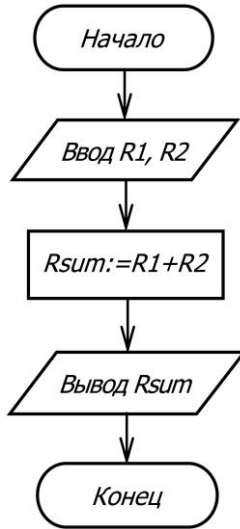


Рис. 2. Последовательное соединение резисторов

1.1. Блок-схема алгоритма расчёта сопротивления электрической цепи.



где: R1 – сопротивление резистора R1
 R2 – сопротивление резистора R2
 Rsum – общее сопротивление электрической цепи

1.2. Программа расчёта электрического сопротивления цепи.

```

program Circuit_Resistance_1;
var
  R1, R2, Rsum : integer;
begin
  Write ('Введите сопротивление резистора R1, R1=');
  ReadLn (R1);
  Write ('Введите сопротивление резистора R2, R2=');
  ReadLn (R2);
  Rsum:= R1 + R2;
  WriteLn ('Общее сопротивление цепи Rsum= ', Rsum );
  ReadLn
end.
  
```

Напишите на компьютере программу расчёта общего сопротивления электрической цепи и рассчитайте его значение по заданным значениям R_1 и R_2 . Результаты расчётов запишите в Таблицу 1.

Таблица 1.

№№ п/п	Значение R_1	Значение R_2	Общее сопротивление электрической цепи, Ом
1.	250 Ом	800 Ом	
2.	10 кОм	1200 Ом	
3.	95 килоом	5 МОм	
4.	110,5 Ω	150,4 Ом	
5.	500 Ом	-500 Ом	

Дополнительные задания.

- Объясните результаты расчётов в п.п. 3 и 4 таблицы.
Какие изменения необходимо внести в программу для получения правильных ответов?
- Объясните результаты расчётов в п. 5 таблицы. Где ошибка?
- Внесите изменения в программу, которые обеспечивают вывод на экран монитора заголовка программы в виде:
«Практическая работа по информатике и физике. Расчёт электрического сопротивления цепи».

Задача 1.2. Составить блок-схему алгоритма и написать программу расчёта электрического сопротивления цепи, которая состоит из двух параллельно соединённых резисторов (Рис. 3).

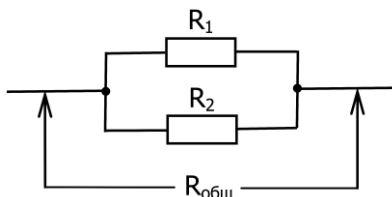
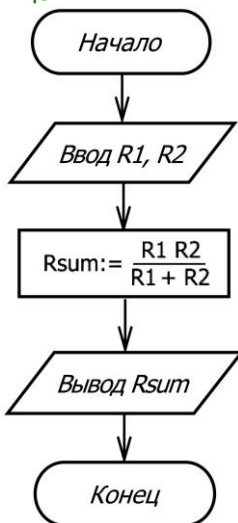


Рис. 3. Параллельное соединение резисторов

Примечание. Общее электрическое сопротивление двух параллельно соединённых резисторов определяется по формуле:

$$R_{\text{общ}} = \frac{R_1 R_2}{R_1 + R_2}$$

2.3. Блок-схема алгоритма расчёта сопротивления электрической цепи.



2.4. Программа расчёта сопротивления электрической цепи.

```

program Circuit_Resistance_2;
var
  R1, R2, Rsum : integer;
begin
  Write ('Введите сопротивление резистора R1, R1 = ');
  ReadLn (R1);
  Write ('Введите сопротивление резистора R2, R2 = ');
  
```

```

Rsum:= (R1*R2)/(R1 + R2);
WriteLn ('Общее сопротивление цепи Rsum= ', Rsum);
WriteLn ('Программа закончила работу ');
WriteLn ('Нажмите клавишу Enter ');
ReadLn
end.
    
```

Напишите на компьютере текст программы расчёта общего сопротивления электрической цепи и рассчитайте его значение по заданным значениям R_1 и R_2 . Результаты расчётов запишите в Таблицу 2.

Таблица 2.

№№ п/п	Значение R_1	Значение R_2	Общее электрическое сопротивление цепи, Ом
1.	200 Ом	200 Ом	
2.	5 кОм	1000 Ом	
3.	400 килоом	1 МОм	
4.	1,5 Ω	50,2 Ом	
5.	-100 Ом	-500 Ом	

Дополнительные задания.

1. Объясните результаты расчётов, полученные в п. 5 таблицы. Где ошибка?
2. Внесите изменения в текст программы, которые обеспечат вывод на экран монитора заголовка программы в виде:
«Лабораторно-практическая работа по информатике и физике. Расчёт электрического сопротивления цепи».

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Задача №1.3. Составить блок-схему алгоритма и написать программу расчёта электрического сопротивления цепи (Рис. 3), которая состоит из трёх последовательно соединённых резисторов (Рис. 4).

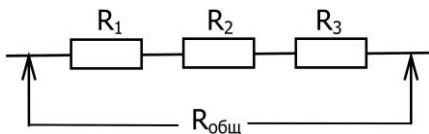


Рис. 4. Последовательное соединение резисторов

Задача №1.4. Составить блок-схему алгоритма и написать программу расчёта электрического сопротивления смешанного соединения резисторов (Рис. 5).

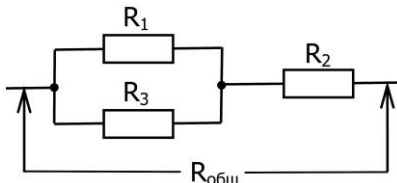


Рис. 5. Смешанное соединение резисторов

Задача №1.5. Составить блок-схему алгоритма и написать программу расчёта электрического сопротивления смешанного соединения резисторов (Рис. 5).

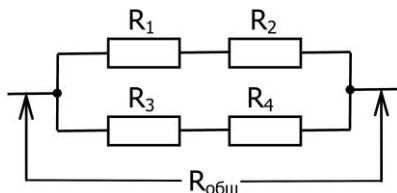


Рис. 5. Смешанное соединение резисторов

Задача №1.6. Составить блок-схему алгоритма и написать программу расчёта электрического сопротивления смешанного соединения резисторов (Рис. 6).

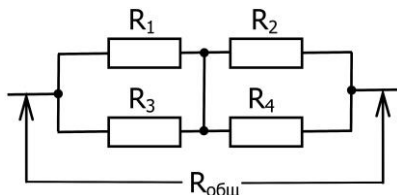


Рис. 6. Смешанное соединение резисторов

Задача №1.7. Составить блок-схему алгоритма и написать программу расчёта электрического сопротивления смешанного соединения резисторов (Рис. 7).

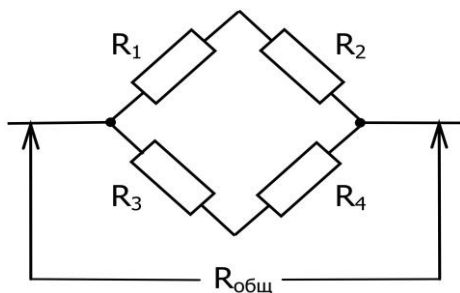


Рис. 7. Смешанное соединение резисторов

Задача №1.8. Составить блок-схему алгоритма и написать программу расчёта электрического сопротивления смешанного соединения резисторов (Рис. 8).

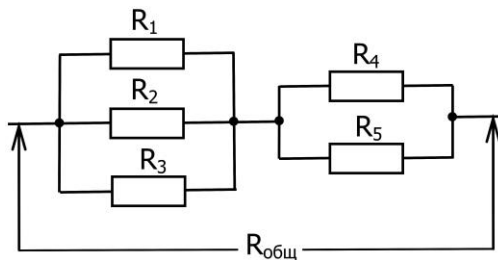


Рис. 8. Смешанное соединение резисторов

ПРАКТИЧЕСКАЯ РАБОТА №2 РАСЧЕТ ВРЕМЕНИ СВОБОДНОГО ПАДЕНИЯ ТЕЛА

Цель работы: Приобрести навыки построения блок-схемы алгоритма, разработки текста программы и расчёта на персональном компьютере времени свободного падения тела с высоты h .

Задача 2.1. Составить блок-схему алгоритма, написать программу и выполнить на компьютере расчёт времени t свободного падения тела (Рис. 9) с заданной высоты h .

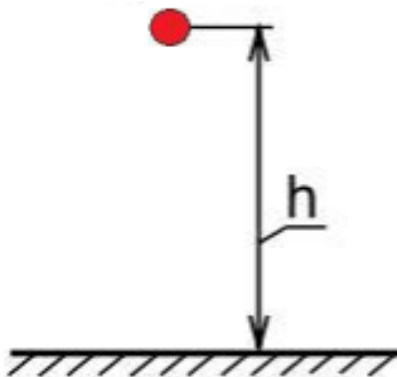


Рис. 9. Свободное падение тела

Определить время свободного падения тела с высоты h можно по формуле:

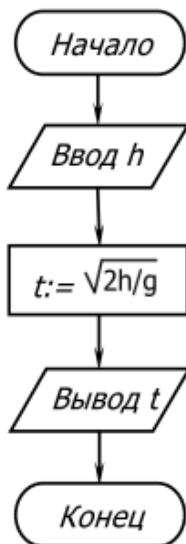
$$t = \sqrt{\frac{2h}{g}}$$

где t – время падения тела, сек,

h – высота падения тела, м,

g – ускорение свободного падения, мс^{-2} .

2.1. Блок-схема алгоритма расчёта времени свободного падения тела.



2.2. Программа расчёта времени свободного падения тела:

```
program Body_fall;
```

```
var
```

```
    h: integer;
```

```
begin
```

```
    Write('Введите высоту падения тела h, h=');
```

```
    ReadLn(h);
```

```
    t:= sqrt(2*h/9.8);
```

```
    WriteLn('Время падения тела t=', t, ' секунд');
```

```
    ReadLn
```

```
end.
```

Вычислите время свободного падения тела по заданным значениям высоты h . Результаты расчётов запишите в таблицу.

№№ п/п	Значение h , метров	Значение t , секунд
1.	100	
2.	324,8	
3.	1500	
4.	35000	

Ответьте на вопросы:

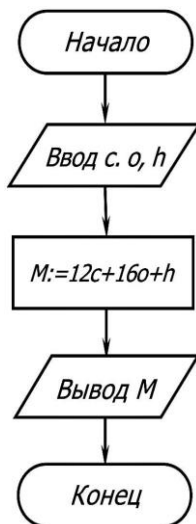
1. Зависит ли время падения тела от его массы?
2. Тело падает с высоты h . Где оно упадёт быстрее: на Земле или на Луне?
3. Чему равно ускорение свободного падения тела вблизи поверхности Земли?
4. Куда направлено ускорение свободного падения тел?
5. От чего зависит ускорение свободного падения?
6. Что называется свободным падением?
7. К какому виду движения относится свободное падение?

ПРАКТИЧЕСКАЯ РАБОТА №3 ОПРЕДЕЛЕНИЕ МОЛЯРНОЙ МАССЫ ВЕЩЕСТВА

Цель работы. Составить блок-схему алгоритма и написать на языке Turbo Pascal программу расчёта молярной массы вещества.

Задача 3.1. Составить блок-схему алгоритма и написать программу расчёта молярной массы вещества (г/моль), которое состоит из углерода (С), водорода (Н) и кислорода (О).

3.2. Блок-схема алгоритма расчёта молярной массы (г/моль) вещества.



3.3. Программа расчёта молярной массы (г/моль) вещества.

```

program Molar_Mass;
var
    C, O, H, M: integer;
begin
    Write ('Введите количество атомов углерода C, C=');
    ReadLn (C);
    Write ('Введите количество атомов кислорода O, O=');
    ReadLn (O);
    Write ('Введите количество атомов водорода H, H=');
    ReadLn (H);
    M:=C*12+O*16+H;
    WriteLn ('Молярная масса вещества равна ', M, ' г/моль');
    ReadLn
end.
    
```

Выполните расчёт молярной массы (г/моль) некоторых веществ.

Название вещества	Химическая формула	Молярная масса вещества, г/моль
Вода	H ₂ O	
Метан	CH ₄	
Пропан	C ₃ H ₈	
Бутан	C ₄ H ₁₀	
Уксусная кислота	CH ₃ COOH	
Этиловый спирт	C ₂ H ₅ OH	
Метиловый спирт	CH ₃ OH	
Угловая кислота	H ₂ CO ₃	

ПРАКТИЧЕСКАЯ РАБОТА №4 РЕШЕНИЕ КВАДРАТНОГО УРАВНЕНИЯ

Целью настоящей практической работы по информатике и математике является повторение студентами существующих способов решения квадратных уравнений, а также приобретение ими практических навыков в разработке соответствующих блок-схем алгоритмов, программ и выполнение расчётов с использованием возможностей языка программирования Turbo Pascal.

Квадратное уравнение – это алгебраическое уравнение вида:

$$ax^2 + bx + c = 0,$$

где x – неизвестная,

a, b, c – коэффициенты квадратного уравнения.

Коэффициенты квадратного уравнения – это произвольные числа, причём $a \neq 0$.

Выражение $ax^2 + bx + c$ называют квадратным трёхчленом.

Корень уравнения – это значение переменной x , которое обращает квадратный трёхчлен в нуль, а квадратное уравнение – в тождество.

Квадратные уравнения могут быть полными и неполными.

Полным квадратным уравнением называют такое, в котором все коэффициенты не равны нулю.

Неполным квадратным уравнением называется такое, в котором хотя бы один из коэффициентов (кроме a) равен нулю.

СПОСОБЫ РЕШЕНИЯ КВАДРАТНЫХ УРАВНЕНИЙ

1. Решение квадратных уравнений по формуле.
2. Решение квадратных уравнений по теореме Виета.
3. Графическое решение квадратного уравнения.

Для нахождения корней квадратного уравнения в общем случае можно использовать формулу:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Дискриминант – это выражение, от значения которого зависит число корней квадратного уравнения.

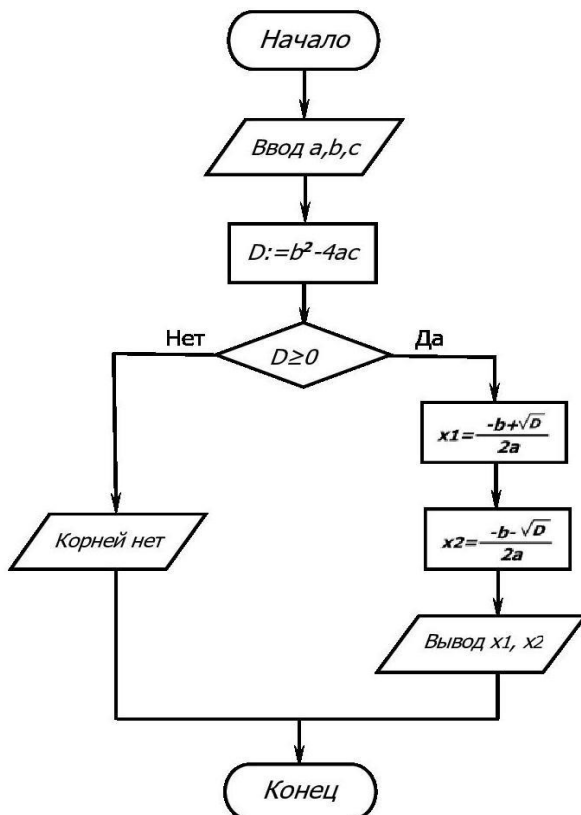
Формула дискриминанта: $D = b^2 - 4ac$

Если $D < 0$, то уравнение не имеет корней.

Если $D = 0$, то уравнение имеет два одинаковых корня.

Если $D > 0$, то уравнение имеет два различных корня.

4.1. Блок-схема алгоритма решения квадратного уравнения.



4.2. Программа расчёта корней квадратного уравнения.

```
program RootsOfQuadraticEquation;
```

```
uses Crt;
```

```
var
```

```
  a, b, c : real;
```

```
  x1, x2 : real;
```

```
  D: real;
```

```
  {Коэффициенты уравнения}
```

```
  {Корни уравнения}
```

```
  {Дискриминант}
```

```

begin
ClrScr;                {Очистка экрана монитора}
WriteLn ('Программа решения квадратного уравнения');
WriteLn;               {Ввод пустой строки}
Write ('Введите значение коэффициента a, a=');
ReadLn (a);
Write ('Введите значение коэффициента b, b=');
ReadLn (b);
Write ('Введите значение коэффициента c, c=');
ReadLn (c);
D:= b*b - 4*a*c;      {Вычисление дискриминанта}
if D >= 0
then
begin
x1:= (- b + sqrt(D)) / (2*a);
x2:= (- b - sqrt(D)) / (2*a);
WriteLn;              {Ввод пустой строки}
WriteLn ('Корни квадратного уравнения:');
WriteLn ('x1=' , x1 );
WriteLn ('x2=' , x2);
end
else
begin
WriteLn ('Действительных корней нет!');
WriteLn;              {Ввод пустой строки}
WriteLn ('Программа завершила работу!');
end;
ReadLn
end.
    
```

Напишите программу на компьютере и вычислите корни квадратного уравнения $ax^2+bx+c=0$ при заданных значениях его коэффициентов a, b, c . Результаты расчётов запишите в таблицу.

№№ п/п	Коэффициенты уравнения			Корни уравнения	
	a	b	c	x_1	x_2
1.	5	3	-26		
2.	-2	6	10		
3.	1	4	3		
4.	4	1	7		
5.	0	5	2		

1. Объясните результаты расчётов, полученные в пунктах 4 и 5.
2. Какие изменения нужно внести в текст программы для получения результатов расчёта корней квадратного уравнения с точностью до 0,1, до 0,001?
3. Какие изменения нужно внести в текст программы для вывода на экран монитора результатов расчёта дискриминанта?
4. Какие изменения нужно внести в текст программы для вывода на экран монитора результатов расчёта корней в виде:
 - Первый корень уравнения $x_1=5.1$
 - Второй корень уравнения $x_2=-2.8$

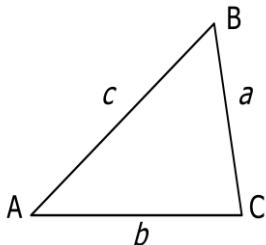
Ответьте на вопросы:

1. Что такое квадратное уравнение?
2. Назовите виды квадратных уравнений.
3. Что такое корни квадратного уравнения?
4. Назовите известные вам способы решения квадратных уравнений.
5. Напишите формулу расчёта корней полного квадратного уравнения.
6. В каком случае корни квадратного уравнения x_1 и x_2 будут равны между собой?

ПРАКТИЧЕСКАЯ РАБОТА №5

РАСЧЕТ ПЛОЩАДИ ТРЕУГОЛЬНИКА ПО ФОРМУЛЕ ГЕРОНА

Задача 5.1. Составить блок-схему алгоритма и написать программу расчёта площади S треугольника по заданным размерам его сторон a, b, c с проверкой условия существования решения.



Определить площадь S треугольника можно по формуле Герона:

$$S = \sqrt{p(p-a)(p-b)(p-c)},$$

где p – полупериметр треугольника

$$p = \frac{a+b+c}{2}$$

5.1. Программа расчёта площади треугольника.

```

program Geron_1;
var
    a, b, c, d, s, p: real;
begin
    Write ('Введите размер стороны a, a=');
    ReadLn (a);
    Write ('Введите размер стороны b, b=');
    ReadLn (b);
    Write ('Введите размер стороны c, c=');
    ReadLn (c);
    p:=(a+b+c)/2;           {Полупериметр треугольника}
    d:= p*(p-a)*(p-b)*(p-c);
    if d<0
    then
        begin
            WriteLn ('Треугольника с такими размерами сторон нет');
            WriteLn ('Повторите, пожалуйста, ввод ... ');
        end
    else
        begin
            s:=sqrt(d);
            Write ('Площадь треугольника s=', s:5:3);
        end
    end
end
    
```

```

end;
WriteLn;
ReadLn;
end.
    
```

По заданным размерам сторон a , b , c треугольника вычислите его площадь. Результаты расчётов запишите в таблицу.

№№ п/п	Размеры сторон треугольника			Площадь треугольника, S
	a	b	c	
1.	10	20	25	
2.	16,5	51,7	49,1	
3.	16500	51700	49100	
4.	12	34	46	
5.	26	40	11	

Дополнительное задание.

1. Объясните результаты расчётов, полученные в п. 4.
2. Объясните результаты расчётов, полученные в п. 5. Где ошибка?

ПРАКТИЧЕСКАЯ РАБОТА №6 РАСЧЕТ ЗАРАБОТНОЙ ПЛАТЫ

Цель работы. Составить блок-схему алгоритма и написать на языке Turbo Pascal программу расчёта заработной платы инженерам-строителям.

Задача 6.1. Составить блок-схему алгоритма и написать программу расчёта заработной платы инженерам-строителям в зависимости от стажа их работы по следующему правилу:

– при стаже работы инженеров-строителей менее 5 лет их часовая заработная плата составляет 10 долларов;

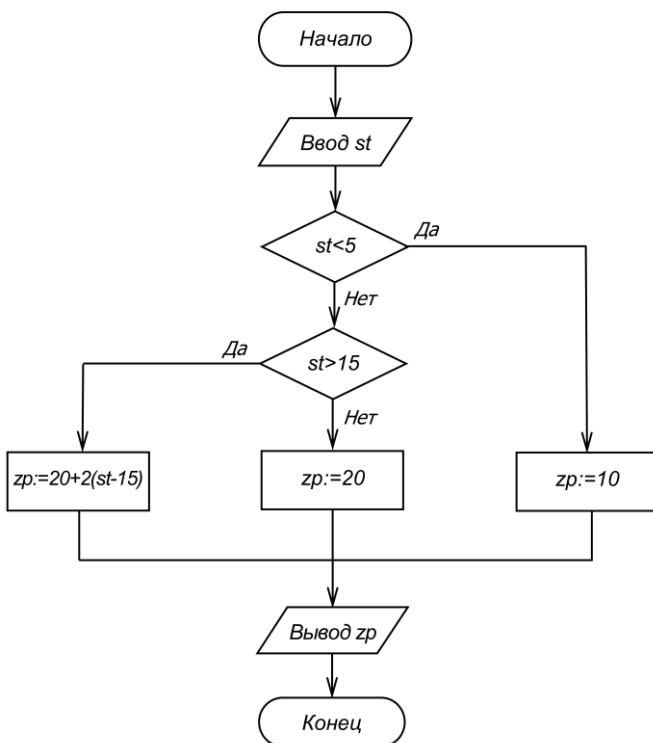
- при стаже работы от 5 до 15 лет – 20 долларов;
- при стаже работы свыше 15 лет часовая заработная плата инженеров-строителей повышается каждый год на 2 доллара.

6.2. Условие задачи в математическом виде.

$$z_p = \begin{cases} 10, & \text{если } st < 5 \\ 20, & \text{если } 5 \leq st \leq 15 \\ 20 + 2(st - 15), & \text{если } st > 15 \end{cases}$$

где z_p – часовая заработная плата,
 st – стаж работы.

6.3. Блок-схема алгоритма начисления (расчёта) заработной платы.



6.4. Программа расчёта заработной платы.

```

program Zarplata;
uses Crt;
var
    st: integer;           {st - стаж работы}
    zp: integer;          {zp – размер заработной платы}
begin
    ClrScr;
    WriteLn ('Программа расчета часовой заработной платы
инженера-строителя');
    WriteLn ('в зависимости от стажа работы');
    WriteLn;               {Ввод пустой строки}
    Write ('Введите стаж работы инженера-строителя st, st =');
    ReadLn (st);
    if st < 5
    then
        WriteLn (' Заработная плата при стаже st=' , st, равна 10 $ в
час')
    else
        if st > 15
        then
            WriteLn ('Заработная плата при стаже st=' , st, ' равна ' ,
20+2*(st-15), ' $ в час')
        else
            WriteLn ('Заработная плата при стаже st=' , st, ' равна ' , '20
$ в час');
    WriteLn;               {Ввод пустой строки}
    WriteLn ('Для выхода в рабочее окно программы нажмите
клавишу <Enter> ');
    ReadLn
end.
    
```

Студенты! Рассчитайте размер заработной платы **zp** инженеров-строителей при указанном стаже их работы **st**. Результаты расчётов запишите в таблицу.

№№ п/п	Стаж работы, <i>st</i>	Размер заработной платы, <i>zp</i>
1.	3 года	
2.	8 лет	
3.	9,5 лет	
4.	20 лет	
5.	-2 года	
6.	110 лет	

- Объясните результаты расчётов, полученные в п.п. 3, 5, 6. Где ошибка?
- Внесите изменения в программу, которые позволят вместо символа «\$» записывать слово «доллар».

Дополнительные задания.

1. Составьте блок-схему алгоритма начисления заработной платы экономисту по следующему правилу:

- при стаже работы экономиста менее 3 лет его часовая заработная плата составляет 5 долларов;
- при стаже работы от 3 до 8 лет – 10 долларов;
- при стаже работы свыше 8 лет его часовая заработная плата составляет 15 долларов.

2. Написать на компьютере программу и выполнить расчёт заработной платы экономиста в зависимости от стажа его работы. Расчёты выполните при следующем стаже работы: 2 года, 6 лет, 10 лет, 20 лет, 30 лет. Объясните результаты расчётов.

**Критерии оценки
выполнения дополнительного задания**

№№ п/п	Перечень выполненных заданий	Количество баллов
1.	Описание условия задачи в математическом виде	1
2.	Построение блок-схемы алгоритма решения задачи	1
3.	Разработка программы расчёта заработной платы	1
4.	Набор и тестирование программы на компьютере; выполнение расчётов	2
ИТОГО:		5

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

Ответьте на вопросы по теме «Алгоритмы»:

- Что такое алгоритм?
- Какие свойства алгоритмов вы знаете?
- Какие способы описания алгоритмов вы знаете?
- Что такое графический способ описания алгоритмов?
- Какие блоки используют при графическом описании алгоритмов?
- Какие виды алгоритмов вы знаете?
- Какой алгоритм называется линейным?
- Какой алгоритм называется разветвляющимся?
- Какой алгоритм называется циклическим?

Ответьте на вопросы по теме «Язык программирования Turbo Pascal»:

- Что такое алфавит языка программирования?
- Что входит в состав алфавита языка Turbo Pascal?
- Какие знаки арифметических действий используют в Turbo Pascal?

- Что такое линейная программа?
- Что такое разветвляющаяся программа?
- Какие числа называются вещественными?
- Какие существуют правила записи чисел в Turbo Pascal?
- Какие существуют правила записи арифметических выражений?
- Какие арифметические функции вы знаете?
- Как записать число в экспоненциальном виде?

ЛИТЕРАТУРА И ИНТЕРНЕТ-ИСТОЧНИКИ

1. Моренко Б.Н. Информатика. Алгоритмы и алгоритмизация: учеб.-метод. пособие для иностранных слушателей дополнительных общеобразовательных программ, инженерно-технической и технологической направленности обучения [Электрон. ресурс] / Б.Н. Моренко, Л.Д. Бабакова. – Режим доступа: <http://de.donstu.ru/CDOCourses/96a0a634-bb48-42f8-a74d-eb70601695f1/3832/4903/3732.pdf>
Рег. номер 3732 от 12.05.2017 г.
2. Моренко Б.Н. Информатика. Линейные программы в Turbo Pascal: метод. указания к лаб.-практ. работам для иностранных студентов предвузовской подготовки, обучающихся по образовательным программам инженерно-технической, технологической и естественнонаучной направленности. [Электрон. ресурс] / Сост.: Б.Н. Моренко, Л.Д. Бабакова. – Режим доступа: <http://de.donstu.ru/CDOCourses/380f223c-4107-4d68-8c9f-693234686953/2924/2727.pdf>
Рег. номер 2727 от 01.02.2016 г.
3. Методические указания к интегрированной практической работе по информатике и математике для иностранных студентов предвузовской подготовки «Решение квадратных уравнений с использованием языка Turbo Pascal» [Электронный ресурс] / Сост.: Б.Н. Моренко, Л.Д. Бабакова. – Режим доступа: <http://de.donstu.ru/CDOCourses/380f223c-4107-4d68-8c9f-693234686953/1603/1384.pdf>
Рег. номер 1384 от 30.09.2013.

4. Моренко Б.Н. Расчет электрического сопротивления цепи: практикум по дисциплине «Информатика» дополнительные общеобразовательные программы для иностранных граждан и лиц без гражданства [Электронный ресурс] / Б.Н. Моренко, Л.Д. Бабакова. – Режим доступа: <http://de.donstu.ru/CDOCourses/380f223c-4107-4d68-8c9f-693234686953/2335/2176.pdf>
Рег. номер 2176 от 06.02.2015.
5. Моренко Б.Н. Информатика. Программы ветвления в Turbo Pascal: практикум по дисциплине «Информатика» для слушателей дополнительных общеобразовательных программ, обеспечивающих подготовку иностранных граждан и лиц без гражданства к освоению профессиональных образовательных программ на русском языке [Электрон. ресурс] / Б.Н. Моренко, Л.Д. Бабакова. – Режим доступа: <http://de.donstu.ru/CDOCourses/380f223c-4107-4d68-8c9f-693234686953/3426/4903/3302.pdf>
Рег. номер 3302 от 09.01.2017 г.
6. Моренко Б.Н. Использование ветвлений в Turbo Pascal при решении экономических задач: метод. указания [Электрон. ресурс] / Сост.: Б.Н. Моренко, Л.Д. Бабакова, О.М. Воскерчьян.
– Режим доступа: <http://de.donstu.ru/CDOCourses/380f223c-4107-4d68-8c9f-693234686953/1465/1171.pdf>
Рег. номер 1171 от 01.04.2013.
7. Моренко Б.Н. Подготовка к итоговой аттестации по информатике : учеб.-метод. пособие / Б.Н. Моренко, Л.Д. Бабакова, О.В. Николенко; ДГТУ. – Ростов н/Д. : ИЦ ДГТУ, 2019. - 52 с.
– Режим доступа: <https://ntb.donstu.ru/content/podgotovka-k-итоговой-аттестации-по-информатике.pdf>