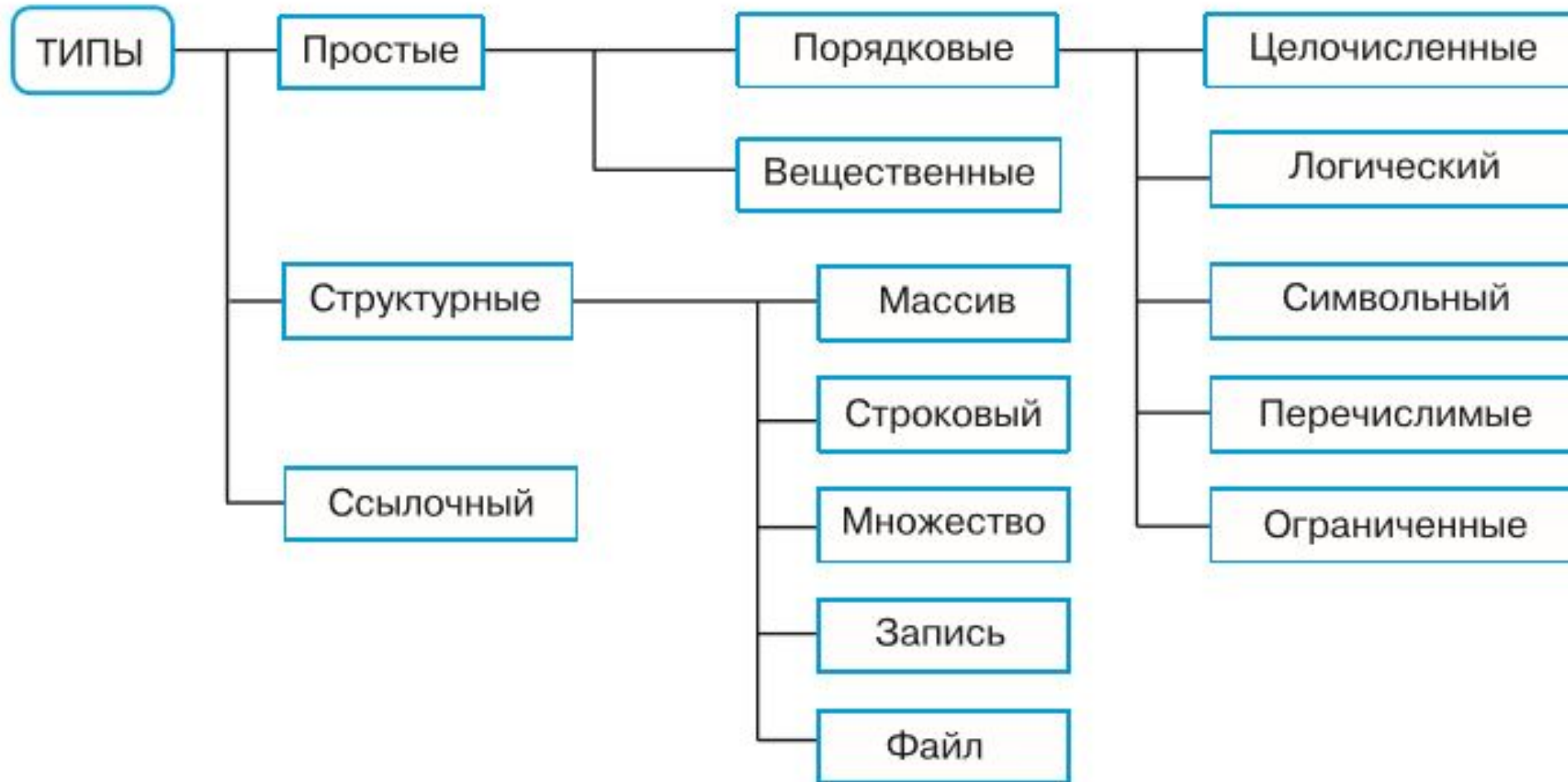


ТИПЫ



Тип	Обозначение типа	К нему относятся	Примеры	Примечание
Целый	<code>int</code>	Целые числа (положительные и отрицательные, а также 0)	4, -45, 0, 687	
Вещественный	<code>float</code>	Вещественные числа ¹ (могут быть с дробной частью)	1.45, 0.00453, -3.789,	Как уже указывалось в главе 2, в Python разделителем целой и дробной частей вещественного числа является точка
Логический	<code>bool</code>	Величины, которые могут принимать значения <code>True</code> («Истина») или <code>False</code> («Ложь»)		
Строковый	<code>str</code>	Последовательность (строка) символов, в том числе один символ или пустая строка (<code>' '</code>)	'Школа', "красный", 'h', ''	Подробно о работе с переменными типа <code>str</code> будет рассказано в главе 11

СИСТЕМА ТИПОВ



Запись инструкции



Что указано в скобках	Пример	На экран будет выведено	Пример
1. Текст	<code>print('Привет!')</code>	Текст без кавычек, включая возможные начальные и конечные пробелы	Привет!
2. Число	<code>print(-2)</code>	Соответствующее число	-2
3. Имя переменной величины	<code>print(x1)</code>	Значение величины	273
4. Выражение	<code>print(a * b)</code>	Значение выражения	1024
5. Метод	<code>print(famil.upper())</code>	Результат применения метода	ЛУКИН

Линейный алгоритм

алг Сумма

нач

цел a, b, c

ввод a, b

c := a + b

вывод c

кон



program Sum;

var a, b, c: integer;

begin

read(a, b);

c := a + b;

write(c)

end.

Pascal
ABC

 python

```
#Ввод исходных данных
print('Задайте первое число')
a = input()
print('Задайте второе число')
b = input()
#Расчет суммы
sum = a + b
#Вывод ответа на экран
print('Сумма этих чисел равна', sum)
```

Условный оператор



```
if ...:  
    print('Это число нечетное')  
else:  
    print('Это число четное')
```

```
если a>b то  
    M:=a  
иначе  
    M:=b  
все
```



```
if a>b then  
    M:=a  
else  
    M:=b;
```



Вложенные условные операторы

```
if a>b then write('A') else if a=b then write('=')  
else write('B');
```

может быть записан с отступами так:

```
if a>b then  
    write('A')  
else  
    if a=b then  
        write('=')  
    else  
        write('B');
```



Деление нацело и остаток

Во многих практических задачах все данные — целые числа. Для них введены две особые операции: **деление нацело** и **остаток от деления**, которые обозначаются как «//» и «%» соответственно. Они имеют такой же приоритет, как умножение и деление.

```
d = 85
a = d // 10    # = 8
b = d % 10    # = 5
```

Обратим внимание на результат выполнения этих операций для отрицательных чисел. Программа

```
print (-7 // 2)
print (-7 % 2)
```



В таких случаях в алгоритмическом языке используют команды `div` и `mod`, а в Паскале — операции с теми же именами (они имеют такой же приоритет, как умножение и деление):

```
t:=175
m:=div(t, 60)    | = 2
s:=mod(t, 60)   | = 55
```

```
t:=175;
m:=t div 60     { = 2 }
s:=t mod 60     { = 55 }
```

С помощью этих операций удобно работать с отдельными цифрами числа. Как мы увидели в главе 2, остаток от деления числа на 10 — это последняя цифра его десятичной записи¹⁾.

```
N:=123
d1:=mod(N, 10)  | =3
```

```
N:=123;
d1:=N mod 10;  { =3 }
```



Операции



$x + y$	Сложение
$x - y$	Вычитание
$x * y$	Умножение
x / y	Деление
$x // y$	Получение целой части от деления
$x \% y$	Остаток от деления
$-x$	Смена знака числа
$\text{abs}(x)$	Модуль числа
$\text{divmod}(x, y)$	Пара $(x // y, x \% y)$
$x ** y$	Возведение в степень
$\text{pow}(x, y[, z])$	x^y по модулю (если модуль задан)