

Практическое занятие №6

Тема: 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 задачи №1.

Дан список размера N и целые числа от K и L ($1 < K < L < N$). Найти сумму всех элементов списка, кроме элементов с номерами от K до L включительно.

Постановка задачи №2.

Дан целочисленный список размера N . Если он является перестановкой, то есть содержит все числа от 1 до N , то вывести 0; в противном случае вывести номер первого недопустимого элемента.

Постановка задачи №3.

Дано множество A из N точек на плоскости и точка B (точки заданы своими координатами x, y). Найти точку из множества A , наиболее близкую к точке B . Расстояние R между точками с координатами (x_1, y_1) и (x_2, y_2) вычисляется по формуле: $R = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. Для хранения данных о каждом наборе точек следует использовать по два списка: первый список для хранения абсцисс, второй — для хранения ординат.

Тип алгоритма: циклический с использованием функции.

Текст программы №1.

```
"""Практическая работа №6 Складов Владимир Группа ИС-23."""
```

```
"""Постановка задачи
```

```
    Дан список размера  $N$  и целые числа от  $K$  и  $L$  ( $1 < K < L < N$ ). Найти сумму  
    всех элементов списка,
```

```
    кроме элементов с номерами от  $K$  до  $L$  включительно."""
```

```
from random import randint
```

```
"""Импортирование библиотек(модулей) random функции randint."""
```

```
def numbers_random(number_1, number_2, number_3): #Функция
```

```
number_random где в списке генерируется список случайных значений.
```

```
    list_1 = [] # Создание пустого списка
```

```
for number in range(0, number_1): #Цикл пока индекс в диапазоне
списка(от 1 до 100).
    list_1.append(randint(1, 100))    # Происходит добавления в список
всех значений с индексом number.
    print('Изначальный список: ', list_1) # Вывод изначального списка
    del list_1[number_3:number_2+1]      # Удаление в списке значений от K
до L
    print('Список после удаления индексов от k до L: ', list_1)    # Вывод
списка после удаления значений от k до l
    return sum(list_1)          # Возвращение СУММЫ списка (итоговое
значение после вычета k and L)

'''Ввод/Вывод данных'''
number_input = int(input('Введите размер списка N: '))
number_L = int(input('Введите число L, оно должно быть меньше N: '))
number_K = int(input('Введите число K, оно должно быть больше 1 и
меньше L: '))
if number_L >= number_input or number_L <= number_K:
    print('Error')
else:
    print(numbers_random(number_input, number_L, number_K))
```

Текст программы №2.

```
'''21 Вариант Складов'''
'''
```

Дан целочисленный список размера N. Если он является перестановкой,
то есть
содержит все числа от 1 до N, то вывести 0; в противном случае вывести
номер
первого недопустимого элемента.'''

```
import random
'''Импортирование библиотеки random'''
N = int(input('Введите значение: ')) # Переменной присваивается значение
5
a = [random.randrange(1,N+1) for i in range(N)] # В переменной образуются
рандомные значения от 1 до 5 в списке
print("N>>:",N) # Вывод N
print(">>",a) # Вывод списка a
```

```
L = sorted(a) # Сортировка списка a
for i in range(1,N+1) : # Цикл проходится по каждому значению от 1 до
вводимого значения
    if i != L[i-1] : # Проверка условия, если i (индекс) не равен списку по
индексу i-1
        break      # То произойдет остановка цикла
if i == N :        # Проверка условия если i == N
    print("0")      # Вывод 0
else:              # Условие иначе
    for i in range(1,N+1) : # Цикл которых проходится по каждому значению
от 1 до вводимого значения
        if i != a[i-1] : # Проверка условия, что i (индекс) не равен списку a
по индексу i-1
            print("first wrong:",a[i-1]) # Вывод первого не совпавшего числа
            break                      # Остановка цикла
```

Текст программы №3.

```
import random
"21 Вариант Складов"
"Постановка задачи.....
Дано множество A из N точек на плоскости и точка B (точки заданы
своими
координатами x, y). Найти точку из множества A, наиболее близкую к
точке B.
Расстояние R между точками с координатами (x1, y1) и (x2, y2)
вычисляется по
формуле:

$$R = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

.
Для хранения данных о каждом наборе точек следует использовать по
два списка: первый
список для хранения абсцисс, второй — для хранения ординат."
```

```
def rasstoyanie(x1, x2, y1, y2): ####Функция которая принимает 4 параметра.
    R = (x2 - x1) + (y2 - y1) ####Формула по которой находится расстояние
    list_2.append(R)          ####Добавление в список значения переменной R
try: ####Блок обработчика исключений, в нём выполняется код, без
ошибок
    B = []    #### Переменная B ссылается на пустой список
```

```
c = int(input('Введите размер списка: '))  ### В переменную c
записываются вводимое значение пользователем в int
list_2 = []  ### Создание пустого списка, переменная list_2 ссылается
на него.
if c % 2 == 0:  ### Условие что вводимое пользователем число является
чётным, тк точки должны иметь x и y координаты
    A = [random.randint(0, 100) for a in range(c)]  ### Создание списка в
котором генерируется рандомное количество
### чисел до вводимого пользователем числа. В диапазоне от 0-100.
else:  ### Условие иначе если не выполняется условие if
    print('Ошибка')  ### Вывод Ошибка

"Ввод данных пользователем в переменную a,b в int формате"
a, b = int(input('Введите x координату для точки B: ')), int(input('Введите
y координату для точки B: '))
B.append(a)  ###Добавление в список значения переменной a - это x
координата.
B.append(b)  ###Добавление в список значения переменной b - это y
координата.
abscisya = []  ### Создание пустого списка в переменной abscisya в нём
хранятся все x
ordinati = []  ### Создание пустого списка в переменной ordinati в нём
хранятся все y
"Цикл который добавляет в список значения абсцисс"
for i in A[0:-1:2]:  ### Пока i в списке A от 0 значения до конечного c
шагом 2
    abscisya.append(i)  ### Добавление в список abscisya значения i
"Цикл который добавляет в список значения ординат"
for number in A[1:-1:2]:  ### Пока i в списке A от 1 значения до конечного
с шагом 2
    ordinati.append(number)  ### Добавление в список ordinaty значения i
ordinati.append(A[-1])  ### Добавление конечного значения из списка A в
список ordinaty
except NameError:  ### Обработчик исключений который выводит ошибку в
случае, если переменной c не существует(то есть
# введено не чётное значение
print('Ошибка')  ###Вывод ошибки
except ValueError:  ### Обработчик исключений, который выводит ошибку,
в случае если введённое число c не в int типе
```

```

print('Ошибка') ##№ Вывод ошибки
else: ### Условие иначе, где выполняется продолжение блока try
    print('Количество всех точек по y в массиве A:', ordinati) ### Вывод
    списка ordinati в котором хранятся все y
    print('Количество всех точек по x в массиве A:', abscisya) ### Вывод
    списка abscisya в котором хранятся все x
    print('Массив A: ', A) ### Вывод всего списка A
    print(f'Координаты точки B: x->{B[0]}, y->{B[1]}') ### Вывод
    координаты точки в списке B, x и y.
    i = 0 ### Счётчик, переменная в которой изначально значение 0
    while i < c // 2: ### Цикл: пока значение в переменной i меньше c
    делённое на 2, чтобы брались пары x-y
        x = abscisya[i] ### Переменная x ссылается на список abscisya в
        который добавляется значения счётчика i
        y = ordinati[i] ### Переменная y ссылается на список ordinati в
        который добавляется значения счётчика i
        i += 1 ### Поэтапное прибавление к счётчику значения 1
        rasstoyanie(B[0], x, B[1], y) ### Вызов функции rasstoyanie в которую
        передаются значения переменных x, y и
        ### Значения списка B по начальному индексу. Значения списка B по
        конечному индексу всего 2 знач. x и y.
    else: ### Условие иначе в котором выводится список всех точек и
    ближайшая точка к точке заданной в списке B
        i = 0 ### Переменная i - счётчик в которой изначально значение это 0
        while i < c // 2: ### Цикл: пока значение в переменной i меньше c
        делённое на 2, чтобы брались пары x-y
            x = abscisya[i] ### Переменная x ссылается на список abscisya в
            который добавляется значения счётчика i
            y = ordinati[i] ### Переменная y ссылается на список ordinati в
            который добавляется значения счётчика i
            if x - B[0] + y - B[1] == min(list_2): ## Проверка условие что
            выражение x - B[0] + y - B[1] равняется
            # минимальному значению в списке list_2 -> min(list_2)
            print(f'Список расстояния точек: {list_2}') ### Вывод всех
            расстояний точек в списке list_2
            print(f'Ближайшие точки к точке B это: x->{x}, y->{y}') ###
            Вывод ближайшей точки к точке в списке B
            i += 1 ### Поэтапное прибавление к счётчику значения 1

```

Протокол работа программы №1.

Ввод:

Введите размер списка N: 5

Введите число L, оно должно быть меньше N: 7

Введите число K, оно должно быть больше 1 и меньше L: 3

Вывод: Ошибка

Ввод:

Введите размер списка N: 5

Введите число L, оно должно быть меньше N: 4

Введите число K, оно должно быть больше 1 и меньше L: 3

Вывод:

Изначальный список: [47, 10, 57, 31, 83]

Список после удаления индексов от k до L: [47, 10, 57]

Сумма списка: 165

Протокол работа программы №2.

Ввод:

Введите значение: 8

Вывод:

N>>: 8

>> [5, 4, 8, 2, 2, 7, 6, 6]

first wrong: 5

Ввод:

Введите значение: 54

Вывод:

N>>: 54

>> [28, 3, 27, 50, 45, 9, 28, 3, 17, 7, 46, 2, 48, 52, 28, 35, 3, 20, 54, 9, 45, 1, 4, 51, 14, 30, 6, 19, 26, 9, 11, 11, 39, 43, 35, 13, 4, 14, 8, 5, 16, 10, 18, 32, 27, 51, 16, 18, 34, 10, 4, 27, 12, 6]

first wrong: 28

Протокол работа программы №3.

Ввод:

Введите размер списка: 6

Введите x координату для точки В: 4

Введите y координату для точки В: 3

Вывод:

Количество всех точке по y в массиве А: [37, 41, 90]

Количество всех точке по x в массиве А: [1, 7, 32]

Массив А: [1, 37, 7, 41, 32, 90]

Координаты точки В: x->4, y->3

Список расстояния точек: [31, 41, 115]

Ближайшие точки к точке В это: x->1, y->37

Ввод:

Введите размер списка: 5

Вывод: Ошибка

Ввод:

Введите x координату для точки В: 6

Введите y координату для точки В: 5

Вывод:

Ошибка

Вывод: в процессе выполнения практического занятия выработал навыки
Студент группы ИС-23 Складов В.Д

составления программ циклической структуры(с использованием циклов и списков) в IDE PyCharm Community. Были использованы языковые конструкции while, if, else, try, except, list, for, def.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.

Готовые программные коды выложены на GitHub.