**Отчет по самостоятельной работе №5**

**по дисциплине МДК 01.01 “Технология разработки программного обеспечения”.**

Выполнил: студент

группы 319

Белоусов Савелий Дмитриевич

Дата 23.12.2024

Сургут 2024

**Введение**

Функции и массивы являются важными элементами программирования в Python. В данной работе будет рассмотрено использование функций и работа с массивами.

**Цель работы**

Цель работы – научиться создавать и использовать функции в Python, а также работать с массивами и операциями над ними.

**Практическая часть**

**Задание 1: Создание функций**

**1. Напишите функцию для вычисления факториала числа.**

python

Copy

def factorial(n):

if n == 0 or n == 1:

return 1

else:

return n \* factorial(n - 1)

# Пример использования

number = 5

print(f"Факториал числа {number} равен {factorial(number)}")

**2. Создайте функцию для нахождения максимального и минимального элемента в списке.**

python

Copy

def find\_min\_max(lst):

if not lst:

return None, None

min\_element = max\_element = lst[0]

for element in lst:

if element < min\_element:

min\_element = element

if element > max\_element:

max\_element = element

return min\_element, max\_element

# Пример использования

numbers = [3, 7, 1, 9, 4, 6]

min\_val, max\_val = find\_min\_max(numbers)

print(f"Минимальный элемент: {min\_val}, Максимальный элемент: {max\_val}")

**Задание 2: Работа с аргументами функций**

**1. Реализуйте функцию с переменным количеством аргументов.**

python

Copy

def sum\_all(\*args):

total = 0

for num in args:

total += num

return total

# Пример использования

result = sum\_all(1, 2, 3, 4, 5)

print(f"Сумма всех аргументов: {result}")

**2. Напишите функцию с аргументами по умолчанию.**

python

Copy

def greet(name="Гость", greeting="Привет"):

return f"{greeting}, {name}!"

# Пример использования

print(greet()) # Привет, Гость!

print(greet("Иван")) # Привет, Иван!

print(greet("Мария", "Здравствуйте")) # Здравствуйте, Мария!

**Задание 3: Работа с массивами**

**1. Создайте массив и выполните операции добавления, удаления и поиска элементов.**

python

Copy

# Создание массива

numbers = [1, 2, 3, 4, 5]

# Добавление элемента

numbers.append(6)

print("После добавления элемента:", numbers)

# Удаление элемента

numbers.remove(3)

print("После удаления элемента:", numbers)

# Поиск элемента

search\_element = 4

if search\_element in numbers:

print(f"Элемент {search\_element} найден в массиве")

else:

print(f"Элемент {search\_element} не найден в массиве")

**2. Реализуйте программу для вычисления среднего значения элементов массива.**

python

Copy

def calculate\_average(lst):

if not lst:

return 0

return sum(lst) / len(lst)

# Пример использования

numbers = [10, 20, 30, 40, 50]

average = calculate\_average(numbers)

print(f"Среднее значение элементов массива: {average}")

**Задание 4: Многомерные массивы**

**1. Напишите программу для работы с двумерными массивами (матрицами).**

python

Copy

# Создание двумерного массива (матрицы)

matrix = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

# Вывод матрицы

for row in matrix:

print(row)

**2. Реализуйте сложение и умножение матриц.**

python

Copy

def add\_matrices(matrix1, matrix2):

result = []

for i in range(len(matrix1)):

row = []

for j in range(len(matrix1[0])):

row.append(matrix1[i][j] + matrix2[i][j])

result.append(row)

return result

def multiply\_matrices(matrix1, matrix2):

result = []

for i in range(len(matrix1)):

row = []

for j in range(len(matrix2[0])):

sum = 0

for k in range(len(matrix2)):

sum += matrix1[i][k] \* matrix2[k][j]

row.append(sum)

result.append(row)

return result

# Пример использования

matrix1 = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

matrix2 = [

[9, 8, 7],

[6, 5, 4],

[3, 2, 1]

]

# Сложение матриц

sum\_matrix = add\_matrices(matrix1, matrix2)

print("Сумма матриц:")

for row in sum\_matrix:

print(row)

# Умножение матриц

product\_matrix = multiply\_matrices(matrix1, matrix2)

print("Произведение матриц:")

for row in product\_matrix:

print(row)

**Задание 5: Рекурсивные функции**

**1. Напишите рекурсивную функцию для вычисления чисел Фибоначчи.**

python

Copy

def fibonacci(n):

if n <= 1:

return n

else:

return fibonacci(n - 1) + fibonacci(n - 2)

# Пример использования

number = 10

print(f"Число Фибоначчи для {number} равно {fibonacci(number)}")

**2. Реализуйте функцию для вычисления суммы цифр числа рекурсивно.**

python

Copy

def sum\_of\_digits(n):

if n == 0:

return 0

else:

return (n % 10) + sum\_of\_digits(n // 10)

# Пример использования

number = 12345

print(f"Сумма цифр числа {number} равна {sum\_of\_digits(number)}")

**Вывод**

В данной работе были рассмотрены основы работы с функциями и массивами в Python. Были созданы и использованы функции для различных задач, а также выполнены операции над массивами и многомерными массивами. Рекурсивные функции были применены для решения задач, связанных с числами Фибоначчи и суммой цифр числа.