

Практическое занятие № 6

Тема: Составление программ со списками в IDE PyCharm Community.

Цели: Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ со списками в IDE PyCharm Community.

Постановка задачи №1: Дан список A размера N и целое число K ($1 < K < N$). Вывести элементы списка, кратные K: AK, A2K, A3K... . Условный оператор не использовать.

Тип алгоритма: циклический

Текст программы:

```
# Дан список A размера N и целое число K ( $1 < K < N$ ).
# Вывести элементы списка, кратные K: AK, A2K, A3K...
# Условный оператор не использовать
import random
a = []
n = input("Введите значение N: ")
while int != type(n):
    try:
        n = int(n)
    except ValueError:
        n = input("Введите значение N: ")
for i in range(n):
    a.append(random.randint(a: 1, b: 5))
k = input('Введите значение K ( $1 < K < N$ ): ')
while True:
    try:
        k = int(k)
        if 1 < k < n:
            break
        else:
            k = int(input('Введите значение K ( $1 < K < N$ ): '))
    except ValueError:
        k = input('Введите значение K ( $1 < K < N$ ): ')
```

```
# Вывод элементов списка, кратных K
elements = a[k-1:k]
print("Список: ", a)
print('элементы списка, кратные K: ', elements)
```

Протокол работы программы:

Введите значение N: 6

Введите значение K($1 < K < N$): 2

Список: [4, 4, 5, 2, 5, 3]

Элементы списка, кратные K: [4, 2, 3]

Process finished with exit code 0

Постановка задачи №2: Дан список A размера N. Найти количество его промежутков монотонности, то есть участков на которых его элементы возрастают или убывают)

Тип алгоритма: циклический

Текст программы:

```
# Дан список A размера N. Найти количество его промежутков монотонности
# то есть участков на которых его элементы возрастают или убывают
import random
1 usage
def count_monotone_intervals(arr):
    increasing = decreasing = 0
    for i in range(1, len(arr)):
        if arr[i] > arr[i - 1]:
            increasing += 1
        elif arr[i] < arr[i - 1]:
            decreasing += 1
    return increasing, decreasing
a = []
# Пример списка A
n = input("Введите значение N: ")
```

```
while int != type(n):
    try:
        n = int(n)
    except ValueError:
        n = input("Введите значение N: ")
for i in range(n):
    a.append(random.randint(a: 1, b: 5))

# Вычисление количества участков возрастания и убывания
increasing_count, decreasing_count = count_monotone_intervals(a)
print("Список: ", a)
print("Количество участков возрастания: ", increasing_count)
print("Количество участков убывания: ", decreasing_count)
```

Протокол работы программы:

Введите значение N: 6

Список: [1, 1, 1, 2, 1, 4]

Количество участков возрастания: 2

Количество участков убывания: 1

Process finished with exit code 0

Постановка задачи №3: Дан список размера N. Осуществить сдвиг элементов списка вправо на одну позицию (при этом A1 перейдет в A2, A2 - в A3.... AN-1 в AN, а исходное значение последнего элемента будет потеряно). Первый элемент полученного списка положить равным 0.

Тип алгоритма: циклический

Текст программы:

```
# Дан список размера N. Осуществить сдвиг элементов
# списка вправо на одну позицию (при этом A1 перейдет в A2, A2 -
# в A3.... AN-1 в AN, а исходное значение последнего элемента будет
# потеряно). Первый элемент полученного списка положить равным 0.
import random
list = []
n = input("Введите значение N: ")
while int != type(n):
    try:
        n = int(n)
    except ValueError:
        n = input("Введите значение N: ")
for i in range(n):
    list.append(random.randint(a=1, b=5))
print("Первоначальный список", list)
list.pop()
list.insert(_index: 0, _object: 0)
print("Измененный список", list)
```

Вывод: В процессе выполнения практической работы я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ со списками в IDE PyCharmCommunity.

Студентка группы ИС-23 Камилова А.Н.

Студентка группы ИС-23 Камилова А.Н.