

Практическое задание № 6.

Тема: составление программ со списками в IDE PyCharm Community.

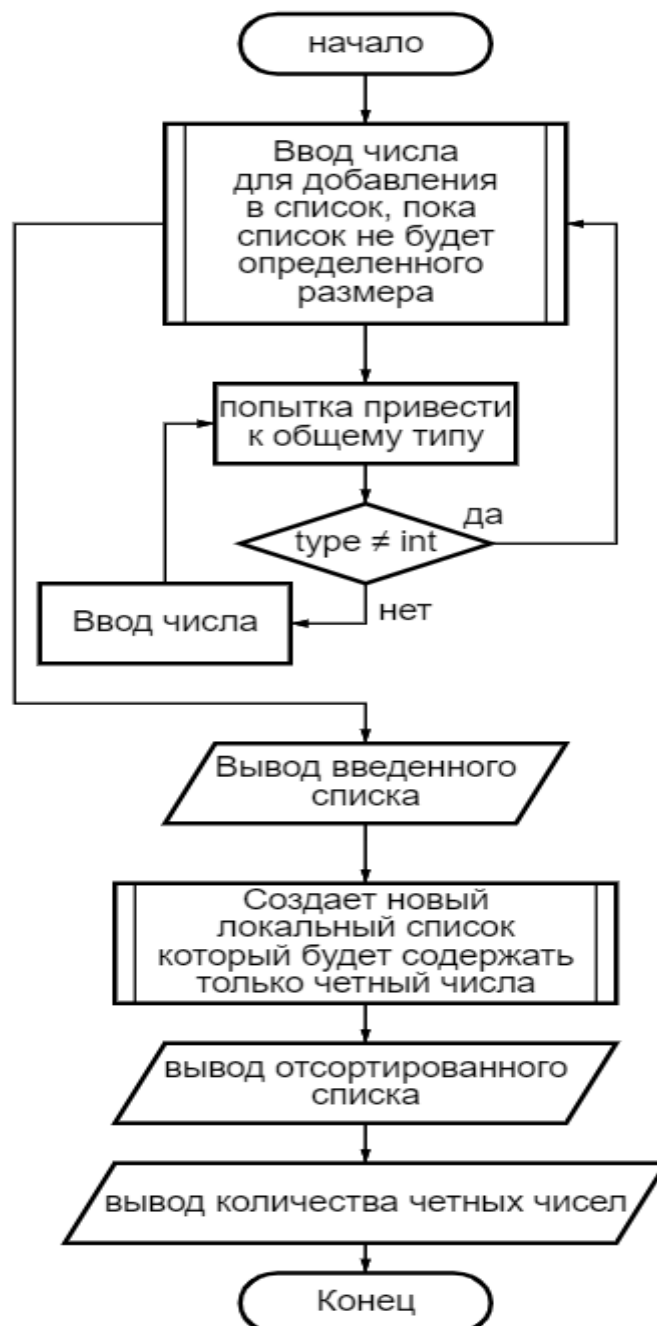
Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ со списками в IDE PyCharm Community.

Постановка задачи:

Дан целочисленный список размера 10. Вывести все содержащиеся в данном списке четные числа в порядке убывания их индексов, а также их количество K.

Тип алгоритма: Циклический.

Блок-схема:



Текст программы:

```
# Дан целочисленный список размера 10.
# Вывести все содержащиеся в данном списке четные числа в порядке убывания их
индексов
# А также их количество K.
def list_append(n: int, list_gen: list) -> list: # Функция генерации списка
    for _ in range(n):
        list_gen.append(input(f"Введите {_ + 1}-е число: "))
        while type(list_gen[_]) != int: # Обработчик исключений элементов списка
            try:
                list_gen[_] = int(list_gen[_])
            except ValueError:
                list_gen[_] = input(f"Введите целое число {_ + 1} без лишних
символов: ")
    return list_gen

def list_changer(old_list): # Функция, сортирующая список
    new_list = []
    for i in reversed(old_list):
        if i % 2 == 0:
            new_list.append(i)
    return new_list

N = 10
lst = list_append(N, [])
print(f"Вот содержимое вашего списка: {lst}")
even_list = list_changer(lst)

print(f"Вот все четные числа из вашего списка: {even_list}")
print(f'Их количество = {len(even_list)}')
```

Протокол работы программы:

Введите 1-е число: 3

Введите 2-е число: 7

Введите 3-е число: 9

Введите 4-е число: 0

Введите 5-е число: 11

Введите 6-е число: 6

Введите 7-е число: 82

Введите 8-е число: 28

Введите 9-е число: 1

Введите 10-е число: 13

Вот содержимое вашего списка: [3, 7, 9, 0, 11, 6, 82, 28, 1, 13]

Вот все четные числа из вашего списка: [28, 82, 6, 0]

Их количество = 4

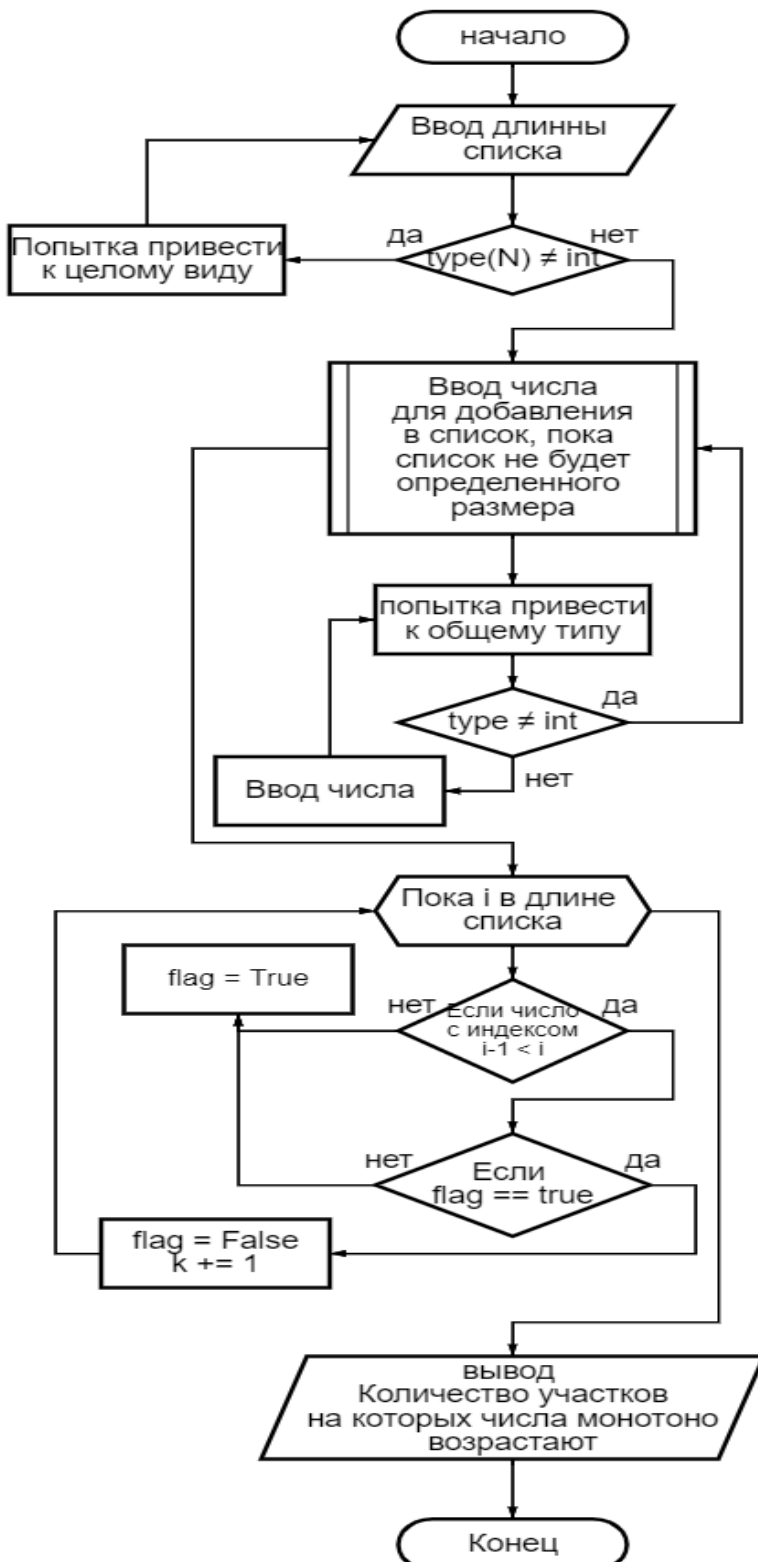
Process finished with exit code 0

Постановка задачи:

Дан список размера N. Найти количество участков, на которых его элементы монотонно возрастают.

Тип алгоритма: Циклический.

Блок-схема:



Текст программы:

```
# Дан список размера N.
# Найти количество участков, на которых его элементы монотонно возрастают.
def list_append(n: int, list_gen: list) -> list: # Функция генерации списка
    for _ in range(n):
        list_gen.append(input(f"Введите {_ + 1}-е число: "))
        while type(list_gen[_]) != int: # Обработчик исключений элементов списка
            try:
                list_gen[_] = int(list_gen[_])
            except ValueError:
                list_gen[_] = input(f"Введите целое число {_ + 1} без лишних
символов: ")
    return list_gen

N = input("Введите длину списка: ")

while type(N) != int: # Обработчик исключений
    try:
        N = int(N)
    except ValueError:
        N = input("Введите целое число: ")

lst = list_append(N, [])

k = 0
flag = True
for i in range(len(lst)): # Цикл пробегает по списку
    if lst[i-1] < lst[i]: # Если 1 число меньше 2, то эти числа возрастают
        if flag: # Если пара чисел i-2 и i-1 возростала, то условие не выполнится
            k += 1
            flag = False # Если числа возрастают монотонно, то следующая пара не
будет считаться участком
        else:
            flag = True

print(f"Количество участков на которых числа монотонно возрастают - {k}")
```

Протокол работы программы:

Введите длину списка: 6

Введите 1-е число: 1

Введите 2-е число: 2

Введите 3-е число: 5

Введите 4-е число: 3

Введите 5-е число: 4

Введите 6-е число: 7

Количество участков на которых числа монотонно возрастают - 2

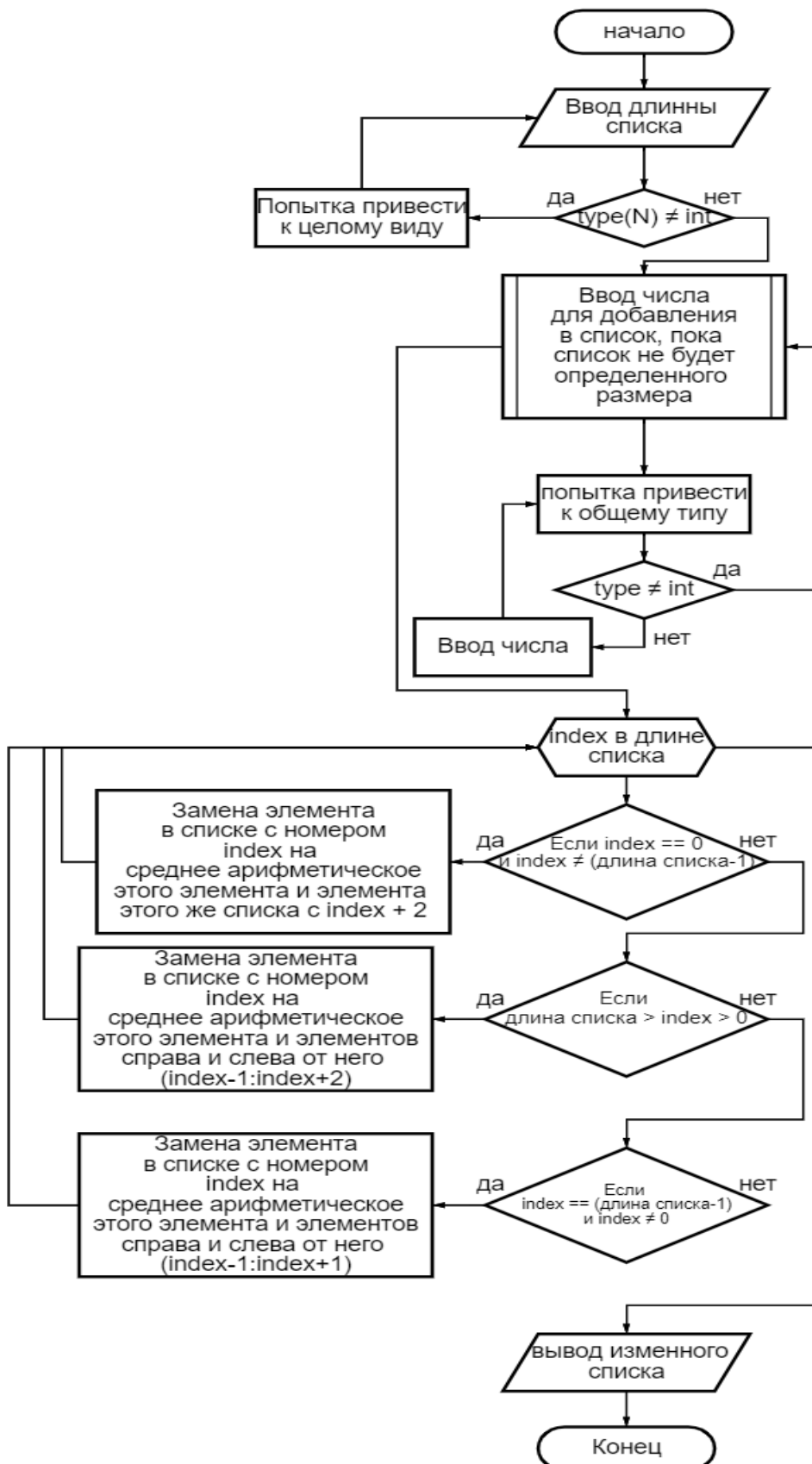
Process finished with exit code 0

Постановка задачи:

Дан список размера N. Заменить каждый элемент списка на среднее арифметическое этого элемента и его соседей.

Тип алгоритма: Циклический.

Блок-схема:



Текст программы:

```
# Дан список размера N.
# Заменить каждый элемент списка на среднее арифметическое этого элемента и его
соседей.
def list_append(n: int, list_gen: list) -> list: # Функция генерации списка
    for _ in range(n):
        list_gen.append(input(f"Введите {(_ + 1)}-е число: "))
        while type(list_gen[_]) != int: # Обработчик исключений элементов списка
            try:
                list_gen[_] = int(list_gen[_])
            except ValueError:
                list_gen[_] = input(f"Введите целое число {_ + 1} без лишних
символов: ")
    return list_gen

N = input("Введите длину списка: ")

while type(N) != int: # Обработчик исключений
    try:
        N = int(N)
    except ValueError:
        N = input("Введите целое число: ")

lst = list_append(N, [])
print(lst)

for index in range(len(lst)): # Цикл, который будет находить среднее арифметическое
# и присваивать значение к элементу списка с номером index
    if index == 0 and index != (len(lst) - 1):
        lst[index] = round(sum(lst[index:index + 2]) / 2, 5)

    if len(lst) - 1 > index > 0:
        lst[index] = round(sum(lst[index - 1:index + 2]) / 3, 5)

    if index == (len(lst) - 1) and index != 0:
        lst[index] = round(sum(lst[index - 1:index + 1]) / 2, 5)

print(lst)
```

Протокол работы программы:

Введите длину списка: 3

Введите 1-е число: 4

Введите 2-е число: 3

Введите 3-е число: 9

[4, 3, 9]

[3.5, 5.16667, 7.08333]

Process finished with exit code 0

Вывод: закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ со списками в IDE PyCharm Community.