

Практическая работа №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: Вариант 11

1. Дан список A размера N (N — четное число). Вывести его элементы с четными номерами в порядке возрастания номеров: A₂, A₄, A₆, ..., A_N. Условный оператор не использовать.

Тип алгоритма: линейный (кроме функции try_except и составления списка — они циклические).

Текст программы:

```
import random

def listing(a, b): # составление списка со случайными числами
    x = 0
    c = []
    while x != a:
        c.append(random.randrange(0, b))
        x += 1
    return c

def try_except(a): # обработка исключений
    while type(a) != int:
        try:
            a = int(a)
        except ValueError:
            print('Введите целое число!')
            a = input()
    return a

N = try_except(input('Введите кол-во элементов списка: '))
Z = try_except(input('Список будет формироваться из диапазона с 0 и до '))
A = listing(N, Z)
print('Список:', A)
print('Элементы с четными номерами в порядке возрастания номеров:', A[1:N + 1:2])
```

Протокол работы программы:

Введите кол-во элементов списка: 4

Список будет формироваться из диапазона с 0 и до 10

Список: [7, 4, 0, 0]

Элементы с четными номерами в порядке возрастания номеров: [4, 0]

Process finished with exit code 0

Постановка задачи: Вариант 11

2. Дано число R и список A размера N. Найти элемент списка, который наиболее близок к числу R (то есть такой элемент A_k, для которого величина |A_k - R| является минимальной).

Тип алгоритма: линейный (кроме функции try_except и составления списка — они циклические).

Текст программы:

```
import random

def try_except(a, b): # обработка исключений на int или float
    if b == int:
        while type(a) != int:
            try:
                a = int(a)
            except ValueError:
                print('Введите целое число!')
                a = input()
    else:
        while type(a) != float:
            try:
                a = float(a)
            except ValueError:
                print('Введите число!')
                a = input()
    return a

def listing(a, b): # составление списка со случайными числами
    x = 0
    c = []
    while x != a:
        c.append(random.randrange(0, b))
        x += 1
    return c

N = try_except(input('Введите кол-во элементов списка: '), int)
Z = try_except(input('Список будет формироваться из диапазона с 0 и до '), int)
R = try_except(input('Введите число R: '), float)
lst = listing(N, Z) # формирование списка чисел

result = lst[0]
result_number = lst[0]
for i in lst:
    if abs(i - R) < abs(result_number - R):
        result = abs(i - R)
        result_number = i

print(f'Список: {lst}')
print(f'Элемент списка наиболее близкий к {R}: {result_number}')
```

Протокол работы программы:

Введите кол-во элементов списка: 6

Список будет формироваться из диапазона с 0 и до 10

Введите число R: 14

Список: [6, 3, 5, 9, 5, 3]

Элемент списка наиболее близкий к 14.0: 9

Process finished with exit code 0

Постановка задачи: Вариант 11

3. Дан список размера N. Осуществить сдвиг элементов списка влево на одну позицию (при этом A_n перейдет в A_{n-1} , A_{n-1} — в A_{n-2} , ..., A_2 — в A_1 , а исходное значение первого элемента будет потеряно). Последний элемент полученного списка положить равным 0.

Тип алгоритма: линейный (кроме функции `try_except` и составления списка — они циклические).

Текст программы:

```
import random

def listing(a, b): # составление списка со случайными числами
    x = 0
    c = []
    while x != a:
        c.append(random.randrange(0, b))
        x += 1
    return c

def try_except(a): # обработка исключений
    while type(a) != int:
        try:
            a = int(a)
        except ValueError:
            print("Введите целое число!")
            a = input()
    return a

N = try_except(input('Введите кол-во элементов списка: '))
Z = try_except(input('Список будет формироваться из диапазона с 0 и до '))
lst = listing(N, Z)

print('Исходный список: ', lst)
lst.pop(0) # удаление 1 элемента и замена последнего на 0
lst[-1] = 0
print('Полученный список: ', lst)
```

Протокол работы программы:

Введите кол-во элементов списка: 5

Список будет формироваться из диапазона с 0 и до 10

Исходный список: [8, 6, 3, 2, 1]

Полученный список: [6, 3, 2, 0]

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия я выработал навыки составления программ со списками в IDE PyCharm Community. Были использованы языковые конструкции: `while`, `for`, `try`, `list`, `def`.

Выполнены разработка, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.