

Информация – это знания или сведения о ком-либо или о чем-либо, которые можно собирать, хранить, передавать, обрабатывать, использовать.

Объекты информации



Для измерения информации вводятся
два параметра



Объем информации
(объемный подход)

Количество информации
(вероятностный подход)

Объемный подход

Если количество информации, содержащейся в сообщении из одного символа, принять за единицу, то объем информации (данных) V в любом другом сообщении будет равен количеству символов (разрядов) в этом сообщении. В памяти компьютера объем информации записывается двоичными знаками и равен количеству требуемых для этой записи двоичных кодов.

Единицы измерения информации

Наименьшая единица измерения - **1 бит**

1 байт = 8 бит

1 Кбайт = 1024 байт

1 Мбайт = 1024 Кбайт = 1 048 576 байт;

1 Гбайт = 1024 Мбайт = 1 073 741 824 байт;

1 Тбайт = 1024 Гбайт = 1 099 511 627 776 байт.

Кодирование информации. Основные понятия



Источник представляет сообщение в алфавите, который называется *первичным*, далее это сообщение попадает в устройство, преобразующее и представляющее его во *вторичном алфавите*.

- *Код* – правило, описывающее соответствие знаков (или их сочетаний) первичного алфавита знаком (их сочетаниями) вторичного алфавита.
- *Кодирование* – перевод информации, представленной сообщением в первичном алфавите, в последовательность кодов.
- *Декодирование* – операция обратная кодированию.
- *Кодер* – устройство, обеспечивающее выполнение операции кодирования.
- *Декодер* – устройство, производящее декодирование.

Операции кодирования и декодирования называются *обратимыми*, если их последовательное применение обеспечит возврат к исходной информации без каких-либо ее потерь.

Информация передается в виде сообщений. Информация может быть по своей физической природе



Любая информация, обрабатываемая в ЭВМ, должна быть представлена двоичными цифрами $\{0,1\}$, т.е. должна быть закодирована комбинацией этих цифр. Различные виды информации (числа, тексты, графика, звук) имеют свои правила кодирования. Коды отдельных значений, относящиеся к различным видам информации, могут совпадать. Поэтому расшифровка кодированных данных осуществляется по контексту при выполнении команд программы.

Транслятор - обслуживающая программа, преобразующая исходную программу, предоставленную на входном языке программирования, в рабочую программу, представленную на объектном языке



Язык, на котором представлена входная программа, называется исходным языком, а сама программа — исходным кодом. Выходной язык называется целевым языком, а выходная (результатирующая) программа — объектным кодом.

Компилятор - это обслуживающая программа, выполняющая трансляцию на машинный язык программы, записанной на исходном языке программирования. Результат компилятора – это exe файл. И может быть запущен в рамках ОС

Интерпретатор - программа или устройство, осуществляющее пооператорную трансляцию и выполнение исходной программы.

Ассемблер - системная обслуживающая программа, которая преобразует символические конструкции в команды машинного языка. Это языки, в которых вместо численного обозначения команд и областей памяти используются буквенные. После ассемблеров наступил рассвет языков так называемого **высокого уровня**.

Транслятор

- генерирует выходную программу (ее часто называют объектной) на языке машинных команд;

- анализирует транслируемую программу, в частности определяет, содержит ли она синтаксические ошибки;

- распределяет память для объектной программы.

Процесс поиска и устранение ошибок называется **отладкой**.

Ошибки

Синтаксические ошибки – это ошибки в записи конструкций языка программирования

Логические ошибки это ошибки, связанные с неправильным содержанием действий и использованием недопустимых значений величин

Семантические ошибки это нарушение логики программы, приводящее к неверному результату.

Для того чтобы решить задачу с помощью ПК,
необходимо пройти **определенные этапы ее решения.**

□Формализация задачи.

□Создание математической модели.

*□Детальное описание алгоритма (текстовое, псевдокод,
□блок-схема).*

□Реализация на языке программирования.

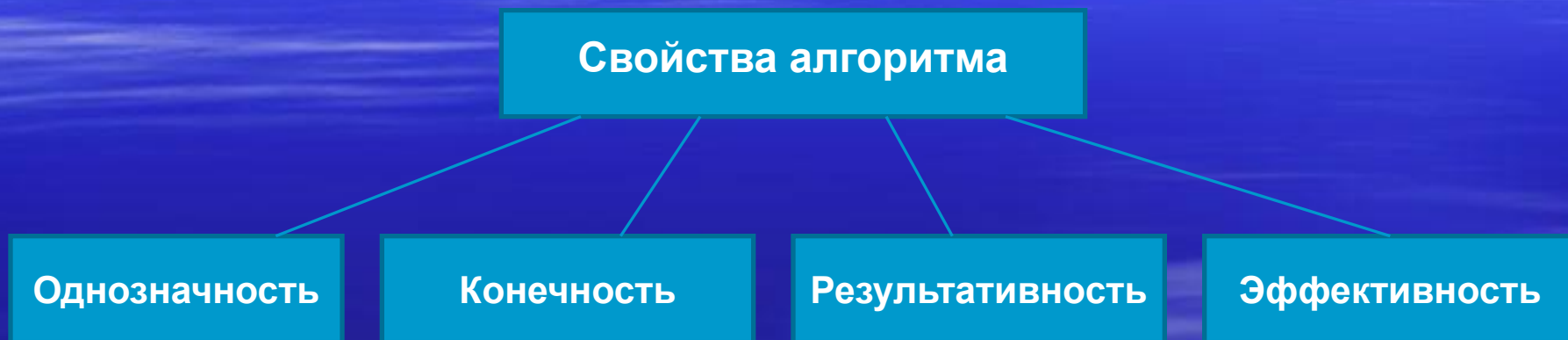
□Отладка программы.

□Тестирование программы.

□Анализ результатов работы.

АЛГОРИТМ. СПОСОБЫ ЗАПИСИ АЛГОРИТМА.

Алгоритм - это конечная последовательность однозначных предписаний, исполнение которых позволяет с помощью конечного числа шагов получить решение задачи, однозначно определяемое исходными данными



Способы записи алгоритма

- ✓ Словесно-формульное описание (на естественном языке с использованием математических формул).
- ✓ Графическое описание в виде блок-схемы (набор связанных между собой геометрических фигур).
- ✓ Описание на каком-либо языке программирования (программа).

Словесно-формульное описание

Запись алгоритма на псевдокоде называется структурным планом.

основные ключевые слова		дополнительные ключевые слова
<u>алг</u> (алгоритм)	<u>рез</u> (результат)	<u>запись</u>
<u>нач</u> (начало)	<u>кон</u> (конец)	<u>истина</u>
<u>арг</u> (аргумент)	<u>знач</u> (значение)	<u>лог</u> (логический)
<u>тип</u>		<u>ложь</u>
<u>вещ</u> (вещественный)	<u>цел</u> (целый)	<u>массив</u>
<u>лит</u> (литерный)	<u>таб</u> (таблица)	<u>множество</u>
<u>сим</u> (символьный)		<u>функция</u>
<u>не</u> <u>то</u> <u>если</u> <u>и</u>		<u>дано</u>
<u>все</u> <u>выбор</u> <u>если</u> <u>иначе</u>		<u>надо</u>
<u>от</u> <u>до</u> <u>шаг</u> <u>для</u>		<u>ввод</u>
<u>при</u> <u>да</u> <u>нет</u>		<u>вывод</u>
<u>пока</u> :=(оператор присваивания)		<u>утв</u> (утверждение)
<u>нц</u> (начало цикла) <u>кц</u> (конец цикла)		

Схема алгоритма – это графическое представление метода решения задачи, в котором используются символы для отображения операций, данных, потока, оборудования и т.д.

При всем разнообразии структур алгоритмов можно выделить четыре типовых структуры

Линейный - алгоритм, в котором все предписания (шаги) выполняются так, как записаны, без изменения порядка следования, строго друг за другом

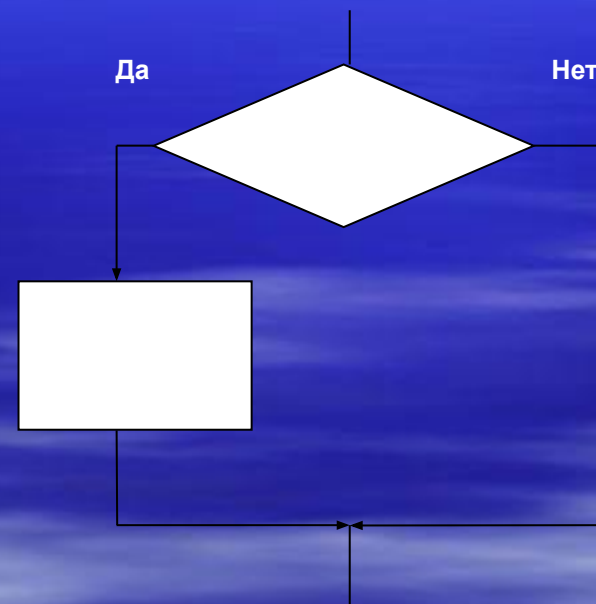
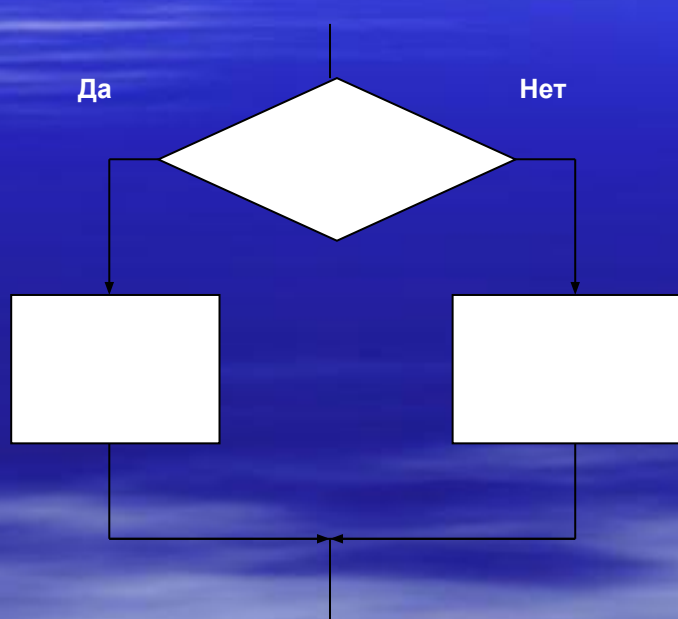
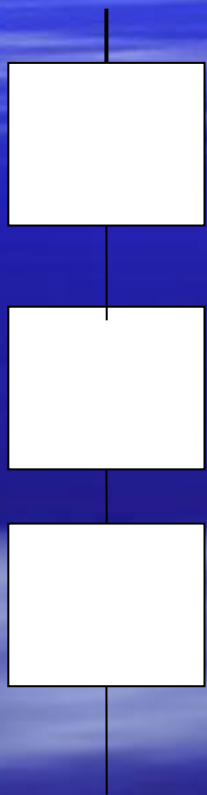
Разветвляющийся - алгоритм, в котором выполнение того или иного действия (шага) зависит от выполнения или не выполнения какого-либо условия

Циклический - алгоритм, в котором некоторая последовательность действий повторяется несколько раз

Типовые структуры алгоритмов

Линейный

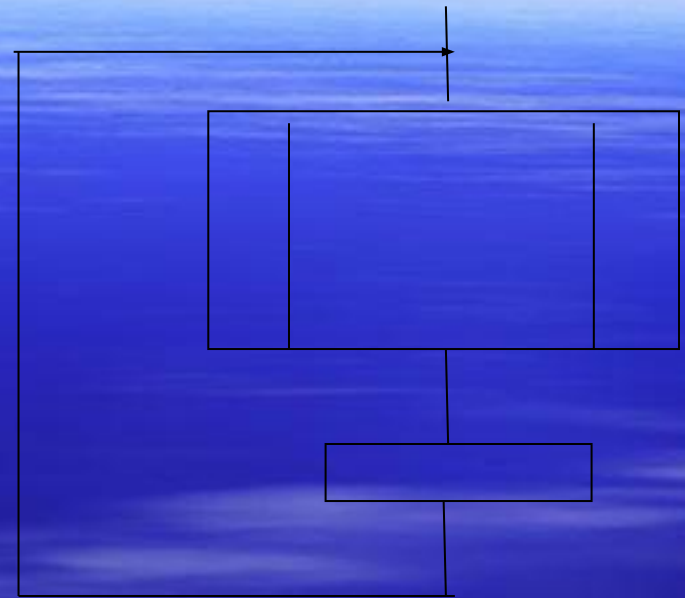
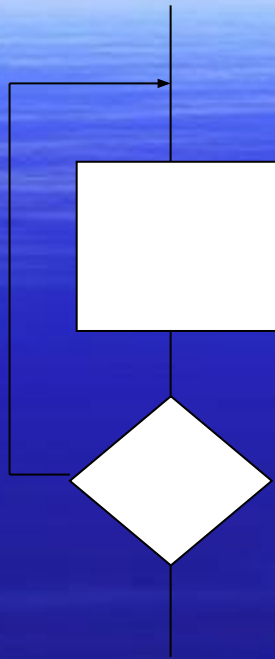
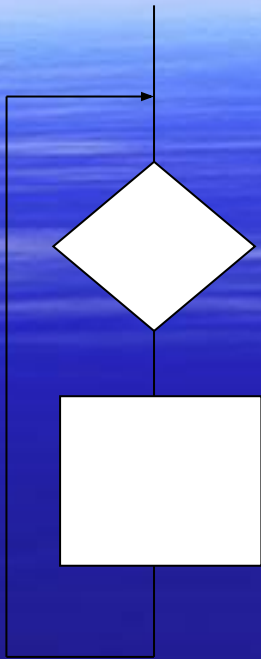
Разветвляющийся



а) - следование;

б, в) – ветвление (полное и неполное).

Циклический



а) – цикл с предусловием;

б) – цикл с постусловием

в) цикл с шагом 1.

Программирование на языке Pascal ABC

Язык Паскаль был создан Никлаусом Виртом в 1968—1969 годах. Язык назван в честь французского математика, физика, литератора и философа Блеза Паскаля, который создал первую в мире механическую машину, складывающую два числа.

Программа - это набор команд (инструкций), которые управляют работой компьютера.

Структура программы на языке программирования PASCAL :

program имя программ

раздел описаний

begin

операторы;

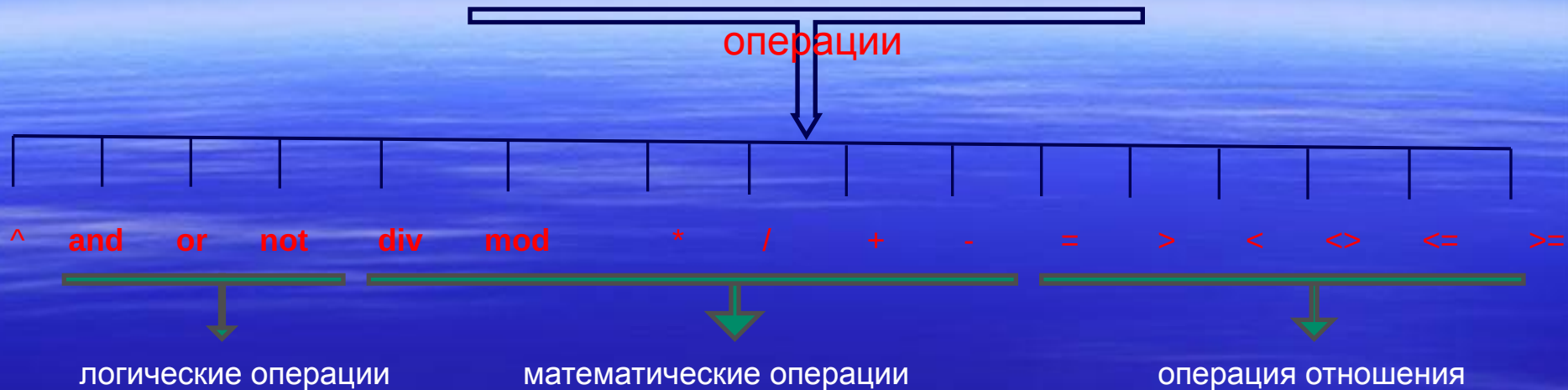
end.

Первая строка называется **заголовком программы** и не является обязательной.

Операторы отделяются один от другого символом "точка с запятой".

Раздел описаний может включать разделы описания переменных, констант, типов, процедур и функций, которые следуют друг за другом в произвольном порядке.

Данные, к которым применяются операции, называются **операндами**.



Простейшими выражениями являются переменные и константы.
Более сложные выражения строятся из более простых с использованием операций, скобок

Выражение, имеющее числовой тип, называется **арифметическим**. Выражение имеет тип integer или real.

Выражение, имеющее тип boolean, называется **логическим**.

Выражение, имеющее тип **string**., называется **строковыми**.

Идентификаторы служат в качестве имен программ, модулей, процедур, функций, типов, переменных и констант.

Идентификатором считается любая последовательность латинских букв или цифр, начинающаяся с буквы.

Любой используемый в блоке идентификатор должен быть предварительно описан. В одном блоке не может быть описано двух переменных, констант или типов с одним именем

В блоке может быть описано несколько процедур или функций с одним именем, но с разным набором параметров

Область действия идентификатора простирается от момента описания до конца блока, в котором он описан.

Блоком называется раздел описаний, после которого следуют операторы, заключенные **в операторные скобки**

begin / end.

Раздел описания переменных начинается со служебного слова **var**. Раздел описаний включает разделы описания переменных, констант, типов, процедур и функций, которые следуют друг за другом в произвольном порядке. Имена в списке перечисляются через запятую.

var <список имен переменных>: тип;

Раздел описания именованных констант начинается со служебного слова **const**

const <имя константы> = <значение>;

или

const <имя константы> : <тип> = <значение>;

Для вывода в окно вывода используются стандартные процедуры

write

или

writeln

Параметры в списке перечисляются через запятую и должны иметь простой тип, либо тип **string**

```
writeln(f, 'abc' ,l);
```

В процедурах вывода **write** и **writeln** после каждого выводимого значения типа может указываться **формат вывода**

```
writeln(f, 'abc' , l:6:2);
```

Комментарий – это любой текст, заключённый в фигурные скобки

```
{ Текст комментария }
```

```
// текст комментария
```

```
writeln(f, 'abc' ,l);//оператор вывода
```

Оператор присваивания имеет вид:

переменная := выражение

Простое логическое выражение состоит из двух переменных или выражений, связанных операцией отношения.

= (равно); <> (не равно);
< (меньше чем); <= (меньше чем или равно);
> (больше чем);
>= (больше чем или равно)

Используя ключевые слова AND (И) или OR (ИЛИ) можно объединить вместе несколько простых логических выражений.

Условный оператор имеет *полную* и *краткую* формы.

Полная форма условного оператора выглядит следующим образом:

```
if <условие> then <оператор1>  
else <оператор2>;
```

Перед ключевым словом else точка с запятой не ставится.

Краткая форма условного оператора имеет вид:

```
if <условие> then <оператор>;
```

Оператор цикла while

Оператор цикла **while** имеет следующую форму:

```
while <условие> do  
    оператор;
```

Оператор цикла repeat

Оператор цикла **repeat** имеет следующую форму:

```
repeat  
    операторы  
until <условие>;
```

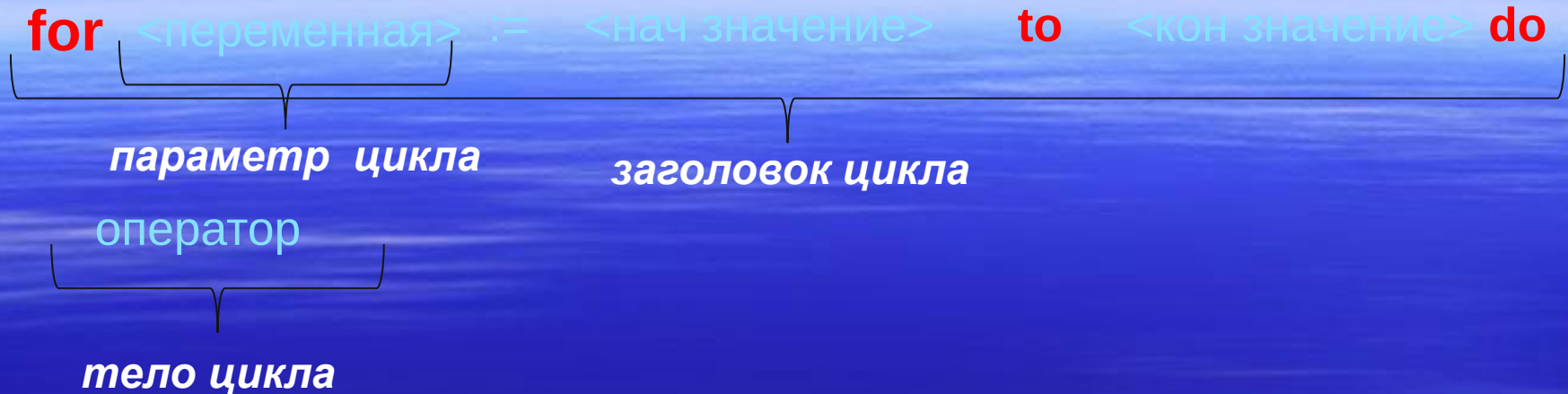
Чтобы прервать зациклившуюся программу, следует использовать комбинацию клавиш

Ctrl-F2

или кнопку.

Оператор цикла for

Оператор цикла **for** имеет одну из двух форм:



или

for <переменная>:=<нач значение> **downto** <кон значение> **do**

оператор

Итерация цикла - однократное повторение тела цикла

Стандартные функции и процедуры

Имя и параметры Действие

Abs(x)	возвращает абсолютное значение (модуль) x
Sqr(x)	возвращает квадрат x
Sqrt(x)	возвращает квадратный корень из x
Sin(x)	возвращает синус x
Cos(x)	возвращает косинус x
Ln(x)	возвращает натуральный логарифм x
Exp(x)	возвращает e в степени x (e=2.718281...)
Power(x,y)	возвращает x в степени y
Round(x)	возвращает результат округления x до ближайшего целого
Int(x)	возвращает целую часть x
Frac(x)	возвращает дробную часть x
Random	возвращает случайное вещественное в диапазоне [0..1)

Функция - имя со списком параметров в виде констант, переменных или выражений

Выражение в скобках называется **аргументом функции**


```

program v; //неизвестно
var x:integer;
y,z:real;
const L=3;
begin
writeln('найти значение 2-х функций');
writeln('-----');
writeln('  x   :   y   | :   z   ');
writeln('-----');
for x:=-5 to 5 do

if x<>0 then
begin
y:=L*sin(x);
z:=cos(x)/x;
writeln('  ',x:2,' :  ',y:4:1,' :  ',z:4:1);
end;

end.

```

найти значение 2-х функций

```

-----
  x   :   y   | :   z
-----
-5   :   2.9   : -0.1
-4   :   2.3   :  0.2
-3   :  -0.4   :  0.3
-2   :  -2.7   :  0.2
-1   :  -2.5   : -0.5
 1   :   2.5   :  0.5
 2   :   2.7   : -0.2
 3   :   0.4   : -0.3
 4   :  -2.3   : -0.2
 5   :  -2.9   :  0.1

```

СПАСИБО ЗА ВНИМАНИЕ