Лабораторная работа № 10

Разветвляющиеся вычислительные процессы

Цель работы: получение навыков написания программ, в которых выбор программой вариантов действий зависит от наступления определенных событий (ситуаций).

1. Теоретическая часть

В линейной программе все операторы выполняются последовательно, один за другим. Таким способом можно записывать только очень простые алгоритмы.

Для того чтобы в зависимости от конкретных значений исходных данных обеспечить выполнение разных последовательностей операторов, в программах на языке Паскаль применяются операторы ветвления *if* и *case*.

Оператор *if* обеспечивает передачу управления на одну из двух ветвей вычислений, а оператор *case* — на одну из произвольного числа ветвей.

Оператор варианта case

Оператор варианта (выбора) *case* используется для реализации разветвляющихся алгоритмов с множественным выбором. Формат оператора:

```
case выражение of
    константы_1 : оператор_1;
    константы_2 : оператор_2;
    …
    константы_n : оператор_n-1
  [ else : оператор_n]
end;
```

При выполнении оператора *case* решение о том, какой именно из списка операторов - оператор_1, оператор_2, оператор_3 и т.д. надо выполнить, принимается после определения значения выражения после слова *case*. При известном значении выражения выполняется тот оператор, который записан после константы, значение которой совпало со значением выражения.

Если требуется выполнить одни и те же действия для нескольких констант, они записываются через запятую либо указывается диапазон значений констант.

Если по какой-либо ветви требуется записать не один, а несколько операторов, они должны образовывать составной оператор, т.е. заключаться между ключевыми словами *begin* и *end*.

Наличие строки, начинающейся со слова *else* в операторе *case*, не обязательно, оператор после *else* (оператор_n) соответствует случаю, когда значение выражения не совпадает ни с одной из констант.

Рассмотрим пример. Пусть необходимо написать программу, реализующую калькулятор на четыре арифметических действия.

Исходными данными для этой программы являются два вещественных операнда и знак операции, представляющий собой символ.

Пусть вид экрана во время выполнения программы будет следующим.

Простейший калькулятор.
Введите первый операнд: **5**
Введите операцию **(+,-,*,/)**: ***
Введите второй операнд: **6**
Результат: **30**

Схема алгоритма работы программы (рис. 10.1) показывает, что в зависимости от значения введенного знака операции необходимо выполнить соответствующее действие над операндами. Программа, реализующая данный алгоритм, представлена на рис. 10.2.

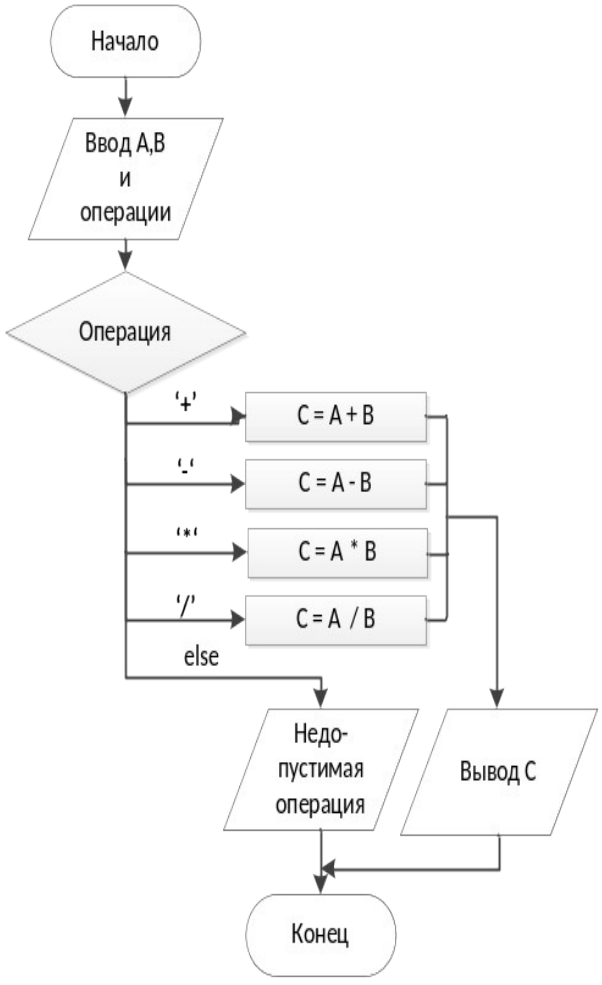


Рис. 10.1. Схема алгоритма для конструкции *case*

```
Program calculator:
  var
    A, B, C : real; {исходные данные и результат}
    op      : char:  {операция}
    error   : boolean; {признак недопустимой
операции}
   begin
     writeln('Простейший калькулятор.');
     write(' Введите первый операнд :  ');
     readln(A):
     write(' Введите операцию (+,-,*,/): ');
     readln(op):
     write(' Введите второй операнд: ');
     readln(B);
     error := false;  {считаем, что введена
действительная операция}
     case op of
        '+' :  C := A + B;
        '-'  :  C := A – B;
        '*' :  C := A * B;
         '/' :  C := A / B;
      else
        begin
          writeln( ' Недопустимая операция '};
          error := true;
        end;
     end; {case}
     if not error
       then writeln(' Результат: ',C:6:2);
   end.  {конец программы}
```

Рис. 10.2. Программа с оператором *case*

Условный оператор if

Полный формат оператора имеет вид:

if выражение then оператор_1 else оператор_2 .

Выражение, записанное после слова *if*, должно иметь логический тип (логическое выражение). При выполнении оператора *if* сначала вычисляется значение этого выражения. Если выражение имеет значение *true*,

выполняется оператор 1, иначе - оператор 2 (рис. 10.3, а). Ветвь *else* оператора *if* может отсутствовать (рис. 10.3, б), и в этом случае оператор имеет следующий сокращенный вид:

if выражение then оператор

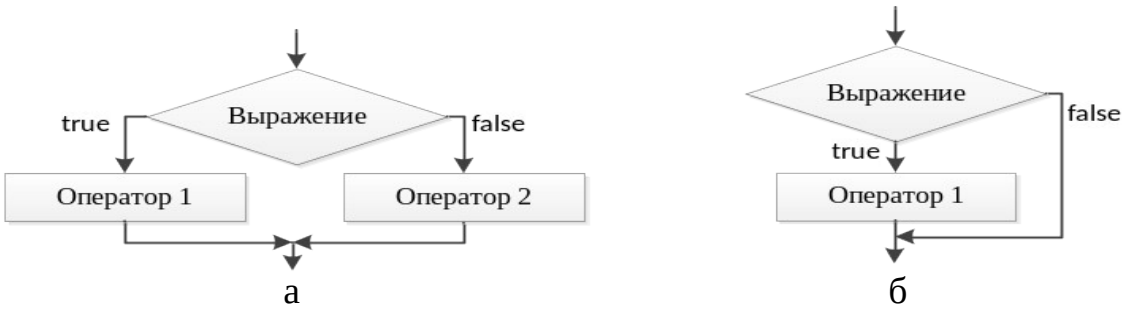


Рис. 10.3. Структурная схема условного оператора

2. Практическая часть

2.1. Требования к выполнению заданий

Для каждого варианта необходимо выполнить 2 задания.

Задание 1. Для этого задания разработать алгоритм и написать программу с использованием оператора *case*.

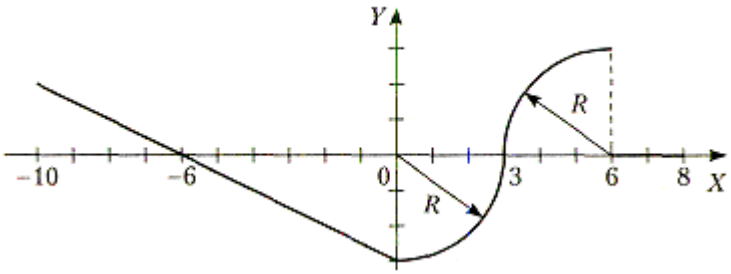
Задание 2. Для этого задания разработать алгоритм и написать программу вычисления значения функции, заданной ее графиком. В программе необходимо использовать оператор *if*. Параметр *R* вводится с клавиатуры.

2.2. Варианты заданий для выполнения

Вариант 1

Задание 1. Для числа от 0 до 35 написать фразу: «Мы посетили N занятий», согласовав окончание слова «занятие» с числом N.

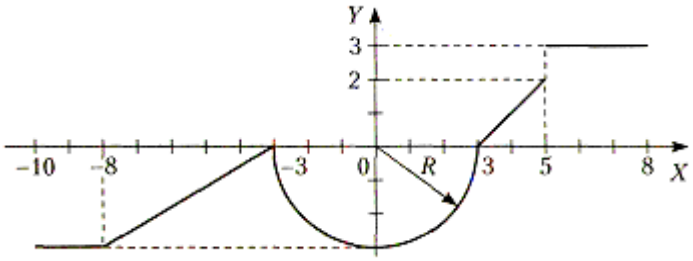
Задание 2. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.



Вариант 2

Задание 1. Составить программу, которая после введенного с клавиатуры числа (в диапазоне от 1 до 99), обозначающего денежную единицу, дописывает слово «копейка» в правильной форме.

Задание 2. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.



Вариант 3

Задание 1. Вывести на экран следующую информацию:

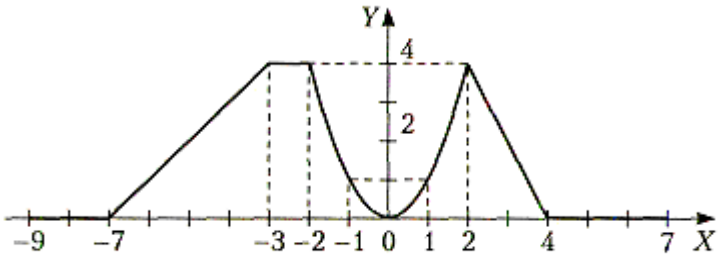
ОВОЩИ И ФРУКТЫ

1. Яблоко 2. Груша 3. Огурец 4. Арбуз

5. Помидор 6. Картофель 7. Хурма 8. Редис

Предоставить пользователю возможность выбора элемента и определить, какой элемент выбрал пользователь: овощ или фрукт.

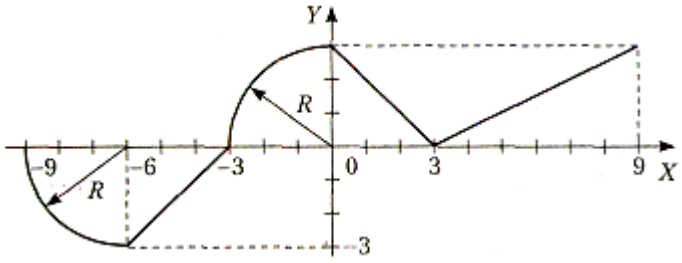
Задание 2. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.



Вариант 4

Задание 1. Для числа от 0 до 35 написать фразу: «Я прочитал N книг», согласовав окончание слова «книга» с числом N.

Задание 2. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.



Вариант 5

Задание 1. Вывести на экран следующую информацию:

Изучение языка Паскаль

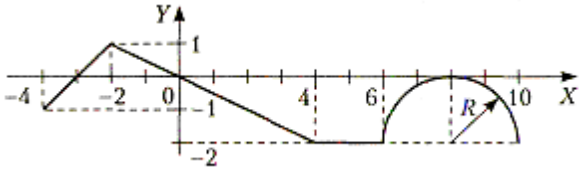
1. While 2. Else 3. Repeat 4. If

5. Begin 6. End 7. Then 8. For

Предоставить пользователю возможность выбора элемента и определить, к какой категории относятся элементы: операторные скобки, циклические или условные операторы.

Задание 2. Написать программу, которая по введенному значению

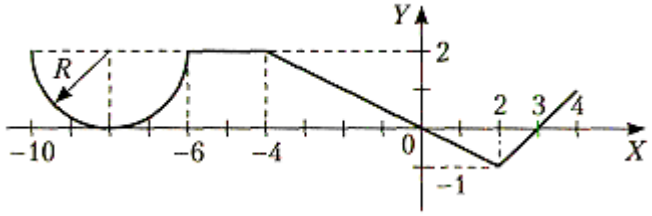
аргумента вычисляет значение функции, заданной в виде графика.



Вариант 6

Задание 1. Для числа от 16 до 75 написать фразу: «Мой стаж N лет», учитывая, что для некоторых N слово «лет» нужно заменить на слово «год» или «года».

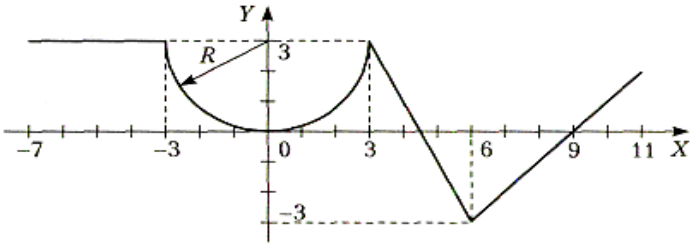
Задание 2. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.



Вариант 7

Задание 1. Для числа от 0 до 31 написать фразу: «Я просмотрел N кинофильмов», согласовав окончание слова «кинофильм» с числом N.

Задание 2. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.



Вариант 8

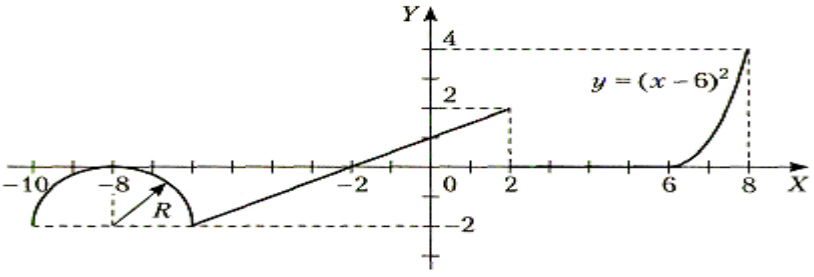
Задание 1. Вывести на экран следующую информацию:

НАПИТКИ

1. Ессентуки 2. Тонус 3. Колесник 4. Я
5. Фруктовый сад 6. Нарзан 7. Мартини 8. Кагор

Предоставить пользователю возможность выбора элемента и определить, что выбрал пользователь: сок, минеральную воду или спиртной напиток.

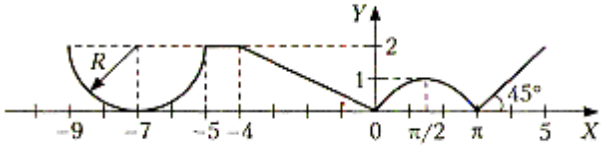
Задание 2. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.



Вариант 9

Задание 1. Пусть пользователь вводит номер дня в неделе. Вывести наименование дня недели и указать, является ли день рабочим или выходным.

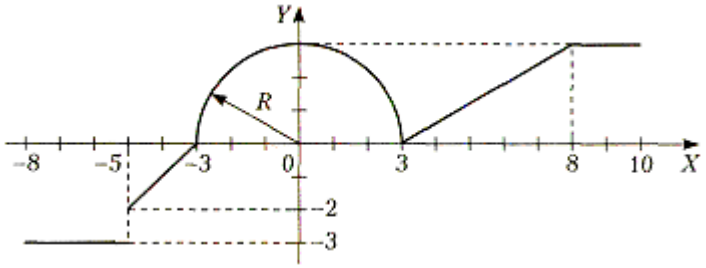
Задание 2. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.



Вариант 10

Задание 1. Составить программу, которая по номеру введенного месяца выводит время года.

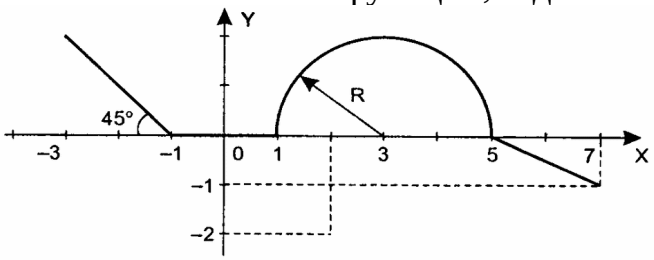
Задание 2. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.



Вариант 11

Задание 1. Составить программу, которая по номеру введенного месяца выводит квартал года.

Задание 2. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.



Вариант 12

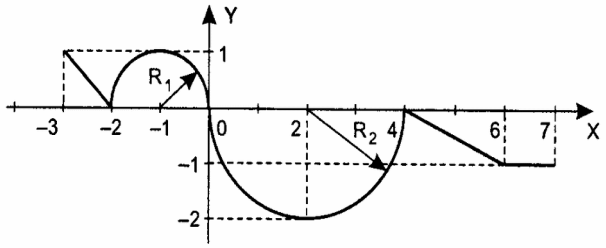
Задание 1. Вывести на экран следующую информацию:

СТРАНА

1. Чили	2. Чад	3. Китай	4. Россия
5. США	6. Норвегия	7. Индия	8. Австралия

Предоставить пользователю возможность выбора элемента (страны) и определить континент, на котором расположена страна.

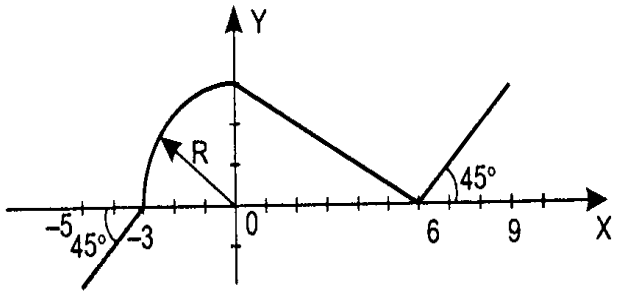
Задание 2. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.



Вариант 13

Задание 1. По дате рождения (месяц и день) определить знак Зодиака.

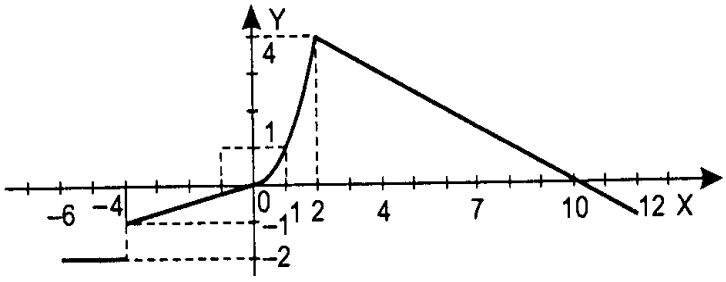
Задание 2. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.



Вариант 14

Задание 1. Составить программу, которая после введенного с клавиатуры числа в диапазоне от 1 до 99 , обозначающего сумму в рублях, дописывает слово «рубль» без кавычек, но в правильном падеже.

Задание 2. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.



2.3. Порядок выполнения лабораторной работы

1. Разработать алгоритмы и программы для заданий в соответствии с вариантом.
2. Подобрать тестовые исходные данные (несколько наборов), для которых заранее известен правильный результат работы программы. Эти тестовые данные будут нужны для выполнения тестирования разработанной программы.
3. Выполнить тестирование программы:

а) выполнить *ручное* тестирование, для чего вручную внимательно проверить алгоритм и программу с целью определения наличия и места

расположения ошибки. Проверку выполнять без запуска программы на выполнение по исходным тестам на бумаге или в редакторе ИС *Free Pascal*;

б) выполнить *экспериментальное* тестирование (тестовые запуски программы) с использованием (с подачей на вход программы) подобранных в пункте 2 тестовых исходных данных.

4. После нахождения и исправления каждой найденной ошибки повторить пункт 3.

5. Запустить программу (оттестированную и исправленную) и зафиксировать в отчете результаты ее работы.

2.4. Требования к содержанию отчета

Отчет о лабораторной работе должен включать:

1. Конспект теоретической части.
2. Схемы алгоритмов для каждого задания.
3. Тексты разработанных программ с комментариями.
4. Результаты тестирования.
5. Копии экранов с полученными результатами.
6. Объяснение полученных результатов.

2.5. Контрольные вопросы

1. Что такое ситуация и таблица ситуаций?
2. Что такое зависимая ситуация? Что такое независимая ситуация?
3. Какие признаки ситуации считаются необходимыми, а какие – достаточными? Как тип признака (необходимый или достаточный) учитывается при описании ситуации?
4. Что такое уточнение описания ситуаций?
5. Какой вид должно иметь описание ситуации в программе?
6. В чем состоит особенность уточнения ситуаций, имеющих общие признаки?
7. Как оператор *case* оформляется на схеме алгоритма?
8. Каков порядок выполнения полной и сокращенной форм оператора *case*?
9. Как оператор *if* оформляется на схеме алгоритма?
10. Каков порядок выполнения полной и сокращенной форм оператора *if?*
11. Как в случае цепочки операторов *if* определить, какому из последовательности операторов *if* соответствует конструкция *else*?
12. В чем суть метода половинного деления при поиске корня уравнения?
13. Как выбираются половины текущего интервала при поиске корня уравнения методом половинного деления?
14. Какие имеются подходы к записи последовательности операторов *if*?