

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.

1. Алгоритм должен выводить элементы из списка (список не имеет ограничений) с чётными номерами(индексом).
2. Алгоритм должен найти элемент из списка a , который наиболее близок к числу R , то есть такой элемент a для которого величина $|a-R|$ является минимальной.
3. Список имеет размер N . Нужно осуществить сдвиг элементов списка влево на одну позицию, при этом исходное значение первого элемента будет утеряно. Также последний элемент полученного списка положить равным 0.

Текст программы:

1. Текст 1-ой программы.

Алгоритм должен выводить элементы из списка (список не имеет ограничений) с чётными номерами(индексом).

```
A = list(range(-10, 10))
s = len(A)
print("Список A = ", A)
```

```
k = 2
while k < s:
    while k < s:
        print("Элемент", A[k], "под номером", k)
        k += 2
    break
```

2. Текст 2-ой программы.

Алгоритм должен найти элемент из списка a , который наиболее близок к числу R , то есть такой элемент a для которого величина $|a-R|$ является минимальной.

```
N = int(input("Введите кол-во элементов в списке: "))
```

```

R = int(input("Введите какое-нибудь целое число к нему будет
подобранное наиболее похожее число из списка: "))
a = list(range(N))

print("Список a = ", a)

min = 0
t = 0 # Переменная t нужна для того, чтобы перемещаться с конца
списка к его началу.
index = 0
for el in a:
    m = abs(a[el] - R)
    index_start = a[el] # Сохраняем 1-ое число.

    index_end = a[-1 - t] # Сохраняем 2-ое число
    n = abs(a[-1 - t] - R)

    t += 1
    if n == 0:
        min = index_end, n
        break
    elif m == 0:
        min = index_start, m
        break
    elif n < m:
        min = index_end, n
    else:
        min = index_start, m
min_1 = list(min)
N_1, M_2 = min_1

print(f"Наболее похожий элемент в списке число {N_1}, так как разница
между {N_1} и", R, "составляет", M_2)

```

3. Текст 3-ей программы.

```

# Список имеет размер N. Нужно осуществить сдвиг элементов списка
влево на одну позицию,
# при этом исходное значение первого элемента будет утеряно. Также
последний элемент полученного списка
# положить равны 0.

```

```

s = int(input("Введите диапазон списка.\n"
              "Введите 1-ое число диапазона: "))
e = int(input("Введите 2-ое число диапазона: "))
A = list(range(s, e))
print("Исходный список A = ", A)

k = 0

```

```

b = 1
while b < len(A):
    A[k] = A[b]
    while b < len(A):
        b += 1
        k += 1
    break
A[-1] = 0
print("Конечный список A = ", A)

```

Протокол работы программы:

1. Протокол PZ_6_1

Список A = [-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Элемент -8 под номером 2

Элемент -6 под номером 4

Элемент -4 под номером 6

Элемент -2 под номером 8

Элемент 0 под номером 10

Элемент 2 под номером 12

Элемент 4 под номером 14

Элемент 6 под номером 16

Элемент 8 под номером 18

Process finished with exit code 0

2. Протокол PZ_6_2

Введите кол-во элементов в списке: 7

Введите какое-нибудь целое число к нему будет подобраное наиболее похожее число из списка: 2

Список a = [0, 1, 2, 3, 4, 5, 6]

Наболее похожий элемент в списке число 2, так как разница между 2 и 2 составляет 0

Process finished with exit code 0

3. Протокол PZ_6_3

Введите диапазон списка.

Введите 1-ое число диапазона: 1

Введите 2-ое число диапазона: 6

Исходный список A = [1, 2, 3, 4, 5]

Конечный список A = [2, 3, 4, 5, 0]

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал(а) навыки составления программ со списками в IDE PyCharm Community. Были использованы языковые конструкции `while`, `if`, `break`, `for`.

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

Готовые программные коды выложены на [GitHub](#).