

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.

Разработать три программы. Первая проверяет чередование чётных и нечётных чисел в данном целочисленном списке. Вторая формирует новый список, каждый элемент которого равен максимальному элементу из элементов списков А и В. Третья находит минимальное расстояние между точками и сами точки. Точки берутся из двух множеств А и В, где точки заданы своими координатами x, y.

Текст первой программы:

```
# Вариант 25
# Дан целочисленный список размера N.
# Проверить, чередуются ли в нём чётные и нечётные числа.
# Если чередуются, то вывести 0, если нет, то вывести порядковый номер первого
# нарушающего закономерность.

from random import randint

i = 0
N = randint(1, 10)
A = []

while i < N:
    A.append(randint(0, 100))
    i += 1

print("Данный список:", A)

for x in A:
    if A.index(x) == N-1:
        y = A[A.index(x) - 1]
    else:
        y = A[A.index(x) + 1]
    if (x % 2 == 0) and (y % 2 == 0):
        print("Порядковый номер числа:", A.index(y) + 1)
        break
    elif (x % 2 != 0) and (y % 2 != 0):
        print("Порядковый номер числа:", A.index(y) + 1)
        break
else:
    print(0)
```

Протокол работы программы:

Данный список: [20, 58, 36, 71, 97, 54, 89, 78, 46, 48]

Порядковый номер числа: 2

Program finished with exit code 0

Текст второй программы:

```
# Вариант 25
# Даны два списка А и В одинакового размера N.
# Сформировать новый список С того же размера,
# каждый элемент которого равен максимальному элементу из элементов списков А и В.

from random import randint

i = 0
N = randint(1, 10)
A = []
B = []
C = []

while i < N:
    A.append(randint(0, 100))
    i+=1
else:
    i = 0

while i < N:
    B.append(randint(0, 100))
    i+=1

for elem in A:
    if elem > B[A.index(elem)]:
        C.append(elem)
    else:
        C.append(B[A.index(elem)])

print("Исходные списки А и В:", A, B, "Список С:", C, sep="\n")
```

Протокол работы программы:

Исходные списки А и В:

[4, 92, 58, 3, 44]

[34, 2, 76, 13, 46]

Список С:

[34, 92, 76, 13, 46]

Program finished with exit code 0

Текст третьей программы:

```
# Вариант 25
# Даны множества А и В, состоящие соответственно из N1 и N2 точек (точки заданы своими координатами x, 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).
# 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.
```

```
from random import randint
from math import sqrt

i = 0
l = 0
N = randint(1, 5)
A = list(((1+elem)*randint(0, 25), (1+elem)*randint(0, 25)) for elem in range(N))
X_A = []
```

```

Y_A = []
B = list({((1+elem)*randint(0, 25), (1+elem)*randint(0, 25)) for elem in
range(N)})
X_B = []
Y_B = []
_R = []
_XY = []

def int_r(num):
    num = int(num + (0.5 if num > 0 else -0.5))
    return num

def rasst(x1, x2, y1, y2):
    R = int_r(sqrt((x2-x1)**2+(y2-y1)**2))
    _R.append(R)
    _XY.append([(x1, y1), (x2, y2)])

for e in A:
    X_A.append(e[0])
for e in A:
    Y_A.append(e[1])

for e in B:
    X_B.append(e[0])
for e in B:
    Y_B.append(e[1])

while i < N:
    while l < N:
        rasst(X_A[i], X_B[l], Y_A[i], Y_B[l])
        l += 1
    i += 1
    l = 0

print("Множество A: \n", A)
print("Множество B: \n", B)
print("Все расстояния: \n", _R)
print("Минимальное расстояние между точками:", min(_R))
print("Точки минимального расстояния (A и B):", *_XY[_R.index(min(_R))])

```

Протокол работы программы:

Множество A:

[(46, 36), (45, 42), (24, 20)]

Множество B:

[(36, 12), (69, 42), (11, 13)]

Все расстояния:

[26, 24, 42, 31, 24, 45, 14, 50, 15]

Минимальное расстояние между точками: 14

Точки минимального расстояния (A и B): (24, 20) (36, 12)

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с списками в IDE PyCharm Community. Были использованы языковые конструкции `while`, `if`, `def`, `for`, `list`, `set`.

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