

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. Сформировать и вывести целочисленный список размера 10, содержащий степени двойки от первой до 10-й: 2, 4, 8, 16,

Текст программы:

```
# Сформировать и вывести целочисленный список размера 10, содержащий
# степени
# двойки от первой до 10-й: 2, 4, 8, 16, ...
my_list = [] # создаем пустой список

for i in range(1, 11): # цикл для вычисления степени
    a = 2 ** i
    my_list_new = my_list.append(a) # добавляем значение в конец списка

print(f'Степень двойки до 10 - {my_list}')
```

Протокол работы программы:

Степень двойки до 10 - [2, 4, 8, 16, 32, 64, 128, 256, 512, 1024]

Задача 2

Постановка задачи. Дан список размера N. Найти номера тех элементов списка, которые больше своего левого соседа, и количество таких элементов. Найденные номера выводить в порядке их убывания.

Текст программы:

```
# Дан список размера N. Найти номера тех элементов список, которые больше
своего
# левого соседа, и количество таких элементов. Найденные номера выводить в
# порядке их убывания.
import random

N = input('Введите количество элементов в списке - ')
t = 0
count = 0

while N != int: # проверка на тип переменной
    try:
        N = int(N)
        break
    except ValueError:
        print('Неправильно ввели, ввести нужно число ')
        N = input('Введите количество элементов в списке - ')

new_List = []
while t < N:
    new_List.append(random.randint(-100, 100)) #добавляем в список рандомные
числа
    t += 1

print(f'Получившийся список: {new_List}')

mini_list = [] # здесь будут храниться подходящие под условие элементы
index_list = [] # здесь индексы подходящих элементов
for index in range(1, len(new_List)):
    if new_List[index] > new_List[index - 1]:
        count += 1
        mini_list.append(new_List[index])
        index_list.append(index)
mini_list.sort(reverse=True) # метод для вывода списка в порядке убывания
index_list.reverse() # метод, выводящий список в обратном порядке
print(f'Количество элементов в списке, которые больше своего левого соседа =
{count}')
```

```
print(f'Список данных чисел порядке убывания: {mini_list}')  
print(f'Индексы данных элементов в обратном порядке: {index_list}')
```

Протокол работы программы:

Введите количество элементов в списке - аа

Неправильно ввели, ввести нужно число

Введите количество элементов в списке - 10

Получившийся список: [73, 19, 95, -32, 9, 28, -41, -97, 84, -52]

Количество элементов в списке, которые больше своего левого соседа = 4

Список данных чисел порядке убывания: [95, 84, 28, 9]

Индексы данных элементов в обратном порядке: [8, 5, 4, 2]

Задача 3

Постановка задачи. Дан список размера N. Обнулить элементы списка, расположенные между его минимальным и максимальным элементами (не включая минимальный и максимальный элементы).

Текст программы:

```
# Дан список размера N. Обнулить элементы списка, расположенные между его  
# минимальным и максимальным элементами (не включая минимальный и  
# максимальный элементы).  
import random  
  
N = input('Введите количество элементов в списке - ')  
t = 0  
  
while N != int: # проверка на тип переменной  
    try:  
        N = int(N)  
        break  
    except ValueError:  
        print('Неправильно ввели, ввести нужно число ')  
        N = input('Введите количество элементов в списке - ')
```

```
my_list = []
while t < N:
    my_list.append(random.randint(-100, 100)) #добавляем в список рандомные
    числа
    t += 1

print(fПолучившийся список: {my_list}')

max_digit = max(my_list)
max_index = my_list.index(max_digit)

min_digit = min(my_list)
min_index = my_list.index(min_digit)

print(fМинимальное значение в списке - {min_digit}\nМаксимальное значение в
списке - {max_digit}')

if min_index > max_index: # удаляем из списка определенный срез
    del my_list[max_index + 1 : min_index]
else:
    del my_list[min_index + 1 : max_index]

print(fОбновленный под условие список: {my_list}')
```

Протокол работы программы:

Введите количество элементов в списке - 11

Получившийся список: [-13, -96, -62, -80, 82, 92, 1, 53, -59, 6, -88]

Минимальное значение в списке - -96

Максимальное значение в списке - 92

Обновленный под условие список: [-13, -96, 92, 1, 53, -59, 6, -88]

Вывод: Практическая работа позволила закрепить знания по программированию и алгоритмам, а также развить навыки работы со списками и циклом for в PyCharm Community.