|  |  |
| --- | --- |
| лого-РГСУ-2015.png | **Российский государственный социальный университет**  **Факультет информационных технологий** |

**ЛАБОРАТОРНАЯ РАБОТА №2**

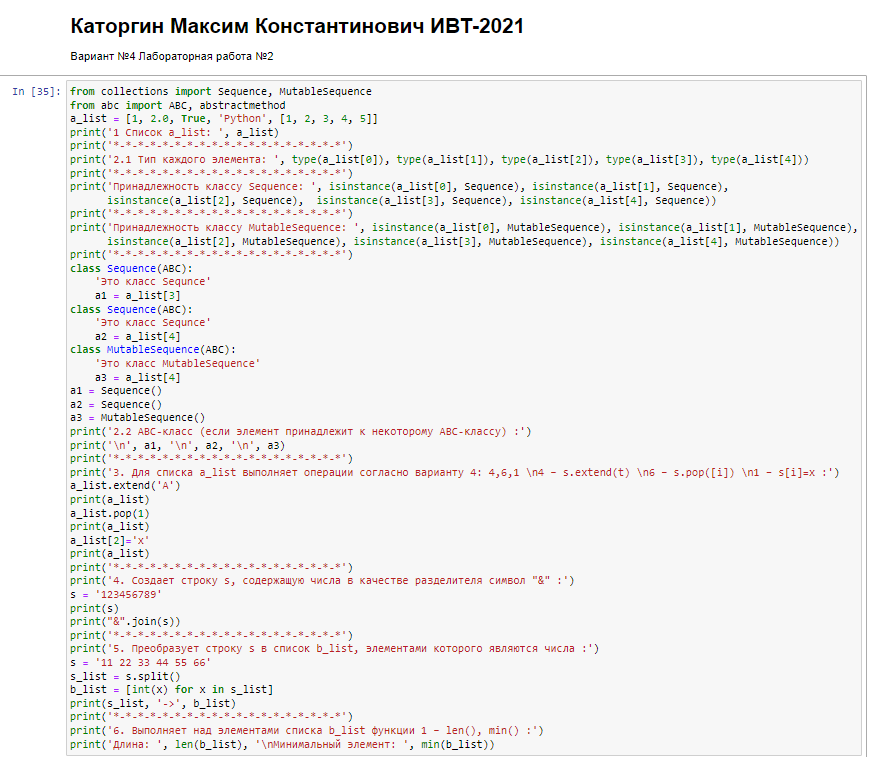
**по дисциплине «Специальные разделы программирования»**

название дисциплины

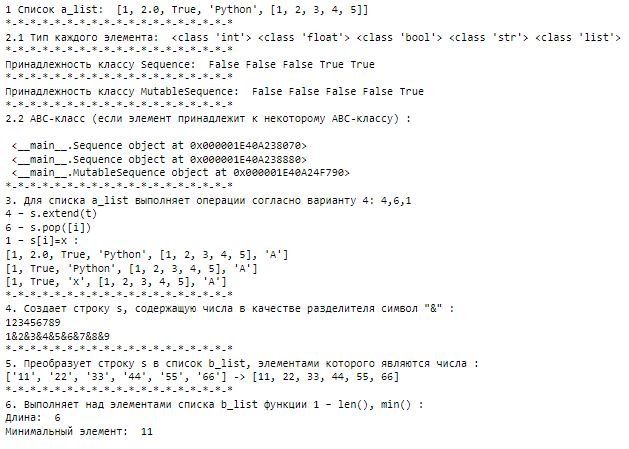
|  |  |
| --- | --- |
| **ФИО студента** | Каторгин Максим Константинович |
| **Направление подготовки** | Информатика и вычислительная техника |
| **Группа** | ИВТ-М-1-Д-2021-1 |

**Москва 2022**

Код в IDE:



Результат работы:



Листинг:

from collections import Sequence, MutableSequence

from abc import ABC, abstractmethod

a\_list = [1, 2.0, True, 'Python', [1, 2, 3, 4, 5]]

print('1 Список a\_list: ', a\_list)

print('\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*')

print('2.1 Тип каждого элемента: ', type(a\_list[0]), type(a\_list[1]), type(a\_list[2]), type(a\_list[3]), type(a\_list[4]))

print('\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*')

print('Принадлежность классу Sequence: ', isinstance(a\_list[0], Sequence), isinstance(a\_list[1], Sequence),

isinstance(a\_list[2], Sequence), isinstance(a\_list[3], Sequence), isinstance(a\_list[4], Sequence))

print('\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*')

print('Принадлежность классу MutableSequence: ', isinstance(a\_list[0], MutableSequence), isinstance(a\_list[1], MutableSequence),

isinstance(a\_list[2], MutableSequence), isinstance(a\_list[3], MutableSequence), isinstance(a\_list[4], MutableSequence))

print('\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*')

class Sequence(ABC):

'Это класс Sequnce'

a1 = a\_list[3]

class Sequence(ABC):

'Это класс Sequnce'

a2 = a\_list[4]

class MutableSequence(ABC):

'Это класс MutableSequence'

a3 = a\_list[4]

a1 = Sequence()

a2 = Sequence()

a3 = MutableSequence()

print('2.2 ABC-класс (если элемент принадлежит к некоторому ABC-классу) :')

print('\n', a1, '\n', a2, '\n', a3)

print('\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*')

print('3. Для списка a\_list выполняет операции согласно варианту 4: 4,6,1 \n4 – s.extend(t) \n6 – s.pop([i]) \n1 – s[i]=x :')

a\_list.extend('A')

print(a\_list)

a\_list.pop(1)

print(a\_list)

a\_list[2]='x'

print(a\_list)

print('\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*')

print('4. Создает строку s, содержащую числа в качестве разделителя символ "&" :')

s = '123456789'

print(s)

print("&".join(s))

print('\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*')

print('5. Преобразует строку s в список b\_list, элементами которого являются числа :')

s = '11 22 33 44 55 66'

s\_list = s.split()

b\_list = [int(x) for x in s\_list]

print(s\_list, '->', b\_list)

print('\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*')

print('6. Выполняет над элементами списка b\_list функции 1 – len(), min() :')

print('Длина: ', len(b\_list), '\nМинимальный элемент: ', min(b\_list))