

§ 19. Реализация алгоритмов работы с целочисленными данными

19.1. Целочисленный тип данных

Часто при решении задач используются целые числа. В языке Python целые числа могут быть сколь угодно большими. Для этого типа данных определены следующие операции:

Математические операции	Запись в Python
+ (сложение)	+
− (вычитание)	−
· (умножение)	*
целочисленное деление	//
нахождение остатка	%
возведение в степень	**

Для целочисленных данных не определена операция деления (как для рациональных чисел). Если использовать эту операцию, то результат всегда будет рациональным, даже если одно число делится на другое без остатка.

Для организации вычислений с целыми числами определены операции // и %. Значения результатов операций // и % могут отличаться от принятых в математике (пример 19.1).

Чтобы облегчить визуальную оценку величины числа, в Python, начиная с версии 3.6, между цифрами разрешается вставлять одиночные символы подчёркивания. Запись $a = 1000000$ эквивалентна записи $a = 1_000_000$. Такую запись можно использовать как при написании текста программы, так и при вводе чисел с клавиатуры.

Пример 19.1. Результат операций // и % для чисел разных знаков.

a	b	a // b	a % b
17	3	5	2
−17	3	−6	1
17	−3	−6	−1
−17	−3	5	−2

При выполнении операции // знак результата определяется так же, как в математике: он положительный, если исходные числа одного знака, и отрицательный, если разного. Значение результата округляется до ближайшего целого в меньшую сторону.

Пример 19.1. Продолжение.

Результат операции `%` может быть отрицательным, хотя в математике под остатком понимают неотрицательное число. Если остаток не равен нулю, то знак числа, которое является результатом операции `%`, определяется знаком делителя. При анализе данных из таблицы можно понять, что

$$a \% b = a - (a // b) * b.$$

Пример 19.2. Решение.

V. Программа:

```
a = int(input('a = '))
b = int(input('b = '))
c = a // b
d = a % b
print('целая часть =', c)
print('остаток =', d)
```

VI. Тестирование программы.

Запустите программу и введите значения $a = 11$ и $b = 4$.

Проверьте, результат должен быть следующим:

```
a = 11
b = 4
целая часть = 2
остаток = 3
```

Пример 19.3. Решение.

V. Программа:

```
c = int(input('c = '))
#минуты
m = c // 60
#секунды
s = c % 60
print(m, s, sep = ':')
```

Эти операции имеют такой же приоритет, как и операции деления и умножения.

Пример 19.2. Даны два целых числа a и b . Напишите программу, которая находит целую часть от деления a на b и остаток от деления a на b .

Этапы выполнения задания.

I. Определение исходных данных: переменные a и b .

II. Определение результатов: переменные c (целочисленное частное) и d (остаток).

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Целочисленное частное находим как результат операции $a // b$, остаток — $a \% b$.

3. Вывод результата.

IV. Описание переменных: все переменные имеют тип `int`.

19.2. Использование целочисленных данных для решения задач

Пример 19.3. Пусть таймер показывает время только в секундах. Написать программу, которая переведёт время в минуты и секунды.

Этапы выполнения задания.

I. Определение исходных данных: переменная c (время в секундах).

II. Определение результатов: переменные m (полное количество минут) и s (остаток секунд).

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Для нахождения полного числа минут нужно найти целую часть от деления исходного числа секунд на 60.

3. Оставшиеся секунды можно найти как остаток от деления исходного числа секунд на 60.

4. Вывод результата.

IV. Описание переменных: все переменные имеют тип `int`.

Пример 19.4. Задано положительное двузначное число. Написать программу, которая поменяет местами первую и вторую цифры числа.

Этапы выполнения задания.

I. Определение исходных данных: переменная a (исходное число).

II. Определение результатов: переменная b (преобразованное число).

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Для преобразования числа необходимо выполнить следующие действия:

Пример 19.3. Продолжение.

VI. Тестирование программы.

Запустите программу и введите значение $c = 137$. Результат должен быть следующим:

```
c = 137
2:17
```

Для $c = 24$ получим:

```
c = 24
0:24
```

Операция нахождения остатка в разных языках программирования реализуется по-разному для отрицательного делимого или делителя. В некоторых языках такая операция не определена (Basic) или остаток всегда неотрицательный, как в математике (Oberon). В других языках остаток может быть отрицательным. Возможны два варианта:

- знак остатка совпадает со знаком делимого (C++, C#, Pascal);
- знак остатка совпадает со знаком делителя (Python, Prolog).

Есть языки, в которых реализованы оба варианта (Java, Haskell). В этом случае операции обозначаются по-разному. Для обозначения операции целочисленного деления часто используют `\`, `//`, `div`, `quot` или знак, которым также обозначают деление для рациональных чисел, `/`. Для операции нахождения остатка используют `%`, `\%`, `mod`, `rem`. Могут использоваться другие обозначения.

Пример 19.4. Решение.**V. Программа:**

```

a = int(input('a = '))
#выделение последней цифры
a1 = a % 10
#выделение первой цифры
a2 = a // 10
b = a1 * 10 + a2
print('результат =', b)

```

VI. Тестирование программы.

Запустите программу и введите значение $a = 25$.

Проверьте, результат должен быть следующим:

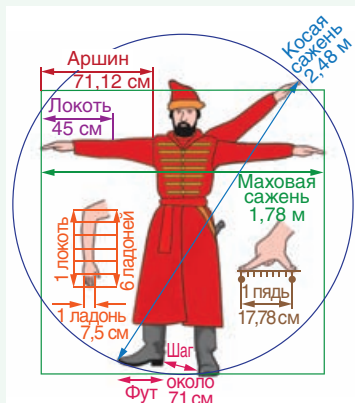
```

a = 25
результат = 52

```

На Руси традиционно применялась русская система мер. Сегодня, несмотря на то, что официально принята метрическая система мер, старорусские названия мер используются в исторических исследованиях и фразеологических оборотах.

Например: 1 аршин = 16 вершков, 1 вершок $\approx 4,5$ см.



1) в переменной $a1$ сохраним вторую цифру числа. Для выделения цифры из числа нужно найти остаток от деления исходного числа на 10 ($a \% 10$);

2) для выделения первой цифры (переменная $a2$) нужно найти целую часть от деления числа на 10;

3) искомое число b получим, если умножим $a1$ на 10 и к полученному произведению прибавим значение переменной $a2$.

3. Вывод результата.

IV. Описание переменных: все переменные имеют тип `int`.

Пример 19.5. В исторической книге, которую читала Таня, длина отреза ткани измерялась в локтях. Напишите программу, которая переведёт локти в метры и сантиметры.

Этапы выполнения задания.

I. Определение исходных данных: переменная l (локти).

II. Определение результатов: переменные m (метры) и s (сантиметры).

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Сначала переведем локти в сантиметры. Для этого количество локтей нужно умножить на 45 и сохранить значение в переменной x .

3. Для определения числа метров найдём целую часть от деления x на 100.

4. Оставшиеся сантиметры можно найти как остаток от деления x на 100.

5. Вывод результата.

IV. Описание переменных: все переменные имеют тип `int`.

В Python, как и в большинстве других языков программирования, запрещено деление на нуль. Если попытаться это сделать, появится сообщение об ошибке `ZeroDivisionError`.

Пример 19.5. Решение.

V. Программа:

```
l = int(input('l = '))
x = l * 45
#метры
m = x // 100
#сантиметры
s = x % 100
print(l, 'локтей =', end = ' ')
print(m, 'м', s, 'см')
```

V. Тестирование программы.

Запустите программу и введите значение $l = 7$. Проверьте, результат должен быть следующим:

```
l = 7
7 локтей = 3 м 15 см
```



1. Какой тип данных можно использовать в Python для работы с целочисленными данными?
2. Какие операции определены для целочисленных данных?
3. Как определяется результат целочисленной операции `//` для чисел с разными знаками?
4. Как определяется результат целочисленной операции `%` для чисел с разными знаками?



Упражнения

1 Вася написал программу, которая переводит длину из метров в километры и метры. Но он не может решить, где нужно использовать `//`, а где — `%`. Помогите ему. Откройте файл и исправьте программу.

```
d = int(input('d = '))
k = d ... 1000
m = d ... 1000
print(d, 'м =', end = ' ')
print(k, 'км', m, 'м')
```

- 2 Ответьте на вопросы по примеру 19.4 на с. 137.
1. При каких значениях переменной a значение переменной b будет таким же?
 2. Всегда ли в результате выполнения программы получается двузначное число? Почему?
 3. Попробуйте ввести трёхзначное число (например, 125). Объясните получившийся результат.
- 3 Напишите программы для решения задач. Используйте операции `//` и `%`.
1. Задано положительное двузначное число. Найдите среднее арифметическое цифр числа.
 2. Задано положительное двузначное число. Найдите разность между количеством десятков и единиц.
 3. Дана масса в граммах. Переведите её в килограммы и граммы.
 4. Площадь участка измеряется в арах. Выразите эту площадь в квадратных километрах.
 5. Размер файла задан в килобайтах. Переведите его в мегабайты и килобайты.
- 4* Для старорусской системы весов известны следующие соотношения:
1 берковец = 10 пудов = 400 фунтов = 38 400 золотников.
- Напишите программу, которая переводит массу, заданную в золотниках, в фунты, пуды и берковцы.