

Практическое занятие № 6

Тема: практического занятия: составление программ со списками в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ со списками в IDE PyCharm Community

Постановка задачи1: Дан список A ненулевых целых чисел размера 10. Вывести значение первого из тех его элементов AK, которые удовлетворяют неравенству $A_K < A_{10}$. Если таких элементов нет, то вывести 0.

```
Текст программы1
# Дан список A ненулевых целых чисел размера 10. Вывести значение первого из
тех
# его элементов AK, которые удовлетворяют неравенству  $A_K < A_{10}$ . Если таких
# элементов нет, то вывести 0.
# Инициализируем пустой список
A = []

# Вводим список A из 10 ненулевых целых чисел
while len(A) < 10:
    try:
        number = int(input(f"Введите элемент A[{len(A)}]: "))
        if number == 0:
            raise ValueError("Число не должно быть нулевым. Попробуйте
снова.")
        A.append(number)
    except ValueError as e:
        print(f"Ошибка: {e}. Пожалуйста, введите целое число.")

# Последний элемент списка
last_element = A[-1]

# Ищем первый элемент, который меньше последнего
result = 0 # По умолчанию, если не найдем, будет 0
for element in A:
    if element < last_element:
        result = element
        break # Выходим из цикла, как только нашли первый подходящий элемент

# Выводим результат
print(result)
```

Терминал1:

C:\Users\User\AppData\Local\Microsoft\WindowsApps\python3.11.exe "D:\колледж\алгоритмы и программирование\Pz Sapronov Daniil\PZ_6\PZ_6_1.py"

Введите элемент A[0]: 5

Введите элемент A[1]: 4

Введите элемент A[2]: 3

Введите элемент A[3]: 2

Введите элемент A[4]: 1

Введите элемент A[5]: 8

Введите элемент A[6]: 9

Введите элемент A[7]: 4

Введите элемент A[8]: 2

Введите элемент A[9]: 5

4

Process finished with exit code 0

Постановка задачи2: Дан список размера N. Найти номера двух ближайших элементов из этого списка (то есть элементов с наименьшим модулем разности) и вывести эти номера в порядке возрастания.

Текст программы2:

```
def list_1(arr):
    try:
        if len(arr) < 2:
            raise ValueError("Список должен содержать как минимум два
элемента.")

        min_diff = float('inf') # Начальное значение для минимальной
разности
        indices = (0, 1) # Начальные индексы

        for i in range(len(arr)):
            for j in range(i + 1, len(arr)):
                diff = abs(arr[i] - arr[j]) # Вычисляем модуль разности
                if diff < min_diff: # Если разность меньше минимальной
                    min_diff = diff
                    indices = (i, j) # Обновляем индексы

        return sorted(indices) # Возвращаем индексы в порядке возрастания

    except ValueError as e:
        print(f"Ошибка: {e}")
    except Exception as e:
        print(f"Произошла непредвиденная ошибка: {e}")

def main():
    try:
        input_list = input("Введите элементы списка, разделенные пробелами:
")
        arr = list(map(float, input_list.split())) # Преобразуем ввод в
список чисел

        b = list_1(arr)
        if b:
            print(f"Индексы двух ближайших элементов: {b}")

    except ValueError:
        print("Ошибка: Пожалуйста, введите только числа, разделенные
пробелами.")
    except Exception as e:
        print(f"Произошла непредвиденная ошибка: {e}")

if __name__ == "__main__":
    main()
```

Терминал2:

C:\Users\User\AppData\Local\Microsoft\WindowsApps\python3.11.exe "D:\колледж\алгоритмы и программирование\Pz Sapronov Daniil\PZ_6\PZ_6_2.py"

Введите элементы списка, разделенные пробелами: 2 3 5 4 8 6 3 2

Индексы двух ближайших элементов: [0, 7]

Process finished with exit code 0

Постановка задачи3: Дан список A размера N и целое число K ($1 < K < 4$, $K < N$). Осуществить циклический сдвиг элементов списка влево на K позиций (при этом A_N перейдет в A_{N-K} , A_{N-1} — в A_{N-K-1} , ..., A_1 — в A_{N-K+1}). Допускается использовать вспомогательный список из 4 элементов.

Текст программы3:

```
# Дан список A размера N и целое число K (1 < K < 4, K < N).
# Осуществить циклический сдвиг элементов списка влево на K позиций (при этом
# AN перейдет в AN_K, AN-1 — в AN-K-1, ..., A1 — в AN-K+1).
# Допускается использовать вспомогательный список из 4 элементов.
def find_positions(N, K):
    positions = []
    for i in range(N):
        positions.append((i + K) % N)
    return positions

# Пример использования
N = int(input("Введите N"))
K = int(input("Введите K"))
positions = find_positions(N, K)
print(positions)
```

Терминал3:

C:\Users\User\AppData\Local\Microsoft\WindowsApps\python3.11.exe "D:\колледж\алгоритмы и программирование\Pz Sapronov Daniil\PZ_6\PZ_6_3.py"

Введите N10

Введите K5

[5, 6, 7, 8, 9, 0, 1, 2, 3, 4]

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с использованием функций в PyCharm Community Edition . Были использованы языковые конструкции if for while def

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub