Студент группы ПОКС-23Галамян Э.Р

Отчет

Практическое занятие № 6

Тема: Составление программ циклической структуры в IDE PyCharm Community.

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

Постановка задачи.

1. Дан целочисленный список размера 10. Вывести вначале все содержащиеся в данном списке четные числа в порядке возрастания их индексов, а затем — все нечетные числа в порядке убывания их индексов.
2. Дан список размера N. Найти количество участков, на которых его элементы монотонно убывают.
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 (x1, y1) и (x2, y2) вычисляется по формуле: R = √(x2 – x1)2 + (y2 – y1)2 .

Тип алгоритма : циклический .

Текст программы 1:

```
# Дан целочисленный список размера 10. Вывести вначале все содержащиеся в данном
# списке четные числа в порядке возрастания их индексов, а затем — все нечетные числа
# в порядке убывания их индексов

import random

spisok = []
chet = []
nechet = []


for i in range(0, 10):
   spisok.append(random.randint(-100, 100))
   if spisok[i] % 2 == 0:
      chet += [spisok[i]]
   elif spisok[i] % 2 == 1:
      nechet += [spisok[i]]


print("Весь список : ", spisok)
print('Все четные числа :', chet)
nechet.reverse()
print("Все не четные числа :", nechet)
```

Протокол работы программы:

Весь список : [48, 18, 39, 24, 45, -40, -67, 31, 28, -45]
Все четные числа : [48, 18, 24, -40, 28]
Все не четные числа : [-45, 31, -67, 45, 39]

Process finished with exit code 0

Текст программы 2:

```
# Дан список размера N. Найти количество участков, на которых его элементы
# монотонно убывают.

import random

n = int(input("Введите размер списка N = "))
spisok = []
i = 0

while i < n:
   spisok.append(random.randint(-100, 100))
   i += 1

print(spisok)


t = 0
for i in range(0, n):
   if spisok[i-1] < spisok[i]:
      if True:
         t += 1
      else:
         False
   else:
      False
print("Колличесвто участков с монотоным убыванием = :", t
```

Протокол работы программы:

```
Весь список :  [48, 18, 39, 24, 45, -40, -67, 31, 28, -45]
Все четные числа : [48, 18, 24, -40, 28]
Все не четные числа : [-45, 31, -67, 45, 39]

Process finished with exit code 0
```

Текст программы 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 (x1, y1) и (x2, y2) вычисляется по формуле:
# R = √(x2 – x1)2 + (y2 – y1)2

import random
import math

n = int(input("Введите количество точек : "))
A = []
N = []
x = []
y = []

while n > 0:
    a = random.randrange(-100, 100)
    b = random.randrange(-100, 100)
    x += [a]
    y += [b]
    N = [a, b]
    A += [N]
    n -= 1

print("Множество A из точек N : ", A)

l = random.randrange(-100, 100)
k = random.randrange(-100, 100)
B = [l, k]
```

```
print("Точка B : ", B)


so = []

sa = []

rast = []


for i in x:

   sa.append((I - i) ** 2)

for p in y:

   so.append((k - p) ** 2)

rast += [int(math.sqrt(x+y)) for x, y in zip(so, sa)]


print("Ближайшая точка к точке B под номером : ", rast.index(min(rast))+1)

print("Растояние от нее до точки B  = ", min(rast))
```

Протокол работы программы:

```
Введите количество точек : 5
Множество A из точек N :  [[5, -60], [37, 25], [-47, 24], [-93, -92], [-2, 97]]
Точка B :  [-53, 27]
Ближайшая точка к точке B под номером :  3
Растояние от нее до точки B  =  6

Process finished with exit code
```

Вывод: в процессе выполнения практического занятия выработал(а) навыки составления программ циклической структуры в IDE PyCharm Community. Были использованы языковые конструкции while, for

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

Готовые программные коды выложены на GitHub.

Студент группы ПОКС-23Галамян Э.Р