Студент группы ИС-23 Старчиков А.Р

**Практическое занятие № 6**

**Тема:** составление программ со списками в IDE PyCharm Community.

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

**Постановка задачи № 1:**

Дан список ненулевых целых чисел размера N. Проверить, чередуются ли в нем положительные и отрицательные числа. Если чередуются, то вывести 0, если нет, то вывести порядковый номер первого элемента, нарушающего закономерность.

**Текст программы:**

*"""Дан список ненулевых целых чисел размера N.  
Проверить, чередуются ли в нем положительные и отрицательные числа.  
Если чередуются, то вывести 0, если нет, то вывести порядковый номер первого элемента, нарушающего закономерность."""*def check(lst):  
 for i in range(len(lst) - 1):  
 if lst[i] \* lst[i + 1] > 0: # проверяем, одного ли знака соседние числа  
 return i + 1 # возвращаем порядковый номер(индекс)  
 return 0 # числа чередуются  
  
  
A = [1, -2, 3, -4, 5, -6]  
print("Список A:", A)  
print(check(A))

**Протокол работы программы:**

Список A: [1, -2, 3, -4, 5, -6]

0

Process finished with exit code 0

**Постановка задачи № 2:**

Дан целочисленный список A размера N (< 15). Переписать в новый целочисленный список B все элементы с нечетными порядковыми номерами (1,3,...) и вывести размер полученного списка B и его содержимое. Условный оператор не использовать.

Студент группы ИС-23 Старчиков А.Р

**Текст программы:**

*"""Дан целочисленный список A размера N (< 15). Переписать в новый целочисленный  
список B все элементы с нечетными порядковыми номерами (1,3,...) и вывести  
размер полученного списка B и его содержимое. Условный оператор не использовать."""*import random  
N = random.randint(1, 14) # Создание случайной длины списка в диапозоне  
A = [random.randint(-100, 100) for i in range(N)] # Создание случайных элементов списка в диапозоне  
print("Список A:", A)  
B = A[::2] # Занесение нечетных элементов в новый список  
print("Размер списка B:", len(B))  
print("Список B:", B)

**Протокол работы программы:**

Список A: [4, 1, 80, -80, 69, -90, -70, -63, 74, 16, -65]

Размер списка B: 6

Список B: [4, 80, 69, -70, 74, -65]

Process finished with exit code 0

**Постановка задачи № 3:**

Дано множество A из N точек (N > 2, точки заданы своими координатами x, у). Найти такую точку из данного множества, сумма расстояний от которой до остальных его точек минимальна, и саму эту сумму. Расстояние R между точками с координатами (x1, y1) и (x2, у2) вычисляется по формуле: R = √(x2 – x1)2 + (у2 – y1)2 . Для хранения данных о каждом наборе точек следует использовать по два списка: первый список для хранения абсцисс, второй — для хранения ординат.

**Текст программы:**

*"""  
Дано множество A из N точек (N > 2, точки заданы своими координатами x, у).  
Найти такую точку из данного множества, сумма расстояний от которой до остальных его точек минимальна, и саму эту сумму.  
Расстояние R между точками с координатами (x1, y1) и (x2, у2) вычисляется по формуле: R = √(x2 – x1)2 + (у2 – y1)2 .  
Для хранения данных о каждом наборе точек следует использовать по два списка: первый список для хранения абсцисс, второй — для хранения ординат.  
"""*import math  
  
x\_cords = [5, 8, 16]  
y\_cords = [3, 4, -9]  
distance\_sum = []  
if len(x\_cords) > 2 and len(y\_cords) > 2: # Проверка, что есть минимум 3 точки  
 for x, y in zip(x\_cords, y\_cords):  
 distances = [] # Cписок для хранения расстояний от текущей точки до всех остальных  
 c\_point = (x, y)  
 for x1, y1 in zip(x\_cords, y\_cords):  
 point = (x1, y1)  
 if point != c\_point: # Проверка, что это не текущая точка  
 distance = math.sqrt((x1-x)\*\*2+(y1-y)\*\*2)  
 distances.append(distance)  
 distances\_sum = sum(distances)  
 distance\_sum.append(distances\_sum)  
 min\_sum = min(distance\_sum)  
 min\_sum\_point\_index = distance\_sum.index(min\_sum) # Индекс точки с минимальной суммой расстояний  
 min\_sum\_point = (x\_cords[min\_sum\_point\_index], y\_cords[min\_sum\_point\_index])  
 print("Точка минимальной суммы расстояний:", min\_sum\_point)  
 print("Минимальная сумма расстояний:", min\_sum)  
else:  
 print("Вы ввели меньше трёх точек")

**Протокол работы программы:**

Точка минимальной суммы расстояний: (8, 4)

Минимальная сумма расстояний: 18.426615182642127

Process finished with exit code 0

**Вывод:** в процессе выполнения практического занятие выработал навыки составления программ со списками в IDE PyCharm Community. Были использованы языковые конструкции цикл for, if, else.  
Выполнены разработка кода, отладка ,тестирование, оптимизация, программного кода.  
Готовые программные коды выложены на GitHub.