

Практическое занятие № 6 ЗАДАЧА 1

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.

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

Текст программы:

```
import random
# -*- coding: utf8 -*-
# ПЗ 6 вариант 26 задание 2
# Дан список ненулевых целых чисел размера N. Проверить, чередуется ли в нем
# положительные и отрицательные
# числа. Если чередуются то вывести 0, если нет, то вывести порядковый номер
# первого элемента нарушающего закономерность

# генерация списка ненулевых чисел
allowed_values = list(range(-5, 5+1))
allowed_values.remove(0)

# can be anything in {-5, ..., 5} \ {0}:
random_list = [random.choice(allowed_values) for i in range(1,
random.randint(2, 5))]
# random_list = [-2,4,3]

# Функция с циклом для проверки созданного цикла
def check_seq(rand_list):
    for i in range(0, len(rand_list)-1):
        if i % 2 == 0:
            if rand_list[i] > 0 and rand_list[i+1] < 0 or rand_list[i] < 0
and rand_list[i + 1] > 0:
                continue
            else:
                return rand_list[i+1]
        else:
            if rand_list[i] < 0 and rand_list[i + 1] > 0 or rand_list[i] > 0
and rand_list[i+1] < 0:
                continue
            else:
                return rand_list[i+1]
    return 0

# Выводы
print(f"[INFO]{random_list}")
print(check_seq(rand_list=random_list))
```

Протокол работы программы:

[INFO] [-4, 4, 5, -5]

5

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с функциями в IDE PyCharm Community

Используется модуль random для генерации списка ненулевых чисел данного размера. В данном случае, мы создаем список из чисел от -5 до 5, исключая 0.

Переменная random_list хранит случайно сгенерированный список ненулевых чисел.

Далее определена функция check_seq(rand_list), которая выполняет проверку чередования положительных и отрицательных чисел в созданном списке.

Функция проходит по элементам списка и проверяет, чередуются ли положительные и отрицательные числа. Если обнаруживается несоответствие чередования, то функция возвращает порядковый номер первого элемента, нарушающего закономерность. Если чередование есть, функция возвращает 0.

Наконец, программа выводит сгенерированный список и результат проверки чередования через вызов функции check_seq.

Если список random_list выглядит, например, как [-2, 4, 3], то функция вернет 0, так как здесь чередуются положительные и отрицательные числа.

Готовые программные коды выложены на GitHub.

Практическое занятие № 6 ЗАДАЧА 2

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.

Дан целочисленный список А размера $N < 15$. Переписать в новый целочисленный список В все элементы с нечетными порядковыми номерами (1,3,5) и вывести размер полученного списка В и его содержимое. Условный оператор не использовать

Текст программы:

```
from random import randint as ri
# -*- coding: utf8 -*-

# ПЗ 6 вариант 26 задание 1
# Дан целочисленный список А размера N < 15. Переписать в новый целочисленный
# список В все элементы с нечетными
# порядковыми номерами (1,3,5) и вывести размер полученного списка В и его
# содержимое. Условный оператор не использовать

A = [ri(1, 100) for i in range(0, ri(1,15))]
B = A[1::2]

print(f"[INFO] начальный список: {A}\n")
print(f"[INFO] полученный список В с нечетными индексами: {B}\n\n[INFO]
размер полученного списка В: {len(B)}")
```

Протокол работы программы:

[INFO] начальный список: [54, 48, 88]

[INFO] полученный список В с нечетными индексами: [48]

[INFO] размер полученного списка В: 1

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с функциями в IDE PyCharm Community

Сначала используется импорт функции `randint` из модуля `random` под псевдонимом `ri` для генерации случайных целых чисел.

Далее создается список `A` случайной длины, содержащий случайные целочисленные значения, и затем создается список `B` путем извлечения элементов из списка `A` с нечетными индексами.

Наконец, программа выводит начальный список `A`, полученный список `B` с нечетными индексами и размер полученного списка `B`.

Наконец, программа выводит сгенерированный список и результат проверки чередования через вызов функции `check_seq`.

Готовые программные коды выложены на [GitHub](#).

Практическое занятие № 6 ЗАДАЧА 3

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.

Дано множество A из N точек ($N > 2$), точки заданы своими координатами (x , y). Найти такую точку из данного множества, сумма расстояний от которой до остальных его точек минимальна, и саму эту сумму.

Текст программы:

```
from itertools import combinations
from random import randint as ri
# -*- coding: utf8 -*-
# ПЗ 6 вариант 26 задание 3
# Дано множество A из N точек (N > 2), точки заданы своими координатами (x,
# y). Найти
# такую точку из данного множества, сумма расстояний от которой до остальных
# его
# точек минимальна, и саму эту сумму.

# Задаем исходное множество точек A
A = [(ri(-10, 10), ri(-10, 10)) for i in range(0, ri(1, 10))] # Пример
входных данных, можно заменить на свои значения

# Функция для вычисления расстояния между двумя точками
def distance(point1, point2):
    return ((point1[0] - point2[0]) ** 2 + (point1[1] - point2[1]) ** 2) **
0.5

# Создаем список всех возможных комбинаций точек для подсчета расстояний
point_combinations = list(combinations(A, 2))

# Инициализируем переменные для хранения наилучшей точки и наименьшей суммы
расстояний
best_point = A[0]
min_distance_sum = float('inf')

# Цикл для вычисления сумм расстояний от каждой точки до остальных
for point in A:
    distance_sum = 0
    for other_point in A:
        if other_point != point:
            distance_sum += distance(point, other_point)
    if distance_sum < min_distance_sum:
        min_distance_sum = distance_sum
        best_point = point

# Выводим найденную точку и сумму расстояний
print(f"Все точки: {A}")
print("Наилучшая точка:", best_point)
print("Минимальная сумма расстояний:", min_distance_sum)
```

Протокол работы программы:

Все точки: $[(-9, 9), (-5, 5), (1, 3)]$

Наилучшая точка: $(-5, 5)$

Минимальная сумма расстояний: 11.98140956982914

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с функциями в IDE PyCharm Community

Используются модули `itertools` для работы с комбинациями точек и `random` для генерации случайных чисел.

Сначала создается список A из случайно сгенерированных точек с координатами (x, y) .

Затем определена функция `distance(point1, point2)`, которая вычисляет расстояние между двумя точками по формуле расстояния между двумя точками в декартовой системе координат.

После этого создается список всех возможных комбинаций точек из множества A для вычисления расстояний между ними.

Далее инициализируются переменные для хранения наилучшей точки и наименьшей суммы расстояний.

Затем проводится цикл, в котором для каждой точки вычисляется сумма расстояний от нее до остальных точек. Если текущая сумма меньше минимальной, то обновляются значения переменных `min_distance_sum` и `best_point`.

Наконец, программа выводит все точки из множества A , а также найденную наилучшую точку и минимальную сумму расстояний до остальных точек из данного множества.

Готовые программные коды выложены на [GitHub](#).