

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 первой задачи.

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

Текст программы:

```

1  # Дан список ненулевых целых чисел размера N. Проверить, чередуются ли в нем
2  # положительные и отрицательные числа. Если чередуются, то вывести 0, если нет, то
3  # вывести порядковый номер первого элемента, нарушающего закономерность.
4
5  N = input("Введите число N: ")
6
7  while type(N) != int: # обработка исключений
8      try:
9          N = int(N)
10         if N < 0:
11             print("Неправильно ввели! Число N должно быть положительным!")
12             N = input("Введите число N:")
13         except:
14             print("Неправильно ввели! Введите целое число!")
15             N = input("Введите число N:")
16
17  numbers = []
18  for i in range(N):
19      num = input(f"Введите число {i+1}: ")
20      while type(num) != int: # обработка исключений
21          try:
22              num = int(num)
23              if num == 0:
24                  print("Неправильно ввели! Введите целое число!")
25                  num = input(f"Введите число {i + 1}: ")
26          except:
27              print("Неправильно ввели! Введите целое число!")
28              num = input(f"Введите число {i+1}: ")
29      numbers.append(num)
30      continue

```

```

33  violation_index = 0
34  for i in range(1, N):
35      if (numbers[i] < 0 and numbers[i-1] < 0) or ( numbers[i] > 0 and numbers[i-1] > 0):
36          violation_index = i + 1
37          break
38
39  print(violation_index)

```

Протокол работы программы:

```

Введите число N: 3
Введите число 1: 1
Введите число 2: -4
Введите число 3: 2
0

Process finished with exit code 0

```

Постановка второй задачи.

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

Текст программы:

```

1  # Дан целочисленный список A размера N (< 15). Переписать в новый целочисленный
2  # список B все элементы с нечетными порядковыми номерами (1,3,...) и вывести
3  # размер полученного списка B и его содержимое. Условный оператор не
4  # использовать.
5
6  N = input("Введите размер списка A: ")
7  while type(N) != int: # обработка исключений
8      try:
9          N = int(N)
10         if N < 0:
11             print("Неправильно ввели! Число N должно быть положительным!")
12             N = input("Введите размер списка A: ")
13         elif N >= 15:
14             print("Неправильно ввели! Число N должно быть строго меньше 15!")
15             N = input("Введите размер списка A: ")
16     except:
17         print("Неправильно ввели! Введите целое число!")
18         N = input("Введите размер списка A: ")
19
20  A = []
21
22  for i in range(N):
23      while True:
24          try:
25              num = int(input(f"Введите элемент {i + 1}: "))
26              A.append(num)
27              break
28          except ValueError:
29              print("Неправильно ввели!")
30
31  B = A[1::2]
32
33  print("Размер списка B:", len(B))
34  print("Содержимое списка B:", B)
35
36

```

Протокол работы программы:

```

Введите размер списка A: 5
Введите элемент 1: 1
Введите элемент 2: 2
Введите элемент 3: 3
Введите элемент 4: 4
Введите элемент 5: 5
Размер списка B: 2
Содержимое списка B: [2, 4]

Process finished with exit code 0

```

Постановка третьей 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:

```

1  # Дано множество A из N точек (N > 2, точки заданы своими координатами x, y). Найти
2  # такую точку из данного множества, сумма расстояний от которой до остальных его
3  # точек минимальна, и саму эту сумму
4
5  import math
6
7  x_coords = []
8  y_coords = []
9
10 N = input("Введите количество точек: ")
11
12 while type(N) != int: # обработка исключений
13     try:
14         N = int(N)
15         if N < 2:
16             print("Неправильно ввели! Количество точек должно быть больше 2!")
17             N = input("Введите число N:")
18     except:
19         print("Неправильно ввели! Введите целое число!")
20         N = input("Введите число N:")
21
22 for i in range(N):
23     x = input(f"Введите координату x для точки {i+1}: ")
24     while type(x) != float:
25         try:
26             x = float(x)
27         except:
28             print("Неправильно ввели! Число должно быть вещественным!")
29             x = input(f"Введите координату x для точки {i+1}: ")
30

```

```

31     y = input(f"Введите координату y для точки {i+1}: ")
32     while type(y) != float:
33         try:
34             y = float(y)
35         except:
36             print("Неправильно ввели! Число должно быть вещественным!")
37             y = input(f"Введите координату x для точки {i+1}: ")
38
39     x_coords.append(x)
40     y_coords.append(y)
41
42     min_total_distance = float('inf')
43     min_point_index = -1
44
45     for i in range(len(x_coords)):
46         total_distance = 0
47         for j in range(len(x_coords)):
48             if i != j:
49                 distance = math.sqrt((x_coords[i] - x_coords[j]) ** 2 + (y_coords[i] - y_coords[j]) ** 2)
50                 total_distance += distance
51             if total_distance < min_total_distance:
52                 min_total_distance = total_distance
53                 min_point_index = i
54
55     print("\nТочка с минимальной суммой расстояний:")
56     print(f" x: {x_coords[min_point_index]}")
57     print(f" y: {y_coords[min_point_index]}")
58     print(f"Минимальная сумма расстояний: {min_total_distance}")
59

```

Протокол работы программы:

```

Введите количество точек: 2
Введите координату x для точки 1: 1.3
Введите координату y для точки 1: 1.5
Введите координату x для точки 2: 2.5
Введите координату y для точки 2: 1.2

Точка с минимальной суммой расстояний:
  x: 1.3
  y: 1.5
Минимальная сумма расстояний: 1.2369316876852983

Process finished with exit code 0

```

Вывод: в процессе выполнения практического занятия выработала навыки составления программ линейной структуры в IDE PyCharm Community. Были использованы языковые конструкции `while`, `try`, `except`, `if`, `else`, `for`.

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

Готовые программные коды выложены на GitHub.