Оглавление

[Задачи на числа и математические выражения 2](#_Toc199353577)

[Задачи на строки 4](#_Toc199353578)

[Задачи на списки 8](#_Toc199353579)

[Задачи на кортежи 11](#_Toc199353580)

[Задачи на словари 13](#_Toc199353581)

[Задачи на множества 16](#_Toc199353582)

[Задачи на линейные алгоритмы 18](#_Toc199353583)

[Задачи на логические выражения 19](#_Toc199353584)

[Задачи на if/else 21](#_Toc199353585)

[Задачи на циклы for и while 25](#_Toc199353586)

[Задачи на протокол итераций и генераторы 29](#_Toc199353587)

[Задачи на функции 31](#_Toc199353588)

[Задачи на ООП: классы и объекты 36](#_Toc199353589)

[Задачи на импорт, модули и пакеты 41](#_Toc199353590)

[Задачи на файлы и каталоги 43](#_Toc199353591)

[Задачи на исключения и ошибки 47](#_Toc199353592)

[Использование утилиты pip и виртуальных окружений 49](#_Toc199353593)

[Задачи на дату и время 52](#_Toc199353594)

[Олимпиадные задачи 53](#_Toc199353595)

[Первые скрипты и программы 55](#_Toc199353596)

# Задачи на числа и математические выражения

**1.1.** Выведите на экран строковые представления числа 53 в двоичной, восьмеричной и шестнадцатеричной системах счисления. 

**1.2.** Даны строковые представления чисел: '011011101' (двоичная СС), '357' (восьмеричная СС) и 'AF23D' (шестнадцатеричная СС). Выведите на экран строковые представления этих чисел в десятичной системе счисления.

**1.3.** Зная, что 1372 является числом в восьмеричной системе счисления, выведите на экран значение этого числа в десятичной системе счисления.

**1.4.** Представьте вещественные числа 275.4, 0.0032 и 3.45 в коде Python в стандартном виде. Напомним, что в математике число в такой форме имеет вид a\*10n, где 1 ≤ a < 10, а n – целое число. Например, для числа 0.123 получим 1.23\*10-1.

**1.5.** Найдите наименьшую обыкновенную дробь, равную вещественному числу 14.375, и выведите ее на экран в формате '14.375 = числитель/знаменатель'.

**1.6.** Дано комплексное число 7.5+1.35j. Выведите на экран его действительную и мнимую части, а также число, сопряженное к данному.

**1.7.** Даны два действительных числа a и b. Вычислите их сумму, разность, произведение и частное при a=3.79 и b=84.93. Округлите результаты до сотых и выведите их на экран.

**1.8.** Вычислите значение выражения (10/2.3 - 34)\*0.7 + 90.5, округлив результат до трех знаков после десятичной точки и найдя его модуль.

**1.9.** Найдите значение выражения в десятичной системе счисления: (1EA16 + 010111012 - 2378 - 68110)\*21023. Выведите результат на экран.

**1.10.** Извлеките квадратный корень из 196 тремя способами.

**1.11.** Выведите на экран результат вычисления sin(π/6) и cos(45°) округлив результат до двух знаков после десятичной точки.

**1.12.** Вычислите значение арифметического выражения при заданных значениях переменных и выведите полученный результат на экран: 5.2a3 + 3b5 - 7.3 при a=3, b=2.5.

**1.13.** Вычислите значение арифметического выражения при заданных значениях переменных и выведите полученный результат на экран: x(3.3 + 2y) - |64:(x + y)| при x=-4.1, y=2.

**1.14.** Вычислите значение арифметического выражения при заданных значениях переменных и выведите полученный результат на экран: |m(2m-1) - 35.5|:(3n + 0.8m)2 при m=2, n=5.

**1.15.** Вычислите значение арифметического выражения при заданных значениях переменных и выведите полученный результат на экран: ln|a2 - 1.5a:b| + e3 при a=5.73, b=1.7.

**1.16.** Вычислите значение арифметического выражения при заданных значениях переменных и выведите полученный результат на экран: log4(7x - 3y) + √(lg|10xy|) при x=4, y=-3.

**1.17.** Вычислите значение арифметического выражения при заданных значениях переменных и выведите полученный результат на экран: √(log2(mn+2 - 3e)):(ln(2m) + lg(4n)) при m=5, n=2.

**1.18.** Запишите арифметическое выражение на языке Python в виде строки, а затем выведите полученный результат на экран: cos2(3a - 1) - sin(5a - b)3.

**1.19.** Запишите арифметическое выражение на языке Python в виде строки, а затем выведите полученный результат на экран: 3tg|√(x + y2) - π| - arctg3(√x + y2).

**1.20.** Запишите арифметическое выражение на языке Python в виде строки, а затем выведите полученный результат на экран: √(arccos3x - arcsin2y)/arctg|x2 - y2| + 5√π.

**1.21.** Вычислите значение арифметического выражения |3√π4 - 8eπ|/ln9.7. Округлите результат до сотых и выведите его на экран.

# Задачи на строки

Не забываем, что строки относятся к неизменяемому типу данных, поэтому изменить их непосредственно не получится. Но всегда можно получить измененную копию строки и заменить ею хранящуюся в переменной исходную строку.

**2.1.** Составьте и выведите на экран пользователя строку, в которой через запятую перечислите целое, вещественное и комплексное числа в формате «'int', 'float', 'complex'». Реализуйте вывод двумя способами, использовав в качестве внешних двойные и одинарные кавычки.

**2.2.** Составьте и выведите на экран пользователя строку-матрешку с содержимым «'4 + "3 + '2 + "1 + '0' " ' " '». Реализуйте вывод четырьмя способами, использовав в качестве внешних все виды кавычек, разрешенных для строк.

**2.3.** Необходимо вывести на экран пользователя строку с адресом '..\new\_dir\tables.py'. Реализуйте вывод двумя способами.

**2.4.** Дана строка '123456789'. Используя операции индексирования и среза выведите на экран третий и пятый символы, а также подстроку '567'. Реализуйте вывод двумя способами: используя только положительные индексы и только отрицательные.

**2.5.** Дана строка 'AaBbCcDd'. Используя срезы с шагом получите две строки: только с заглавными и только со строчными буквами. Выведите их на экран.

**2.6.** Измените строку 'кот', записав ее символы в обратном порядке. Выведите результат на экран.

**2.7.** Дана строка '0123456789'. Удалите из нее первый, пятый и последний символы. Выведите результат на экран.

**2.8.** При помощи строк '.', '!' и ' ' (один пробел) сформируйте новую строку '..... ! ! !....' и выведите ее на экран. Разрешается использовать только операторы + и

**2.9.** Дана строка 'Вот и пришла осень золотая!'. Возьмите ее в кавычки «» и выведите результат на экран. Шестнадцатиричные коды кавычек в Юникоде: 00AB и 00BB.

**2.10.** Подсчитайте количество символов в строке '1a\u0398\t\u03B43s'. Сделайте это сперва устно, а затем проверьте себя программно.

**2.11.** Удалите в строке ' a b c   d e f ' все пробелы и выведите результат на экран.

**2.12.** Дана строка "..-.--..-.--". После каждой точки допишите еще по одной точке и выведите результат на экран.

**2.13.** Дана строка '131231442145'. Подсчитайте в ней количество символов '1' и выведите результат на экран.

**2.14.** Посчитайте устно, а затем проверьте себя, выведя на экран сумму индексов символа 'а' в строках 'повар' и 'Пайтон'.

**2.15.** Даны две переменные: a = 73 и b = 95. Используя метод форматирования строк, выведите на экран их сумму и произведение в форматах 'a + b = c' и 'a\*b = c'. Повторите тоже самое, но использовав для этого F-строки.

**2.16.** Дана строка 'ABCDEF'. Поменяйте в ней первый и последний символы местами и выведите итоговую строку на экран.

**2.17.** Допишите в конец строки 'автопрогон' ее длину и выведите результат на экран.

**2.18.** Дана строка 'в Ереване'. Подсчитайте в ней количество русских символов 'е' в любом регистре и выведите результат на экран.

**2.19.** Подсчитайте количество слов в строке 'Вот и пришла осень золотая!'. Разделителем слов считайте пробел. Выведите результат подсчета на экран.

**2.20.** Найдите количество вхождений подстроки 'ab' в строку 'ab bc ab bd ab be' и поменяйте в каждом случае символы 'a' и 'b' местами. Выведите результаты на экран.

**2.21.** Определите совпадает ли количество круглых открывающихся и круглых закрывающихся скобок в строке 'abs(math.sin(pow((5\*a - b), 3)))'. Выведите результат проверки на экран в виде сообщения.

**2.22.** Определите есть ли в строке 'Земля-2022' хотя бы один символ арабской цифры от нуля до девяти и выведите результат проверки на экран.

**2.23.** Подсчитайте общее количество алфавитных символов Юникода в строке 'a1, b2, c3, e4, d5, f6.' и выведите результат на экран.

**2.24.** Подсчитайте количество цифр 7 дробной части десятичного числа 377.37177785772707.

**2.25.** Составьте строку, состоящую из заглавных букв латинского алфавита, и выведите ее на экран. В таблице символов Юникода заглавные буквы латинского алфавита занимают диапазон номеров с 65 по 90.

**2.26.** Выведите на экран сумму числовых кодов символов строки 'два пижона'.

**2.27.** Подсчитайте количество русских букв в строке 'Camel - это же верблюд, а не ёж!' и выведите результат на экран. В таблице символов Юникода строчные буквы русского алфавита занимают диапазон номеров с 1040 по 1103, а также 1025 и 1105 для букв Ё и ё.

**2.28.** Дана строка '\_\*..\*..\*\_'. Замените в ней все символы '\*' числами, которые соответствуют их номеру вхождения в строку (первая слева имеет номер вхождения один и т.д.). Выведите измененную строку на экран.

**2.29.** Подсчитайте, сколько видов букв присутствует в строке 'синхрофазотрон', и выведите результаты на экран.

**2.30.** Дана строка 'акваланг'. Замените в ней все символы 'а' числами, которые соответствуют индексу этого символа в строке. Выведите измененную строку на экран.

**2.31.** Подсчитайте количество каждого вида символов в строке 'выхухоль' и выведите результаты на экран.

**2.32.** Измените строку 'Чья это корова там замычала?' таким образом, чтобы символы в словах были записаны в обратном порядке. Разделителем слов считайте пробел. Выведите результат на экран.

**2.33.** Дано строковое представление времени таймера '03:25:57'. Выведите на экран количество секунд, оставшихся до момента срабатывания таймера. Используйте формат сообщения «До срабатывания таймера осталось {n} сек.».

# Задачи на списки

Не забываем, что списки, в отличие от строк и кортежей, мы можем изменять непосредственно.

**3.1.** Используя операции индексирования и среза выведите на экран первый и третий элементы списка [1, 2, 3, 4, 5], а также срез списка из первых трех элементов. Реализуйте вывод двумя способами: используя только положительные и только отрицательные индексы.

**3.2.** Дан список [[1, 2, ['Ok!', 3]], ['list', 4], 5]. Выведите на экран строку 'Ok!', использовав синтаксис доступа к элементу списка по его индексу.

**3.3.** Создайте список-матрешку, в который поместите два элемента: целое число и вложенный список, в который поместите еще два элемента: вещественное число и вложенный список, в который поместите еще два элемента: комплексное число и вложенный список, в который поместите еще два элемента: строку и пустой список. Выведите на экран конечную строку.

**3.4.** Дан список ['Санкт', '+', 'Петербург']. Исправьте плюс на дефис и выведите название города на экран использовав доступ к элементам списка по индексам.

**3.5.** Дан список ['a', '1', 'b', '2', 'c', '3']. Разбейте его на два списка: только с буквами и только с числами. Сам список затем удалите, а новые списки выведите на экран, каждый на отдельной строке.

**3.6.** Замените в списке [1, 2, 3, 4, 5] первые три элемента их суммой. Затем добавьте в конец списка число 7. Переместите первый элемент конечного списка в предпоследнюю позицию списка. Выведите список на экран. Используйте в решении только доступ по индексу и срезы.

**3.7.** Дан список [3, 4, 1, 1, 5, 1]. Выведите на экран: количество всех элементов в списке, количество единиц в списке, индекс первого вхождения единицы в список.

**3.8.** Дан пустой список []. Используя методы списков: добавьте в него строку 'a', в конец списка еще и строку 'b', расширьте список за счет списка ['c', 'e'], вставьте в список перед 'e' строку 'd'. Выведите конечный результат на экран.

**3.9.** В предыдущей задаче должен был получиться список ['a', 'b', 'c', 'd', 'e']. Используя методы списков: удалите из него последний элемент, инверсируйте список (измените порядок следования элементов на обратный), удалите элемент со значением 'c', очистите список.

**3.10.** Дан список неповторяющихся целых чисел [3, -54, 25, 8, 0]. Выведите на экран: сумму его элементов, разность максимального и минимального значений.

**3.11.** Дан список чисел [7, 7, -4, 2.5, -.9, 0]. Выведите на экран количество отрицательных элементов.

**3.12.** Дан список неповторяющихся вещественных чисел [0.3, -2.4, 4.5, 0.0, -3.1]. Выведите на экран индекс минимального элемента.

**3.13.** Дан список целых чисел [9, -3, 5, 0, -3]. Выведите на экран сумму абсолютных значений его элементов.

**3.14.** Дан список неповторяющихся целых чисел [9, -210, 0, 500, -37]. Поменяйте местами максимальный и минимальный элементы списка. Выведите измененный список на экран.

**3.15.** Дан список целых чисел [3, 2, 3, 2, 1, 4]. Расположите числа в порядке убывания их значений, а затем добавьте в конец списка его длину. Выведите измененный список на экран.

**3.16.** Дан список [True, 5, 'go', 3+0.1j]. Циклически сдвиньте все его элементы на одну позицию вправо так, чтобы li[0] перешел на место li[1], li[1] — на место li[2],.., а последний элемент перешел на место li[0]. Выведите измененный список на экран.

**3.17.** Дан список натуральных чисел [13, 5, 5, 8, 16, 4]. Удалите из него первое четное число, имеющее нечетный индекс. Выведите измененный список на экран.

**3.18.** Дан список строк ['уж', 'змея', '', 'питон']. Удалите из него все строки с длиной менее пяти символов и выведите результат на экран.

**3.19.** Дан список ['три', [2, 5], 0.7, 'home']. Создайте список типов элементов данного списка и выведите результат на экран.

**3.20.** Дан список [1.0, 'ok', True, 'ok', 7, 1]. Удалите из него повторяющиеся элементы (в списке не должно остаться дублей). Выведите измененный список на экран. Будьте внимательны! Значения 1.0, True и 1 – это не дубли!

**3.21.** Даны два списка строк ['1', 'True', 'ok'] и ['no', 'True', 'no']. Сформируйте новый список, который будет состоять из строк, которые присутствуют хотя бы в одном из списков. Выведите полученный список на экран.

**3.22.** Дан список [1, None, 3.0, [1, 2], 'b']. Поменяйте в нем соседние элементы местами. Т.к. число элементов в списке нечетное, последний элемент оставьте на своем месте. Выведите измененный список на экран.

# Задачи на кортежи

Не забываем, что кортежи относятся к неизменяемому типу данных, поэтому изменить их непосредственно не получится. Но всегда можно получить измененную копию кортежа и заменить ею хранящийся в переменной исходный кортеж. Следует заметить, что практически все задачи для списков можно адресовать и кортежам, т.к. они легко конвертируемы в списки и обратно при помощи конструкторов list(tpl) и tuple(li). Однако кортежи в основном предназначены для хранения неизменяемых данных, поэтому для них первостепенными следует считать задачи по извлечению значений и получению индексов элементов.

**4.1.** Создайте кортеж, в качестве элементов которого используйте целое, вещественное и комплексное числа, кортеж с одним элементом в виде строки, пустой кортеж и пустой список. Выведите на экран строку вложенного кортежа.

**4.2.** Создайте кортеж-матрешку, в который поместите два элемента: целое число и вложенный кортеж, в который поместите еще два элемента: вещественное число и вложенный кортеж, в который поместите еще два элемента: комплексное число и вложенный кортеж, в который поместите еще два элемента: строку и пустой кортеж. Выведите на экран конечную строку. Обратите внимание на схожесть решения с задачей 3.3.

**4.3.** Используя операции индексирования и среза выведите на экран первый и третий элементы кортежа (1, 2, 3, 4, 5), а также срез кортежа из первых трех элементов. Реализуйте вывод двумя способами: используя только положительные и только отрицательные индексы. Обратите внимание на схожесть решения и результатов с задачей 3.1.

**4.4.** Дан кортеж ((1, 2, ('Ok!', 3)), ('tuple', 4), 5). Выведите на экран строку 'Ok!', использовав синтаксис доступа к элементу кортежа по его индексу. Обратите внимание на схожесть решения и результатов с задачей 3.2.

**4.5.** Дан кортеж (3, 's', 1, 5, 's'). Выведите на экран: количество всех элементов кортежа, количество строк 's', индекс первого вхождения 's' в кортеж.

**4.6.** Дан кортеж (['кит', 1, 3], 5). Замените в списке 'кит' на 'кот', удалите единицу, а также измените значение последнего элемента списка, возведя тройку в квадрат. Выведите кортеж на экран. Попробуйте изменить второй элемент кортежа, умножив его на два. Объясните результат.

# Задачи на словари

**5.1.** Создайте словарь d = {'1': 0, 2: 0, '3': 0} тремя способами. Выведите полученный словарь на экран.

**5.2.** Создайте словарь содержащий данные о человеке. В качестве ключей используйте строки 'имя', 'возраст' и 'пол'. Значения придумайте самостоятельно. Выведите полученный словарь на экран.

**5.3.** Дан словарь d = {'1': 1.29, '2': 0.43}. Используя доступ к элементам словаря по ключу, найдите произведение 1.29\*0.43, после чего добавьте результат в словарь, а затем выведите значение нового элемента на экран.

**5.4.** Найдите представления ключей, значений и элементов словаря d = {1: 1, '2': 2, '3': 3, 4: 4}. Выведите их на экран в виде списков.

**5.5.** Сколько элементов будет содержать словарь d\_1 = {'1': 1, '2': 2} после добавления к нему элементов словаря d\_2 = {'2': 'two', '3': 3}? Каковы будут значения элементов итогового словаря? Проверьте свой ответ программно.

**5.6.** Удалите из словаря d = {'a': 1, 'b': 2, 'c': 3} элементы d['a'] и d['c'], использовав для второго элемента оператор удаления, а для первого имеющийся у словарей метод. Выведите результат на экран, после чего очистите словарь.

**5.7.** Требуется создать словарь colors для хранения информации о представлениях цветов HTML в виде специальных названий, а также в цветовой модели RGB в десятичном и шестнадцатеричном форматах. Сохраните в словаре информацию для черного, серого и белого цветов. В HTML эти цвета могут задаваться, соответственно:

* по названиям: black, grey и white;
* в 10-м формате: rgb(0, 0, 0), rgb(128, 128, 128) и rgb(255, 255, 255);
* в 16-м формате: #000000, #808080 и #FFFFFF.

При создании хранилища в родительском словаре используйте ключи 'черный', 'серый' и 'белый', а во вложенных – 'name', 'rgb10' и 'rgb16'. Выведите на экран представление серого цвета по имени в виде строки 'color: name;', а также в 10-м формате в виде строки 'color: rgb(a, b, c);' и в 16-м формате в виде строки 'color: #RRGGBB;'.

**5.8.** Создайте словарь-матрешку, в который поместите элемент со строковым ключом в формате целого числа и значением в виде вложенного словаря, в который поместите элемент со строковым ключом в формате вещественного числа и значением в виде вложенного словаря, в который поместите элемент со строковым ключом в формате комплексного числа и значением в виде вложенного словаря, в который поместите элемент со строковым ключом в формате пустого списка и значением в виде вложенного словаря, в который поместите элемент со строковым ключом в формате кортежа с одним элементом и значением в виде строки 'Ok!'. Выведите на экран конечную строку.

**5.9.** Дан список ['Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб', 'Вс']. Создайте словарь с ключами 'd1', 'd2',.., соответствующими индексам элементов списка. Выведите на экран элемент словаря, который соответствует среде, в формате ключ: значение.

**5.10.** Дан словарь с натуральными ключами {1: 'Один', 2: 'Два', 3: 'Три', 100: 'сто'}. Выведите на экран сумму ключей, максимальное и минимальное значения ключей, а также их количество.

**5.11.** Дан словарь со значениями констант {'pi': 3.14, 'e': 2.71, 'fi': 1.62}. Выведите на экран значения констант, которые превышают число 2.5.

**5.12.** Поменяйте в словаре d = {1: '1', 2: '2', 3: '3', 4: '4'} ключи и значения местами. Выведите итоговый словарь на экран.

**5.13.** Удалите из словаря d = {1: 1, '2': 2, '3': 3, 4: 4} все элементы со строковыми ключами. Помните, что изменение размеров словаря во время итерации может привести к ошибке, поэтому для работы используйте копию словаря. Выведите итоговый словарь на экран.

**5.14.** Дан словарь с товарами. Выведите на экран все товары, цена которых не превышает 100 рублей, а текущий остаток не менее 10 кг.

**Условие**

goods = {

"apple": {"name": "Яблоки", "cost": 25, "kg": 30},

"pear": {"name": "Груши", "cost": 50, "kg": 5},

"plum": {"name": "Сливы", "cost": 55, "kg": 25},

"cherry": {"name": "Вишни", "cost": 110, "kg": 15}

}

# Задачи на множества

**6.1.** Создайте множество, в качестве элементов которого используйте целое, вещественное и комплексное числа, а также строки в формате кортежа, списка и словаря, содержащих по одному элементу. Выведите полученное множество на экран.

**6.2.** Дано множество {'a', 'b', 'c'}. Удалите из него строку 'c', а затем добавьте элемент 'd'. Выведите итоговое множество на экран.

**6.3.** Даны множества {'a', 'b'} и {'b', 'c'}. Множества простые, поэтому сперва назовите результат их объединения, пересечения, разности и симметричной разницы устно, а затем проверьте себя, выведя результаты на экран.

**6.4.** Решите предыдущую задачу, использовав для получения результатов соответствующие методы класса set.

**6.5.** Дано множество целых чисел {-3, 8, 15, -5, 0, 7}. Выведите на экран произведение максимального и минимального элементов, сумму элементов не превышающих 7.

**6.6.** Определите есть ли в строке 'Купить, нельзя оставить!' хотя бы одна точка или запятая. Выведите результат проверки на экран.

**6.7.** Даны три слова 'аквариум', 'мармелад' и 'рама'. Выведите на экран сперва все виды букв, которые присутствуют во всех словах сразу, а затем все виды букв, которые присутствуют в любом из них.

**6.8.** Даны три числа: 30, 5 и 25. Выведите на экран наименьшее из них, использовав для этого программную проверку.

**6.9.** Даны два списка ников ['ivan2022', 'sveta', 'ivan', 'kot23', 'sveta', 'ivan'] и ['koly', 'enot37', 'luzer', 'kot23', 'sveta']. Сколько уникальных ников представлено в обоих списках? В каком списке уникальных ников содержится больше? Выведите результаты на экран.

**6.10.** Сформируйте и выведите на экран множество из всех подходящих для этого элементов списка [True, 0, 1, False, 2+3.4j, 5, '5', (6, 7), [8, 9], {'10': 10}]. Объясните конечный результат. Используйте для решения задачи генератор множеств и класс Hashable модуля collections.abc для проверки хешируемости элементов списка.

# Задачи на линейные алгоритмы

**7.1.** Перевести граммы в килограммы и вывести результат на экран. Значение граммов ввести с клавиатуры.

**7.2.** Даны две переменные x = 10 и y = 55. Поменяйте их значения местами. Выведите значения переменных на экран до и после замены.

**7.3.** С клавиатуры вводится расстояние L в метрах. Необходимо найти и вывести на экран количество полных километров в нем.

**7.4.** Вычислить и вывести на экран длину окружности и площадь круга одного и того же заданного радиуса R, который необходимо ввести с клавиатуры в сантиметрах. Результаты должны округляться до сотых.

**7.5.** С клавиатуры вводится некоторое число байт. Определите количество байт, которое нужно к нему добавить, чтобы получить целое число килобайт и мегабайт. Выведите результаты на экран.

**7.6.** Вычислить и вывести на экран период колебания маятника длиной L с точностью до сотых. Для рассчетов использовать формулу T = 2π√(L/g), где g – ускорение свободного падения (9.81 м/c2). Значение длины маятника в метрах необходимо ввести с клавиатуры.

**7.8.** Найдите количество секунд, прошедших с начала суток, если на вход сценария время поступило в виде строки '07:45:33'. Выведите результат на экран в виде сообщения «С начала суток прошло n сек.».

**7.9.** С клавиатуры вводится целое число. Необходимо вывести число, обратное введенному по порядку составляющих его цифр. Например, если было введено число 12345, программа должна вывести пользователю на экран число 54321.

**7.10.** Получите и преобразуйте текущую системную дату, возвращаемую методом date.today() модуля стандартной библиотеки datetime, из формата «год-месяц-день» в формат «день.месяц.год». Выведите оба формата даты на экран.

# Задачи на логические выражения

Не забываем, что в отличие от некоторых других языков программирования в Python логические константы True и False являются регистрозависимыми. Кроме того, к логическому типу автоматически приводятся результаты выражений в заголовках инструкций if и while. В результате мы можем использовать более короткие схемы вместо полноценных логических выражений. Например, заголовок «while len(li) > 0:» можно заменить на «while li:», т.к. после опустошения списка, он автоматически преобразуется в False и цикл прерывается.

**8.1.** Записать и вывести на экран условие, которое является истинным, когда положительное целое n кратно двум или трем.

**8.2.** Записать и вывести на экран условие, которое является истинным, когда целое n не кратно трем и оканчивается нулем.

**8.3.** Записать и вывести на экран условие, которое является истинным, когда список li имеет хотя бы один элемент.

**8.4.** Записать и вывести на экран условие, которое является истинным, когда хотя бы одно из чисел x, y и z больше 80.

**8.5.** Записать и вывести на экран условие, которое является истинным, когда оба числа a и b одновременно положительны или отрицательны.

**8.6.** Записать и вывести на экран условие, которое является истинным для пустого словаря d.

**8.7.** Записать и вывести на экран условие, которое является истинным, когда строка s является ключом словаря D, но отсутствует в списке строк L.

**8.8.** Записать и вывести на экран логическое выражение, описывающее область определения функции у = √a + 1/(a - 7) + 1/(b + 5) + 1/(b - 2).

**8.9.** Записать и вывести на экран условие, которое является истинным, только когда одно из чисел x, y и z меньше нуля.

**8.10.** Записать и вывести на экран условие, которое является истинным, когда переменная принадлежит к классу целых или вещественных чисел.

**8.11.** Известно, что обрабатываемая строка может содержать только целые числа, например, '-32', '59'. Запишите и выведите на экран условие, которое является истинным только для строк, содержащих двузначное число.

**8.12.** Запишите и выведите на экран условие, которое является истинным только для положительных целых чисел, которые делятся сразу и на два, и на четыре. При этом используйте не прямое деление, которое для больших чисел даст значительный отрицательный эффект по производительности, а следующие правила математики:

* целое положительное число делится на два, если оно четное;
* целое положительное число делится на четыре, если число, образованное последними двумя цифрами этого числа делится на четыре.

Показать решение.

# Задачи на if/else

**9.1.** Определите, не является ли введенная пользователем с клавиатуры строка пустой. Результат проверки в виде True или False выведите на экран.

**9.2.** Написать программу, которая будет делить введенные пользователем два вещественных числа и выводить результат на экран, сообщая об ошибке в случае деления на ноль.

**9.3.** Определить, является ли введенное с клавиатуры натуральное число кратным трем. Вывести результат на экран.

**9.4.** Рассчитать стоимость покупки с учетом скидки в 35%, которая предоставляется покупателю, если сумма покупки превышает 20 у.е. Сумму покупки ввести с клавиатуры, а результаты округлить до сотых (копейки, центы и т.д.). Вывести на экран итоговую стоимость и размер предоставленной скидки.

**9.5.** Напишите программу, которая будет проверять, является ли введенное пользователем значение цифрой десятичной системы счисления. Результат проверки должен выводиться пользователю на экран в виде сообщения.

**9.6.** Напишите программу, которая будет проверять, является ли введенное пользователем значение цифрой десятичной системы счисления. В случае положительного результата, на экран пользователя должно выводиться название цифры, например, «3 - это три». В противном случае должно выводиться предупреждение «Введите цифру 10-й СС».

**9.7.** Напишите скрипт, который будет проверять пользователя на умение перемножать однозначное натуральное число на двузначное, запрашивая ввод ответа с клавиатуры. Результат проверки ответа пользователя должен выводиться на экран в виде соответствующего сообщения. В случае неправильного ответа, программа должна выводить сообщение об ошибке, дополнительно сообщая пользователю верный результат. Для получения случайных целых чисел из диапазона 2 - 99 импортируйте из модуля random стандартной библиотеки метод randrange.

**Условие**

Введите ответ: 2\*65 = 120

Вы ошиблись, т.к. 2\*65 = 130

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

Введите ответ: 2\*47 = 94

Верно! 2\*47 = 94

**9.8.** Известна денежная сумма. Необходимо разменять её купюрами по 100, 10 и монетой 2 руб., если это возможно. Сумма должна вводиться с клавиатуры, а результат расчета выводиться на экран в формате «Размен: 100xa, 10xb, 2xc» или «Извините, но размен невозможен!».

**Условие**

Введите сумму в рублях: 156

Размен: 100x1, 10x5, 2x3

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

Введите сумму в рублях: 35

Извините, но размен невозможен!

**9.9.** Даны три числа: 130, 25 и 70. Выведите на экран наименьшее из них, использовав для этого программную проверку.

**9.10.** Определите четверть координатной плоскости, которой принадлежит точка A(x, y). Координаты точки должны вводиться с клавиатуры.

**9.11.** По длинам трех отрезков, вводимых пользователем с клавиатуры, определить возможность существования треугольника, составленного из этих отрезков. Если такой треугольник существует, то определить, является ли он разносторонним, равнобедренным или равносторонним.

**9.12.** Напишите скрипт, который по введенному пользователем числу от 1 до 12, будет выводить на экран сообщение в виде названия месяца и времени года. Если пользователь введет недопустимое число, программа должна выдавать сообщение об ошибке.

**9.13.** Напишите скрипт, который будет тестировать пользователя на знание синтаксиса условной инструкции if. На экран пользователю должны выводиться два случайных формата записи из списка ['if/elseif/else', 'if/elif/else', 'A = Y if X else Z'] и запрос на ввод верного варианта в виде «Введите 0, 1 или 2: ». После чего программа должна оценить ответ и в случае ошибки написать правильный. Для получения случайных вариантов из списка импортируйте из модуля random стандартной библиотеки метод randint.

**Условие**

Какой из двух форматов подходит для инструкции if: if/elseif/else или if/elif/else?

Укажите номер верного формата или введите ноль, если оба варианта не верны.

Введите 0, 1 или 2: 1

Ошибка! Верный вариант: if/elif/else.

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

Какой из двух форматов подходит для инструкции if: if/elseif/else или A = Y if X else Z?

Укажите номер верного формата или введите ноль, если оба варианта не верны.

Введите 0, 1 или 2: 0

Верно!

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

Какой из двух форматов подходит для инструкции if: if/elif/else или A = Y if X else Z?

Укажите номер верного формата или введите ноль, если оба варианта не верны.

Введите 0, 1 или 2: 1

Верно!

**9.14.** Сформируйте список li из 10-ти случайных натуральных чисел не превышающих 100. Для этого используйте инструкции from random import randint и for k in range(1, 11): li.append(randint(1, 100)). Далее, используя для обхода элементов списка цикл for, выведите на экран: максимальное значение четных чисел, если их меньше, чем нечетных; максимальное значение нечетных чисел, если их меньше, чем четных; максимальное значение из двух, если в списке содержится равное количество четных и нечетных чисел. Использовать для решения задачи другие встроенные функции запрещается!

**Условие**

li: [5, 20, 41, 63, 24, 21, 10, 62, 91, 19]

Четных чисел меньше.

Макс. четное: 62

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

li: [28, 58, 94, 59, 27, 30, 80, 18, 4, 53]

Нечетных чисел меньше.

Макс. нечетное: 59

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

li: [88, 32, 50, 99, 17, 12, 78, 95, 55, 47]

Чисел одинаковое кол-во.

Макс. из них нечетное: 99

# Задачи на циклы for и while

Все задачи должны решаться как с использованием цикла for, так и цикла while. Дополнительно разрешается использовать и другие инструменты для обхода наборов элементов.

**10.1.** Посчитайте количество символов в строке 'Python - это Питон!', использовав счетчики на основе циклов for и while.

**10.2.** Сколько кортежей содержится в списке [(1, 2), (3), (4,), (5+6), (7+8,)]? Проверьте свой ответ программно, используя циклы. Выведите кортежи на экран в одну строку.

**10.3.** Найдите сумму всех элементов списка [1, '2', 3, 4, '5'], предварительно приводя строки к целым числам.

**10.4.** Дано множество целых чисел {12, 33, 24, 7, 19}. Выведите на экран только четные числа.

**10.5.** Используя циклы, проверьте при помощи оператора in наличие символов строки 'abcde123' в строке 'bad\_cat\_23', выводя результаты проверки на экран в виде «Символ "a" есть в "bad\_cat\_23".» или «Символа "n" нет в "bad\_cat\_23".».

**10.6.** Вычислить и вывести на экран сумму кубов натуральных чисел от 1 до n включительно. Верхний предел должен вводиться с клавиатуры и не должен превышать числа 100.

**10.7.** Вычислить факториал заданного натурального числа. Факториал числа n вычисляется по формуле: n! = 1\*2\*3\*...\*n. Число n должно вводиться с клавиатуры и не должно превышать числа 30.

**10.8.** Подсчитайте количество арабских цифр и алфавитных символов Юникода в строке 'Станция «Альфа-2015»' и выведите результаты на экран.

**10.9.** Даны два списка строк ['1', 'True', 'ok'] и ['no', 'True', 'no']. Сформируйте новый список, который будет состоять из строк, которые присутствуют хотя бы в одном из списков. Выведите полученный список на экран.

**10.10.** Выведите на экран через пробел все символы строки 'abcdefABCDEF', которые находятся в нижнем регистре, не используя при этом срез строки.

**10.11.** Cгенерируйте и выведите на экран мозаичное изображение гексагональной сетки, напоминающее мелкоячеистую проволочную сетку.

**Условие**

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_

\\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/ \\_/

**10.12.** Выведите на экран в порядке возрастания и через пробел все цифры вложенных списков конструкции [[[1, 2, 3], [4, 5, 6], [7, 8, 9]], [[10, 11, 12], [13, 14, 15], [16, 17, 18]], [[19, 20, 21], [22, 23, 24], [25, 26, 27]]]. Не забываем делать это при помощи циклов for и while (должны получиться тройные матрешки из вложенных циклов).

**10.13.** Выведите на экран цифры вводимого с клавиатуры натурального числа, пропуская цифры 2 и 5.

**10.14.** Дан словарь {1: [1, 2, 3, 4], 2: [5, 6, 7, 8]}. Удалите из вложенных списков все четные значения элементов. Выведите измененный словарь на экран.

**10.15.** Выведите на экран таблицу умножения чисел от одного до девяти.

**Условие**

----+-------------------------------------

| х | 1 2 3 4 5 6 7 8 9 |

----+-------------------------------------

| 1 | 1 2 3 4 5 6 7 8 9 |

| 2 | 2 4 6 8 10 12 14 16 18 |

| 3 | 3 6 9 12 15 18 21 24 27 |

| 4 | 4 8 12 16 20 24 28 32 36 |

| 5 | 5 10 15 20 25 30 35 40 45 |

| 6 | 6 12 18 24 30 36 42 48 54 |

| 7 | 7 14 21 28 35 42 49 56 63 |

| 8 | 8 16 24 32 40 48 56 64 72 |

| 9 | 9 18 27 36 45 54 63 72 81 |

----+-------------------------------------

**10.16.** Напишите программу вывода на экран всех простых чисел, не превосходящих заданного натурального числа n. Простым называется натуральное число большее единицы, которое делится только на единицу и само себя. Если у натурального числа есть делители большие единицы, то все они не превышают корня квадратного из этого числа.

# Задачи на протокол итераций и генераторы

**11.1.** Имеется список вещественных чисел [1.3, 2.5, 3.1]. Создайте множество квадратов этих чисел, использовав для этого генератор множеств. Выведите полученное множество на экран.

**11.2.** Создайте список нечетных чисел, принадлежащих отрезку [10; 20], использовав для этого генератор списков. Выведите полученный список на экран.

**11.3.** Преобразуйте кортеж чисел (3.27, 5.755, 7.321) в словарь, в качестве ключей которого будут использованы числа кортежа, а значениями будут их округленные до десятых значения. Используйте для решения задачи генератор словарей. Выведите полученный словарь на экран.

**11.4.** Выведите на экран элементы списка [1, 2, 3], использовав для этого функцию next().

**11.5.** С помощью генератора-выражений получите итератор для последовательности четных чисел из диапазона от 1 до 20 включительно. В ручном режиме выведите первые три числа последовательности, использовав метод \_\_next\_\_ итератора.

**11.6.** Напишите функцию-генератор gen\_func(n), которая при вызове будет возвращать итератор для последовательности четных чисел из диапазона от 1 до n включительно. Вызовите функцию, передав аргументу значение 20, и в ручном режиме выведите первые три числа сгенерированной последовательности, использовав метод \_\_next\_\_ итератора.

**11.7.** Выведите на экран сумму числовых кодов символов строки 'генераторы', использовав в решении генератор списков.

**11.8.** Дан список [(1, 2), (3+4), (5,), (6+7, 8), (9+10,)]. Оставьте в нем только элементы, которые являются кортежами и содержат более одного элемента. Используйте для решения задачи генератор списков. Выведите итоговый и результирующий списки на экран.

**11.9.** Сформируйте множество из всех подходящих для этого элементов списка [3, 'ok', [1, 2], (True, False), {'flag': 1}] и выведите его на экран. Используйте для решения задачи генератор множеств и класс Hashable модуля collections.abc для проверки хешируемости элементов списка.

**11.10.** Сформируйте словарь, в котором ключами будут строковые значения чисел из диапазона целых чисел от 20 до 30 кратные трем, но не кратные пяти, а значениями квадраты этих чисел. Используйте для решения задачи генератор словарей.

**11.11.** Напишите скрипт, который будет находить сумму цифр вводимого пользователем целого числа. Проверьте его на числах -9720, 0 и +3531. Используйте для решения задачи генератор выражения.

**11.12.** Напишите функцию-генератор gen\_fib(n: 'int>0') для генерирования последовательности из n чисел Фибоначчи. Вместо проверки аргумента просто укажите аннотацию, сообщающую о том, что аргумент должен быть целым неотрицательным числом. Используя созданный генератор, выведите на экран 10-е число Фибоначчи, а также сумму первых 10-ти чисел последовательности.

Числа Фибоначчи представляют собой ряд чисел, в котором первое число равно нулю, второе единице, а каждое последующее число равно сумме двух предыдущих, т.е. 0 1 1 2 3 5 8 и т.д.

# Задачи на функции

**12.1.** Создайте простейшую в мире функцию simple\_pass, а затем выведите на экран ее тип.

**12.2.** Напишите функцию hello\_friend, принимающую в качестве аргументов имя и фамилию пользователя и возвращающую строку в формате «Привет, {name} {surname}!». Вызовите функцию и выведите результат вызова на экран.

**12.3.** Создайте еще одну версию функции hello\_friend из предыдущего задания, также принимающую в качестве аргументов имя и фамилию пользователя и возвращающую строку в формате «Привет, {name} {surname}!». Но на этот раз функция должна принимать только именованные аргументы со значениями по умолчанию. Вызовите функцию без аргументов и с аргументами, попутно выводя результаты вызовов на экран.

**12.4.** Создайте пользовательскую функцию print\_args, принимающую произвольное количество аргументов и выводящую их затем на экран. Вызовите функцию, одновременно передав ей в качестве значений целое число, вещественное число, строку и пустой список.

**12.5.** Создайте пользовательскую функцию print\_named\_args, принимающую произвольное количество именованных аргументов и выводящую их затем на экран в формате «key -> value». Вызовите функцию, передав ей в качестве значений комплексное число, кортеж с логическими значениями и пустое множество.

**12.6.** Создайте и вызовите пользовательскую функцию-матрешку my\_func\_1, состоящую из четырех вложенных друг в друга определений аналогичных функций. Каждая функция должна выводить сообщение в формате 'In my\_func\_{номер функции}.', а также содержать определение и вызов следующей вложенной функции (в последней функции эта часть будет отсутствовать).

**Условие**

In my\_func\_1.

In my\_func\_2.

In my\_func\_3.

In my\_func\_4.

In my\_func\_5.

**12.7.** Создайте функцию, принимающую в качестве аргументов два числа x и y=1 и возвращающую частное d этих чисел в виде строки 'x/y = d'. Снабдите функцию аннотациями: float для x, 'y != 0' для y и 'x/y' для возвращаемого функцией результата. Добавьте в функцию строку документирования «Возвращает строку вида 'x/y = d'.». Выведите на экран результаты вызовов func(5, 0) и func(5, 10), документацию функции через соответствующий атрибут, а также все аннотации, использовав для этого цикл for.

**12.8.** Напишите рекурсивную функцию recursive\_func(text), которая будет выводить символы передаваемой ей строки на экран через пробел.

**12.9.** Напишите рекурсивную функцию, которая будет вычислять сумму цифр натурального числа. Используя полученную функцию, выведите на экран суммы цифр чисел 7 и 39574.

**12.10.** Напишите собственный аналог встроенной функции abs для нахождения модуля числа. Проверьте ее на числах -5, 0 и 10.

**12.11.** Напишите функцию, рассчитывающую стоимость поездки на такси в зависимости от расстояния. В качестве аргументов функция должна принимать именованные параметры: км – расстояние поездки в километрах, мин\_цена – базовый тариф, включающий подачу такси и первые три километра пути, цена\_за\_км – цена за каждый километр, начиная с четвертого. Рассчитайте и выведите на экран стоимость поездки за 17.5 км, если по умолчанию базовый тариф составляет 2 у.е., а цена за километр сверх минимума – 0.3 у.е.

**12.12.** Напишите функцию, которая будет генерировать случайный пароль. В пароле должно быть от 8 до 15 символов Юникода из диапазонов 48-57 (цифры), 65-90 (буквы латинского алфавита в верхнем регистре) и 97-122 (буквы латинского алфавита в нижнем регистре). Сгенерируйте и выведите на экран три пароля.

**12.13.** Напишите простейшую функцию-калькулятор, которая будет выполнять с двумя переданными ей числами следующие действия: сложение, вычитание, умножение и деление с указанной точностью. Соответственно, функция должна принимать два позиционных числовых аргумента для чисел, один именованный строковый аргумент op (по умолчанию функция должна выполнять операцию сложения) и один именованный числовой аргумент prec для требуемой точности результата (по умолчанию три знака после запятой). В случае деления на ноль функция должна возвращать NaN, а при попытке выполнения непредусмотренной операции – сообщение об ошибке «Неподдерживаемый тип операции!». Выведите на экран результаты вызовов функции с точностью до сотых для числовых выражений: -13.756 + 59.291, 599 - 783, -7/55, 7/0 и 57.75\*33.2.

**12.14.** В григорианском календаре, который был введен в 1582 году папой римским Григорием XIII обычный год состоит из 365 дней. Но т.к. период обращения Земли вокруг Солнца составляет 365.2425 дней, для устранения накапливающейся ошибки было решено ввести в календарь дополнительные «високосные года» в 366 дней. Таковыми условились считать года кратные 4. Однако этого оказалось недостаточно, поэтому для сведения ошибки к минимуму в григорианский календарь ввели дополнительную коррекцию: каждый год, который делится без остатка на 100, является високосным только тогда, когда он делится еще и на 400. Например, високосным считается 1760 год, т.к. он делится на четыре, но год 1700 високосным не является, т.к. он делится на четыре и сто, но не делится на четыреста. Зная все эти нюансы, напишите функцию, которая будет определять, является ли переданный ей год високосным. Проверьте и выведите на экран годы 1782, 1900, 2000 и 2023.

**12.15.** Напишите функцию для определения длины гипотенузы прямоугольного треугольника с помощью теоремы Пифагора. В качестве аргументов функция должна принимать длины катетов в виде двух положительных чисел. Возвращаемым значением должна быть длина гипотенузы, округленная с точностью до сотых. В случае передачи аргументов недопустимого типа или диапазона функция должна возвращать -1. Выведите на экран результаты вызовов функции для положительных и отрицательных значений аргументов.

**12.16.** Используя функцию import\_module модуля [importlib](https://okpython.net/python/python_spravochnik/python_spravochnik.html#importlib) стандартной библиотеки напишите функцию imp\_mods(mods), импортирующую в глобальную область видимости текущего модуля необходимые пакеты и модули. В качестве аргумента функция должна принимать кортеж или список имен пакетов и модулей для импорта. Импортируйте при помощи созданной функции модули random и math, а затем выведите на экран факториал случайного положительного целого числа не превышающего десяти.

**12.17.** Напишите функцию для нахождения НОД произвольного количества чисел. Затем найдите и выведите на экран наибольший общий делитель чисел 165, 435 и 300.

**12.18.** Напишите функцию, которая будет тестировать скорость выполнения другой функции за указанное количество итераций. В качестве именованных аргументов функция должна принимать: f – объект другой функции, args=[] – список или кортеж аргументов тестируемой функции (по умолчанию без аргументов), i=1000 – количество итераций тестирования (по умолчанию тысяча итераций). На выходе функция должна возвращать список [<имя функции>, <количество проведенных итераций>, <затраченное время>]. Протестируйте скорость выполнения встроенных функций sum и min за 20000 итераций на списке целых чисел из диапазона от 1 до 10000. Выведите результаты тестирования на экран, как показано в условии.

**Условие**

Функция: sum

Кол-во итераций: 20000

Время выполнения: 1.2101598

Функция: min

Кол-во итераций: 20000

Время выполнения: 1.6920748

# Задачи на ООП: классы и объекты

**13.1.** Создайте простейший в мире класс SimplePass. Затем создайте экземпляр класса и выведите на экран его тип.

**13.2.** Определите класс A, включающий:

* строку документирования класса '''Класс A''';
* метод set\_a() для установки значения атрибута a;
* метод get\_a() для получения значения этого атрибута.

Выведите на экран документацию класса. Затем создайте первый экземпляр класса и при помощи определенных методов установите и выведите на экран значение его атрибута a. Далее создайте второй экземпляр класса, после чего также установите и выведите на экран значение атрибута a, но уже при помощи прямого доступа к атрибуту по точке.

**13.3.** Определите класс B, включающий:

* строку документирования класса '''Класс B''';
* конструктор, инициализирующий атрибут данных b создаваемых экземпляров;
* метод get\_b() для получения значения этого атрибута.

Выведите на экран документацию класса. Затем создайте экземпляр класса obj и при помощи метода экземпляра выведите на экран значение его атрибута b.

**13.4.** Определите класс C, наследующий классы A (задача №2) и B (задача №3) и включающий:

* строку документирования класса '''Класс C = A + B''';
* конструктор, инициализирующий дополнительно атрибуты данных a и c создаваемых экземпляров;
* собственные методы set\_b() и set\_c() для установки значений соответствующих атрибутов;
* собственный метод get\_c() для получения значения атрибута c.

Выведите на экран документацию класса. Затем создайте экземпляр класса obj, после чего при помощи соответствующих методов экземпляра выведите на экран значения его атрибутов a, b и c.

**13.5.** Определите класс D, включающий:

* статический метод stat\_print\_dict, выводящий на экран словарь атрибутов переданного ему объекта класса;
* метод класса cls\_print\_dict, выводящий на экран словарь атрибутов своего класса.

Создайте экземпляр класса obj и, вызвав оба метода из этого экземпляра, выведите на экран словарь атрибутов класса D. Объясните различие в использовании методов.

**13.6.** Определите класс E, наследующий класс D (задача №5) и включающий единственный атрибут данных класса e = 'Класс E'. Создайте экземпляр obj\_1 класса D и, вызвав оба метода из этого экземпляра, выведите на экран словарь атрибутов класса. Затем создайте экземпляр obj\_2 класса E и также, вызвав оба метода из этого экземпляра, выведите на экран словарь атрибутов этого класса. Объясните результаты.

**13.7.** Определите класс F, наследующий класс A (задача №2), включающий:

* конструктор, обновляющий строку документации создаваемых экземпляров на 'Объект класса F';
* расширенный метод set\_a() для установки значения атрибута a, который должен дополнительно выводить сообщение 'Атрибут a установлен!'.

Выведите на экран документацию класса. Затем создайте экземпляр класса obj, после чего выведите его документацию и далее, при помощи соответствующих методов экземпляра, установите и выведите на экран значение его атрибута a. При расширении метода set\_a() используйте по-очереди три варианта синтаксиса для доступа к методу суперкласса (незадействованные инструкции закомментируйте).

**13.8.** Определите класс PiNum, хранящий значение числа Пи и включающий:

* конструктор, инициализирующий текущую точность представления числа Пи создаваемым экземпляром (по умолчанию два знака после запятой);
* переопределяемый магический метод \_\_str\_\_, возвращающий строку с текущим значением числа Пи;
* управляемый атрибут max\_pi, хранящий значение числа Пи с максимальной точностью в 13 знаков после запятой и недоступный для изменения или удаления;
* метод set\_pi\_prec, устанавливающий значение атрибута cur\_pi экземпляра для хранения значения числа Пи с текущей точностью (по умолчанию два знака после запятой).

Используя созданный класс, создайте экземпляр числа Пи с точностью в три знака после запятой и выведите его строковое представление на экран. Измените точность представления числа до пяти знаков после запятой и выведите новое значение на экран. Также выведите на экран значение числа Пи с максимальной точностью.

**13.9.** Определите класс Circle, представляющий окружность и включающий:

* статический метод, переводящий метры в сантиметры или наоборот;
* конструктор, инициализирующий радиус r экземпляра и атрибут pi для хранения числа Пи с точностью в три знака после запятой (для получения требуемого значения используйте класс PiNum из предыдущей задачи);
* методы экземпляров для получения длины и площади окружности с точностью в три знака после запятой.

Используя созданный класс, рассчитайте и выведите на экран длину и площадь окружности в сантиметрах зная, что ее радиус равен 2.57 метра.

**13.10.** Определите суперкласс Сотрудник, включающий:

* конструктор, инициализирующий имя работника, его должность (по умолчанию None) и оклад (по умолчанию 0);
* метод экземпляра для повышения оклада на какую-то часть (например, на 0.3, т.е. на 30%) с округлением результата до копеек;
* магический метод \_\_str\_\_ для перегрузки строкового представления объекта, который должен выводить данные о работнике в формате 'Атрибут: объект.атрибут' по одной записи на каждой строке.

Также определите подкласс Менеджер, наследующий суперкласс Сотрудник и переопре­деляющий метод повышения оклада таким образом, чтобы он еще больше повышал оклад за счет дополнительного бонуса в виде какой-то части оклада. Далее:

* создайте экземпляр иван\_менеджер созданного подкласса с начальным окладом в 1700 рублей;
* повысьте сотруднику оклад за счет стандартной надбавки в 0.335 и бонуса за должность в 0.25;
* выведите строковое представление объекта экземпляра с информацией о сотруднике на экран.

Показать решение.

**13.11.** Создайте абстрактный класс геометрической фигуры Shape с конструктором, принимающим длину стороны и высоту, проведенную к этой стороне. Определите в классе абстрактный метод area(), который будет использоваться подклассами для расчета площади соответствующих им геометрических фигур. Далее создайте классы Triangle и Rectangle, наследующие суперкласс Shape и реализующие его абстрактный метод под свои нужды. Продемонстрируйте использование созданных классов для нахождения площади треугольника и прямоугольника по известным значениям длины стороны и высоты, проведенной к этой стороне.

**13.12.** Определите класс Counter, реализующий десятичный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне, включая границы диапазона. В классе должны быть предусмотрены следующие возможности:

* конструктор для инициализации счетчика значениями по умолчанию (стартовое значение, нижняя и верхняя границы диапазона),
* метод для его инициализации произвольными значениями,
* а также методы для увеличения и уменьшения текущего значения счетчика.

Все методы класса должны принимать только именованные параметры и проверять выход текущего значения счетчика за допустимый диапазон. Создайте экземпляр счетчика со значениями по умолчанию и выведите на экран его начальные параметры. Далее проверьте его работу циклом в пределах диапазона, увеличивая и выводя на экран его текущее значение от минимально возможного до максимального. Затем переустановите счетчик, задав отрицательную нижнюю и положительную верхнюю границы, а также установив положительное стартовое значение для отсчета. Опять же, проверьте его работу циклом, уменьшая и выводя на экран его текущее значение от стартового до минимально возможного. Задайте заведомо большее количество итераций циклов в обоих случаях, обеспечив прерывание их работы при попытке выхода счетчика за пределы диапазона.

# Задачи на импорт, модули и пакеты

**14.1.** Импортируйте из стандартной библиотеки модуль [math](https://okpython.net/python/python_spravochnik/python_spravochnik.html#math). Используя необходимые атрибуты модуля выведите на экран результаты вычисления выражений: cosπ (см. косинус и число Пи), √e (см. число e) и 5! (см. факториал).

**14.2.** Выведите на экран текущее время. Для решения задачи импортируйте из модуля [datetime](https://okpython.net/python/python_spravochnik/python_spravochnik.html#datetime) стандартной библиотеки класс datetime. Далее используйте цепочку datetime.now().time(), которая как раз и вернет текущее время.

**14.3.** Выведите на экран текущую версию интерпретатора в формате «Python major.micro.minor.». Для выполнения задания импортируйте атрибут version\_info модуля [sys](https://okpython.net/python/python_spravochnik/python_spravochnik.html#sys) стандартной библиотеки.

**14.4.** Импортируйте из модуля [random](https://okpython.net/python/python_spravochnik/python_spravochnik.html#random) стандартной библиотеки функции randint и choice, использовав для них псевдонимы rnd и chc. Создайте список li и заполните его пятью случайными целыми числами от 1 до 20. Выведите на экран сам список, а также случайный элемент списка.

**14.5.** Выполните следующее упражнение:

* создайте вручную корневой каталог проекта project с исполняемым файлом main.py;
* в полученном каталоге создайте два пакета pkg\_1 и pkg\_2, добавив в них соответст­вующие модули mod\_1.py с инструкцией x = 1 и mod\_2.py с инструкцией x = 2;
* используя главный файл проекта выведите на экран сумму значений переменных модулей.

Показать решение.

**14.6.** Выполните следующее упражнение:

* создайте вручную корневой каталог проекта project с исполняемым файлом main.py;
* в полученном каталоге создайте два пакета pkg\_1 и pkg\_2, добавив в них соответст­вующие модули mod\_1.py с инструкцией x = 1 и mod\_2.py с инструкцией y = 2;
* в пакет pkg\_2 добавьте подпакет pkg\_3, содержащий модуль mod\_3.py с инструкцией z\_2 = z\_1 = 3;
* добавьте в модуль mod\_1.py инструкцию импорта переменной z\_1, а в модуль mod\_2.py инструкцию импорта переменной z\_2;
* используя главный файл проекта выведите на экран сумму значений всех переменных модулей mod\_1.py и mod\_2.py.

Показать решение.

**14.7.** Используя функцию import\_module модуля [importlib](https://okpython.net/python/python_spravochnik/python_spravochnik.html#importlib) стандартной библиотеки напишите функцию imp\_mods(mods), импортирующую в глобальную область видимости текущего модуля необходимые пакеты и модули. В качестве аргумента функция должна принимать кортеж или список имен пакетов и модулей для импорта. Импортируйте при помощи созданной функции модули random и math, а затем выведите на экран факториал случайного положительного целого числа не превышающего десяти.

# Задачи на файлы и каталоги

**15.1.** Используя встроенную функцию open(), создайте в каталоге скрипта файл text\_1.txt и запишите туда строку 'Пайтон практика: упражнения и задачи.', не забыв закрыть файл после операции. Выведите содержимое записанного файла на экран.

**15.2.** Используя соответствующую встроенную функцию, откройте в каталоге скрипта файл text\_2.txt в режиме записи и чтения, запишите туда строку '0123456789', после чего считайте записанное содержимое и выведите его на экран, не забыв закрыть файл после операции. Далее, откройте файл в режиме дозаписи и чтения, допишите туда строку 'abcdef' и также выведите обновленное содержимое файла на экран.

**15.3.** Используя соответствующую встроенную функцию создайте в каталоге скрипта файл li.py и запишите туда строку li\_1 = [1, 2]. Далее, используя менеджер контекста with/as, допишите в файл строку li\_2 = ['red', 'green'] и, не выходя за область видимости менеджера, выведите на экран содержимое файла используя для чтения цикл for.

**15.4.** Используя менеджер контекста и соответствующую встроенную функцию, создайте в каталоге скрипта файл mod.py и запишите туда инструкцию определения функции sum\_func, принимающую два числа и возвращающую их сумму. Используя вновь созданный модуль выведите на экран сумму чисел 10 и 20. Затем допишите в файл инструкцию определения функции mul\_func, принимающую два числа и возвращающую их произведение. Используя обновленный модуль выведите на экран произведение чисел 10 и 20.

**15.5.** Используя возможности модуля pathlib cоздайте в каталоге скрипта каталог test\_dir, а в нем пустой файл test\_file.txt. Запишите в файл строку 'Python Ok или Ok Python.', после чего выведите содержимое файла на экран. Удалите каталог и файл. Результаты всех операций должны отображаться на экране, как показано в кодовом блоке ниже.

**Условие**

Каталог и файл созданы!

Нажмите любую клавишу...

Python Ok или Ok Python?

Файл записан и прочитан!

Нажмите любую клавишу...

Все удалено!

Нажмите любую клавишу...

**15.6.** Напишите скрипт, в котором объявите три функции:

* create\_db(db\_name) – должна принимать строковый аргумент и создавать пустое хранилище db\_name, если оно еще не существует.
* set\_amount(db\_name, num, money) – должна проверять, существует ли в хранилище счет num (строка), после чего либо создавать и пополнять его с выводом сообщения «Счет №{num} создан, баланс пополнен!», либо обновлять баланс существующего счета с выводом сообщения «Баланс счета №{num} обновлен!».
* get\_amount(db\_name, num) – должна выводить текущий баланс счета num (строка) на экран в форме «Текущий баланс счета №{num}: {money} руб.».

После объявления функций создайте хранилище my\_bank. Создайте в нем банковский счет 3335, пополнив его баланс на 5000 рублей, после чего выведите баланс на экран. Проделайте тоже самое для счета 3336, положив на счет сперва 300 рублей, а потом доложив еще 700 рублей.

Для решения задачи используйте модуль shelve, а также класс Path модуля pathlib, которые импортируйте в глобальное пространство скрипта.

**Условие**

Счет N3335 создан, баланс пополнен!

Текущий баланс счета N3335: 5000 руб.

Счет N3336 создан, баланс пополнен!

Текущий баланс счета N3336: 300 руб.

Баланс счета N3336 обновлен!

Текущий баланс счета N3336: 1000 руб.

**15.7.** Выполните следующее упражнение:

* создайте в папке скрипта корневой каталог упражнения root\_dir;
* в полученном корневом каталоге создайте еще четыре каталога dir\_1, dir\_2, dir\_3, и dir\_4, добавив в них соответст­вующие пустые файлы f\_1.txt, f\_2.txt, f\_3.txt и f\_4.txt (не нужно создавать все вручную, используйте цикл);
* выведите сообщение 'Все каталоги и файлы созданы!' и призыв к действию 'Нажмите любую клавишу...\n';
* переместите каталог dir\_2 в dir\_1 и переименуйте файл f\_2.txt в f\_1\_2.txt;
* переместите каталог dir\_4 в dir\_3 и переименуйте файл f\_4.txt в f\_3\_4.txt;
* выведите сообщение 'Все перемещено и переименовано!' и призыв к действию 'Нажмите любую клавишу...\n';
* удалите каталог dir\_3 и файл f\_1.txt
* выведите сообщение 'dir\_3 удален, f\_1.txt удален!' и призыв к действию 'Нажмите любую клавишу...\n';
* поместите каталог dir\_1 в одноименный архив в том же каталоге (т.е. в root\_dir), а сам каталог dir\_1 удалите;
* выведите сообщение 'dir\_1 заархивирован, сам dir\_1 удален!' и призыв к действию 'Нажмите любую клавишу...\n';
* распакуйте архив обратно в каталог root\_dir, сам архив удалите (в root\_dir должен остаться только распакованный каталог dir\_1);
* выведите сообщение 'dir\_1 разархивирован, сам архив удален!' и призыв к действию 'Нажмите любую клавишу...'.

Для выполнения задания используйте функции модуля shutil и методы класса pathlib.Path. Выполняйте упражнение поэтапно, при необходимости удаляя корневой каталог упражнения воизбежание ошибок.

**15.8.** Напишите функцию mk\_pkg(path, root\_path='./'), создающую по переданному в виде строки относительному пути path готовый python-пакет (т.е. каталог с файлом инициализации \_\_init\_\_.py). В случае отсутствия в пути каких-либо промежуточных пакетов или наличии каталогов без файлов инициализации, функция должна создавать их. Необязательный аргумент root\_path='./' должен принимать строку с каталогом, относительно которого будет отсчитываться местоположение целевого каталога (по умолчанию в качестве локального корня должен использоваться каталог скрипта). Когда функция будет готова, проверьте ее, предварительно создав в каталоге скрипта обычный каталог pkg\_1, а затем осуществив два вызова функции: mk\_pkg('pkg\_1/pkg\_2') и mk\_pkg('pkg\_1/pkg\_3/pkg\_4').

# Задачи на исключения и ошибки

**16.1.** Напишите пользовательскую функцию pow3\_func(x), которая должна принимать в качестве аргумента число x и возводить его в куб. В случае успешных вычислений функция должна выводить на экран сообщение в формате «x^3 = res» и возвращать res. В противном случае должно выводиться соответствующее сообщение об ошибке и возвращаться None. Проверьте функцию, вызвав ее с аргументами 5 и '5'.

**16.2.** Напишите пользовательскую функцию write\_file(f\_obj, s), которая должна принимать в качестве аргумента объект открытого файла f\_obj и пытаться записать в него строку s. В случае успешного завершения операции функция должна выводить на экран сообщение «Файл записан!». В противном случае должно выводиться соответствующее сообщение об ошибке. При любом исходе функция должна закрывать файловый объект перед завершением вызова и выводить сообщение «Файл закрыт!». Проверьте работу функции. Для этого создайте в папке скрипта пустой файл nums.py и запишите туда строку '1 2 3'. Далее откройте файл на чтение и попытайтесь записать в него строку '4 5 6'. Для решения задачи используйте в теле функции полную инструкцию try/except/else/finally.

**16.3.** Напишите пользовательскую функцию avg(x, y), которая должна принимать в качестве аргументов два положительных целых или вещественных числа, а возвращать их среднее геометрическое (z = pow(x\*y, 0.5)). В случае передачи недопустимых значений функция должна возбуждать исключение ValueError. После того, как функция будет готова, используйте ее в скрипте, который будет запрашивать у пользователя два числа, обрабатывать ввод и выдавать либо среднее геометрическое чисел, либо сообщение об ошибке. Для решения задачи используйте в теле функции инструкцию raise, а в скрипте try/except.

**16.4.** Не используя возможности модуля abc стандартной библиотеки, создайте абстрактный класс геометрической фигуры Shape с конструктором, принимающим длину стороны и высоту, проведенную к этой стороне. Определите в классе абстрактный метод area(), который будет использоваться подклассами для расчета площади соответствующих им геометрических фигур. Для реализации абстрактности метода используйте возбуждение исключения NotImplementedError. Далее создайте классы Triangle и Rectangle, наследующие суперкласс Shape и реализующие его абстрактный метод под свои нужды. Продемонстрируйте использование созданных классов для нахождения площади треугольника и прямоугольника по известным значениям длины стороны и высоты, проведенной к этой стороне. Закомментируйте в одном из классов переопределяемый метод area(), использовав вместо него инструкцию pass, и повторно запустите скрипт. Объясните результат. Еще раз посмотрите решение задачи 13.11.

# Использование утилиты pip и виртуальных окружений

**17.1.** Выполните в консоли своей операционной системы с помощью менеджера пакетов pip следующее упражнение:

* выведите текущую версию pip;
* выведите справку по нему;
* выведите список пакетов, установленных в текущем окружении;
* установите пакет cowsay;
* выведите информацию о пакете cowsay;
* повторно выведите список пакетов, установленных в окружении;
* удалите пакет cowsay;
* еще раз выведите список пакетов, установленных в окружении.

Если имеется такая возможность, выполните упражнение как в ОС Windows, так и в ОС Ubuntu.

**17.2.** Напишите класс PipManager, в котором определите следующие методы:

* show\_pip\_version() – вывод версии менеджера;
* show\_pip\_help() – вывод справки по менеджеру;
* install\_package(package\_name) – установка пакета;
* upgrade\_package(package\_name) – обновление пакета;
* show\_package\_info(package\_name) – вывод информации о пакете;
* show\_packages() – вывод списка установленных пакетов;
* uninstall\_package(package\_name) – удаление пакета.

Используя возможности созданного класса, дополните скрипт так, чтобы он смог выполнить предыдущее упражнение в автоматическом режиме.

**17.3.** Создайте вручную каталог проекта my\_prj. Откройте командную оболочку, перейдя в каталог проекта, и выполните в ней следующее упражнение:

* создайте виртуальное окружение в подкаталоге my\_venv проекта, использовав текущую версию Python;
* активируйте виртуальное окружение проекта;
* обновите менеджер пакетов pip окружения;
* выведите его версию;
* деактивируйте виртуальное окружение.

Если имеется такая возможность, выполните упражнение как в ОС Windows, так и в ОС Ubuntu.

**17.4.** Создайте вручную каталог проекта my\_prj (или очистите старый). Откройте командную оболочку, перейдя в каталог проекта, и выполните в ней следующее упражнение:

* создайте виртуальное окружение в подкаталоге venv\_2 проекта, использовав текущую версию Python;
* активируйте виртуальное окружение проекта;
* обновите менеджер пакетов pip окружения;
* установите пакет pillow;
* выведите список установленных пакетов;
* сохраните зависимости в файл requirements.txt в корневой каталог проекта;
* деактивируйте виртуальное окружение;
* удалите каталог виртуального окружения;
* заново создайте виртуальное окружение, но уже в подкаталоге venv\_3 проекта, использовав текущую версию Python;
* активируйте виртуальное окружение проекта;
* обновите менеджер пакетов pip окружения;
* восстановите все зависимости проекта из файла requirements.txt;
* выведите список установленных пакетов;
* деактивируйте виртуальное окружение.

Если имеется такая возможность, выполните упражнение как в ОС Windows, так и в ОС Ubuntu.

**17.5.** Создайте каталог проекта my\_prj (или очистите старый). В нем создайте файл main.py, в котором напишите скрипт вывода информации об используемой версии Python. Откройте командную оболочку, перейдя в каталог проекта, и выполните в ней следующее упражнение:

* запустите файл на исполнение;
* создайте виртуальное окружение в подкаталоге my\_venvs/venv\_1 проекта, использовав при этом версию интерпретатора Python3.9;
* активируйте виртуальное окружение;
* запустите файл на исполнение;
* деактивируйте виртуальное окружение;
* внесите изменения в код файла так, чтобы при запуске на исполнение мог задействоваться интерпретатор виртуального окружения проекта, а не системный;
* запустите файл на исполнение под интерпретатором виртуального окружения проекта.
* запустите файл на исполнение под системным интерпретатором.

Если имеется такая возможность, выполните упражнение как в ОС Windows, так и в ОС Ubuntu.

# Задачи на дату и время

**18.1.** Выведите на экран текущие дату и время, использовав для этого возможности модуля datetime стандартной библиотеки.

**18.2.** Выведите на экран в [ISO 8601](https://ru.wikipedia.org/wiki/ISO_8601) формате текущие дату и время с точностью до секунд. Для вывода даты используйте формат «Дата: {iso\_date}», а для времени – «Время: {iso\_time}».

**18.3.** Выведите на экран текущую дату и время в форматах «DD.MM.YY HH:MM:SS», «DD/MM/YY HH:MM:SS», «DD-MM-YY» и «HH:MM:SS».

**18.4.** Выведите на экран текущие компоненты даты и времи по отдельности: год в формате «YYYY», месяц в формате «MM», название месяца на русском языке, число, название дня недели на русском языке, часы, минуты и секунды.

**18.5.** Допустим, что сегодня 13 марта 2023 года. Сколько дней осталось до нового года? А сколько до следующего Международного женского дня?

**18.6.** Допустим, что сегодня 14 марта 2023 года, Катя работает по графику пятидневной рабочей недели и находится в заслуженном отпуске, который заканчивается через 17 календарных дней (не включая текущей даты). Придется ли ей выходить на работу сразу или же отпуск продлится за счет выходных дней?

**18.7.** Напишите функцию, которая будет проверять входную строку с датой и временем на соответствие формату [ISO 8601](https://ru.wikipedia.org/wiki/ISO_8601), возвращая при этом соответствующее булево значение True или False. Используйте функцию для проверки строк '2011-11-04T00:05:23' и '2011-11-04T00:05:23.'

**18.8.** Установите сторонний модуль [isodate](https://pypi.org/project/isodate/). Используя его возможности, напишите функцию, которая будет проверять входную строку с промежутком времени в формате 'PyYmMdDThHmMsS', где y, m, d, h, m, s – это количество лет, месяцев и т.д., на соответствие формату ISO 8601, возвращая при этом соответствующее булево значение True или False. Используйте функцию для проверки строк 'P3Y7M4DT12H30M19S' и '2021-05-14T09:05:23'.

# Олимпиадные задачи

В разделе представлены адаптированные задачи начального и среднего уровня сложности по олимпиадному программированию. Здесь вам предстоит уже самостоятельно определяться с выбором способа решения той или иной задачи в пределах объема знаний и навыков, полученных вами в ходе решения задач во всех предыдущих разделах задачника. Кроме того, могут понадобиться знания из других областей, например, математики, а также наблюдательность и смекалка.

Далеко не факт, что мои решения будут оптимальными и совпадать с вашими. Какое решение мне пришло в голову первым, то я и указал. Так что не судите строго и присылайте ваши решения, если они более рациональные и отличаются от предложенных.

**19.1.** Дана строка целых чисел '36 787 0 9993754\n2253', отделённых друг от друга неравномерным количеством пробелов и переводов строк. Необходимо для каждого числа, начиная с последнего и заканчивая первым, вывести на экран в отдельной строке его квадратный корень с точностью до пяти знаков после десятичной точки.

**19.2.** С клавиатуры вводится целое положительное число n < 10000. Напишите скрипт, который будет находить и выводить на экран минимальное положительное целое число q такое, что произведение цифр этого числа будет в точности равняться числу n. В случае отсутствия такого числа q скрипт должен выводить на экран сообщение «Искомого числа q не существует!». Например, для числа 70 искомым числом будет 257.

**19.3.** Представьте, что у вас имеется некоторое количество камней n (1 ≤ n ≤ 100) известного веса w1, w2, w3,.., где wi – это целые положительные числа от 1 до 1000000 включительно. Напишите скрипт, который распределит камни в две кучи так, чтобы разность весов этих двух куч была минимальной. Количество камней и веса должны генерироваться случайным образом. Сам скрипт должен возвращать число, равное минимальной разности весов двух куч. Например, для трех камней с весами 15, 20 и 3 скрипт должен возвращать число 2. На выполнение скрипта дается не более одной секунды.

# Первые скрипты и программы

В разделе представлены простейшие скрипты и программы, которые помогут вам закрепить первоначальные практические навыки, полученные в ходе решения задач во всех предыдущих разделах задачника. Если ранее вы уже писали программы на другом языке программирования, можете пробовать осилить все предлагаемые варианты самостоятельно. Что касается начинающих программистов, то им вначале следует внимательно разобраться в готовом решении и только затем приступать к обязательному самостоятельному набору кода. Важно понимать, что простое чтение исходного кода никакой практической пользы не принесет. Поэтому код набирать нужно в любом случае!

**1. Шаблон №1.** Напишите скрипт hello\_world.py, который будет выводить на экран обычное приветствие «Привет, мир!». Оформите содержимое скрипта следующим образом:

* В начале скрипта напишите три комментария, описывающие порядок импортирования модулей и пакетов и начинающиеся словами «Импортируем модули..».
* За ними определите простейшую функцию simple\_func и простейший класс SimpleClass, использовав инструкцию pass.
* Далее определите основную функцию скрипта main, в теле которой вызовите функцию simple\_func, создайте экзмепляр класса SimpleClass, осуществите вывод целевого сообщения и завершите все пустой инструкцией input, чтобы окно скрипта автоматически не закрывалось после вывода сообщения.
* В конец скрипта поместите условную инструкцию, которая будет осуществлять вызов основной функции, но только в том случае, если модуль будет запускаться как основная программа, а не импортироваться другим модулем.

Посмотрите решение данной задачи и запомните один из способов оформления программ: импорт модулей (стандартная библиотека, сторонние модули, собственные модули), определение констант, функций и классов, определение основной функции программы, содержащей последовательность действий для реализации целей программы, и, в самом низу модуля, вспомогательные инструкции с вызовом основной функции для запуска всей программы. Если модуль создается с возможностью использования не только в качестве программы, но и в качестве импорти­руемого модуля, вызов основной функции следует помещать в тело условной инструкции, проверяющей условие « \_\_name\_\_ == '\_\_main\_\_'».

**2. Шаблон №2.** Дополните код предыдущей задачи таким образом, чтобы после показа целевого сообщения пользователю выводился запрос на продолжение работы программы (символ 'y') либо выход из нее (символ 'n'). Для решения задачи используйте цикл с флагом «while flag:», прерывающийся по значению флага, либо бесконечный цикл «while True:», регулируемый условной инструкцией if в теле цикла.

**3. Менеджер пакетов Python.** Напишите программу-оболочку для менеджера пакетов pip, которая будет выполнять команды из предлагаемого пользователю меню (см. пример запуска программы). В программе создайте класс PipInstaller с методами менеджера пакетов pip (см. пункт «[Установка сторонних библиотек](https://okpython.net/python/python_uchebnik/osnovy/ustanovka_bibliotek_i_okruzhenija.html#point_1)» нашего учебника). Дайте методам следующие имена: show\_pip\_version, show\_pip\_help, upgrade\_pip, install\_package, reinstall\_package, upgrade\_package, show\_package\_info, show\_packages\_list, uninstall\_package, create\_requirements и use\_requirements. Последние два метода должны запрашивать путь к каталогу создания/расположения файла requirements.txt.

**Условие**

Менеджер пакетов Python, версия 1.0.

Список доступных комманд:

показать версию pip: 1,

показать краткую справку по pip: 2,

обновить менеджер pip: 3,

установить пакет: 4,

установить пакет принудительно: 5,

обновить пакет: 6,

удалить пакет: 7,

показать информацию о пакете: 8,

показать список установленных пакетов: 9,

создать requirements.txt: 10,

установить пакеты из requirements.txt: 11,

выйти из программы: -1.

Сделайте выбор: 4

Введите имя пакета: pillow

Collecting pillow

Using cached Pillow-9.3.0-cp311-cp311-win\_amd64.whl (2.5 MB)

Installing collected packages: pillow

Successfully installed pillow-9.3.0

[notice] A new release of pip available: 22.3 -> 22.3.1

[notice] To update, run: python.exe -m pip install --upgrade pip

Для продолжения нажмите 'Ввод'...

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

Список доступных комманд:

показать версию pip: 1,

показать краткую справку по pip: 2,

обновить менеджер pip: 3,

установить пакет: 4,

установить пакет принудительно: 5,

обновить пакет: 6,

удалить пакет: 7,

показать информацию о пакете: 8,

показать список установленных пакетов: 9,

создать requirements.txt: 10,

установить пакеты из requirements.txt: 11,

выйти из программы: -1.

Сделайте выбор: 10

Укажите путь к целевому каталогу с последним слешем:/test/

Файл успешно создан в каталоге/test/ !

Для продолжения нажмите 'Ввод'...

**4. Таблица умножения.** Напишите программу «тренажер\_таблицы\_умножения.py» для заучивания таблицы умножения. Программа должна:

* выводить свое название и версию;
* выводить меню, в котором пользователю будет предлагаться вывод таблицы умножения для повторения, запуск тренажера или выход из программы (см. пример запуска программы);
* после каждой попытки ответа пользователя программа должна выводить сообщение о правильности ответа и запрашивать подтверждение на продолжение работы программы либо выход из нее, а в случае неправильного ответа помимо сообщения об ошибке программа должна выводить еще и правильный ответ.

Вывод таблицы умножения и запуск тренажера оформите в виде функций, которые затем будут использоваться в теле основной функции программы.

**Условие**

Тренажер таблицы умножения, версия 1.0.

Вывести таблицу умножения: t

Запустить тренажер: r

Выйти из программы: q

>t

Таблица умножения.

----+-------------------------------------

| х | 1 2 3 4 5 6 7 8 9 |

----+-------------------------------------

| 1 | 1 2 3 4 5 6 7 8 9 |

| 2 | 2 4 6 8 10 12 14 16 18 |

| 3 | 3 6 9 12 15 18 21 24 27 |

| 4 | 4 8 12 16 20 24 28 32 36 |

| 5 | 5 10 15 20 25 30 35 40 45 |

| 6 | 6 12 18 24 30 36 42 48 54 |

| 7 | 7 14 21 28 35 42 49 56 63 |

| 8 | 8 16 24 32 40 48 56 64 72 |

| 9 | 9 18 27 36 45 54 63 72 81 |

----+-------------------------------------

Вывести таблицу умножения: t

Запустить тренажер: r

Выйти из программы: q

>r

4\*8 = 32

Верно!

Продолжить/Меню (any\_key/m):

7\*4 = 30

Вы ошиблись!

7\*4 = 28

Продолжить/Меню (any\_key/m): m

Вывести таблицу умножения: t

Запустить тренажер: r

Выйти из программы: q >

**5. Игра «Камень, ножницы, бумага».** Напишите программу для игры в «Камень, ножницы, бумага» с компьютером. Оформление и общий алгоритм хода игры представлены в примере запуска программы ниже. Для решения задачи вам понадобятся модули random, sys (выход из программы через sys.exit()), time (использование задержек через time.sleep(0.5)). Что касается функций, то их можете не использовать вообще, оформив все через циклы с прерыванием.

**Условие**

Игра «Камень, ножницы, бумага», версия 1.0.

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

Каждый из игроков может выбрать «Камень»,

«Ножницы» или «Бумагу». Победитель определяется

по следующим правилам: «Камень» бьет «Ножницы»,

«Ножницы» — «Бумагу», а «Бумага» бьет «Камень».

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

Победы: 0, проигрыши 0, ничьи 0.

Выберите (К)амень, (Н)ожницы, (Б)умагу или (В)ыход?

> к

«Камень» против..

1...

2...

3...

«Бумага»

Вы проиграли!

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

Победы: 0, проигрыши 1, ничьи 0.

Выберите (К)амень, (Н)ожницы, (Б)умагу или (В)ыход?

> н

«Ножницы» против..

1...

2...

3...

«Бумага»

Вы выиграли!

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

Победы: 1, проигрыши 1, ничьи 0.

Выберите (К)амень, (Н)ожницы, (Б)умагу или (В)ыход?

> б

«Бумага» против..

1...

2...

3...

«Камень»

Вы выиграли!

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

Победы: 2, проигрыши 1, ничьи 0.

Выберите (К)амень, (Н)ожницы, (Б)умагу или (В)ыход?

> в

Спасибо за игру!

**6. Квадратное уравнение.** Напишите программу, которая по введенным пользователем коэффициентам квадратного уравнения ax2 + bx + c = 0 будет выдавать ответ с точностью до тысячных и готовое решение. Программа должна:

* выводить свое название и версию, а также краткое описание;
* принимать на ввод коэффициенты уравнения в виде действительных чисел, например, -3.27, или в виде обыкновенных дробей, например, -5/7;
* проверять вводимые значения на допустимость ввода, предлагая в случае ошибки повторный ввод коэффициента или запрашивая выход из программы в противном случае;
* выводить решение с пояснениями включая случаи, когда один или несколько коэффициентов равны нулю;
* после вывода решения выводить запрос на продолжение работы программы или выход из нее.

Показать решение.

**Условие**

Решение квадратных уравнений, версия 1.0.

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

1. Для решения уравнения необходимо ввести коэффициенты

в формате десятичной дроби, например, -1.35, либо в формате

обыкновенной дроби с числителем и знаменателем, например, -3/5.

2. Результат будет рассчитан с точностью до 0,001.

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

Введите коэффициенты для a\*x^2 + b\*x + c = 0.

a = -5

b = -3

c = 2

Решаем уравнение -5.0\*x^2 - 3.0\*x + 2.0 = 0.

Находим дискриминант по формуле b^2 - 4\*a\*c:

D = b^2 - 4\*a\*c = 49.0.

Т.к. дискриминант больше нуля, уравнение имеет два корня:

x1 = (-b + √D)/(2\*a) = (-(-3.0) + √49.0)/(2\*(-5.0)) = -1.0,

x2 = (-b - √D)/(2\*a) = (-(-3.0) - √49.0)/(2\*(-5.0)) = 0.4.