

Лабораторная работа №3

Тема: «Работа с функциями. Хранений функций. Пространство имен. Создание, поиск и использование модулей. Описание основных встроенных модулей».

Требования к выполнению лабораторной работы №3

1. Изучите теоретическую часть к третьей лабораторной работе:
 - a. Теоретическая часть к третьей лабораторной работе.
 - b. Лекция №3.
2. Создайте новый проект.
3. Запустите примеры из лабораторной работы.
4. Выполните задание согласно вашему варианту:
 - a. Вычислите свой вариант (*согласно формуле ниже*).
Если сделали не свой вариант => работа не засчитывается.
 - b. Каждое задание представляет собой отдельный скрипт формата:
`lab_{номер_ЛР}_{номер_задания}_{номер_варианта}.py`, пример:
`lab_3_1_2.py`
 - c. Отправьте выполненное задание в ОРИОКС (*раздел Домашние задания*).

Формат защиты лабораторных работ:

1. Продемонстрируйте выполненные задания.
2. Ответьте на вопросы по вашему коду.
3. При необходимости выполните дополнительное (*дополнительные*) задания от преподавателя.
4. Ответьте (*устно*) преподавателю на контрольные вопросы.

Список вопросов

1. В какой момент создается новый объект функции?
2. Почему следует избегать модификации изменяемых аргументов?
3. Что такое интроспекция?
4. Что такое аннотация функций?
5. Что такое анонимная функция?
6. Правило LEGB.
7. Генераторные функции.
8. Генераторные выражения.
9. Что такое модуль?
10. Как работает поиск модуля при импортировании?
11. Для чего нужны файлы `__init__.py`?

Задания

Во всех заданиях:

1. Необходимо проверять корректность вводимых данных и выводить соответствующие сообщения об ошибках.
2. Во всех пользовательских функциях использовать аннотации.

№ Варианта = номер_студенческого % 2 + 1

Вариант №1

Задание №1. Напишите функцию, которая принимает произвольное число аргументов (каждый аргумент – список) и возвращает пересечение множеств.

Задание №2. Создайте функцию *reducer*, которая сокращает правильную дробь. На вход функции подается кортеж из двух элементов m и n (числитель и знаменатель). Результатом функции является сокращенная дробь – кортеж из двух элементов m' и n' (числитель и знаменатель).

Ограничение: m и n – целые положительные числа, $m < n$

Необходимо проверять корректность вводимых данных.

Пример:

Аргументы функции		Результат	
m	n	m'	n'
3	5	3	5
2	4	1	2
6	10	3	5
2	20	1	10

Задание №3. Создайте генераторную функцию *hofstadter_f_m*, которая возвращает «Женские и мужские последователи Хофштадтера».

Женские (*F*) и мужские (*M*) последователи Хофштадтера определяются следующим образом:

$$F(0) = 1$$

$$M(0) = 0$$

$$F(n) = n - M(F(n - 1))$$

$$M(n) = n - F(M(n - 1))$$

Функция-генератор *hofstadter_f_m* должна принимать один аргумент – *n* и возвращать объект-генератор, который при итерировании должен выводить первые *n* членов двух последовательностей (*F* и *M*).

Пример:

<i>n</i>	Результат
2	(1, 0), (1, 0)
3	(1, 0), (1, 0), (2, 1)
4	(1, 0), (1, 0), (2, 1), (2, 2)
5	(1, 0), (1, 0), (2, 1), (2, 2), (3, 2)

Задание №4. Создайте функцию *nearest_date*, которая принимает неограниченное количество аргументов (каждый аргумент – дата в формате "*dd.mm.YYYY*") и возвращает дату, которая наиболее близка к текущей (сегодняшней). Если две даты симметрично находятся возле текущей даты, то вернуть нужно дату, которая ещё не наступила.

Пример (для текущей даты = 06.09.2022):

Аргументы функции	Результат
"05.09.2022", "07.09.2022"	"07.09.2022"
"01.01.2050", "12.04.2011", "31.12.1970"	"12.04.2011"

Вариант №2

Задание №1. Напишите функцию, которая принимает произвольное число аргументов (каждый аргумент – список) и возвращает True если пересечение списков пустое, иначе возвращает False

Задание №2. Создайте функцию *binom*, которая принимает один аргумент n – целое число и возвращает строку (*формулу*) – разложение на отдельные слагаемые целой неотрицательной степени n суммы двух переменных.

Пример:

n	Результат
0	1
1	$a+b$
2	$a^2+2ab+b^2$
3	$a^3+3a^2b+3ab^2+b^3$
-3	$1/(a^3+3a^2b+3ab^2+b^3)$

Задание №3. Создайте генераторную функцию *hofstadter_q*, которая последовательность Q Хофштадтера.

Последовательность определяется следующим образом:

$$Q(1) = Q(2) = 1$$

$$Q(n) = Q(n - Q(n - 1)) + Q(n - Q(n - 2)), \quad n > 2$$

Функция-генератор *hofstadter_q* должна принимать один аргумент – n и возвращать объект-генератор, который при итерировании должен выводить первые n членов последовательностей.

Пример:

n	Результат
2	1, 1
3	1, 1, 2
7	1, 1, 2, 3, 3, 4, 5

Задание №4. Создайте функцию *nearest_time*, которая принимает неограниченное количество аргументов (*каждый аргумент – время в формате "HH:MM:SS"*) и возвращает время, которое наиболее близко к текущему (*на время запуска скрипта*). Если два времени симметрично находятся возле текущего, то вернуть нужно время, которое находилось в списке аргументов раньше.

Пример (для текущего времени = 23:50:00):

<i>Аргументы функции</i>	<i>Результат</i>
"23:52:00", "21:00:59"	"23:52:00"
"21:22:33", "00:15:20", "14:47:50"	"00:15:20"
"23:55:00", "12:00:00", "23:45:00"	"23:55:00"
"23:45:00", "23:55:00", "12:00:00"	"23:45:00"