

Практическое занятие №

Тема: Составление программ со списками в IDE PyCharm Community.

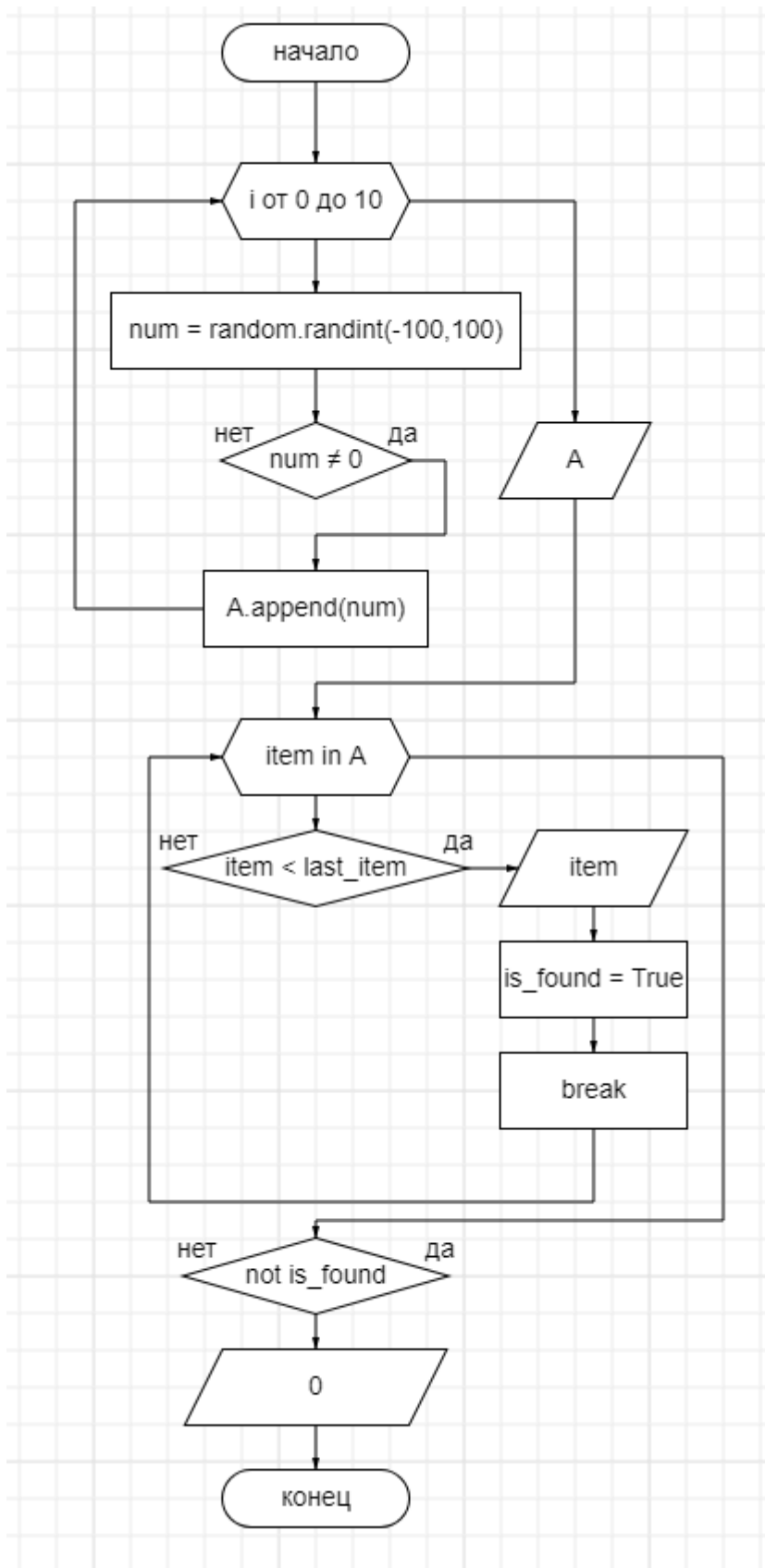
Цель: Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ со списками в IDE PyCharm Community.

Постановка задачи №1.

Дан список A ненулевых целых чисел размера 10. Вывести значение первого из тех его элементов A_K , которые удовлетворяют неравенству $A_K < A_{10}$. Если таких элементов нет, то вывести 0.

Тип алгоритма: циклический.

Блок-схема алгоритма №1:



Текст программы №1:

```
import random

A = []
counter = 0

# Цикл генерирующий список из 10 случайных ненулевых чисел
while counter != 10:
    num = random.randint(-100, b: 100)
    if num != 0:
        A.append(num)
        counter += 1

print(A)

last_item = A[9]

is_found = False

# Цикл, проходящий по всем элементам списка и проверяющий правдивость условия
for item in A:
    if item < last_item:
        print(f'Элемент удовлетворяющий условию>> {item}')
        is_found = True
        break

if not is_found:
    print(0)
```

Протокол работы программы:

[87, 44, 77, 45, -9, -37, -43, -33, 68, -97]

0

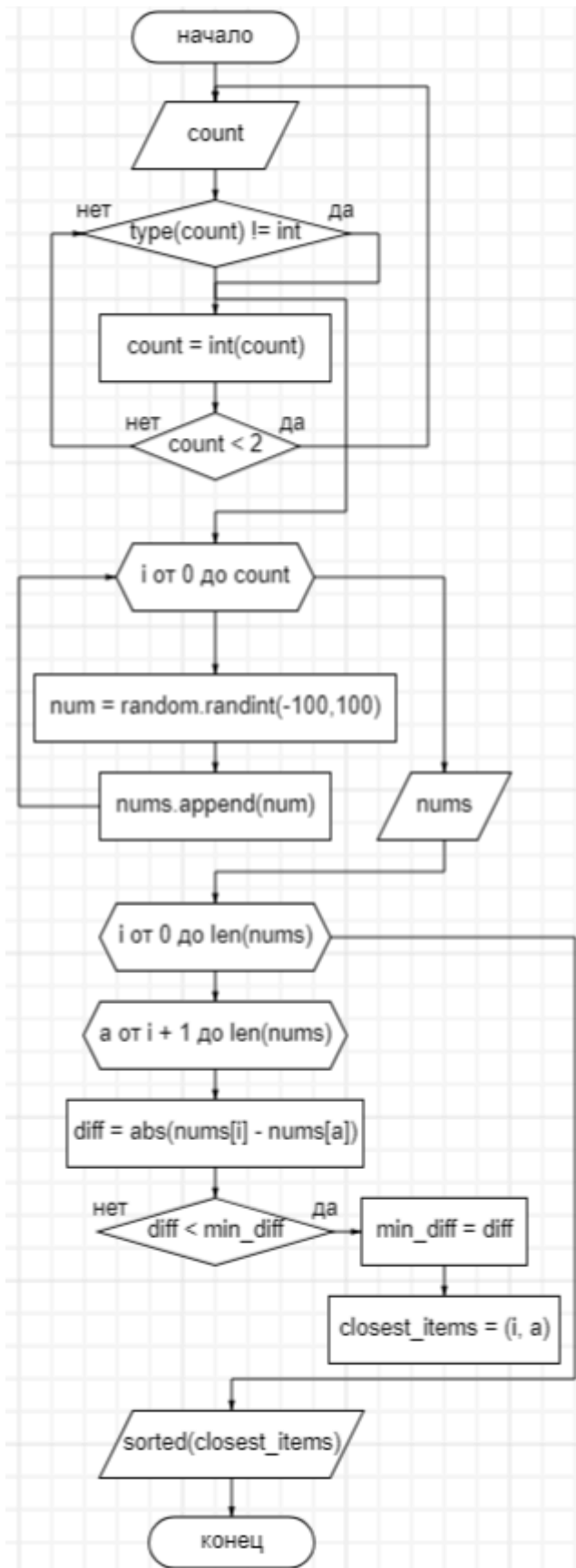
Process finished with exit code 0

Постановка задачи №2.

Дан список размера N . Найти номера двух ближайших элементов из этого списка (то есть элементов с наименьшим модулем разности) и вывести эти номера в порядке возрастания.

Тип алгоритма: циклический

Блок-схема алгоритма №2:



Текст программы №2:

```
# Импортируем нужные библиотеки
import random

count = input('Введите количество элементов списка>> ')

# Обработка исключений
while type(count) != int:
    try:
        count = int(count)
        # Проверка на правильную длину списка
        if count < 2:
            print("Неправильно ввели!\nСписок должен содержать не менее 2 элементов!")
            count = input('Введите количество элементов списка>> ')
    except ValueError:
        print("Неправильно ввели!")
        count = input('Введите количество элементов списка>> ')

nums = []

# Цикл генерирующий список из N случайных чисел
for i in range(count):
    num = random.randint(-100, b: 100)
    nums.append(num)

print(nums)

# Задаем минимальную разницу и начальные индексы
min_diff = 200.1
closest_items = (0, 1)

#Цикл, находящий минимальную разницу путем сравнения 2х элементов
for i in range(len(nums)):
    for a in range(i + 1, len(nums)):
        diff = abs(nums[i] - nums[a])

        if diff < min_diff:
            min_diff = diff
            closest_items = (i, a)

print(f'Индексы элементов с минимальным модулем разности>> {sorted(closest_items)}')
```

Протокол работы программы:

Введите количество элементов списка>> 10

[-3, -43, -81, 97, -38, 75, 20, 62, -41, 25]

Индексы элементов с минимальным модулем разности>> [1, 8]

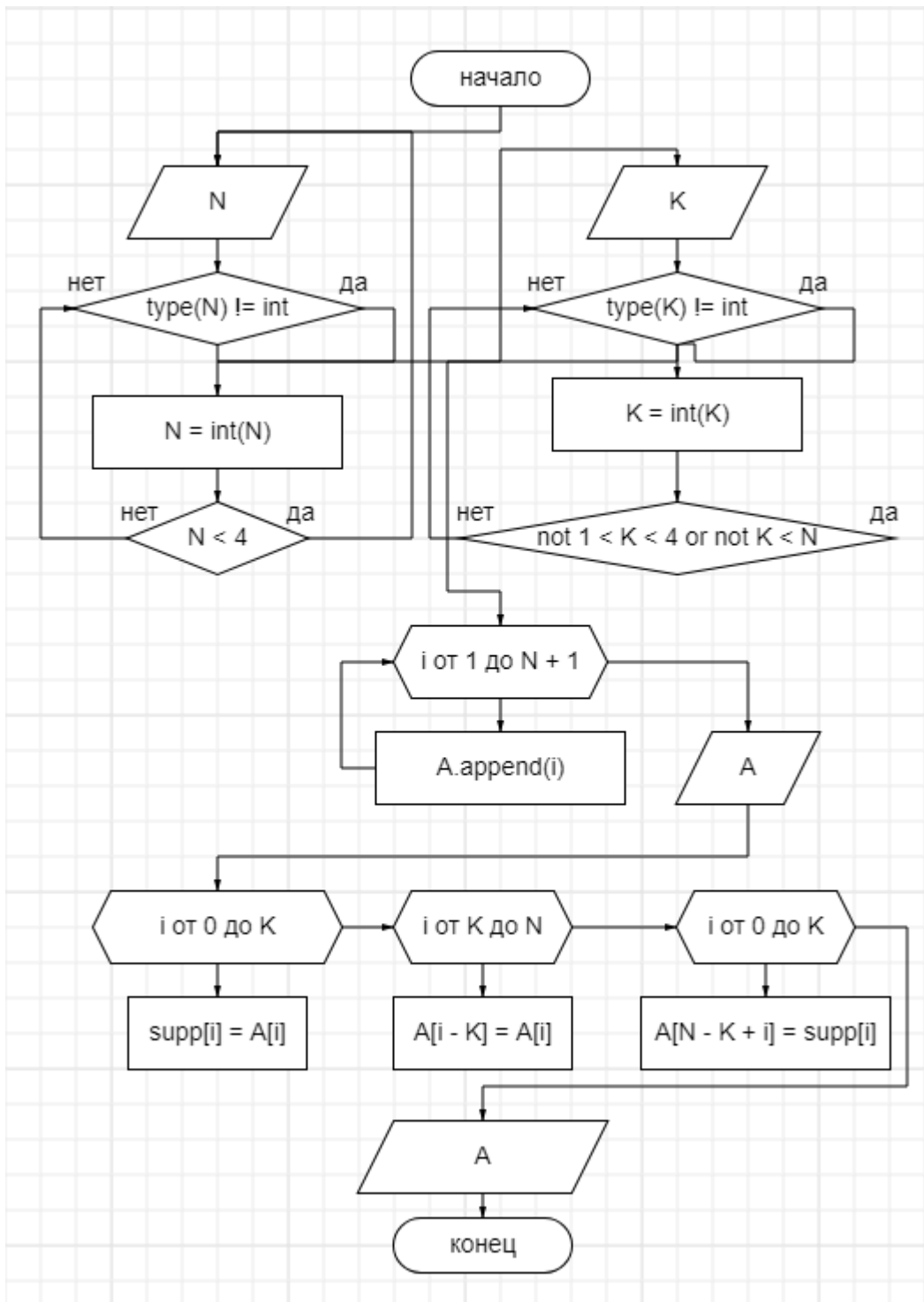
Process finished with exit code 0

Постановка задачи №3.

Дан список A размера N и целое число K ($1 < K < 4$, $K < N$). Осуществить циклический сдвиг элементов списка влево на K позиций (при этом A_N перейдет в A_{N-K} , A_{N-1} — в A_{N-K-1} , ..., A_1 — в A_{N-K+1}). Допускается использовать вспомогательный список из 4 элементов.

Тип алгоритма: циклический

Блок-схема алгоритма №3:



Текст программы №3:

```
# Ввод данных

N = input('Введите длину списка>> ')

# Обработка исключений
while type(N) != int:
    try:
        N = int(N)
        # Проверка на правильную длину списка
        if N < 4:
            print("Неправильно ввели!\nСписок не может содержать меньше 4 элементов!")
            N = input('Введите длину списка>> ')
    except ValueError:
        print("Неправильно ввели!")
        N = input('Введите длину списка>> ')

K = input('Введите количество позиций сдвига>> ')

# Обработка исключений
while type(K) != int:
    try:
        K = int(K)
        # Проверка на правильную длину списка
        if not 1 < K < 4 or not K < N:
            print("Неправильно ввели!\nЧисло K не соответствует условиям!")
            K = input('Введите количество позиций сдвига>> ')
```

```
except ValueError:

    print("Неправильно ввели!")

    K = input('Введите количество позиций сдвига>> ')

A = []

supp = [0, 0, 0, 0] # Вспомогательный список

# Цикл для создания списка из N элементов
for i in range(1, N + 1):

    A.append(i)

print(f'\nНачальный список>> {A}')

# Перезаписываем элементы из списка A до K в список supp
for i in range(K):

    supp[i] = A[i]

# Сдвиг оставшихся элементов влево
for i in range(K, N):

    A[i - K] = A[i]

# Вставим элементы из списка supp в конец
for i in range(K):

    A[N - K + i] = supp[i]

print(f'Конечный список>> {A}')
```

Протокол работы программы:

Введите длину списка>> 10

Введите количество позиций сдвига>> 3

Начальный список>> [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Конечный список>> [4, 5, 6, 7, 8, 9, 10, 1, 2, 3]

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления функций в IDE PyCharm Community. Были использованы языковые конструкции `while`, `if`, `for`. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовый программный код выложен на GitHub.