

§ 4. Выполнение арифметических действий над элементами массива

4.1. Вычисление сумм и произведений элементов массива

Операции, выполняемые с элементами массива, соответствуют операциям, которые выполняются над базовым типом элементов массива (пример 4.1).

Пример 4.2. Задан одномерный массив из целых чисел. Найти сумму и произведение элементов этого массива.

I. Исходные данные: массив a и количество элементов n .

II. Результат: S — сумма элементов и P — произведение элементов массива.

III. Алгоритм решения задачи.

1. Ввод исходных данных. Массив вводится поэлементно с клавиатуры.

2. Определение начального значения для суммы ($S := 0$) и для произведения ($P := 1$).

3. В цикле добавляем очередной элемент массива к сумме и к произведению.

4. Вывод результата.

IV. Описание переменных: a — `array[1..10] of integer`; n , S , P — `integer`.

Пример 4.3. Известны отметки по информатике всех учащихся 10 Б класса за первую четверть. Успеваемость в классе будем считать хорошей, если средний балл больше 7, плохой, если средний балл ниже 4, в остальных случаях — успеваемость средняя. Определить успеваемость класса по заданным отметкам.

I. Исходные данные: массив a для хранения отметок и количество учащихся n .

Пример 4.1.

Если базовым типом элементов массива является тип `integer`, то для элементов массива допустимы следующие операции: `+`, `-`, `*`, `div`, `mod`.

Если в массиве хранятся числа типа `real`, то допустимыми будут операции `+`, `-`, `*`, `/`.

Если в массиве хранятся строки, то для каждого его элемента допустимы строковые функции и процедуры.

Пример 4.2.

V. Программа:

```
var a: array[1..10] of integer;
    n, S, P: integer;
begin
  write('Введите n = ');
  readln(n);
  writeln('Вводите элементы');
  for var i := 1 to n do
    read(a[i]);
  S := 0;
  P := 1;
  for var i := 1 to n do
  begin
    S := S + a[i];
    P := P * a[i];
  end;
  writeln('Сумма = ', S);
  writeln('Произведение = ', P);
end.
```

VI. Тестирование.

Окно вывода

```
Введите n = 5
Вводите элементы
3 2 44 -1 3
Сумма = 51
Произведение = -792
```

VII. Анализ результатов. Проверить правильность вычислений можно на калькуляторе.

Пример 4.3.

V. Программа:

```

var a: array[1..30] of integer;
n, S: integer; Sr: real;
begin
write('Количество учащихся ');
readln(n);
writeln('Вводите отметки');
for var i := 1 to n do
read(a[i]);
S := 0;
for var i := 1 to n do
S := S + a[i];
Sr := S / n;
if Sr > 7 then
writeln('Хорошая')
else
if Sr < 4 then
writeln('Плохая')
else
writeln('Средняя');
end.

```

VI. Тестирование.

Окно вывода
Количество учащихся 5
Вводите отметки
10 5 6 8 9
Хорошая

Пример 4.4.

V. Программа:

```

var Kol, Cen: array[1..50] of
integer;
n, Sum: integer;
begin
write('Введите количество
видов товаров ');
readln(n);
for var i := 1 to n do
begin
writeln('Введите количество
товара', i, ' и его цену ');
read(Kol[i], Cen[i]);
end;
Sum := 0;
for var i := 1 to n do
Sum := Sum + Kol[i]*Cen[i];
writeln('Суммарная стоимость
товаров =', Sum);
end.

```

II. Результат: одно из слов — «хорошая», «средняя», «плохая» в зависимости от значения среднего балла.

III. Алгоритм решения задачи.

1. Ввод исходных данных. Сначала вводим количество учащихся в классе, затем массив отметок (поэлементно с клавиатуры).

2. Для определения успеваемости нужно вычислить средний балл (переменная Sr). Средний балл определяется как сумма (переменная S) всех отметок, деленная на количество учащихся в классе. Начальное значение для суммы — $S := 0$.

3. В цикле добавляем очередной элемент массива к сумме.

4. Делим полученную сумму на количество учащихся в классе.

5. Проверяем значение среднего балла и выводим результат.

IV. Описание переменных: a — `array[1..30] of integer`; n, S — `integer`; Sr — `real`.

4.2. Вычисление сумм и произведений при работе с двумя массивами

Пример 4.4. На складе хранятся товары. Для каждого вида товара известно количество единиц товара и цена за единицу товара. Определить суммарную стоимость всех товаров, хранящихся на складе.

I. Исходные данные: Cen — одномерный массив для хранения цены единицы товара каждого вида, Kol — массив для хранения количества товара каждого вида, n — количество видов товаров.

II. Результат: Sum — значение суммарной стоимости товаров на складе.

III. Алгоритм решения задачи.

1. Ввод исходных данных. Для каждого вида товара задается его цена и количество.

2. Стоимость всех товаров одного вида определяется как произведение количества на цену. Суммарная стоимость — сумма всех таких произведений. Начальное значение суммы $\text{Sum} := 0$; В цикле к сумме прибавляются произведения $\text{Kol}[i] * \text{Cen}[i]$.

3. Вывод результата.

IV. Описание переменных: Cen , Kol — **array**[1..50] of integer; n , Sum — integer.

4.3*. Использование массива, элементы которого являются константами

Пример 4.5. Задано натуральное число n ($n < 5000$). Определить, является ли это число простым.

I. Исходные данные: n — натуральное число.

II. Результат: вывод сообщения «простое» или «составное».

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Известно, что число n является простым, если оно не делится ни на одно простое число, не большее \sqrt{n} . Максимальное число по условию — 5000, $\sqrt{5000} \approx 70,7107$. Создадим массив констант s_n из простых чисел, не больших 71.

3. В цикле будем делить число n на каждое из чисел, не больших \sqrt{n}

Пример 4.4. Продолжение.

VI. Тестирование.

Окно вывода

```
Введите количество видов товаров 3
Введите количество товара 1 и его цену
5 2
Введите количество товара 2 и его цену
7 3
Введите количество товара 3 и его цену
4 5
Суммарная стоимость товаров =51
```

Пример 4.5.

Условие $s_n[i] \leq \sqrt{n}$ проверяется долго за счет вызова функции \sqrt{n} . Это условие обычно заменяют эквивалентным: $s_n[i] * s_n[i] \leq n$.

V. Программа:

```
const s_n: array of integer =
(2, 3, 5, 7, 11, 13, 17, 19, 23, 29,
1, 37, 41, 43, 47, 53, 59, 61, 67, 71);
var n,i: integer;
begin
  writeln('Введите число');
  read(n);
  i := 0;
  while (s_n[i] * s_n[i] <= n)
    and (n mod s_n[i] > 0) do
    i := i + 1;
  if s_n[i] * s_n[i] > n then
    writeln('Простое')
  else
    writeln('Составное')
end.
```

VI. Тестирование.

Окно вывода

```
Введите число
2027
Простое
```

Окно вывода

```
Введите число
2021
Составное
```

VII. Анализ результатов. Проверить правильность вычисления можно на калькуляторе или посмотреть в таблице простых чисел¹.

¹ Технические таблицы:

<http://tehtab.ru/guide/guidemathematics/guidemathematicsfigurestable/simplefigures/>
(дата доступа: 10.02.2019).

Пример 4.6.

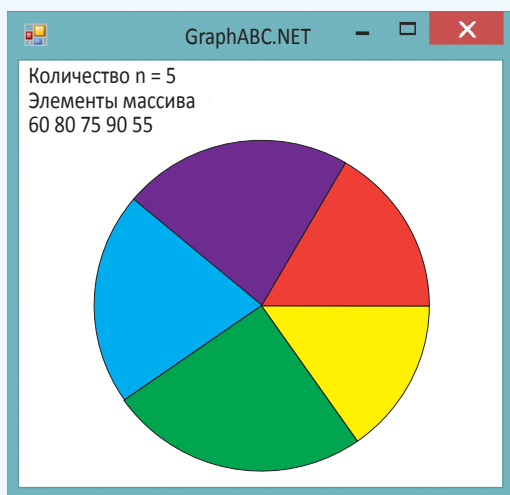
V. Программа:

```

uses graphABC;
var a: array[1..10] of integer;
    n, S, u0, u1: integer;
begin
write('Количество n =');
readln(n);
writeln(n);
writeln('Элементы массива');
for var i := 1 to n do
  read(a[i]);
for var i := 1 to n do
  write(a[i], ' ');
S := 0;
for var i := 1 to n do
  S := S + a[i];
u0 := 0;
for var i := 1 to n do
begin
u1:= u0+trunc(a[i]*360/S);
SetBrushColor(clRandom);
Pie(150,150,100,u0,u1);
u0 := u1;
end;
end.

```

VI. Тестирование.



и хранящихся в массиве констант. Если число n не разделилось ни на одно из них, то оно — простое, иначе — составное.

4. Проверяем, с каким условием закончил работу цикл: число является простым, если последний просмотренный элемент массива больше \sqrt{n} (число ни на что не разделилось).

5. Вывод результата.

IV. Описание переменных: s_n — const **array of** integer; n, i — integer.

4.4. Построение круговой диаграммы

Пример 4.6. Задан одномерный массив из целых чисел. Построить круговую диаграмму по числовым данным, хранящимся в массиве. Например, для 5 элементов массива — 60, 80, 75, 90, 55.

I. Исходные данные: массив a для хранения данных и n — количество данных.

II. Результат: круговая диаграмма.

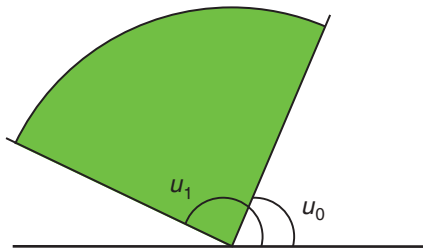
III. Алгоритм решения задачи.

1. Ввод исходных данных. Массив вводится поэлементно с клавиатуры.

2. Круговая диаграмма состоит из n секторов. Градусная мера сектора определяется числовым значением соответствующего элемента в массиве. Суммарное значение всех элементов массива (переменная S) соответствует величине в 360° . Тогда значению элемента массива $A[i]$ будет соответствовать величина — $A[i] * \frac{360}{S}$.

3. Вычисляем сумму всех элементов массива.

4. В цикле строим секторы, градусная мера которых равна целой части величины $A[i] * \frac{360}{S}$.



Для построения сектора нужно знать величины двух углов: u_0 и u_1 . Значение $u_1 = u_0 + A[i] * \frac{360}{S}$. Вначале $u_0 = 0$. Затем в цикле меняем значение u_0 на u_1 . Цвет сектора будем задавать случайным образом. Для вычисления целой части можно использовать функции `trunc` и `round`.

IV. Описание переменных: `a` — `array[1..10] of integer`; `n`, `S`, `u_0`, `u_1` — `integer`.

Пример 4.6. Продолжение.

VII. Постройте по этим данным диаграмму в Excel и сравните.

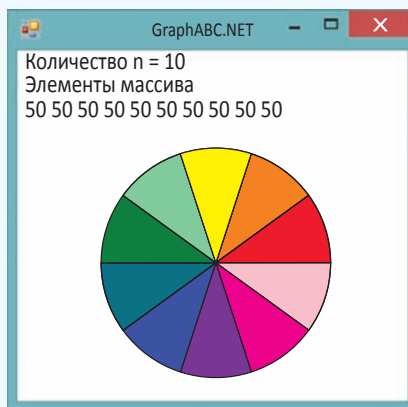
Для задания цвета сектора можно использовать массив, содержащий цветные константы:

```
const d_color: array [1..10]
of Color = (clRed, clOrange,
clYellow, clLightGreen, clGreen,
clTeal, clBlue, clDarkViolet,
clMagenta, clPink);
```

Команду задания цвета сектора нужно будет заменить на:

```
SetBrushColor(d_color[i]);
```

Результат:



- 1. Какие операции допустимы для элементов массива целых чисел?
- 2. Какие операции допустимы для элементов массива вещественных чисел?
- 3. Как записать данные в массив констант?

Упражнения

1 Для задачи из примера 4.2 выполните перечисленные задания.

1. Заполните таблицу.

	n	a	S	P
1	3	-2 -3 -5		
2	5	1 2 3 4 5		
3	10	1 -3 -2 3 4 3 2 4 3 2		

2. Добавьте в таблицу свои значения n и a .
 3. Попробуйте подобрать такие значения элементов массива, чтобы $S = P$, для $n = 2, 5$.
 4. Для $n = 10$ ввели все элементы массива, равные 9. Какой результат получили? Почему? Что нужно исправить в программе для получения правильного результата?
- 2 Для задачи из примера 4.3 добавьте вывод среднего балла.
 - 3 В ходе хоккейного матча удалялись игроки обеих команд. Для каждого удаленного игрока известно время его отсутствия на поле. Определите, какая из команд провела больше времени на скамейке штрафников.
 - 4 Для задачи из примера 4.5 выполните следующее задание:
Введите число 5557. Почему появилась ошибка? Дополните массив констант простыми числами так, чтобы программа могла выдавать ответ для чисел, меньших 10 000. (Для этого можно воспользоваться самой программой или таблицей простых чисел.)
 - 5 Для задачи из примера 4.6 выполните перечисленные задания.
 1. Внесите в программу изменения так, чтобы цвет сектора выбирался из массива констант.
 - 2*. Измените программу так, чтобы диаграмма всегда строилась в центре графического окна. Диаметр круга определяется меньшей из двух величин — шириной или высотой окна.
 - 6* В массивах x и y хранятся координаты точек. Постройте многоугольник, заданный этими координатами. Запросите у пользователя номера двух точек и постройте диагональ многоугольника, соединяющую эти точки.

§ 5. Поиск элементов с заданными свойствами

Человек постоянно сталкивается с задачами поиска требуемой информации. Типичным примером может служить работа со справочниками или библиотечной картотекой. В современном мире информацию ищут с использованием сети Интернет.

Чтобы поиск был результативным и быстрым, разрабатывают эффективные алгоритмы поиска. Важную роль в процессе поиска информации играет способ хранения данных. Одной из самых простых структур для этого является массив.

5.1. Линейный поиск

Рассмотрим, как осуществляется поиск для данных, хранящихся в массиве.

Среди разновидностей простейших задач поиска, встречающихся на практике, можно выделить следующие типы:

1. Найти хотя бы один элемент, равный заданному элементу X . В результате необходимо получить i —