

Практическое занятие № 5

Тема: составление программ со списками в IDE PyCharm Community.

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

Постановка задачи.

Дан список A размера N. Найти максимальный элемент из его элементов с нечетными номерами: A₁, A₃, A₅...

Текст программы:

```
1  # Программа выводит на экран только нечетные элементы списка
2  import random
3  a = []
4  n = int(input('Введите количество элементов в списке: '))
5
6  while n:
7      a.append(random.randint(0, 100))
8      n -= 1
9  print(a)
10 b = []
11 for i in range(len(a)):
12     if (i + 1) % 2 != 0:
13         b.append(a[i])
14 b.sort()
15
16 Max = b[-1]
17 index = int
18 for i in range(len(a) - 1):
19     if a[i] == Max:
20         index = i
21
22 print('Максимальное значение у элемента №', index + 1, '=', Max)
```

Протокол работы программы:

Введите количество элементов в списке: 7

[82, 9, 21, 74, 97, 2, 6]

Максимальное значение у элемента № 5 = 97

Process finished with exit code 0

Задача №2

Дан целочисленный список A размера N (< 15). Переписать в новый целочисленный список B все элементы с порядковыми номерами, кратными трем (3, 6, ...), и вывести размер полученного списка B и его содержимое. Условный оператор не использовать.

Текст программы:

```
1  # Программа выводит только элементы списка, номер которых кратен трем
2  import random
3
4  a = []
5  while True:
6      n = random.randint(0, 14)
7      if n >= 15:
8          print('Ошибка ввода')
9      else:
10         break
11
12  while n:
13      a.append(random.randint(0, 15))
14      n -= 1
15  print(a)
16  b = []
17  o = 2
18  while o <= len(a) - 1:
19      try:
20          b.append(a[o])
21          o += 3
22      except IndexError:
23          pass
24  print(b)
25  print('В списке', len(b), 'элементов')
```

Протокол работы программы:

[8, 9, 6, 3, 2, 4, 3, 15, 6, 0, 13, 11, 9, 0]

[6, 4, 6, 11]

В списке 4 элементов

process finished with exit code 0

Задача №3

Дано множество A из N точек ($N > 2$, точки заданы своими координатами x, y). Найти наименьший периметр треугольника, вершины которого принадлежат различным точкам множества A, и сами эти точки (точки выводятся в том же порядке, в котором они перечислены при задании множества A).

Текст программы:

```

1  # Программа выдает наименьший периметр треугольника, созданного на основе данных точек
2  import math
3  import random
4  X = []
5  Y = []
6  A = [X, Y]
7  P = []
8
9
10 def addpoint(x, y):
11     X.append(x)
12     Y.append(y)
13
14
15 def perimeter(x1, y1, x2, y2, x3, y3):
16     a = math.sqrt(math.pow(x2 - x1, 2) + math.pow(y2 - y1, 2))
17     b = math.sqrt(math.pow(x1 - x3, 2) + math.pow(y1 - y3, 2))
18     c = math.sqrt(math.pow(x3 - x2, 2) + math.pow(y3 - y2, 2))
19
20     p = a + b + c
21     return p
22
23
24 n = int(input('Введите кол-во элементов в списке: '))
25 while n:
26     addpoint(random.randint(-15, 15), random.randint(-15, 15))
27     n -= 1
28
29 print(A)
30 Min = 0
31
32 Min1 = ()
33 Min2 = ()
34 Min3 = ()
35
36 for i in range(len(A[0])):
37     for e in range(len(A[0])):
38         for s in range(len(A[0])):
39             if A[0][i] != A[0][e] and A[0][i] != A[0][s] and A[0][e] != A[0][s] and A[1][i] != A[1][e] and
40                 P.append(perimeter(A[0][i], A[1][i], A[0][e], A[1][e], A[0][s], A[1][s]))
41                 if P[Min] > P[-1]:
42                     Min = len(P)
43                     Min1 = (A[0][i], A[1][i])
44                     Min2 = (A[0][e], A[1][e])
45                     Min3 = (A[0][s], A[1][s])
46
47
48 print('Самый маленький периметр =', P[Min], 'с точками', Min1, Min2, Min3)

```

Протокол работы программы:

Введите кол-во элементов в списке: 3

[[10, 6, 5], [3, -1, 0]]

Самый маленький периметр = 12.902019706710776 с точками (6, 1) (10, 3) (5, 0)

process finished with exit code 0

Вывод: В ходе разработки программы закреплены усвоенные понятия, знания, алгоритмы и основные принципы составления программ со списками в IDE PyCharm Community.