

9



Н. Д. Угринович
И. А. Серёгин
О. А. Полежаева

ИНФОРМАТИКА

Рабочая тетрадь

2

УЧЕНИ

9 КЛАССА

ШКОЛЫ



ИЗДАТЕЛЬСТВО

БИНОМ

Н. Д. Угринович,
И. А. Серёгин, О. А. Полежаева

ИНФОРМАТИКА

9 класс

Рабочая тетрадь

Часть 2



Москва
БИНОМ. Лаборатория знаний

УДК 004.9
ББК 32.97
У27

Угринович Н. Д.

У27 Информатика. 9 класс : рабочая тетрадь : в 2 ч. Ч. 2 /
Н. Д. Угринович, И. А. Серёгин, О. А. Полежаева. —
М. : БИНОМ. Лаборатория знаний, 2017. — 96 с. : ил.

ISBN 978-5-9963-3183-3 (Ч. 2)

ISBN 978-5-9963-3184-0

Рабочая тетрадь входит в состав УМК по информатике для 7–9 классов наряду с тетрадями для 7 и 8 классов, учебниками, методическим пособием для учителя и примерной рабочей программой. Тетрадь выполнена в форме, предусматривающей последовательное выполнение учащимися заданий на компьютере и запись ответов на вопросы, а также оформление отчета. Благодаря такой форме организации занятий педагог может оценить полноту выполнения заданий и сформировать у учащихся культуру оформления практической деятельности проектного и исследовательского характера.

Тетрадь состоит из работ по всем темам учебника для 9 класса с опорой на теоретический материал учебника. Дано подробное описание занятий.

УДК 004.9
ББК 32.97

Учебное издание

Угринович Николай Дмитриевич
Серёгин Игорь Александрович
Полежаева Ольга Александровна

ИНФОРМАТИКА

9 класс

Рабочая тетрадь

В 2 частях

Часть 2

Научный редактор *М. С. Цветкова*
Ведущий редактор *О. А. Полежаева*
Художник *Н. А. Новак*
Технический редактор *Е. В. Денюкова*
Корректор *Е. Н. Клитина*
Компьютерная верстка: *Л. В. Катуркина*

Подписано в печать 27.12.16. Формат 70х100/16. Усл. печ. л. 7,8.
Тираж 3000 экз. Заказ № 39642.

ООО «БИНОМ. Лаборатория знаний»
127473, Москва, ул. Краснопролетарская, д. 16, стр. 1,
тел. (495)181-53-44, e-mail: binom@lbz.ru
<http://www.lbz.ru>, <http://metodist.lbz.ru>

Отпечатано в соответствии с качеством предоставленных издательством
электронных носителей в АО «Саратовский полиграфкомбинат».
410004, г. Саратов, ул. Чернышевского, 59. www.sarpk.ru

ISBN 978-5-9963-3183-3 (Ч. 2)
ISBN 978-5-9963-3184-0

© ООО «БИНОМ. Лаборатория знаний», 2017

Условные обозначения

В работе с тетрадью вам помогут навигационные значки:



— выбор одного или нескольких ответов;



— запись короткого ответа;



— запись развёрнутого ответа;



— работа на компьютере;



— поиск информации;



— построение графов и схем;



— заготовка из электронного приложения.

Ряд заданий выделен в блоки с пометкой «Задания для дополнительного выполнения».

Лабораторная работа № 8

УЧИМСЯ ПРОГРАММИРОВАТЬ ЗАДАЧИ С МАССИВАМИ

(К параграфу учебника:

1.4. Арифметические, строковые и логические выражения)

Тема: изучение программирования алгоритмов с массивами.

Цель работы: научиться составлять программы, содержащие массивы.

Программное обеспечение: компьютер с установленной операционной системой Windows или Linux; текстовый процессор Microsoft Word; среда программирования на языке Паскаль.



1. Что такое массив?



2. Что такое размер массива?



3. Какими величинами характеризуется каждый элемент массива?



4. Где размещается описание массива в программе?



5. Как задаётся размер массива в программах на Паскале?



6. Как описывается массив?

- 

```
k:=0; {ввод начального значения}
for i:=1 to n do
    if a[i]<0 then k:=k+1; {подсчёт количества}
writeln('Количество отрицательных чисел = ', k); {вывод результата}
```

Пример 3. Требуется написать фрагмент программы уменьшения в два раза значений всех элементов массива, имеющих нечётные индексы.

Решение

Фрагмент программы на Паскале:

```
i:=1;
repeat
  a[i]:=a[i] div 2;      {уменьшение в 2 раза значений
                          элементов, имеющих нечётные индексы}
  i:=i+2;
until i>n;
for i:=1 to n do
  write(a[i], ' '); {вывод получившегося массива}
writeln;           {перевод курсора на новую строку}
```

Возможно и другое решение:

```
for i:=1 to (n+1) div 2 do
  a[i*2-1]:=a[i*2-1] div 2;
for i:=1 to n do
  write(a[i], ' '); {вывод получившегося массива}
writeln;           {перевод курсора на новую строку}
```

Пример 4. Требуется написать фрагмент программы уменьшения в два раза всех чётных значений элементов массива.

Решение

Фрагмент программы на Паскале:

```
for i:=1 to n do
  if a[i] mod 2 = 0 then a[i]:=a[i] div 2;
for i:=1 to n do
  write(a[i], ' '); {вывод получившегося массива}
writeln;           {перевод курсора на новую строку}
```

Пример 5. Требуется написать фрагмент программы, которая находит максимальный элемент массива и его индекс.

*Решение***Фрагмент программы на Паскале:**

```

max:=a[1];
imax:=1;
for i:=2 to n do
  if a[i]>max then
    begin
      max:=a[i];
      imax:=i;
    end;
write('Максимальный элемент = ', max,
      ' его индекс = ', imax);      {вывод результата}

```

14. Решение задач на Паскале. Самостоятельно составьте и выполните программу решения следующей задачи на Паскале.

Задача. Напишите программу поиска второго по величине (т. е. следующего по величине за максимальным) элемента в числовом массиве из 30 различных элементов.

- 1) Дан шаблон программы решения задачи. Дополните его — составьте программу.

```

var
  -----;
  -----;
  -----;

begin
  {получение и вывод массива}
  for i:=1 to 30 do
    -----
    -----;
    -----;
    -----;

  writeln;
  {первые два элемента массива становятся двумя
                                     текущими максимумами}

  max1:=a[1];
  max2:=a[2];
  if max2>max1 then begin
    max1:=a[2];
    max2:=a[1];
  end;

```



```

{проверяем очередной элемент}
for i:=3 to 30 do
  begin
    {если он больше второго максимума, назначаем его
                                     вторым максимумом}
    if a[i]>max2 then max2:=a[i];
    {если теперь первый максимум окажется меньше
                                     второго, то меняем их местами}
    if max1<max2 then begin
      x:=max1;
      max1:=max2;
      max2:=x;
    end;
  end;
  {выводим второй максимум}
  writeln (max2);
end.

```



2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.



3) Сделайте и сохраните скриншот решения задачи.



15. В таблице *Dat* хранятся данные о численности учеников в классах (*Dat*[1] — число учеников в первом классе, *Dat*[2] — во втором и т. д.). Определите, какое число будет выведено в результате работы следующей программы (из материалов ГИА).

```

var k, m: integer;
    dat: array[1..11] of integer;
begin
  dat[1]:=20; dat[2]:=25;
  dat[3]:=19; dat[4]:=25;
  dat[5]:=26; dat[6]:=22;
  dat[7]:=24; dat[8]:=28;
  dat[9]:=26; dat[10]:=21;
  dat[11]:=27;
  m:=0;
  for k:=1 to 11 do
    if dat[k]>22 then
      begin
        m:=m+1
      end

```

```
end;
writeln(m)
end.
```

Ответ:

16. Значения двух массивов $A[1..100]$ и $B[1..100]$ задаются с помощью следующего фрагмента программы.

```
for n:=1 to 100 do
  A[n]:=n-10;
for n:=1 to 100 do
  B[n]:=A[n]*n
```

Сколько элементов массива B будут иметь положительные значения? (Из материалов ЕГЭ.)

Ответ:

17. Значения двух массивов $A[1..100]$ и $B[1..100]$ задаются с помощью следующего фрагмента программы.

```
for n:=1 to 100 do
  A[n]:= (n-80)*(n-80);
for n:=1 to 100 do
  B[101-n]:=A[n];
```

Какой элемент массива B будет наибольшим? (Из материалов ЕГЭ.)

Ответ:

18. В программе используется одномерный целочисленный массив A с индексами от 0 до 10. Ниже представлен фрагмент программы, в которой значения элементов сначала задаются, а затем меняются.

```
for i:=0 to 10 do
  A[i]:=i;
for i:=0 to 10 do
  begin
    A[10-i]:=A[i];
    A[i]:=A[10-i];
  end;
```

Чему будут равны элементы этого массива после выполнения фрагмента программы? (Из материалов ЕГЭ.)

Ответ:





19. В программе описан одномерный целочисленный массив A с индексами от 0 до 10. Ниже представлен фрагмент этой программы, в котором значения элементов массива сначала задаются, а затем меняются.

```
for i:=0 to 10 do  
  A[i]:=i-1;  
for i:=1 to 10 do  
  A[i-1]:=A[i];  
A[10]:=10;
```

Как изменятся элементы этого массива после выполнения фрагмента программы? В ответе укажите только число — номер верного варианта. (Из материалов ЕГЭ.)

- 1) все элементы, кроме последнего, окажутся равны между собой
- 2) все элементы окажутся равны своим индексам
- 3) все элементы, кроме последнего, будут сдвинуты на один элемент вправо
- 4) все элементы, кроме последнего, уменьшатся на единицу

Ответ:



20. В программе описан одномерный целочисленный массив с индексами от 0 до 10. Ниже представлен фрагмент программы, обрабатывающей данный массив.

```
s:=0;  
n:=10;  
for i:=1 to n do begin  
  if A[i]-A[i-1]>10 then  
    s:=s+A[i];  
end
```

В начале выполнения этого фрагмента в массиве находились числа 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, т. е. $A[0] = 0$, $A[1] = 10$ и т. д. Чему будет равно значение переменной s после выполнения данного фрагмента программы? В ответе укажите только число — номер верного варианта. (Из материалов ЕГЭ.)

- 1) 0 2) 10 3) 30 4) 550

Ответ:



21. Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 100 — баллы учащихся выпускного класса за итоговый тест по информатике. Для получения положительной оценки за тест требовалось набрать не менее 20 баллов. Напишите программу, которая позволит найти и вывести минимальный балл среди учащихся, получивших за тест положительную оценку. Известно, что в классе хотя бы один учащийся получил за тест положительную оценку. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

```
const
    N=30;
var
    a: array [1..N] of integer;
    i, j, min: integer;
begin
    for i:=1 to N do readln(a[i]);
    ...
end.
```

В качестве ответа необходимо привести фрагмент программы, который должен находиться на месте многоточия. (Из материалов ЕГЭ.)

Ответ:

Отчёт по лабораторной работе № 8

Распечатайте и вклейте скриншот решения задачи:

**Выводы**

Структура данных «массив» нужна для того, чтобы:

Оценка учителя: -----

Лабораторная работа № 9

УЧИМСЯ ПРОГРАММИРОВАТЬ ЗАДАЧИ С МАССИВАМИ: СОРТИРОВКА; ДВУМЕРНЫЕ МАССИВЫ

(К параграфу учебника:

1.4. Арифметические, строковые и логические выражения)

Тема: изучение программирования алгоритмов с массивами: сортировки; работы с двумерными массивами.

Цель работы: научиться использовать алгоритмы сортировок для упорядочения массивов; научиться работать с двумерными массивами (матрицами).

Программное обеспечение: компьютер с установленной операционной системой Windows или Linux; текстовый процессор Microsoft Word; среда программирования на языке Паскаль.

1. Что такое матрица?
2. Какими величинами характеризуется каждый элемент матрицы?
3. Как описывается матрица?
4. Как организуется вывод матрицы?
5. Может ли матрица содержать один элемент?
6. Может ли матрица не содержать ни одного элемента?





7. Может ли в матрице в каждой строке быть разное количество элементов? -----



8. Могут ли в некоторую матрицу входить следующие элементы?

1	1.41	1.37	2
7	14	3.14	9



9. Решение задач на Паскале. Самостоятельно составьте и выполните программы решения следующих задач на Паскале.

Задача 1. Напишите программу формирования матрицы размера $n \times m$ следующего вида:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
...				



- 1) Дан шаблон программы решения задачи. Дополните его — составьте программу.

```

program matritca_1;
var
  a: _____ [1..100, 1..100] _____;
  i, j, m, n: _____;
begin
  writeln ('Введите размер матрицы');
  _____;
  for i:= ___ to ___ do
    for j:= ___ to ___ do
      _____;
  for i:= 1 to n do
    _____
    for j:=1 to m do
      _____;
    writeln;
  _____
end.
```

2) Наберите в среде разработки программ и выполните программу. Если обнаружите ошибки, исправьте их.

3) Сделайте и сохраните скриншот решения задачи.

Задача 2. Напишите программу сортировки одномерного массива по возрастанию. Выберите, по своему усмотрению, либо алгоритм сортировки «пузырьком», либо алгоритм сортировки выбором.

1) Дан шаблон программы решения задачи. Дополните его — составьте программу.

```
program sortirovka;
uses crt;
var
  a: _____;
  i, k, n, _____ : _____;
begin
  randomize;
  writeln ('Введите размер массива');
  _____;
  {получение массива случайных чисел}
  for i:= ___ to ___ do
    _____;
  {вывод полученного массива}
  for i:=1 to n do
    _____;
    _____;
  {сортировка}
```



```

{вывод отсортированного массива}

```

```

for i:=1 to n do

```

```

    -----;

```

```

    -----

```

2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.

3) Сделайте и сохраните скриншот решения задачи.

Для дополнительного выполнения

Задача 3. Создайте квадратную нулевую матрицу размера $n \times n$. Заполните главную диагональ цифрами 8, а побочную — цифрами 9.

1) Дан шаблон программы решения задачи. Дополните его — составьте программу.

```

program matritca_3;

```

```

var

```

```

    a: ----- [ ----- ] -----;

```

```

    i, j, n: -----;

```

```

begin

```

```

    writeln ('Введите размер квадратной матрицы');

```

```

    -----;

```

```

    for i:= ___ to ___ do

```

```

        for j:= ___ to ___ do

```

```

            -----;

```

```

        for i:= ___ to ___ do

```

```

            -----

```

```

                a[ ___ ]:=8;

```

```

                a[ ___ ]:=9;

```

```

-----
for i:= 1 to n do

```

```

-----
    for j:= 1 to n do

```

```

        -----;

```

```

    -----

```

```

end.

```

2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.

3) Сделайте и сохраните скриншот решения задачи.

Задача 4. Заполните матрицу размера $n \times m$ случайными числами из отрезка $[1..100]$. Найдите позицию минимального элемента среди максимальных элементов каждого ряда.

1) Дан шаблон программы решения задачи. Дополните его — составьте программу.

```

program matritca_5;

```

```

uses crt;

```

```

var

```

```

    a: -----

```

```

    i, j, k, n, m, imax, jmax, iimin, jjmin, min: -----

```

```

begin

```

```

    randomize;

```

```

    writeln ('Введите размер квадратной матрицы');

```

```

    -----;

```

```

    {получение матрицы случайных чисел}

```

```

    for i:= ___ to ___ do

```

```

        for j:= ___ to ___ do

```

```

            -----;

```

```

    {вывод полученной матрицы}

```

```

    for i:=1 to n do

```



```

-----
    for j:=1 to m do
        -----
        -----
    -----
{основная часть}
min:=101;    {Подумайте почему}
for i:= 1 to n do
    -----
    {поиск позиции максимального элемента в каждой
                                         строке матрицы}

    imax:= ___;
    jmax:= ___;
    for j:=1 to m do
        if _____ then
            -----
            imax := ___;
            jmax := ___;
            -----;
    {поиск позиции минимального из максимальных
                                         элементов}

    if _____ then
        -----
        min:= _____;
        iimin:= _____;
        jjmax:= _____;
        -----;
        -----;
    writeln _____;
    -----

```

2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.

3) Сделайте и сохраните скриншот решения задачи.



10. Значения двумерного массива задаются с помощью вложенного оператора цикла в представленном фрагменте программы. Чему будет равно значение $B(2, 4)$? (Из материалов ЕГЭ.)



```
for n:=1 to 5 do
  for k:=1 to 5 do
    B[n, k]:=n+k;
```

Ответ:

11. Все элементы двумерного массива A размером 10×10 элементов первоначально были равны 0. Затем значения элементов меняются с помощью вложенного оператора цикла в представленном фрагменте программы. Сколько элементов массива в результате будут равны 1? (Из материалов ЕГЭ.)



```
for n:=1 to 4 do
  for k:=n to 4 do
    begin
      A[n, k]:=A[n, k]+1;
      A[k, n]:=A[k, n]+1;
    end
```

Ответ:

12. Значения двумерного массива размером 7×7 задаются с помощью вложенного оператора цикла в представленном фрагменте программы. Сколько элементов массива будут иметь положительные значения? (Из материалов ЕГЭ.)



```
for n:=1 to 7 do
  for k:=1 to 7 do
    B[n, k]:=k-n;
```

Ответ:



13. Дан фрагмент программы. Чему будет равно значение $B[1, 4]$? (Из материалов ЕГЭ.)

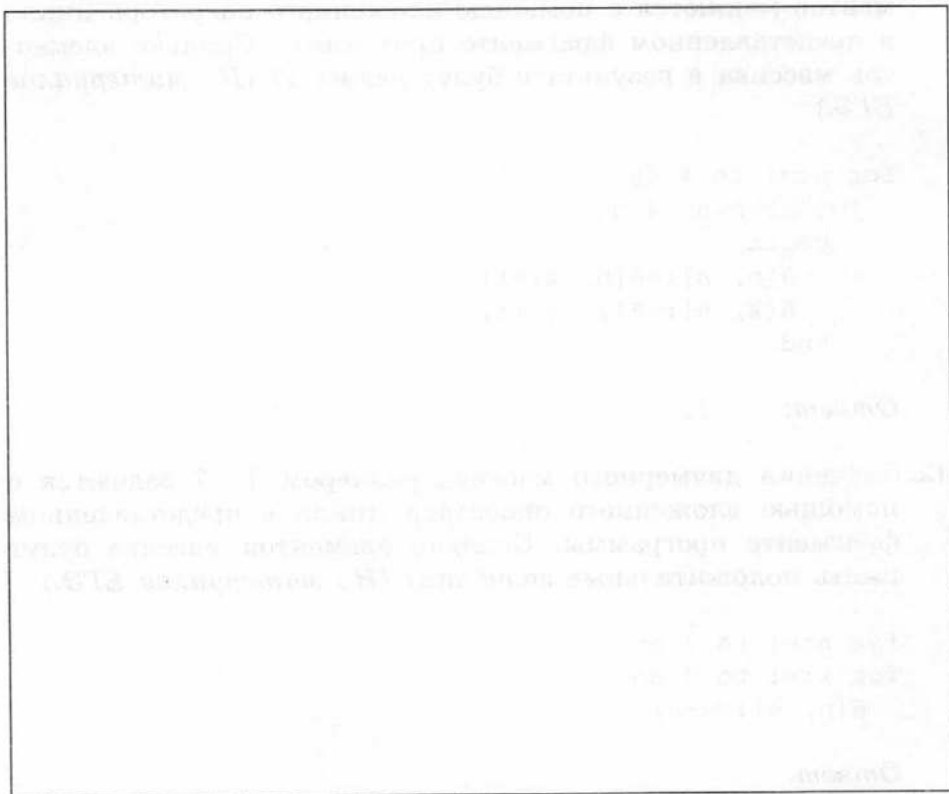
```
for n:=1 to 5 do
  for m:=1 to 4 do
    begin
      k:=n-m+5;
      B[n, m]:=n*n*n-m*m+k;
    end
```

Ответ: -----

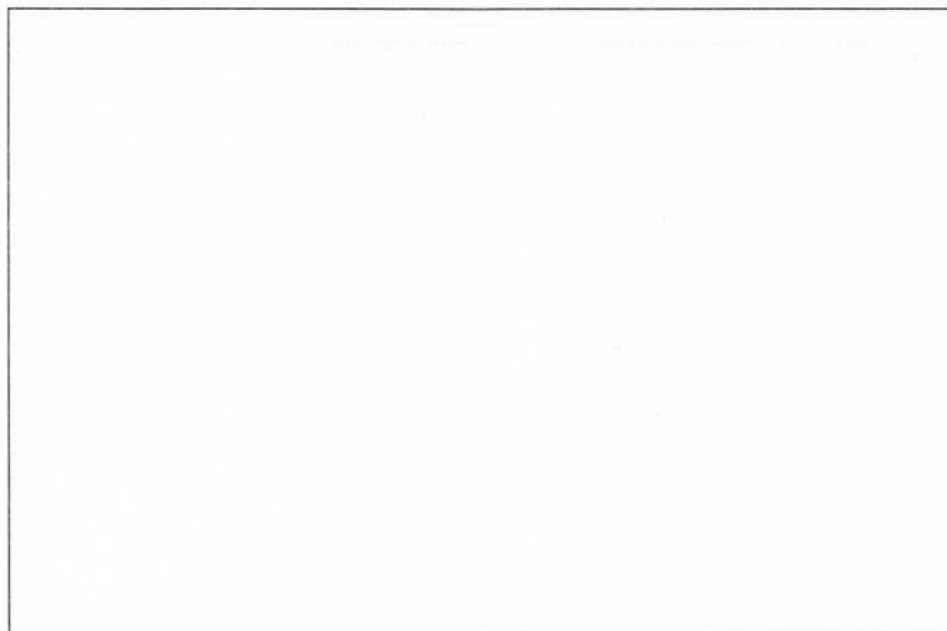
Отчёт по лабораторной работе №9



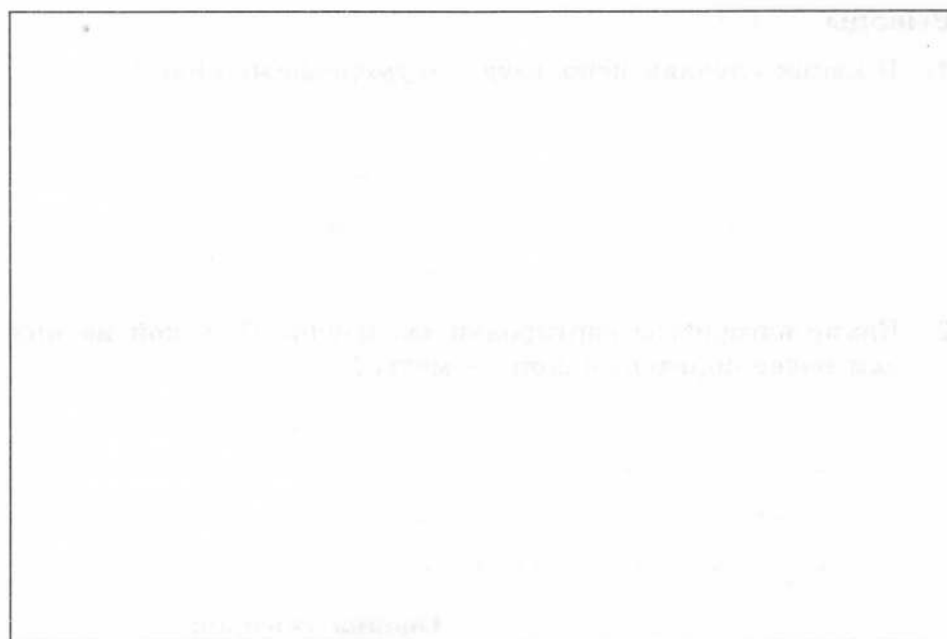
1. Распечатайте и вклейте скриншот решения задачи 1:



2. Распечатайте и вклейте скриншот решения задачи 2:



3. Распечатайте и вклейте скриншот решения задачи 3 (дополнительно):





4. Распечатайте и вклейте скриншот решения задачи 4 (дополнительно):



Выводы

1. В каких случаях используют двумерные массивы?

2. Какие алгоритмы сортировки вы изучили? Какой из них вам более понятен, какой — менее?

Оценка учителя: -----

Лабораторная работа № 10

УЧИМСЯ ПРОГРАММИРОВАТЬ ЗАДАЧИ С СИМВОЛЬНЫМ И СТРОКОВЫМ ТИПАМИ ДАННЫХ

(К параграфу учебника:

1.5. Функции в языках объектно-ориентированного
и процедурного программирования)

Тема: изучение программирования алгоритмов с символьным
и строковым типами данных.

Цель работы: научиться использовать в программах символь-
ный и строковый типы данных.

Программное обеспечение: компьютер с установленной опера-
ционной системой Windows или Linux; текстовый процессор
Microsoft Word; среда программирования на языке Паскаль.

1. Чему будет равно значение переменной *y* после выполне-
ния оператора `y:=CHR(ORD(X))`?



2. Определите значения следующих функций (таблицу ASCII
вы можете найти в любом справочнике по программирова-
нию или в Интернете):



CHR(68)

ORD('d')

PRED(1)

SUCC('Я')

3. Чему будет равно значение переменной *b* после выполне-
ния операторов `a:='компьютер'; b:=copy(a, 2, 2)`?



.....



4. Что делает приведённый фрагмент программы?

```
readln(a);
for i:=1 to length(a)-2 do
  if (a[i]='э') and (a[i+1]='в') and (a[i+2]='м') then
    writeln(i);
```

Как по-другому можно записать фрагмент этой программы?

5. В приведённом ниже фрагменте алгоритма, записанном на алгоритмическом языке, переменные a, b, c имеют тип «строка», а переменные i, k — тип «целое».

Используются следующие функции.

Длина (a) — возвращает количество символов в строке a. (Тип «целое».)

Извлечь (a, i) — возвращает i-й (слева) символ в строке a. (Тип «строка».)

Склеить (a, b) — возвращает строку, в которой записаны сначала все символы строки a, а затем — все символы строки b. (Тип «строка».)

Значения строк записываются в одинарных кавычках. (Например: a := 'дом'.)

(Из материалов ОГЭ.)

Фрагмент алгоритма:

```
i:=Длина(a)
k:=2
b:='А'
пока i>0
нц
  c:=Извлечь(a, i)
```

```
b:=Склеить (b, c)
i:=i-k
кц
b:=Склеить (b, 'Т')
```

Какое значение будет у переменной *b* после выполнения вышеприведённого фрагмента алгоритма, если значение переменной *a* было 'ПОЕЗД'? В ответе запишите только число — номер верного варианта.

- 1) 'АДЕПТ' 2) 'АДЗЕОП' 3) 'АДТЕПТ' 4) 'АДЗОТ'

Ответ:

Напишите эту программу на языке Паскаль:

```
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
```

6. Примеры решения задач на Паскале. Ознакомьтесь с примерами решения задач на Паскале.

Пример 1. Требуется написать программу, вычисляющую сумму двух введённых чисел: *a* и *b*. После каждого вычисления должен появляться запрос о том, нужно ли сложить ещё какие-нибудь два числа («Завершить программу? (Д/Н)»). После того как пользователь нажмёт клавишу «Д» или «Н», программа либо продолжит вычисления, либо закончит свою работу.

*Решение***Программа на Паскале:**

```

var a, b, s: real;
    c, d: char;
begin
  repeat
    write('Введите два числа '); readln(a, b);
    s:=a+b;
    writeln(s:0:2); {количество знаков целой части
                    устанавливается автоматически}
    writeln('Завершить программу? (Д/Н) ');
    readln(c);
    d:=upcase(c)
  until c='Д';      {программа завершит работу,
                    если будет введена буква Д}

end.

```

Пример 2. Вводится предложение. Между словами ровно по одному пробелу. Нужно определить, сколько в этом предложении слов.

Решение

Количество слов на 1 больше количества пробелов в предложении (если никаких лишних пробелов нет). Поэтому подсчитаем количество пробелов и увеличим его на 1 (или начнём считать не с 0, а с 1).

Программа на Паскале:

```

var i, x, k: byte;
    a: string;
begin
  writeln('Введите предложение'); readln(a);
  x:=length(a); {определяем длину предложения}
  k:=1;
  for i:=1 to x do
    if a[i]=' ' then k:=k+1;
  writeln('Количество слов = ', k);
end.

```

Пример 3. Вводится предложение. Между словами ровно по одному пробелу. Нужно заменить в этом тексте все слова «ЭВМ» на слово «компьютер».

Решение

Так как слово «ЭВМ» не может быть частью другого слова, то ищем все его вхождения и заменяем на слово «компьютер».

Программа на Паскале:

```
var p: byte;
    a: string;
begin
  writeln('Введите текст');
  readln(a);
  x:=length(a); {определяем длину текста}
  p:=pos('ЭВМ', a); {определяем позицию первого
                    вхождения слова ЭВМ}

  while p<>0
  begin
    delete(a, p, 3);
    insert('компьютер', a, p);
    p:=pos('ЭВМ', a); {определяем позицию очередного
                      вхождения слова ЭВМ}

  end;
  writeln(a);
end.
```

7. Решение задач на Паскале. Самостоятельно составьте и выполните программы решения следующих задач на Паскале.

Задача 1. Определите количество слов в предложении, если между словами, а также в начале и в конце предложения может стоять любое произвольное количество пробелов.

- 1) Дан набор команд в произвольном порядке (других команд в программе быть не должно). Соберите программу путём выбора правильных команд. В фигурные скобки впишите комментарии к программе.

```
program kol-vo slov;
k:=k+1;{ ----- }
writeln('Введите текст');
```





- ```

program pereverttysh;
writeln('Введите текст');
end.

writeln(ss);
readln(st);

begin
p:=length(st);
ss:=st[i]+ss; { ----- }
for i:=1 to p do
ss:=''; { ----- }
var st, ss: string; i, p: integer;

```

[illegible]



2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.



3) Сделайте и сохраните скриншот решения задачи.

Для дополнительного выполнения

**Задача 3.** Выведите все слова текста в столбик.



1) Дан набор команд в произвольном порядке (других команд в программе быть не должно). Соберите программу путём выбора правильных команд. В фигурные скобки впишите комментарии к программе.

```
program vstolbik;
end;
if (st[i]<>' ') and (st[i]=' ') then { _____ }
end.
writeln(ss);
if (st[i]=' ') and (st[i]<>' ') then { _____ }
readln(st);
begin
p:=length(st);
begin
ss:=copy(st, i+1, i-k); { _____ }
var st, ss: string; i, k, p: integer;
begin
end;
k:=i; { _____ }
for i:=1 to p do
st:=' '+st+' '; { _____ }
writeln('Введите текст');
```

**Текст программы:**

-----  
-----

[illegible]

- 



- ```
program poslprobel;  
p:=length(st);  
writeln('Введите текст');  
end.  
  
writeln(k);  
readln(st);  
var st: string; k, p: integer;
```

```

k:=k-1; { _____ }
begin
k:=p; { _____ }
while st[k]<>' ' do { _____ }

```

Текст программы:

```

-----
-----
-----
-----
-----
-----
-----
-----
-----
-----

```

- 2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.
- 3) Сделайте и сохраните скриншот решения задачи.

Задача 5. Шифр Цезаря. Замените каждую букву текста на английском языке следующей по алфавиту («z» заменяется на «a»). Остальные символы не заменяйте.

- 1) Дан шаблон программы решения задачи. Дополните его — составьте программу.

```

program shifr_tcezarya;
var st, ss: string;
    i, p, a, r, z, k :integer;
    c: char;

```

```

begin
    {st - исходный текст; ss - получаемый зашифрованный
                                     текст}

    writeln('Введите текст');

    -----
    ----- {находим длину текста}
    ----- {определяем "пустой" текст}
    ----- {двигаемся по исходному тексту}

    -----
    r:= ----- {к коду i-й буквы текста
                                     прибавляем 1}

    z:= ----- {находим код последней
                                     английской буквы - "z"}

    k:= ----- {находим приращение кода с
                                     учётом возможного перехода
                                     на первую букву}

    a:= ----- {находим код первой
                                     английской буквы - "a"}

    c:= ----- {к коду первой английской
                                     буквы добавляем приращение и находим
                                     новое значение шифруемой буквы}

    ss:= -----; {добавляем закодированную
                                     букву в новый текст}

    -----
    ----- {выводим полученный
                                     {зашифрованный текст}

    -----

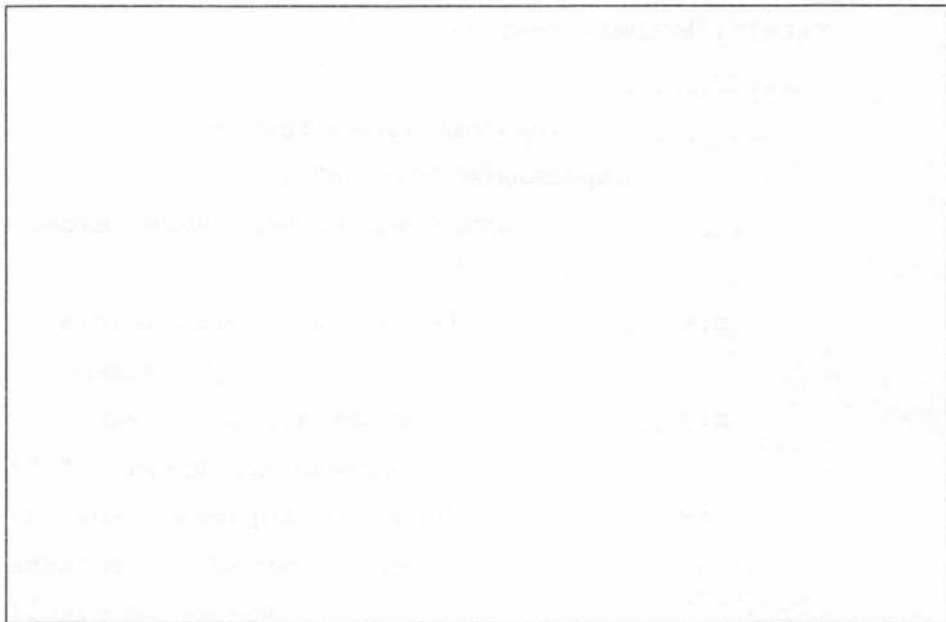
```

- 2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.
- 3) Сделайте и сохраните скриншот решения задачи.

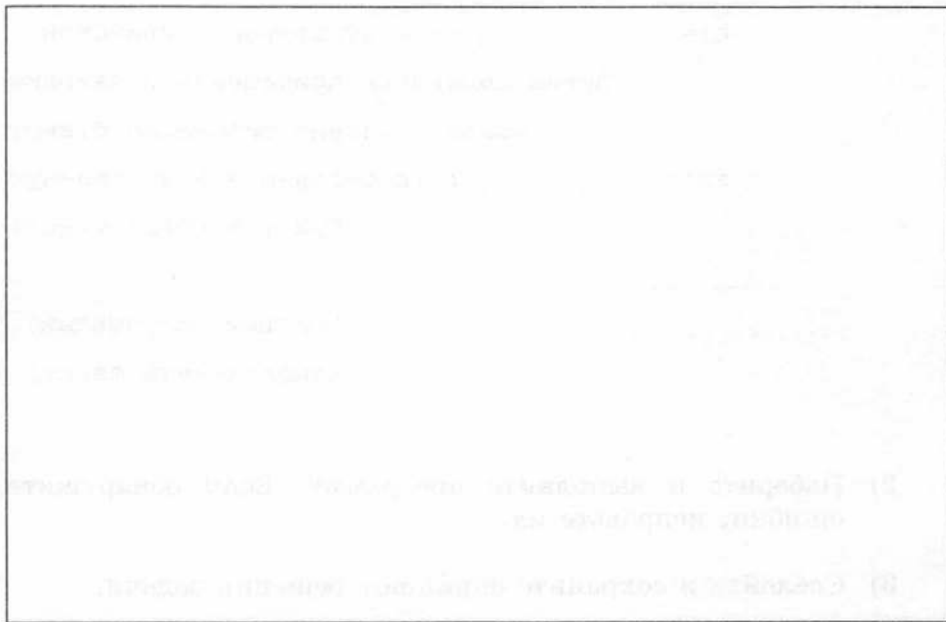


Отчёт по лабораторной работе № 10

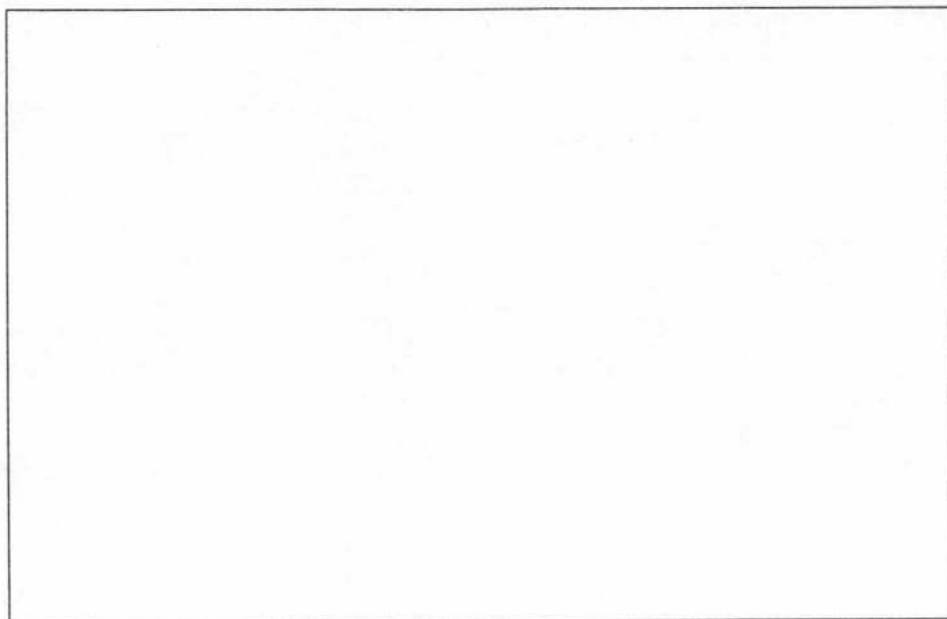
1. Распечатайте и вклейте скриншот решения задачи 1:



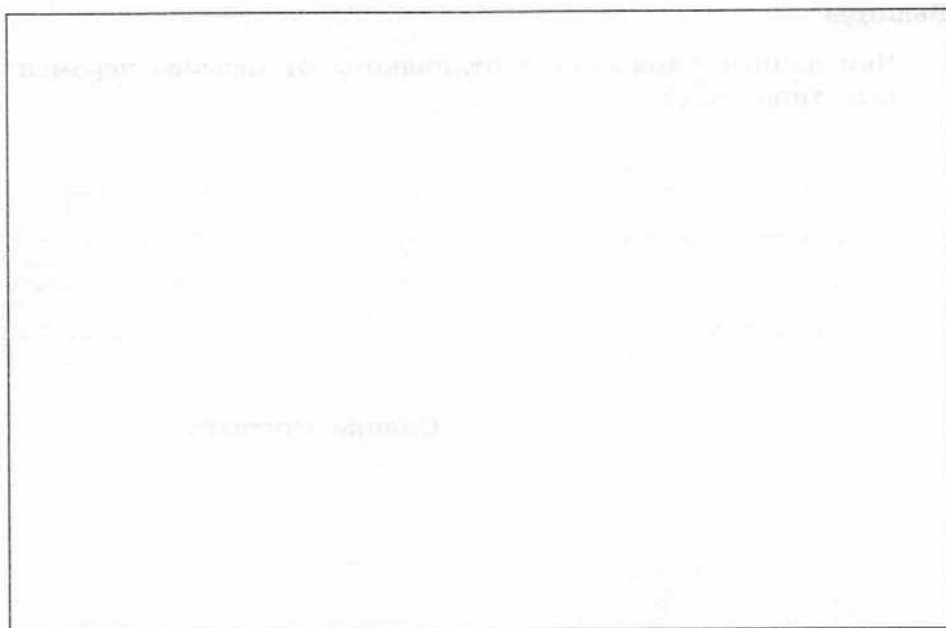
2. Распечатайте и вклейте скриншот решения задачи 2:



- 3. Распечатайте и вклейте скриншот решения задачи 3 (дополнительно):**

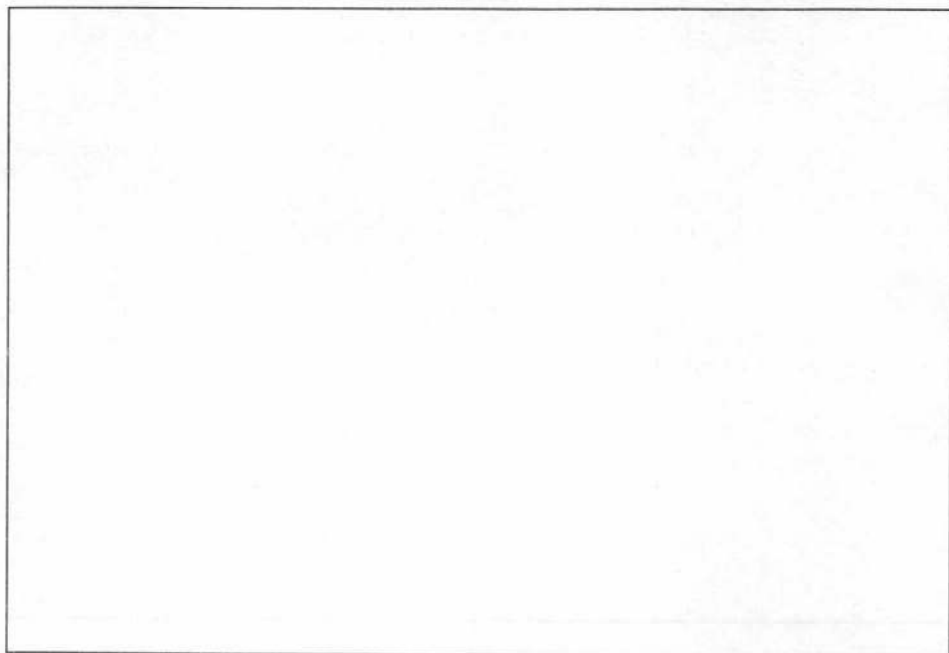


- 4. Распечатайте и вклейте скриншот решения задачи 4 (дополнительно):**





5. Распечатайте и вклейте скриншот решения задачи 5 (дополнительно):



Выводы

Чем данные типа `string` отличаются от массива переменных типа `char`?

Оценка учителя: -----

Лабораторная работа № 11

УЧИМСЯ ПРЕОБРАЗОВЫВАТЬ СТРОКОВЫЕ ВЕЛИЧИНЫ В ЧИСЛОВЫЕ И ОБРАТНО

(К параграфу учебника: 1.5. Функции в языках
объектно-ориентированного и процедурного программирования)

Тема: изучение преобразования строковых величин.

Цель работы: научиться преобразовывать строковые величины в числовые и обратно.

Программное обеспечение: компьютер с установленной операционной системой Windows или Linux; текстовый процессор Microsoft Word; среда программирования на языке Паскаль.

- 1. Примеры решения задач на Паскале.** Ознакомьтесь с примерами решения задач на Паскале.

Пример 1. Проверка корректности ввода данных. Требуется написать программу, запрашивающую у пользователя повторение ввода до тех пор, пока он не введёт корректное число.

Решение

Программа на Паскале:

```
var s: string;
    err: integer;
    a: real;
    err: integer; {err - код ошибки (error)}
begin
  repeat
    write ('Введите число:');
    readln(s);
    val(s, a, err); {преобразуем строку в число}
```

```
if err<>0 then
  writeln('Неверный символ в позиции ', err);
until err=0;
writeln ('Число ', a, ' введено корректно.');
```

end.

Пример 2. Имеется строка вида <число>+<число>=. Оба числа целые.

Нужно подсчитать сумму этих чисел и дописать её в конец строки.

Решение

Программа на Паскале:

```
var st, ss: string;
    a1, a2, p, k, err: integer;
begin
  writeln ('Введите текст');
  readln (st);
  p:=length(st);
  k:=pos('+', st);
  ss:=copy(st, 1, k-1);
  val(ss, a1, err);
  ss:=copy(st, k, p-k);
  val(ss, a2, err);
  a1:=a1+a2;
  str(a1, ss);
  ss:=st+ss;
  writeln(ss);
end.
```

- 2. Решение задач на Паскале.** Решите поэтапно — по блокам — одну из предложенных ниже задач (выберите задачу 1 или 2 по согласованию с учителем).

Задача 1. На вход программе подаются сведения о сдаче теста учениками 9-х классов некоторой средней школы. В первой строке сообщается количество учеников N , которое не меньше 10, но не превосходит 100. Каждая из следующих N строк имеет формат: <Фамилия> <Имя> <Результат>, где <Фамилия> — строка, состоящая не более чем из 20 символов, <Имя> — строка, состоящая не более чем из 15 симво-

лов, <Результат> — число в пределах от 0 до 100 баллов. <Фамилия> и <Имя>, а также <Имя> и <Результат> разделены одним пробелом.

Пример входной строки:

Иванов Пётр 75

Требуется написать программу, которая будет выводить на экран фамилию и имя лучшего ученика и его результат. Если максимальный результат набрали несколько человек, то вывести фамилию и имя первого в этом списке.

(Из материалов ЕГЭ, упрощённый вариант.)

План решения задачи по блокам

Блок 1. Введите строку и разделите входную запись на две части: первая — фамилия и имя, вторая — результат. Для этого найдите место второго (последнего) пробела и всё, что стоит после него, скопируйте в дополнительную строку, после чего переведите эту строку в число. Всё, что стоит до этого пробела, также занесите в строку (можно в эту же дополнительную строку, так как предыдущий результат вы уже перевели в число).

Блок 2. Заведите две переменные: числовую — для максимального результата и строковую — для фамилии и имени лучшего ученика. Предварительно обнулите эти переменные. Организуйте цикл `for` просмотра результатов всех учеников (количество учеников — N). В цикле получите данные об ученике и отделите результат от фамилии с именем (блок 1), далее сравните полученный результат с текущим максимальным. Если полученный результат лучше текущего максимума, то сделайте его текущим максимумом (числовая переменная) и замените значение строковой переменной.

Блок 3. По окончании цикла выведите текущий максимальный результат и имя и фамилию лучшего ученика.

1) Ответьте на вопросы.

- Сколько переменных строкового типа должно быть в вашей программе и каково назначение каждой?

Количество строковых переменных:



Обозначение и назначение строковых переменных:

- Какие основные числовые переменные понадобятся в вашей программе?

Обозначение и назначение числовых переменных:

- 2) Напишите коды описанных выше блоков программы.

Блок 1 — ввод данных тестирования и разделение строки на результат и фамилию с именем:



Блок 2 — задание начального значения переменных и организация цикла поиска максимума:

```

-----
-----
-----
-----
-----
-----

```

Блок 3 — вывод текущего максимального результата и имени с фамилией лучшего ученика:

```

-----
-----

```

- 3) Составьте из блоков текст итоговой программы, включая описание переменных:

```

-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----

```



-
-
-
-
-
-
-
-
-
-
-
-
- 4) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.
- 5) Сделайте и сохраните скриншот решения задачи.

Задача 2. На автозаправочных станциях (АЗС) продаётся бензин с маркировкой 92, 95 и 98. В городе М был проведён мониторинг цены бензина на различных АЗС. Напишите эффективную по времени работы и по используемой памяти программу, которая будет определять для каждого вида бензина, сколько АЗС продают его дешевле всего. На вход программе в первой строке подаётся N — количество данных о стоимости бензина. В каждой из последующих N строк находится информация в формате:

<Компания> <Улица> <Марка> <Цена>

где:

<Компания> — строка, состоящая не более чем из 20 символов без пробелов;

<Улица> — строка, состоящая не более чем из 20 символов без пробелов;

<Марка> — одно из чисел: 92, 95 или 98;

<Цена> — целое число в диапазоне от 1000 до 5000, обозначающее стоимость одного литра бензина в копейках.

<Компания> и <Улица>, <Улица> и <Марка>, а также <Марка> и <Цена> разделены ровно одним пробелом.

Пример входной строки:

Синойл Цветочная 95 2250

Программа должна выводить через пробел 3 числа — количество АЗС, продающих дешевле всего 92-й, 95-й и 98-й бензин соответственно.

Если бензин какой-то марки нигде не продавался, то следует вывести 0.

Пример выходных данных:

12 1 0

(Из материалов ЕГЭ.)

Решите задачу поэтапно, отвечая на вопросы и записывая основные блоки программы.

Фраза «Напишите эффективную по времени работы и по используемой памяти программу» может означать, что в программе не нужно запоминать всю информацию в массиве, а эффективней обрабатывать информацию по мере её поступления, запоминая только необходимую на данный момент информацию, каждый раз обновляя её.

- 1) По условию задачи, входная строка имеет вид:

<Компания> <Улица> <Марка> <Цена>

Какая часть вводимой информации является ненужной для решения задачи?

- 2) Как убрать ненужную информацию?





Напишите часть программы, которая убирает из строки ненужную информацию:

Что осталось в строке?



- 3) На какие две части нужно разбить оставшуюся информацию и какие провести преобразования?

Напишите фрагмент программы, который разбивает оставшуюся информацию на нужные блоки и выполняет необходимые преобразования:

- 4) Какие переменные, помимо отражающих текущую марку бензина и её цену, вам потребуются для обработки в цикле всей информации и вывода количества АЗС, продающих бензин каждой марки по минимальной цене?



- 5) Какие проверки для каждой марки бензина нужно выполнять в цикле, чтобы каждый раз обновлялась информация о минимальной цене и количестве АЗС, продающих бензин по этой минимальной цене?



Запишите, как должны меняться значения переменных для марки бензина, если:

- а) цена на очередной АЗС больше текущей минимальной:

- б) цена на очередной АЗС равна текущей минимальной:

- в) цена на очередной АЗС меньше текущей минимальной:

Напишите фрагмент программы, который реализует описанные выше проверки:



- 5) Как организовать предыдущие проверки для всех марок бензина?

Напишите фрагмент программы, который реализует описанные выше проверки для всех марок бензина:

- 6) Какой вид цикла потребуется для решения задачи и почему?



- 7) Какие последовательные блоки должны выполняться в цикле?



Ввод строки данных

- 8) Начальные значения каких переменных нужно задать до начала цикла?





9) Что ещё нужно ввести до начала цикла?



10) Напишите программу целиком:

- 11) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.
- 12) Сделайте и сохраните скриншот задачи.

Отчёт по лабораторной работе № 11

Распечатайте и вклейте скриншот решения задачи:



**Выводы**

Обработка символов и строк в программировании является одной из основных задач в программировании потому, что:

Оценка учителя: -----

Лабораторная работа № 12

УЧИМСЯ СТРОИТЬ ГРАФИЧЕСКИЕ ИЗОБРАЖЕНИЯ НА ЯЗЫКЕ ПАСКАЛЬ

(К параграфу учебника:

1.6. Графические возможности объектно-ориентированного языка программирования Visual Basic)

Тема: изучение построения графических изображений на языке Паскаль.

Цель работы: научиться рисовать простые фигуры на языке Паскаль.

Программное обеспечение: компьютер с установленной операционной системой Windows или Linux; текстовый процессор Microsoft Word; среда программирования на языке Паскаль.

- 1. Решение задач на Паскале.** Самостоятельно составьте и выполните программы решения следующих задач на Паскале.

Задача 1. Определите, что будет выведено на экран в результате выполнения следующих программ.

Программа 1:

```
program ol;  
uses crt, graphabc;  
var t: integer;  
begin  
  for t:=1 to 10 do  
    circle(320, 200, t*10);  
end.
```

Программа 2:

```
program o2;  
uses crt, graphabc;  
var t: integer;  
begin  
  for t:=10 downto 1 do  
    circle(320, 200, t*10);  
end.
```

- 1) Программа 1 должна выводить:





- 2) Наберите программу и проверьте её работу на компьютере.

Программа 1 выводит:





- 3) Сделайте и сохраните скриншот результата работы программы 1.

- 4) Программа 2 отличается от программы 1 тем, что:



- 5) Программа 2 должна выводить:





- 6) Проверьте работу программы 2 на компьютере.

Программа 2 выводит:





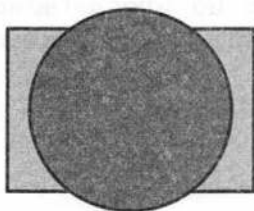
- 7) Сделайте и сохраните скриншот результата работы программы 2.



2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.

3) Сделайте и сохраните скриншот решения задачи (текст программы должен быть виден).

Изображение 2 (красный круг поверх зелёного прямоугольника):



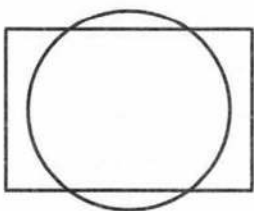
1) Текст программы:



2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.

3) Сделайте и сохраните скриншот решения задачи (текст программы должен быть виден).

Изображение 3:



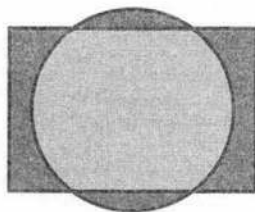
- 1) Текст программы:



- 2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.
- 3) Сделайте и сохраните скриншот решения задачи (текст программы должен быть виден).



Изображение 4 (пересечение круга и прямоугольника — зелёного цвета, область фигуры вне пересечения — красного цвета):



Для построения данного рисунка постарайтесь применить минимальное количество команд (5 — отлично, 6–7 — хорошо, 8 и более — удовлетворительно).

- 1) Текст программы:





- 2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.
- 3) Сделайте и сохраните скриншот решения задачи (текст программы должен быть виден).

Для дополнительного выполнения

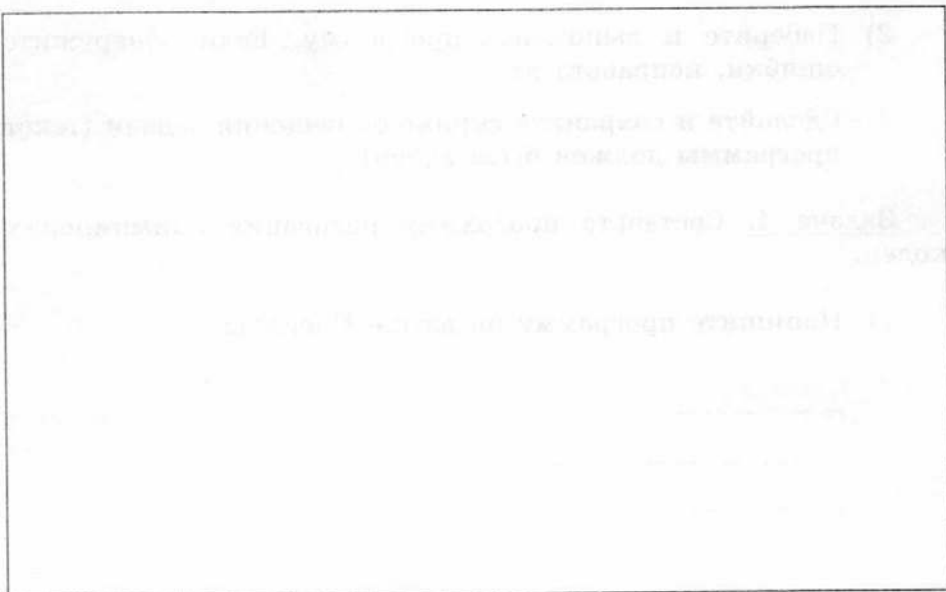
Задача 3. Составьте программу рисования домика и фигуры человечка около него.



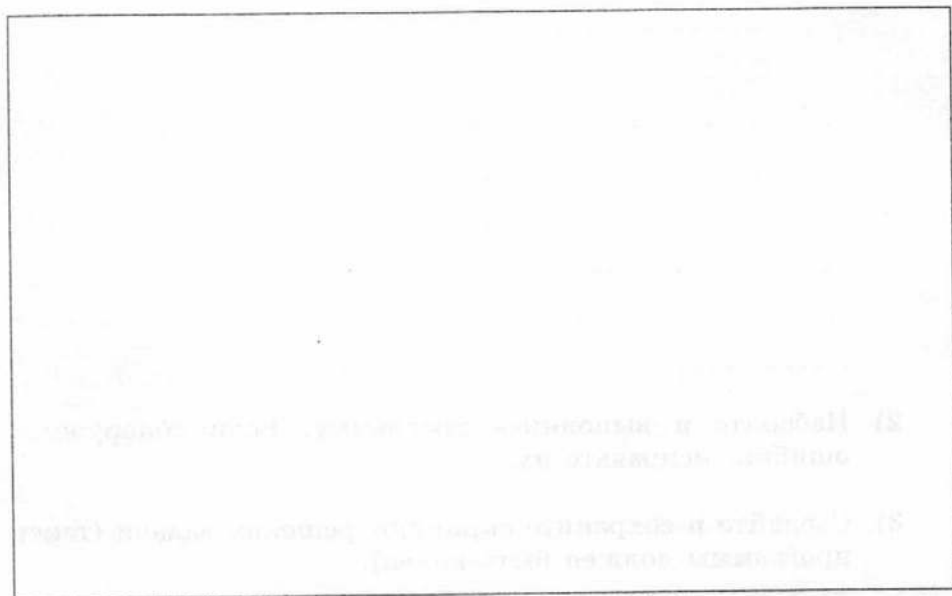
- 1) Напишите программу на языке Паскаль:

Отчёт по лабораторной работе № 12

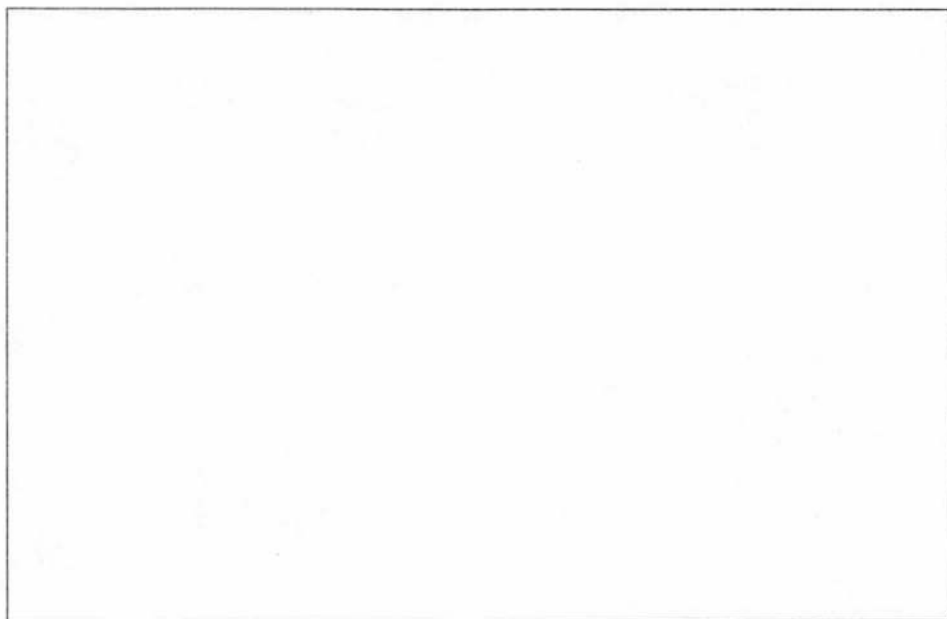
1. Распечатайте и вклейте скриншот результата работы программы 1 (задача 1):



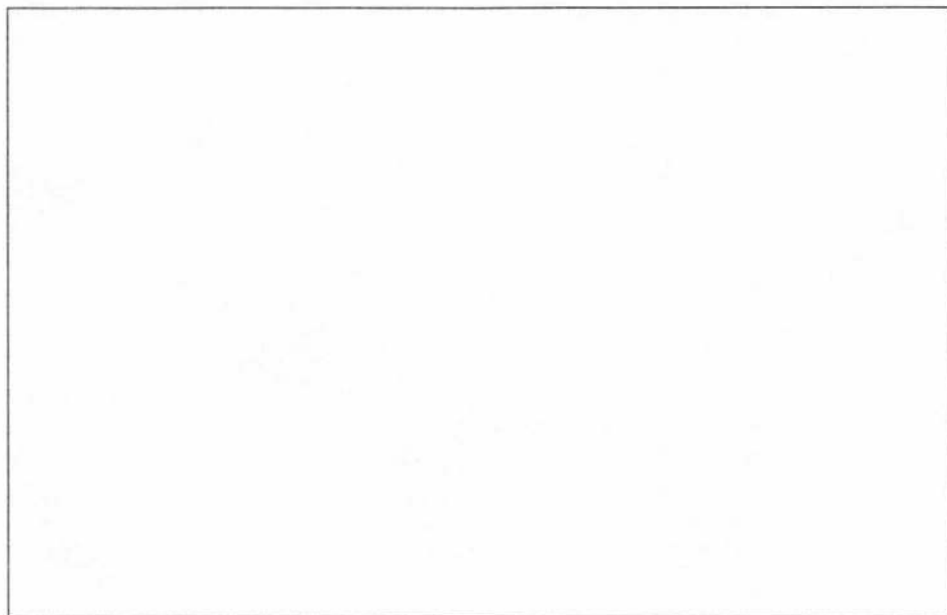
2. Распечатайте и вклейте скриншот результата работы программы 2 (задача 1):



3. Распечатайте и вклейте скриншот решения задачи 2 (изображение 1):

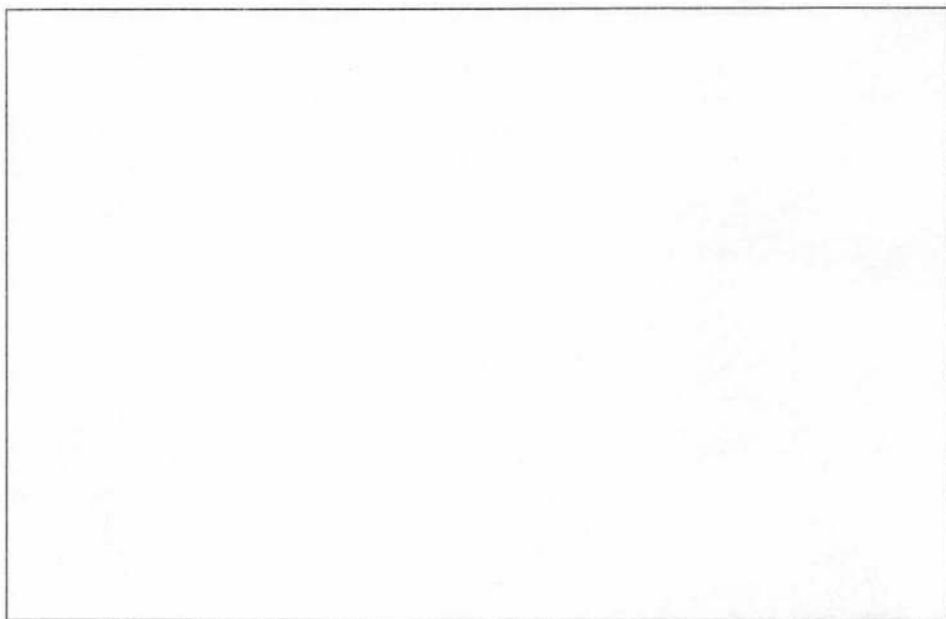


4. Распечатайте и вклейте скриншот решения задачи 2 (изображение 2):

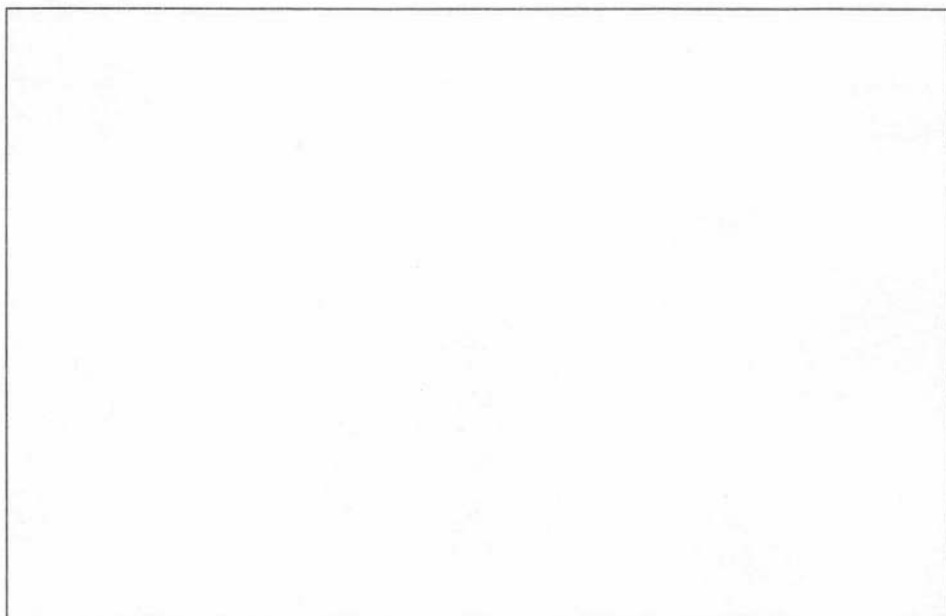




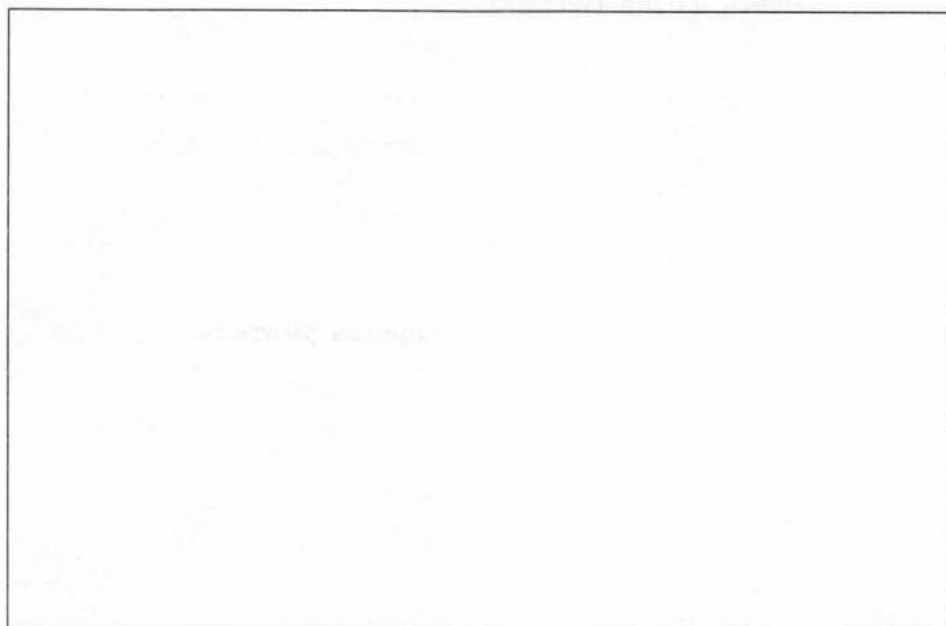
5. Распечатайте и вклейте скриншот решения задачи 2 (изображение 3):



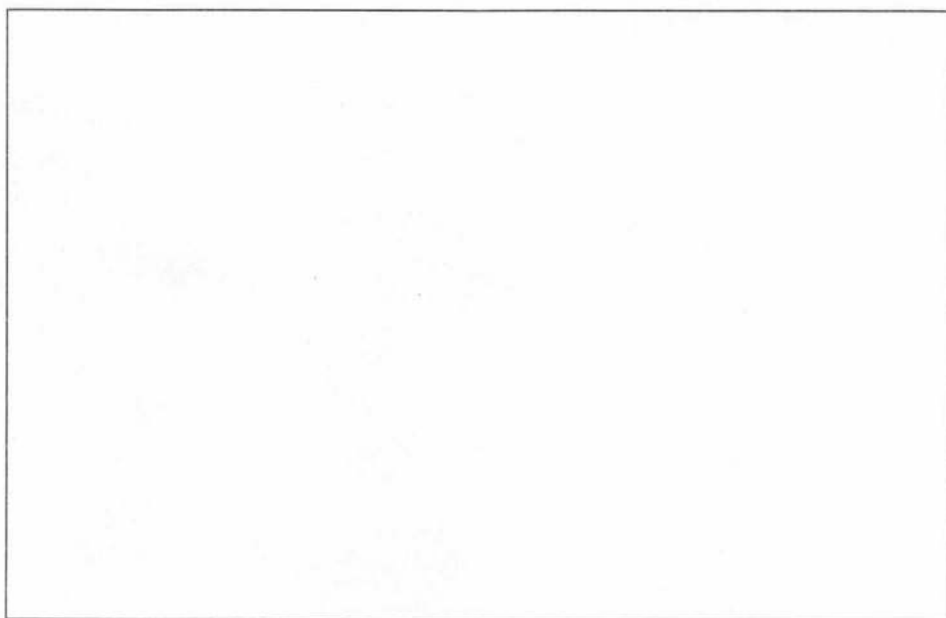
6. Распечатайте и вклейте скриншот решения задачи 2 (изображение 4):



7. Распечатайте и вклейте скриншот решения задачи 3 (дополнительно):



8. Распечатайте и вклейте скриншот решения задачи 4 (дополнительно):



**Выводы**

Какие изображения можно создавать с помощью графических средств языка Паскаль?

Оценка учителя: -----

Лабораторная работа № 13

УЧИМСЯ РАБОТАТЬ С СИСТЕМОЙ КООРДИНАТ НА ЯЗЫКЕ ПАСКАЛЬ

(К параграфу учебника:

1.6. Графические возможности объектно-ориентированного языка программирования Visual Basic)

Тема: изучение работы с системой координат на языке Паскаль.

Цель работы: научиться рисовать компьютерную систему координат на языке Паскаль.

Программное обеспечение: компьютер с установленной операционной системой Windows или Linux; текстовый процессор Microsoft Word; среда программирования на языке Паскаль.

Решение задачи на Паскале. Самостоятельно составьте и выполните программу решения следующей задачи на Паскале.

Задача. Напишите программу построения компьютерной системы координат. Должны быть выведены оси координат, сделаны засечки и проставлены цифры напротив засечек.

Алгоритм построения:

- Подключите в разделе `uses` графический блок — `graphabc`.
- Заведите переменную (`p`) для хранения значения шага, с которым должны идти засечки (например `p = 50`), чтобы при изменении значения шага не нужно было менять его во многих местах программы.
- Выведите линии осей `X` и `Y`. Для того чтобы осталось место для размещения цифр, отступите от верхней стороны координатной плоскости на 15 пикселей, а от левой стороны — на 20 пикселей.

Используйте для этого команду

```
line(x1, y1, x2, y2)
```

Так, например, для оси X можно использовать команду

```
line(20, 15, 620, 15)
```

Аналогично запишите команду вывода линии для оси Y .

- Ввиду того что оси X и Y имеют разную длину, на них будет разное количество засечек и, соответственно, чисел. Для установки засечек и чисел сделайте два похожих цикла.

На оси X поставьте 12 засечек (при шаге 50), поэтому в цикле **for** со счётчиком, меняющимся от 1 до 12, поставьте вертикальные засечки той же командой

```
line(x1, y1, x2, y2)
```

Длина засечек — например 10. Шаг засечек — p . Изменение пропорционально шагу цикла.

Команда будет выглядеть, например, так:

```
line(20+p*i, 15, 20+p*i, 25 );
```

Для вывода цифровых значений напротив засечек используйте команду вывода текста:

```
TextOut(x, y, 'текст').
```

Для этого получите числа засечек, умножив шаг p на изменение параметра цикла, и переведите числа в строковую переменную (*string*). Сделать это можно, применив команду

```
str(i*p, st)
```

(i — шаг цикла, p — шаг засечек, st — строка для вывода).

Команда вывода цифр будет выглядеть так:

```
TextOut(10+p*i, 0, st).
```

Напишите аналогичную группу команд для засечек и чисел на оси Y .



1) Дополните шаблон — составьте программу.

```
program komp_sistema_koordinat;
uses _____;
var i, p: integer;
    st: string;
begin
    p:= _____ ; {шаг засечек}
    {выводим линии осей X и Y}
    line(20, 15, 620, 15); {вывод оси X}
    line(20, 15, 20, 380); {вывод оси Y}

    for i:= ___ to ____ do
        begin
            _____ {засечки на оси X}
            str _____ {получение чисел шкалы по оси X}
            textout _____ {вывод шкалы X}
        end;

    for i:=1 to 6 do
        begin
            _____ {засечки на оси Y}
            _____ {получение чисел шкалы по оси Y}
            _____ {вывод шкалы Y}
        end;
    end.
```

2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.



3) Сделайте и сохраните скриншот решения задачи (текст программы должен быть виден).



Отчёт по лабораторной работе № 13

Распечатайте и вклейте скриншот решения задачи:

**Выводы**

Какие принципы построения компьютерной системы координат вы можете выделить?

Оценка учителя: -----

Лабораторная работа № 14

УЧИМСЯ СОЗДАВАТЬ ДВИЖУЩИЕСЯ ОБЪЕКТЫ НА ЯЗЫКЕ ПАСКАЛЬ

(К параграфу учебника:

1.6. Графические возможности объектно-ориентированного языка программирования Visual Basic)

Тема: изучение создания движущихся объектов на языке Паскаль.

Цель работы: научиться создавать анимацию на языке Паскаль.

Программное обеспечение: операционная система Windows или Linux; текстовый процессор Microsoft Word; среда программирования на языке Паскаль.

Решение задачи на Паскале. Самостоятельно составьте и выполните программу решения следующей задачи на Паскале.

Задача 1. Физкультурники. На поле расположите семь физкультурников. Все они произвольным образом должны выполнять упражнения: смену четырёх разных положений. Последовательность движений — случайная для каждого физкультурника. Упражнения выполняются до нажатия любой клавиши. Четыре положения физкультурников заданы в файлах *a1.bmp*, *a2.bmp*, *a3.bmp*, *a4.bmp* (место размещения файлов спросите у учителя).

WWW

Алгоритм построения:

- Подключите в разделе **uses** графический блок **graphabc** и текстовый блок **crt** (так как команда управления клавиатурой находится в этом блоке).
- Названия файлов запоминайте в массиве (**m**) из четырёх элементов (так как у вас четыре картинки), но ввиду того, что в программе имя будет переведено в число, массив будет целочисленный.

- Вначале в цикле **for** запомните имена всех четырёх файлов в массиве. Имена файлов картинок — *a1.bmp*, *a2.bmp*, *a3.bmp*, *a4.bmp*.

Используйте команду

```
m[i]:=LoadPicture('a'+intToStr(i)+'.bmp')
```

Эта функция загружает рисунок из файла с указанным именем и возвращает дескриптор (описатель) рисунка — целое число.

Здесь:

'a' — буква в имени файла,

i — счётчик цикла **for**,

intToStr(i) — команда перевода целого числа в строковую переменную,

'.bmp' — символы расширения файлов картинок.

- Задайте начальное положение физкультурников по осям X и Y.
- Для правильного расположения картинок определите их размеры — ширину и высоту (на самом деле, высота здесь, в общем-то, не нужна) командами:

```
h:=PictureHeight(m[1]);
```

```
w:=PictureWidth(m[1])
```

(берём за основу первую картинку, так как все они имеют одинаковые размеры).

- Теперь, когда все начальные значения заданы, создайте большой цикл, который закончится, только если пользователь нажмёт какую-нибудь клавишу. Здесь удобнее всего использовать цикл **repeat...until**, где в конце цикла проверять, была ли нажата какая-нибудь клавиша; если была, то внешний цикл заканчивается.
- Внутри большого цикла создайте цикл для всех семи физкультурников. Здесь удобнее всего использовать цикл **for** с параметром, изменяющимся от 1 до 7.
- В цикле **for** выводите картинки командой:

```
DrawPicture(m[Random(4)+1], x+i*w, y)
```

Здесь картинки выводятся случайным образом, а к **Random(4)** прибавляется единица, так как **Random** генерирует числа, начиная с 0, а в именах картинок число меняется, начиная с единицы.

$x+i*w$ — к начальному положению по оси X прибавляется ширина картинки (w), умноженная на её номер.
 y — положение картинок по оси Y .

- После выполнения движения всеми семью физкультурниками сделайте задержку командой `sleep(500)` (500 — параметр замедления. Чем он больше, тем медленнее будет движение).
- И наконец, завершите внешний цикл **repeat** командой **until keypressed**. Эта команда определяет, была ли нажата какая-нибудь клавиша.

1) Дополните шаблон — составьте программу.



```
program fizkulturniki;
uses _____;
var
    m: _____;
    x, y, h, w, i: integer;
begin
    for i:=1 to 4 do
        m[i] := _____;
    x:=10;
    y:=10;
    h:= _____;
    w:= _____;
    repeat
        for i:=1 to 7 do
            _____;
            _____;
        until _____;
    end.
```

- 2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.
- 3) Сделайте и сохраните скриншот решения задачи (текст программы должен быть виден).





4) Поясните действие следующих команд и функций.

`h:=PictureHeight(...)` —

`w:=PictureWidth(...)` —

`intToStr(x)` —

`sleep(t)` —

`a:=LoadPicture('...')` —

`DrawPicture(имя, x, y)` —

`KeyPressed` —

Задача 2. Полёт кольца. Из центра поля размером 500×500 пикселей должно вылетать кольцо и лететь в левый верхний угол. После вылета за пределы экрана кольцо должно возвращаться обратно, после чего снова лететь в угол экрана — и так до нажатия любой клавиши. При каждой смене направления движения кольцо должно менять свой цвет. Размер и толщину кольца подберите самостоятельно.

Алгоритм построения:

- Подключите в разделе **uses** графический блок **graphabc** и текстовый блок **crt** (так как команда управления клавиатурой находится в этом блоке).
- Задайте размеры экрана по осям *X* и *Y* командой `setwindowsize(500, 500)`.
- Для исключения эффекта «моргания» используйте команду `lockdrawing`.
Для того чтобы на экране не мигал курсор, используйте команду `hidecursor`.

- Задайте начальное положение кольца (в середине экрана, $x = 250$, $y = 250$), его радиус (например, $r = 20$) и толщину командой `setpenwidth(8)`.
- Так как кольцу нужно будет двигаться то в одном, то в другом направлении, задайте начальное направление ($ii = 1$).
- Теперь, когда все начальные значения заданы, создайте большой цикл, который закончится, только если пользователь нажмёт какую-нибудь клавишу. Здесь удобнее всего использовать цикл `repeat...until`, где в конце проверять, была ли нажата какая-нибудь клавиша; если была, то внешний цикл заканчивается.
- Внутри внешнего цикла вначале задайте параметры цвета кольца. Так как принята система раскраски RGB, случайным образом задайте каждый из этих цветовых параметров командами:

```
rr1:=random(255)
gg1:=random(255)
bb1:=random(255)
```

- Теперь можно запустить цикл движения в одном направлении (из центра в угол или обратно). Здесь удобней всего использовать цикл `for`. В зависимости от того, хотите ли вы, чтобы кольцо «ударялось» о границу экрана или «вылетало» за экран, цикл должен меняться от 1 до $(250 - \text{диаметр кольца})$ или до $(250 + \text{диаметр кольца})$.
- В цикле `for`:
 - вначале каждый раз очищайте экран командой `clearwindow`;
 - задавайте раскраску кольца командой `setpencolor(rgb(rr1, gg1, bb1))`;
 - рассчитывайте положение кольца по осям X и Y . Для этого текущее положение по X и Y уменьшайте на 1 (точнее, не на 1, а на величину приращения, определяемого направлением движения ii);
 - рисуйте кольцо командой `circle(x, y, r)`;
 - замедляйте движение командой `sleep(5)` (5 — параметр замедления. Чем он больше, тем медленнее будет движение);
 - для того чтобы не было видно моргания в результате использования команды `clearwindow`, нужно использовать команду `redraw`;

- после окончания цикла **for** запишите команду изменения направления на противоположное, умножив старое направление на (-1);
- и, наконец, завершите внешний цикл **repeat** командой **until keypressed**.



1) Дополните шаблон — составьте программу.

```

program polyot_koltca;
uses _____; var i, rrl, ggl, bbl, ii, xx1,
yy1, r: integer;
begin
    _____;
    _____;
    _____;
    xx1:= _____ ; {начальные координаты}
    yy1:= _____ ;
    r:= _____ ;
    _____;
    ii:= _____ ; {начальное направление движения}
    repeat
        rrl:= _____;
        ggl:= _____;
        bbl:= _____;
        for i:= _____ to _____ do
            begin
                _____;
                _____;
                xx1:= _____;
                yy1:= _____;
                circle _____;
                sleep(5);
                _____;
            end
        until keypressed;
    end
end

```

```

        end;
        ii:=-ii;
    until _____;
end.

```

- 2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.
- 3) Сделайте и сохраните скриншот решения задачи (текст программы должен быть виден).
- 4) Поясните действие следующих команд и функций.



setwindowsize(x, y) — _____

w:=PictureWidth(...) — _____

lockdrawing — _____

hidecursor — _____

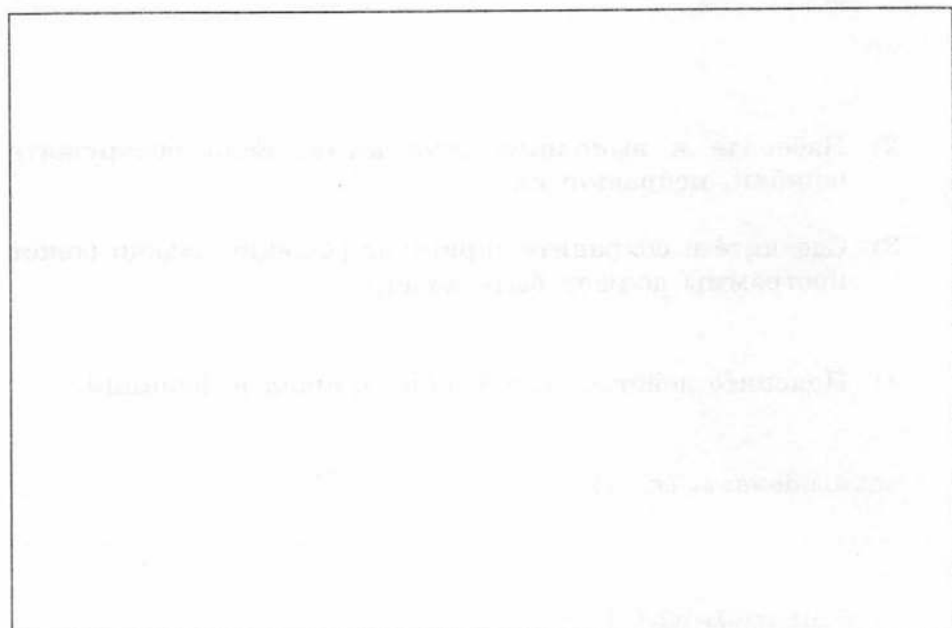
a:=LoadPicture('...') — _____

rgb(r, g, b) — _____

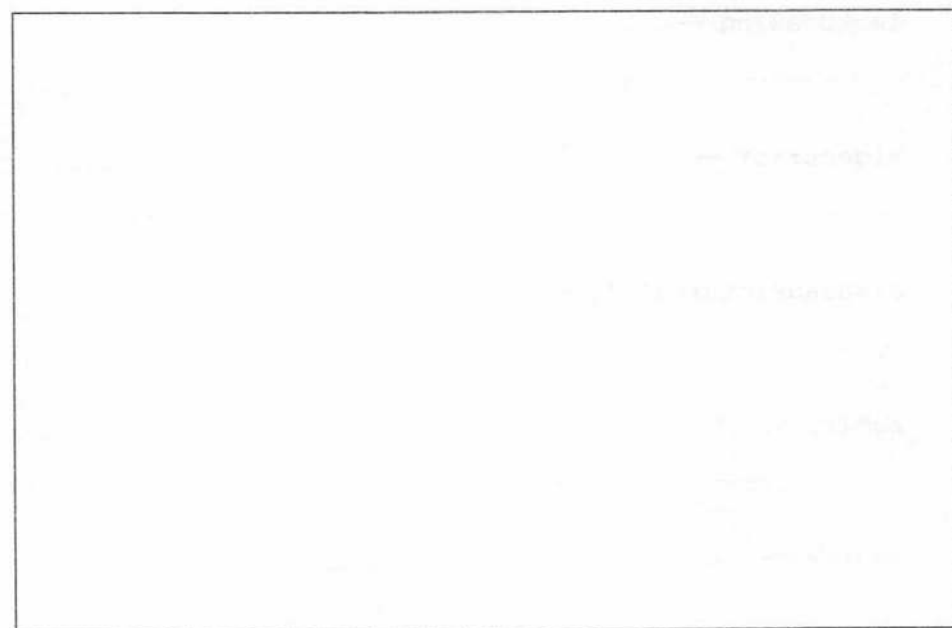
redraw — _____

Отчёт по лабораторной работе № 14

1. Распечатайте и вклейте скриншот решения задачи 1:



2. Распечатайте и вклейте скриншот решения задачи 2:



Выводы

Какие возможности анимации в языке Паскаль вы изучили?



Оценка учителя: -----

Лабораторная работа № 15

УЧИМСЯ ВЫПОЛНЯТЬ ПРЕОБРАЗОВАНИЯ ЛОГИЧЕСКИХ ФОРМУЛ

(К параграфу учебника: 3.1. Алгебра логики)

Тема: знакомство с выполнением преобразований логических формул.

Цель работы: научиться выполнять преобразования логических формул.

Программное обеспечение: не требуется.

Справочная информация

Законы алгебры логики

№	Законы для операций		Название закона
	Логическое умножение	Логическое сложение	
1	$A \cdot B = B \cdot A$	$A + B = B + A$	Коммутативный закон
2	$A \cdot B \cdot C = (A \cdot B) \cdot C = A \cdot (B \cdot C) = (A \cdot C) \cdot B$	$A + B + C = (A + B) + C = A + (B + C) = (A + C) + B$	Ассоциативный закон
3	$A \cdot (B + C) = A \cdot B + A \cdot C$	$A + (B \cdot C) = (A + B) \cdot (A + C)$	Дистрибутивный закон
4	$A \cdot A = A$	$A + A = A$	Закон идемпотентности
5	$A \cdot \bar{A} = 0$	$A + \bar{A} = 1$	Законы непротиворечия и исключённого третьего
6	$A \cdot 0 = 0$	$A + 0 = A$	Правила операций с константами
7	$A \cdot 1 = A$	$A + 1 = 1$	

№	Законы для операций		Название закона
	Логическое умножение	Логическое сложение	
8	$\overline{\overline{A}} = A$	$\overline{\overline{\overline{A}}} = \overline{A}$	Законы двойного и тройного отрицания
9	$\overline{1} = 0$	$\overline{0} = 1$	Свойства констант
10	$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$	Законы де Моргана
11	$A + A \cdot B = A$	$A \cdot (A + B) = A$	Закон поглощения
12	$(A + B) \cdot (\overline{A} + \overline{B}) = B$	$A \cdot B + \overline{A} \cdot \overline{B} = B$	Закон склеивания
13	$A \cdot (\overline{A} + B) = A \cdot B$	$A + \overline{A} \cdot B = A + B$	Закон свёртки
Преобразование небазовых операций в базовые			
14	$A \oplus B = \overline{A} \cdot B + A \cdot \overline{B}$		Операция исключающее ИЛИ
15	$A \leftrightarrow B = (A + \overline{B}) \cdot (\overline{A} + B)$		Операция эквиваленция
16	$A \rightarrow B = \overline{A} + B$		Операция импликация

1. Примеры решения задач на преобразование логических выражений. Ознакомьтесь с примерами решения задач.



Пример 1. С помощью таблицы истинности докажите правильность логического тождества: $\overline{A} + A \cdot B = \overline{A} + B$.

Решение

Расставим приоритеты логических операций в левой и правой частях:

$$\overset{1}{\overline{A}} + \overset{3}{A} \cdot \overset{2}{B} = \overset{4}{\overline{A} + B}$$

Построим таблицу истинности для всех операций:

Исходные переменные		Операции в левой части			Операция в правой части
A	B	$\overline{A}$	$A \cdot B$	$\overline{A} + A \cdot B$	$\overline{A} + B$
0	0	1	$0 \cdot 0 = 0$	$1 + 0 = 1$	$1 + 0 = 1$
0	1	1	$0 \cdot 1 = 0$	$1 + 0 = 1$	$1 + 1 = 1$
1	0	0	$1 \cdot 0 = 0$	$0 + 0 = 0$	$0 + 0 = 0$
1	1	0	$1 \cdot 1 = 1$	$0 + 1 = 1$	$0 + 1 = 1$

Как мы видим, значения истинности в левой и правой частях уравнения при всех сочетаниях значений переменных A и B совпадают, следовательно, тождество доказано.

Пример 2. Упростите логическое выражение:

$$F = (X_1 + \overline{X}_3) \cdot (X_1 + X_3) \cdot (X_2 + \overline{X}_3).$$

Решение

В фигурных скобках здесь и далее указаны номера логических законов (см. выше справочную информацию).

$$\begin{aligned} F = \{3\} &= (X_1 + \overline{X}_3 \cdot X_3) \cdot (X_2 + \overline{X}_3) = \{5\} = (X_1 + 0) \cdot (X_2 + \overline{X}_3) = \\ &= \{6\} = X_1 \cdot (X_2 + \overline{X}_3). \end{aligned}$$

Пример 3. Для какого из указанных значений числа X истинно логическое выражение:

$$((X < 5) \rightarrow (X < 3)) \wedge ((X < 2) \rightarrow (X < 1))?$$

- 1) 1 2) 2 3) 3 4) 4

В ответе запишите число — номер верного варианта.

Решение

Исходная логическая формула состоит из двух сложных логических выражений, объединённых логическим умножением, которое истинно, если истинны оба выражения.

Преобразовав импликацию $\{16\}$ слева, получим:

$$(X \geq 5 \vee X < 3).$$

Выражение истинно при следующих вариантах ответа:
 $X = 1$ и $X = 2$.

Преобразовав импликацию $\{16\}$ справа, получим:
 $(X \geq 2 \vee X < 1)$. Выражение истинно при следующих вариан-
 тах ответа: $X = 2$, $X = 3$ и $X = 4$.

Оба этих выражения одновременно истинны только при
 $X = 2$. Значит, исходное высказывание истинно при $X = 2$.

Ответ: 1.

Пример 4. Дан фрагмент таблицы истинности логической
 функции F :

X	Y	Z	F
1	1	1	1
1	1	0	1
1	0	1	1

Какое выражение соответствует F ? В ответе запишите
 число — номер верного варианта.

- 1) $X \vee \neg Y \vee Z$
- 2) $X \wedge Y \wedge Z$
- 3) $X \wedge Y \wedge \neg Z$
- 4) $\neg X \wedge Y \wedge \neg Z$

Решение

Построим фрагмент таблицы истинности для всех при-
 ведённых выражений:

X	Y	Z	$X \vee \neg Y \vee Z$	$X \wedge Y \wedge Z$	$X \wedge Y \wedge \neg Z$	$\neg X \wedge Y \wedge \neg Z$	F
1	1	1	1	1	0	0	1
1	1	0	1	0	1	0	1
1	0	1	1	0	0	0	1

Из таблицы видим, что значения истинности совпадают
 со значениями F только у первого выражения, следовательно,
 верный ответ: 1) $X \vee \neg Y \vee Z$.

Ответ: 1.



2. Докажите тождество с помощью таблицы истинности.

$$\overline{A \cdot B + A \cdot \overline{B}} = \overline{A} \cdot \overline{B} + A \cdot B$$

A	B	$\overline{A}$	$\overline{B}$						
0	0	1	1						



3. Упростите логические выражения с помощью законов алгебры логики.

а) $(A \vee B) \wedge (A \vee C) \wedge (B \vee C)$

Ответ: -----

б) $\overline{A \cdot (B + \overline{C}) \cdot (A + \overline{B} \cdot \overline{C}) + A \cdot \overline{B}}$

Ответ: -----

в) $(A \rightarrow B \wedge C) \leftrightarrow A \vee C$

Ответ: -----

В заданиях 4–12 в качестве ответа запишите число — номер верного варианта.

4. Для какого из указанных значений числа X истинно выражение $(X > 2) \& \neg (X > 3)$?

1) 1 2) 2 3) 3 4) 4

(Из материалов ОГЭ.)

Ответ: -----

5. Для какого из указанных значений числа X истинно выражение $(X > 2) \& ((X < 4) \vee (X > 4))$?

1) 1 2) 2 3) 3 4) 4

(Из материалов ОГЭ.)

Ответ: -----

6. Для какого из указанных значений числа X истинно выражение $(X < 4) \& (X > 1) \& (X \neq 2)$?

1) 1 2) 2 3) 3 4) 4

(Из материалов ОГЭ.)

Ответ: -----

7. Для какого из указанных значений числа X истинно выражение $((X > 3) \vee (X < 3)) \rightarrow (X < 1)$?

1) 1 2) 2 3) 3 4) 4

(Из материалов ОГЭ.)

Ответ: -----

8. Какое логическое выражение равносильно выражению $\neg (A \wedge B) \wedge \neg C$?

1) $\neg A \vee B \vee \neg C$
 2) $(\neg A \vee \neg B) \wedge \neg C$
 3) $(\neg A \vee \neg B) \wedge C$
 4) $\neg A \wedge \neg B \wedge \neg C$

(Из материалов ЕГЭ.)

Ответ: -----





9. Укажите, какое логическое выражение равносильно выражению $\neg(A \vee \neg B \vee C)$.

- 1) $\neg A \vee B \vee \neg C$
- 2) $A \wedge \neg B \wedge C$
- 3) $\neg A \vee \neg B \vee \neg C$
- 4) $\neg A \wedge B \wedge \neg C$

(Из материалов ЕГЭ.)

Ответ: _____



10. Символом F обозначено одно из указанных ниже логических выражений от двух аргументов: X , Y .
В таблице приведены значения функции F для всевозможных значений истинности аргументов X и Y :

X	Y	F
0	0	0
0	1	1
1	0	0
1	1	0

Каким выражением может быть F ?

- 1) $X \& \neg Y$
- 2) $X \vee \neg Y$
- 3) $\neg X \& Y$
- 4) $\neg X \vee Y$

(Из материалов ЕГЭ.)

Ответ: _____



11. Дан фрагмент таблицы истинности выражения F :

X	Y	Z	F
1	1	1	1
1	1	0	1
1	0	1	1

Каким выражением может быть F ?

- 1) $X \wedge Y \wedge Z$
- 2) $\neg X \vee \neg Y \vee Z$
- 3) $X \vee Y \vee Z$
- 4) $X \wedge Y \wedge \neg Z$

(Из материалов ЕГЭ.)

Ответ: _____

12. Для какого из приведённых чисел истинно высказывание:

НЕ (Первая цифра чётная) **И** (Последняя цифра нечётная)?

1) 1234 2) 6843 3) 3561 4) 4562

(Из материалов ЕГЭ.)

Ответ:

13. Ниже в табличной форме представлен фрагмент базы данных о результатах тестирования учащихся (максимальное количество баллов — 100):

Фамилия	Пол	Математика	Физика	Информатика	Иностранный язык
Абрикосова	Ж	57	93	43	62
Бананов	М	34	64	74	58
Вишнев	М	46	57	64	63
Грушева	Ж	37	74	67	86
Дынникова	Ж	94	78	36	48
Жёлудева	Ж	74	84	82	92

Сколько записей в данном фрагменте удовлетворяют условию:

Пол = 'Ж' И Физика < Иностранный язык?

В ответе укажите одно число — найденное количество записей.

(Из материалов ЕГЭ.)

Ответ:

14. В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке убывания количества страниц, которые найдёт поисковый сервер по каждому запросу.

№ запроса	Запрос
1	Азовское Каспийское Чёрное
2	(Азовское Чёрное) & Каспийское
3	Азовское Чёрное
4	Азовское & Чёрное & Каспийское

(Из материалов ЕГЭ.)

Ответ:



Отчёт по лабораторной работе № 15

1. На основе чего можно доказать равносильность логических выражений?

.....

.....

.....

.....

2. Какие способы доказательства равносильности логических выражений вы знаете?

.....

.....

.....

Оценка учителя:

Лабораторная работа № 16

УЧИМСЯ ПРОЕКТИРОВАТЬ ЦИФРОВЫЕ СХЕМЫ ИЗ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ

(К параграфу учебника:

3.3. Логические основы устройства компьютера)

Тема: знакомство с проектированием схем из логических элементов.

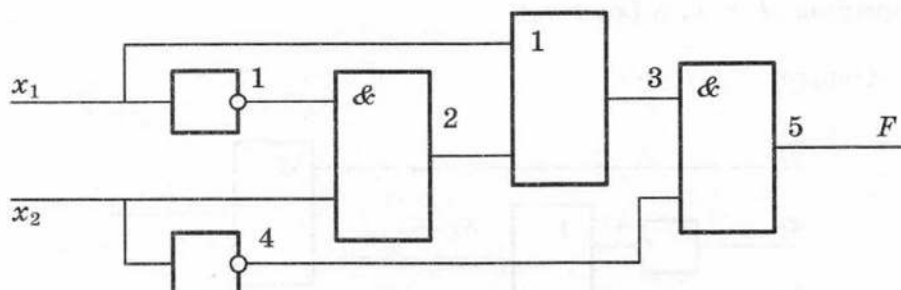
Цель работы: научиться проектировать схемы из логических элементов.

Программное обеспечение: не требуется.

1. Примеры решения задач на проектирование схем из логических элементов. Познакомьтесь с примерами решения задач.



Пример 1. Дана логическая схема. Требуется построить логическое выражение, описывающее эту схему.



Решение

Цифрами обозначены выходы логических элементов. Запишем формулы на выходах элементов:

1. $\bar{x}_1$
2. $\bar{x}_1 \wedge x_2$
3. $x_1 \vee \bar{x}_1 \wedge x_2$
4. $\bar{x}_2$
5. $\bar{x}_2 \wedge (x_1 \vee \bar{x}_1 \wedge x_2)$

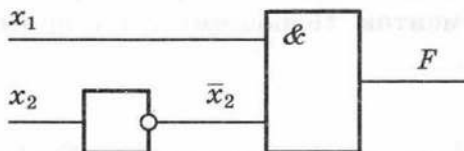
То есть $F = \bar{x}_2 \wedge (x_1 \vee \bar{x}_1 \wedge x_2)$.

В других обозначениях: $F = \bar{x}_2 \cdot (x_1 + \bar{x}_1 \cdot x_2)$.

Полученную функцию можно упростить, используя теоремы алгебры логики (в фигурных скобках указаны номера законов из справочной таблицы, приведённой в работе № 15):

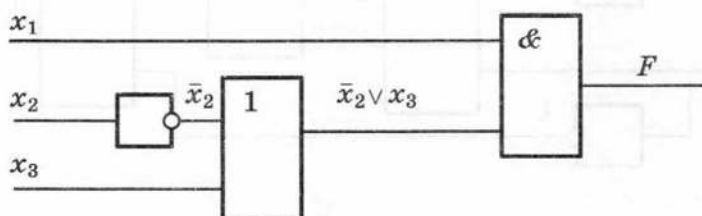
$$F = \bar{x}_2 \cdot (x_1 + \bar{x}_1 \cdot x_2) = \{13\} = \bar{x}_2 \cdot (x_1 + x_2) = \{3\} = x_1 \cdot \bar{x}_2 + x_2 \cdot \bar{x}_2 = \{5\} = x_1 \cdot \bar{x}_2 + 0 = x_1 \cdot \bar{x}_2.$$

Соответственно, и схема, реализующая эту функцию, может быть проще:



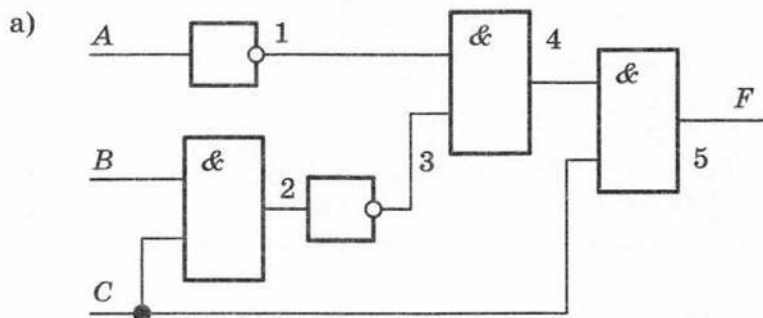
Пример 2. Требуется построить логическую схему для формулы: $F = x_1 \wedge (\bar{x}_2 \vee x_3)$.

Ответ





2. Запишите промежуточные формулы, которые получаются на выходах логических элементов (на схемах выходы обозначены числами). Ниже приведены схемы из логических элементов. Упростите формулы (если это возможно) и нарисуйте схемы для упрощённых формул. Выполните одно из заданий а), б) или в) по указанию учителя. Остальные задания можно выполнить дополнительно.



Формулы на выходах:

1.
2.
3.
4.



Итоговая формула:

Упрощение формулы:

.....

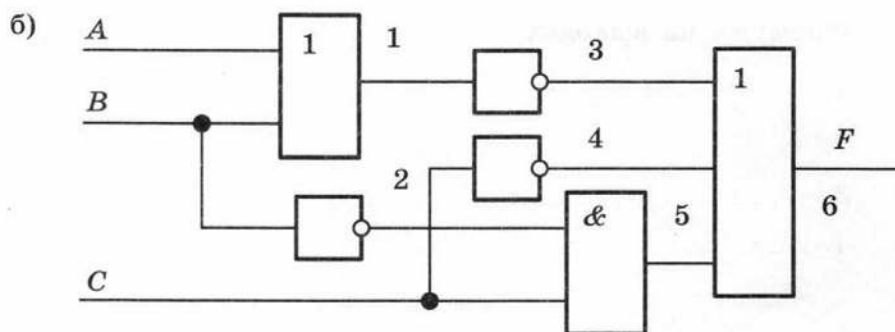
.....

.....

Упрощённая формула:



Схема, реализующая упрощённую формулу:



Формулы на выходах:

1.
2.
3.
4.
5.

Итоговая формула:

Упрощение формулы:

Упрощённая формула:

Схема, реализующая упрощённую формулу:



3. Составьте таблицу истинности для схемы. Как вы думаете, каково может быть применение этой схемы?

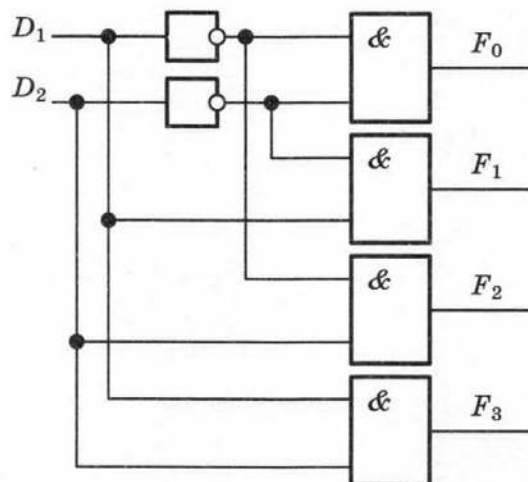


Таблица истинности:

Входы		Выходы			
D_1	D_2	F_0	F_1	F_2	F_3
0	0				
0	1				

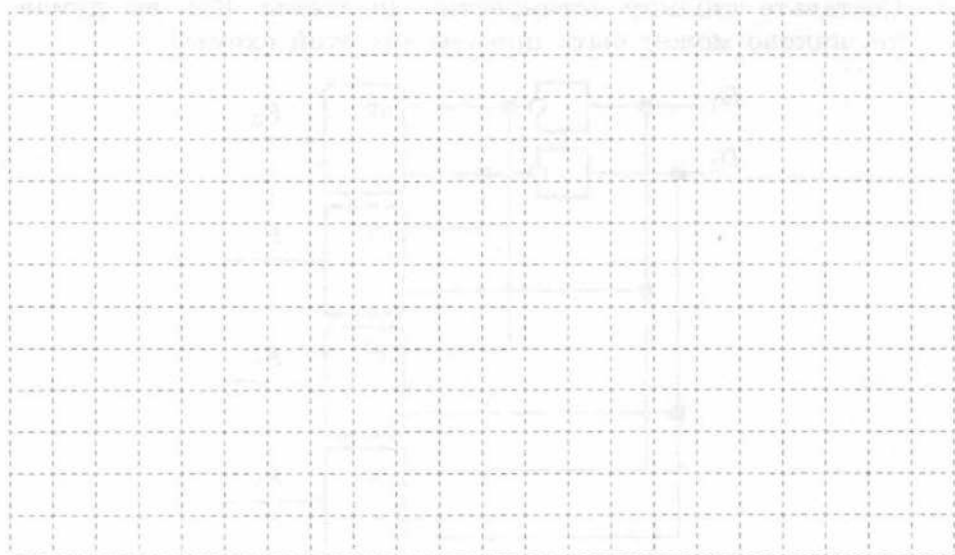
Возможное применение схемы:

4. Нарисуйте схемы по заданным формулам, упростите формулы (если это возможно) и нарисуйте схемы для упрощённых формул.

Выполните одно из заданий а), б) или в) по указанию учителя. Остальные задания можно выполнить дополнительно.

а) $F = (x_1 \vee x_3) \cdot (x_1 \vee \bar{x}_3) \cdot (\bar{x}_2 \vee x_3)$

Логическая схема:



Упрощение формулы:



Упрощённая формула:

Схема, реализующая упрощённую формулу:



б) $F = \bar{x}_1 \cdot \bar{x}_2 \vee x_1 \cdot \bar{x}_2 \vee x_1 \cdot x_2 \cdot x_3$

Логическая схема:



.....

.....

.....

Упрощённая формула: _____

Схема, реализующая упрощённую формулу:

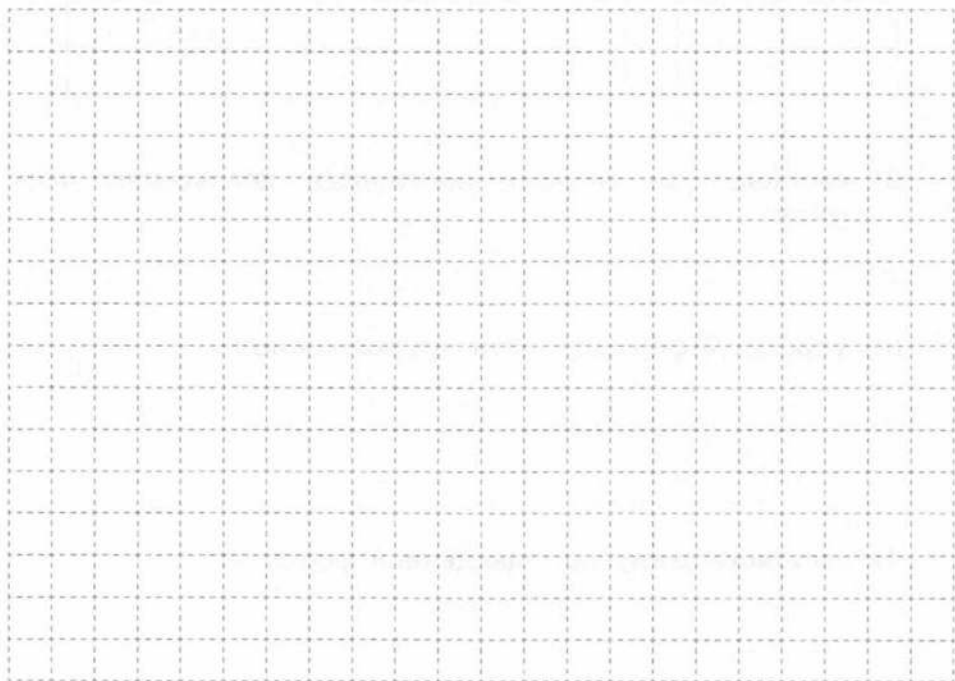
В) $F = x_1 \cdot x_2 \vee x_2 \cdot x_3 \vee \bar{x}_1 \cdot x_3$

Логическая схема:

Упрощение формулы:

Упрощённая формула: -----

Схема, реализующая упрощённую формулу:



Для дополнительного выполнения

5. Во время соревнований каждый из трёх судей нажимает кнопку на своем пульте, если он считает, что спортсмен выполнил упражнение правильно. В противном случае судья кнопку не нажимает. Упражнение считается выполненным правильно, если: 1) все трое судей нажали кнопки или 2) кнопки нажали двое судей, но при этом один из них — главный судья. Итоговый результат выдается на световое табло.

Составьте схему работы светового табло по результатам нажатия кнопок тремя судьями. Для этого:



1) составьте таблицу истинности:

Судьи			Результат включения табло
A (главный)	B	C	F
0	0	0	



2) составьте по таблице истинности логическую формулу:

$F =$

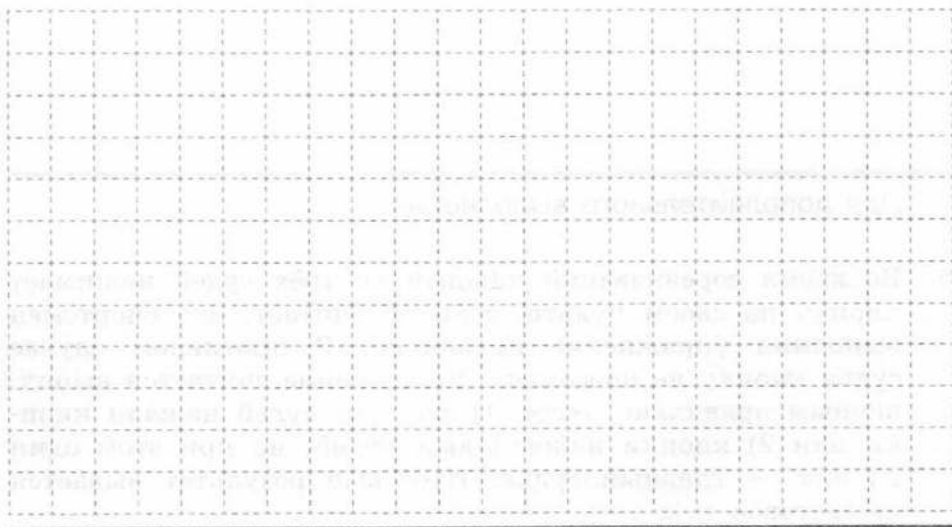


3) упростите формулу (если это возможно):

.....



4) составьте схему по упрощённой формуле:



Отчёт по лабораторной работе № 16



1. Алгебра логики лежит в основе вычислительной техники потому, что:

2. Подумайте и запишите, как с помощью цифровой техники можно реализовать решение логических уравнений.

3. Как можно оптимизировать логические схемы?

Оценка учителя: -----

ОГЛАВЛЕНИЕ

.....	
<i>Лабораторная работа № 8. Учимся программировать задачи с массивами</i> (К параграфу учебника: 1.4. Арифметические, строковые и логические выражения)	4
<i>Лабораторная работа № 9. Учимся программировать задачи с массивами: сортировка; двумерные массивы</i> (К параграфу учебника: 1.4. Арифметические, строковые и логические выражения)	13
<i>Лабораторная работа № 10. Учимся программировать задачи с символьным и строковым типами данных</i> (К параграфу учебника: 1.5. Функции в языках объектно-ориентированного и процедурного программирования)	23
<i>Лабораторная работа № 11. Учимся преобразовывать строковые величины в числовые и обратно</i> (К параграфу учебника: 1.5. Функции в языках объектно-ориентированного и процедурного программирования)	37
<i>Лабораторная работа № 12. Учимся строить графические изображения на языке Паскаль</i> (К параграфу учебника: 1.6. Графические возможности объектно-ориентированного языка программирования Visual Basic)	51
<i>Лабораторная работа № 13. Учимся работать с системой координат на языке Паскаль</i> (К параграфу учебника: 1.6. Графические возможности объектно-ориентированного языка программирования Visual Basic)	63
<i>Лабораторная работа № 14. Учимся создавать движущиеся объекты на языке Паскаль</i> (К параграфу учебника: 1.6. Графические возможности объектно-ориентированного языка программирования Visual Basic)	67
<i>Лабораторная работа № 15. Учимся выполнять преобразования логических формул</i> (К параграфу учебника: 3.1. Алгебра логики)	76
<i>Лабораторная работа № 16. Учимся проектировать цифровые схемы из логических элементов</i> (К параграфу учебника: 3.3. Логические основы устройства компьютера)	85