МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

ГЛАВНОЕ УПРАВЛЕНИЕ ПО ОБРАЗОВАНИЮ

МОГИЛЕВСКОГО ОБЛАСТНОГО ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

|  |  |
| --- | --- |
| Специальность | 2– 40 01 01 |
| Учебная группа | ПО-455 |
|  |  |

Учебная дисциплина Инструментальное

программное обеспечение

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**

**РАЗРАБОТКА, ОТЛАДКА И ИСПЫТАНИЕ АЛГОРИТМОВ И ПРОГРАММ ОБРАБОТКИ СТРОК**

Выполнил Савич А.О.

Проверил Денисовец Д. А.

2021

1. **Цель работы**

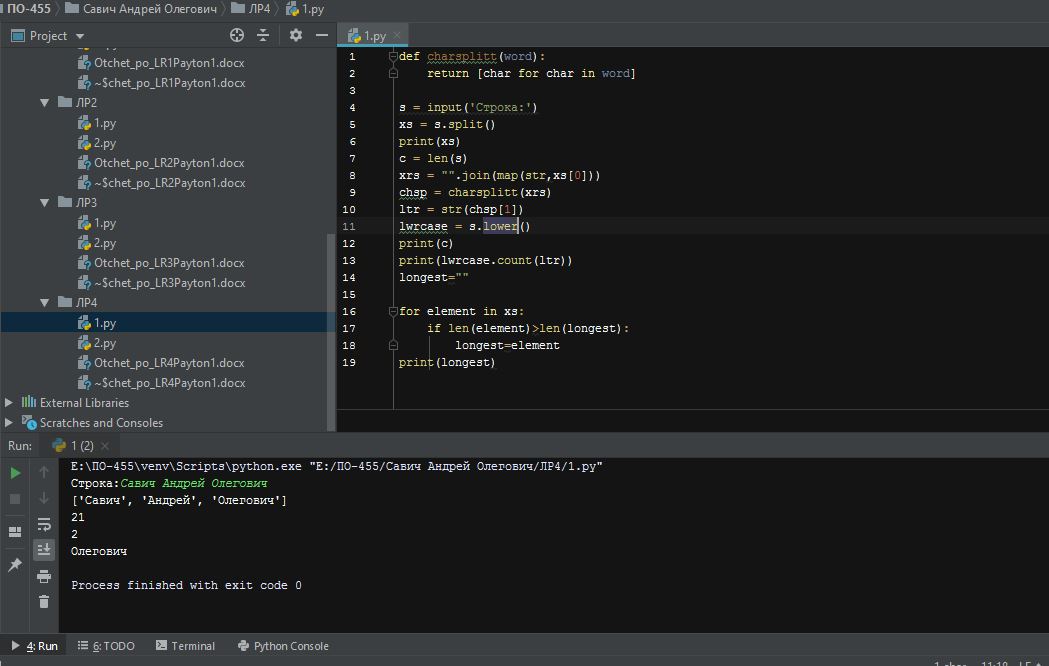
Формирование умений разрабатывать, производить отладку и испытание алгоритмов и программ обработки строк

**2 Индивидуальное задание**

# Вариант 8

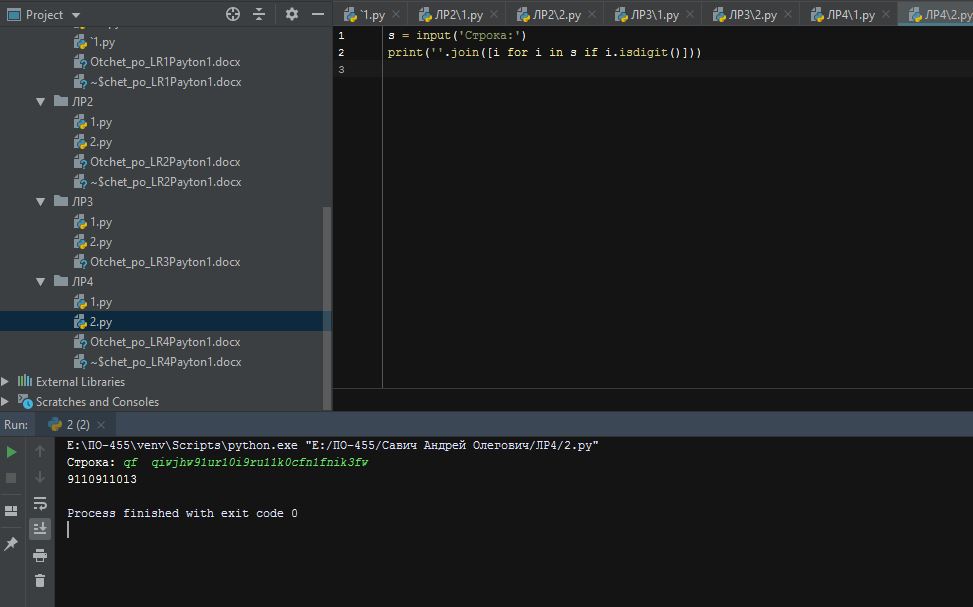
1 Разработайте алгоритм и напишите программу решения задачи на языке программирования Python в соответствии с заданным вариантом.

8 Введите вашу фамилию, имя и отчество как строку символов. Определите длину строки и количество букв, совпадающих со второй буквой фамилии. Выведите самое длинное слово.



1 Разработайте алгоритм и напишите программу решения задачи на языке программирования Python в соответствии с заданным вариантом.

8 Дана строка символов. Выведите на экран цифры, содержащиеся в строке.



**3 Контрольные вопросы**

1 Опишите операции над строками в языке Python.

Над строками можно выполнять типичные операции. Для этого в языке Python перегружены соответствующие операторы.

|  |  |  |
| --- | --- | --- |
| **Оператор (операция)** | **Использование в программах** | **Объяснение** |
| **+** | s1+s2 | Конкатенация |
| **\*** | s\*2 | Повторение |
| **[ ]** | s[i] | Обращение к символу строки s по индексу i |
| **[:]** | s[i:j] | Вытягивание подстроки из позиции *i* до позиции *j* |

Ниже в таблице приведен перечень операторов для работы с строками.

2 Приведите синтаксис создания строк в языке Python.

Чтобы объявить переменную типа строка достаточно использовать оператор присваивания **=**. Общая форма объявления переменной следующая

variable\_name = string

где

* *variable\_name* – имя создаваемой переменной. В дальнейшем это имя используется в программе и связано со строкой string;
* *string* – строка (литерал), размещенная между одинарными или двойными кавычки.

3 Охарактеризуйте функции для работы с символами в языке Python.

В этом уроке рассмотрим следующие встроенные функции, условно разбив их на группы:

* функции для работы с символами – ord(), chr(), len()
* математические функции – abs(), round(), divmod(), pow(), max(), min(), sum()

Функция ord() позволяет получить номер символа по таблице Unicode. Соответственно, принимает она в качестве аргумента одиночный символ, заключенный в кавычки:

>>> ord('z')

122

>>> ord('ф')

1092

>>> ord('@')

64

Функция chr() выполняет обратное действие. Она позволяет получить символ по его номеру:

>>> chr(87)

'W'

>>> chr(1049)

'Й'

>>> chr(10045)

'✽'

Чтобы не путать ord() и chr(), помните, что функция – это действие. Ее имя как бы отвечает на вопрос "Что сделать?". Order – это порядок. Значит, мы хотим получить порядковый номер элемента в ряду. А чтобы получить номер, должны передать символ. Character – это символ. Значит, мы хотим получить символ. Поэтому должны передать порядковый номер.

Функция len() в качестве аргумента принимает объект, который состоит из более простых объектов, количество которых она подсчитывает. Числа – это простые объекты, их нельзя передавать в len(). Строки можно:

>>> len('abc')

3

>>> s1 = '------'

>>> s2 = '\_\_\_\_\_\_\_'

>>> len(s1) > len(s2)

False

>>> len(s1)

6

>>> len(s2)

7

Кроме строк в len() можно передавать другие, еще не изученные нами, структуры данных.

Функция abs() возвращает абсолютное значение числа:

>>> abs(-2.2)

2.2

>>> abs(9)

9

Если требуется округлить вещественное число до определенного знака после запятой, то следует воспользоваться функцией round():

>>> a = 10/3

>>> a

3.3333333333333335

>>> round(a,2)

3.33

>>> round(a)

3

Если второй аргумент не задан, то округление идет до целого числа. Есть одна специфическая особенность этой функции. Второй аргумент может быть отрицательным числом. В этом случае округляться начинают единицы, десятки, сотни и т. д., то есть целая часть:

>>> round(5321, -1)

5320

>>> round(5321, -3)

5000

>>> round(5321, -4)

10000

Функция именно округляет согласно правилу округления из математики, а не отбрасывает. Поэтому 5 тысяч неожиданно округляются до десяти.

>>> round(3.76, 1)

3.8

>>> round(3.72, 1)

3.7

>>> round(3.72)

4

>>> round(3.22)

3

Если нужно просто избавиться от дробной части без округления, следует воспользоваться функцией int():

>>> int(3.78)

3

Нередко функцию round() используют совместно с функцией print(), избегая форматирования вывода:

>>> a = 3.45673

>>> print("Number: %.2f" % a)

Number: 3.46

>>> print("Number:", round(a,2))

Number: 3.46

В последнем случае код выглядит более ясным.

Функция divmod() выполняет одновременно деление нацело и нахождение остатка от деления:

>>> divmod(10, 3)

(3, 1)

>>> divmod(20, 7)

(2, 6)

Возвращает она кортеж, извлечение данных из которого мы пока не изучали. В других языках нередко встречаются две отдельные функции: div() и mod(). Первая делит нацело, вторая находит остаток от целочисленного деления (деления по модулю). В Python и многих других языках для этого используются специальные символы-операнды:

>>> 10 // 3

3

>>> 10 % 3

1

Функция pow() возводит в степень. Первое число – основание, второе – показатель:

>>> pow(3, 2)

9

>>> pow(2, 4)

16

То же самое можно проделать так:

>>> 3\*\*2

9

>>> 2\*\*4

16

Однако pow() может принимать третий необязательный аргумент. Это число, на которое делится по модулю результат возведения в степень:

>>> pow(2, 4, 4)

0

>>> 2\*\*4 % 4

0

Преимуществом первого способа является его более быстрое выполнение.

Функции max(), min() и sum() находят соответственно максимальный, минимальный элемент и сумму элементов аргумента:

>>> max(10, 12, 3)

12

>>> min(10, 12, 3, 9)

3

>>> a = (10, 12, 3, 10)

>>> sum(a)

35

В sum() нельзя передать перечень элементов, должна быть структура данных, например, кортеж. В min() и max() также чаще передают один так называемый итерируемый объект:

>>> max(a)

12

4 Опишите функции и методы для работы со строками в языке Python.

## Таблица "Функции и методы строк"

| **Функция или метод** | **Назначение** |
| --- | --- |
| **S = 'str'; S = "str"; S = '''str'''; S = """str"""** | Литералы строк |
| **S = "s\np\ta\nbbb"** | Экранированные последовательности |
| **S = r"C:\temp\new"** | Неформатированные строки (подавляют экранирование) |
| **S = b"byte"** | Строка байтов |
| **S1 + S2** | Конкатенация (сложение строк) |
| **S1 \* 3** | Повторение строки |
| **S[i]** | Обращение по индексу |
| **S[i:j:step]** | Извлечение среза |
| **len**(S) | Длина строки |
| **S.find**(str, [start],[end]) | Поиск подстроки в строке. Возвращает номер первого вхождения или -1 |
| **S.rfind**(str, [start],[end]) | Поиск подстроки в строке. Возвращает номер последнего вхождения или -1 |
| **S.index**(str, [start],[end]) | Поиск подстроки в строке. Возвращает номер первого вхождения или вызывает ValueError |
| **S.rindex**(str, [start],[end]) | Поиск подстроки в строке. Возвращает номер последнего вхождения или вызывает ValueError |
| **S.replace**(шаблон, замена[, maxcount]) | Замена шаблона на замену. maxcount ограничивает количество замен |
| **S.split**(символ) | Разбиение строки по разделителю |
| **S.isdigit**() | Состоит ли строка из цифр |
| **S.isalpha**() | Состоит ли строка из букв |
| **S.isalnum**() | Состоит ли строка из цифр или букв |
| **S.islower**() | Состоит ли строка из символов в нижнем регистре |
| **S.isupper**() | Состоит ли строка из символов в верхнем регистре |
| **S.isspace**() | Состоит ли строка из неотображаемых символов (пробел, символ перевода страницы ('\f'), "новая строка" ('\n'), "перевод каретки" ('\r'), "горизонтальная табуляция" ('\t') и "вертикальная табуляция" ('\v')) |
| **S.istitle**() | Начинаются ли слова в строке с заглавной буквы |
| **S.upper**() | Преобразование строки к верхнему регистру |
| **S.lower**() | Преобразование строки к нижнему регистру |
| **S.startswith**(str) | Начинается ли строка S с шаблона str |
| **S.endswith**(str) | Заканчивается ли строка S шаблоном str |
| **S.join**(список) | Сборка строки из списка с разделителем S |
| **ord**(символ) | Символ в его код ASCII |
| **chr**(число) | Код ASCII в символ |
| **S.capitalize**() | Переводит первый символ строки в верхний регистр, а все остальные в нижний |
| **S.center**(width, [fill]) | Возвращает отцентрованную строку, по краям которой стоит символ fill (пробел по умолчанию) |
| **S.count**(str, [start],[end]) | Возвращает количество непересекающихся вхождений подстроки в диапазоне [начало, конец] (0 и длина строки по умолчанию) |
| **S.expandtabs**([tabsize]) | Возвращает копию строки, в которой все символы табуляции заменяются одним или несколькими пробелами, в зависимости от текущего столбца. Если TabSize не указан, размер табуляции полагается равным 8 пробелам |
| **S.lstrip**([chars]) | Удаление пробельных символов в начале строки |
| **S.rstrip**([chars]) | Удаление пробельных символов в конце строки |
| **S.strip**([chars]) | Удаление пробельных символов в начале и в конце строки |
| **S.partition**(шаблон) | Возвращает кортеж, содержащий часть перед первым шаблоном, сам шаблон, и часть после шаблона. Если шаблон не найден, возвращается кортеж, содержащий саму строку, а затем две пустых строки |
| **S.rpartition**(sep) | Возвращает кортеж, содержащий часть перед последним шаблоном, сам шаблон, и часть после шаблона. Если шаблон не найден, возвращается кортеж, содержащий две пустых строки, а затем саму строку |
| **S.swapcase**() | Переводит символы нижнего регистра в верхний, а верхнего – в нижний |
| **S.title**() | Первую букву каждого слова переводит в верхний регистр, а все остальные в нижний |
| **S.zfill**(width) | Делает длину строки не меньшей width, по необходимости заполняя первые символы нулями |
| **S.ljust**(width, fillchar=" ") | Делает длину строки не меньшей width, по необходимости заполняя последние символы символом fillchar |
| **S.rjust**(width, fillchar=" ") | Делает длину строки не меньшей width, по необходимости заполняя первые символы символом fillchar |
| **S.format**(\*args, \*\*kwargs) | Форматирование строки |

**Вывод:** Научился разрабатывать, производить отладку и испытание алгоритмов и программ обработки строк