

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 циклической структуры в IDE PyCharm Community.

Постановка задачи:

1. Даны целые числа N (>2), A и B . Сформировать и вывести целочисленный список размера 10, первый элемент которого равен A , второй равен B , а каждый последующий элемент равен сумме всех предыдущих.

Текст программы:

```
# # Даны целые числа N (>2), A и B. Сформировать и
вывести целочисленный список
# # размера 10, первый элемент которого равен A,
второй равен B, а каждый
# # последующий элемент равен сумме всех предыдущих.
# #

a = input("Введите целое число №1, которое будет > 2: ")
b = input("Введите целое число №2, которое будет > 2: ")

while type(a) != int:
    try:
        a = int(a)
        if a < 2 :
            print("Число должно быть больше 2!!!")
            a = input("Введите целое число №1, которое
будет > 2: ")
    except ValueError:
        print("Неправильно ввели")
        a = input("Введите целое число ")
```

```

while type(b) != int:
    try:
        b = int(b)
        if b < 2:
            print("Число должно быть больше 2!!!")
            b = input("Введите целое число №2, которое
будет > 2: ")
        except ValueError:
            print("Неправильно ввели")
            b = input("Введите целое число")

List = [0] * 10
List[0] = a
List[1] = b
for i in range(2, 10):
    List[i] = sum(List[:i])

print(List)

```

Протокол работы программы:

Введите целое число №1, которое будет > 2: 5
Введите целое число №2, которое будет > 2: 6
[5, 6, 11, 22, 44, 88, 176, 352, 704, 1408]

Process finished with exit code 0

№2

Постановка задачи:

2. Дан список размера N. Найти максимальный из его локальных минимумов (локальный минимум — это элемент, который меньше любого из своих соседей).

Текст программы:

```
# Дан список размера N. Найти максимальный из его
локальных минимумов
# (локальный минимум — это элемент, который меньше
любого из своих соседей) .

import random

spisok = []

n = input("Введите размер списка: ")

while type(n) != int:                # Обработка
исключений
    try:
        n = int(n)
        if n < 1:
            print("Размер списка должен быть больше
1")
            n = input("Введите размер списка > 1: ")
    except ValueError:
        print("Неправильно ввели!")
        n = input("Введите размер списка: ")

for f in range(n):
    spisok += [random.randint(1, 100)]

print("Изначальный список:", spisok)
```

```

lokal_minimumy = []

# Проверяем первый элемент
if n > 1 and spisok[0] < spisok[1]:
    lokal_minimumy.append(spisok[0])

# Проверяем локальные минимумы для элементов с 1 по
n-2
for i in range(1, n - 1):
    if spisok[i] < spisok[i - 1] and spisok[i] <
spisok[i + 1]:
        lokal_minimumy.append(spisok[i])

# Проверяем последний элемент
if n > 1 and spisok[n - 1] < spisok[n - 2]:
    lokal_minimumy.append(spisok[n - 1])

print("Локальные минимумы:", lokal_minimumy)

# Выводим максимальный из локальных минимумов, если
они есть
if lokal_minimumy:
    max_lokal_min = max(lokal_minimumy)
    print("Максимальный из локальных минимумов:",
max_lokal_min)
else:
    print("Локальных минимумов нет.")

```

Протокол работы программы:

Введите размер списка: 6

Изначальный список: [14, 82, 45, 94, 66, 63]

Локальные минимумы: [14, 45, 63]

Максимальный из локальных минимумов: 63

Process finished with exit code 0

№3

Постановка задачи:

3. Дан список размера N и целое число K ($1 < K < N$). Осуществить сдвиг элементов списка вправо на K позиций (при этом A_1 перейдет в A_{K+1} , A_2 — в A_{K+2} , .. A_{N-K} — в A_N , а исходное значение K последних элементов будет потеряно). Первые K элементов полученного списка положить равными 0.

Текст программы:

```
# Дан список размера N и целое число K (1 < K < N) .  
Осуществить сдвиг элементов  
# список вправо на K позиций (при этом A1 перейдет в  
AK+1, A2 — в AK+2, ..AN-K — в  
# AN, а исходное значение K последних элементов будет  
потеряно) . Первые K  
# элементов полученного списка положить равными 0.  
  
import random
```

```
Spisok = []

n = input("Введите размер списка: ") # размер списка

while type(n) != int:
    try:
        n = int(n)
    except ValueError:
        print("Неправильно ввели!")
        n = input("Введите размер списка: ")

k = input("Введите на сколько вы хотите сдвинуть  
элементы (1 < k < n) : ") # количество сдвигов

while type(k) != int:
    try:
        k = int(k)
        if k <= 1 or k >= n:
            print("Ошибка: К должно быть в пределах (1  
< k < n)")
            k = input("Введите на сколько вы хотите  
сдвинуть элементы 1 < k < n : ")
        except ValueError:
            print("Неправильно ввели!")
            k = input("Введите на сколько вы хотите  
сдвинуть элементы: ")

for u in range(n):
    Spisok += [random.randint(1, 100)]

print("Изначальный список", Spisok)

# Сдвиг вправо на К позиций
for i in range(n - k - 1, -1, -1):
    Spisok[i + k] = Spisok[i]
```

```
# Заполнение первых K элементов нулями
for i in range(k) :
    Spisok[i] = 0

# Вывод результата
print("Список после сдвига:", Spisok)
```

Протокол работы программы:

Введите размер списка: 6

Введите на сколько вы хотите сдвинуть элементы ($1 < k < n$) : 5

Изначальный список [60, 36, 77, 3, 3, 50]

Список после сдвига: [0, 0, 0, 0, 0, 60]

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработала навыки составления программ со списками в IDE PyCharm Community. Были использованы языковые конструкции `while`, `for`, `try`, `except`. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.