Студент группы ПОКС-22 Смирнов Ю.Н.

**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.

**Задача 1.**

**Постановка задачи.**

Разработать программу, которая должна проверять, чередуются ли в списке размера N четные и нечетные числа. Если чередуются, то вывести 0, если нет, то вывести порядковый номер первого элемента, нарушающего закономерность.  
**Тип алгоритма:** смешанный

**Текст программы:**

# программа должна проверять, чередуются ли в списке размера N четные и нечетные числа.  
# Если чередуются, то вывести 0, если нет, то вывести порядковый номер первого элемента,  
# нарушающего закономерность  
import random  
  
n = int(input('Введите длину списка: '))  
mylist = [random.randint(1, 100) for el in range(n)]  
# генерируем список с рандомными целыми числами от 1 до 100, длиной n  
print(f'Список: {mylist}')  
result = 0  
for i in range(1, len(mylist)):  
 if mylist[i - 1] % 2 == mylist[i] % 2:  
 # если остаток от деления на два предыдущего элемента списка  
 # равен остатку от деления на два текущего элемента списка, то:  
 result = i  
 break  
print(result if result == 0 else f'Последовательность нарушилась на {result} элементе списка')

**Протокол работы программы:**

Введите длину списка: 7

Список: [100, 73, 47, 63, 56, 20, 25]

Последовательность нарушилась на 2 элементе списка

Process finished with exit code 0

**Задача 2**

**Постановка задачи:**

Разработать программу, которой дается два списка длиной N. Сформировать новый список С того же размера, каждый элемент которого равен максимальному из элементов списков A и B.

**Тип алгоритма:** смешанный

**Текст программы:**

# программе дается два списка длиной N.  
# Сформировать новый список С того же размера, каждый элемент которого равен  
# максимальному из элементов списков A и B  
import random  
  
n = int(input('Введите длину списков: '))  
a = [random.randint(1, 100) for i in range(n)]  
b = [random.randint(1, 100) for i in range(n)]  
print(f'Списки:\n{a}\n{b}')  
# генерируем два списка с рандомными целыми числами от 1 до 100, длиной n  
c = [max(a[i], b[i]) for i in range(n)]  
# генерируем список, i-ый элемент которого равен максимальному из i-ых элементов списков a и b  
print(f'Получившийся список: {c}')

**Протокол работы программы:**

Введите длину списков: 7

Списки:

[62, 58, 72, 52, 39, 76, 73]

[35, 24, 36, 36, 98, 40, 44]

Получившийся список: [62, 58, 72, 52, 98, 76, 73]

Process finished with exit code 0

**Задача 3**

**Постановка задачи:**

Разработать программу, которой даны два множества A и B, состоящие соответственно из N1 и N2 точек (точки заданы своими координатами x и y). Найти минимальное расстояние между точками этих множеств и сами точки, расположенные на этом расстоянии (вначале выводится точка из множества А, затем точка из множества В).

**Тип алгоритма:** смешанный

**Текст программы:**

# Программе даны два множества A и B, состоящие соответственно из N1 и N2 точек  
# (точки заданы своими координатами x и y). Найти минимальное расстояние между точками этих множеств и сами точки,  
# расположенные на этом расстоянии (вначале выводится точка из множества А, затем точка из множества В).  
import random  
  
n1 = int(input('Введите n1: '))  
n2 = int(input('Введите n2: '))  
a = {(random.randint(-100, 100), random.randint(-100, 100)) for i in range(n1)}  
b = {(random.randint(-100, 100), random.randint(-100, 100)) for j in range(n2)}  
print(f'\nМножество A: {a}')  
print(f'Множество B: {b}')  
# генерируем два множества с точками, координаты которых лежат в диапазоне от -100 до 100,  
# длиной n1 и n2 соответственно  
minR = 99999 # берем минимальное расстояние равное 99999, чтобы потом сравнивать с ним другие расстояния  
minR\_point1 = tuple() # создаем два кортежа, в которых будем хранить точки, между которыми минимальное расстояние  
minR\_point2 = tuple()  
for point1 in a:  
 for point2 in b:  
 if ((point2[0] - point1[0]) \*\* 2 + (point2[1] - point1[1]) \*\* 2) \*\* (1 / 2) < minR:  
 # сравниваем минимальное расстояние между двумя точками с прошлым минимальным расстоянием  
 minR = ((point2[0] - point1[0]) \*\* 2 + (point2[1] - point1[1]) \*\* 2) \*\* (1 / 2)  
 minR\_point1 = point1  
 minR\_point2 = point2  
print(f'\nМинимальное расстояние между точками = {minR}')  
print(f'Точки:\n'  
 f'Из множества А: {minR\_point1},\n'  
 f'Из множества В: {minR\_point2}')

**Протокол работы программы:**

Введите n1: 6

Введите n2: 12

Множество A: {(-75, -12), (-20, 68), (10, 53), (-90, 76), (-8, -35), (90, 11)}

Множество B: {(45, 78), (32, -96), (-37, 64), (-13, -74), (70, 92), (2, 99), (94, 22), (97, 67), (-2, -77), (33, 69), (-92, 82), (11, -28)}

Минимальное расстояние между точками = 6.324555320336759

Точки:

Из множества А: (-90, 76),

Из множества В: (-92, 82)

Process finished with exit code 0

**Вывод:** в процессе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ со списками в IDE PyCharm Community.