**Министерство образования и науки Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**(ФГБОУ ВО «АмГУ»)**

Факультет математики и информатики

Кафедра информационных и управляющих систем

Направление подготовки 09.03.04 - Программная инженерия

Направленность (профиль) образовательной программы: Программная инженерия

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**

на тему: **Списки или массивы в Питоне**

|  |  |  |
| --- | --- | --- |
| Исполнитель  студент группы 357-об | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | Д.Е. Буханов |
| Проверил | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | Е.В. Дегтярёв |

Благовещенск 2023

**1 ОБЩИЕ ЗАДАНИЯ**

**1 Задание.**

Создайте список целых чисел от -20 до 30 (генерация).

def main():

    # Генерируем список

    lst = [i for i in range(-20, 31)]

    print(f"Сгенерированный список: \n{lst}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 1 – Листинг рабочей программы «file1.py»

Таблица 1

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Константы | Сгенерированный список:  [-20, -19, -18, -17, -16, -15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 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, 28, 29, 30] |

**2 Задание.**

Создайте список целых чисел от -10 до 10 с шагом 2 (генерация).

lst = [i for i in range(-10 , 11, 2)]

print('Список: {0} '.format(lst))

Рисунок 2 – Листинг рабочей программы «file2.py»

Таблица 2

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Константы | Список: [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10] |

**3 Задание.**

Создайте список из 20 пятерок (генерация)

def main():

    # Генерируем список

    lst = [5] \* 20

    print(f"Сгенерированный список: \n{lst}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 3 – Листинг рабочей программы «file3.py»

Таблица 3

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Константы | Сгенерированный список:  [5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5] |

**4 Задание.**

Создайте список из сумм троек чисел от 0 до 10, используя генератор списка (0 + 1 + 2, 1 + 2 + 3, …)

lst = [i+i+1+i+2 for i in range (0, 11)]

print('Список: {0}'.format(lst))

Рисунок 4 – Листинг рабочей программы «file4.py»

Таблица 4

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| Константы | Список: [3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33] |

**5 Задание.**

Заполните массив элементами арифметической прогрессии. Ее первый элемент, разность и количество элементов нужно ввести с клавиатуры

def progres(a\_1, n, d):

    return a\_1 + ((n - 1) \* d)

def main():

    a\_1 = float(input("Введите первый элемент ариф. прогрессии: "))

    d = float(input("Введите разность ариф. прогрессии: "))

    n = int(input("Введите количество элементов ариф. прогрессии: "))

    lst = []

    for n in range(1, n + 1):

        a\_new = progres(a\_1, n, d)

        lst.append(a\_new)

    print(f"Массив с арифм. прогрессией: \n{lst}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 5 – Листинг рабочей программы «file5.py»

Таблица 5

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 2  3  4 | Массив с арифм. прогрессией:  [2.0, 5.0, 8.0, 11.0] |

**6 Задание.**

Заполните массив случайными числами в диапазоне 20…100 и подсчитайте отдельно число четных и нечетных элементов.

import random

number\_of\_t = int(input('Введите количество раз:'))

n = 0

sum\_ch = 0

sum\_nech = 0

lst = [random.randint(20,100) for a in range (number\_of\_t)]

for b in range(len(lst)):

    number = lst [b]

    number = int(number)

    if number % 2 == 0:

        sum\_ch += 1

    else:

        sum\_nech += 1

print('Чётных элементов: {0}, число нечётных элементов: {1}'.format(sum\_ch,sum\_nech))

Рисунок 6 – Листинг рабочей программы «file6.py»

Таблица 6

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 4 | Чётных элементов: 1, число нечётных элементов: 3 |

**2 ИНДИВИДУЛЬНОЕ ЗАДАНИЕ**

**2.1 Задание.**

Составить программу, которая считывает сначала количество оценок, потом по очереди сами эти оценки, затем выводит их же в том же порядке (используем список). Найдите среднюю оценку за урок.

def main():

    count\_number = int(input("Введите количество цифр: "))

    sp = [] # список для оценок

    for i in range(count\_number):

        sp.append(int(input("Введите значение оценки: ")))

    print(f"Среднее арифметическое {round(sum(sp) / len(sp), 3)}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 7 – Листинг рабочей программы «main.py»

Таблица 7

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 3  2  2  3 | Среднее арифметическое 2.333 |



Блок схема 1

**2.2 Задание.**

Для списка X, состоящего из 15 элементов, найти наибольший элемент списка и его порядковый номер

def main():

    sp = []

    for i in range(15):

        sp.append(float(input("Введите значение числа: ")))

    print(f"Максимальный элемент: {max(sp)}")

    print(f"Номер максимального элемента: {sp.index(max(sp)) + 1}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 8 – Листинг рабочей программы «main\_2.py»

Таблица 8

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 1  1  1  1  1  1  1  1  2  3  3  4  4  5  66 | Максимальный элемент: 66.0  Номер максимального элемента: 15 |



Блок схема 2

**2.3 Задание.**

В списке из 12 вещественных чисел найти наименьший элемент и поменять его местами с первым элементом.

def main():

    sp = []

    for i in range(12):

        sp.append(float(input("Введите значение числа: ")))

    print(f"Первоначальный список: {sp}")

    ind = sp.index(min(sp))

    sp[0], sp[ind] = sp[ind], sp[0]

    print(f"Измененый список: {sp}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Рисунок 9 – Листинг рабочей программы «main\_3.py»

Таблица 9

|  |  |
| --- | --- |
| Данные: | |
| Входные (ввод) | Выходные (вывод) |
| 3456  34  5623  456  3456  23456  0  3456  3245  234  3  43 | Первоначальный список: [3456.0, 34.0, 5623.0, 456.0, 3456.0, 23456.0, 0.0, 3456.0, 3245.0, 234.0, 3.0, 43.0]  Измененый список: [0.0, 34.0, 5623.0, 456.0, 3456.0, 23456.0, 3456.0, 3456.0, 3245.0, 234.0, 3.0, 43.0] |



Блок схема 3