Студент группы ИС-27 Ермоленко В.Р.

Практическое занятие No 5

Тема: Составление программ циклической структуры с функцией в IDE PyCharm Community.

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

Задача 1
Постановка задачи.

Тип алгоритма: циклический с функцией

Текст программы:

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

# Вводим функцию для подсчета четных чисел
def chetnue_numbers(numbers):
num_list = []

for index in range(len(numbers) -1, -1, -1):
if numbers[index] % 2 == 0:
num_list.append(numbers[index])
return num_list

# Обработка исключений
try:
numbers = [int(input(f"Input element {i + 1} (number) : ")) for i in 
range(10)]

num_list = chetnue_numbers(numbers)
K = len(num_list)

print("Chet numbers in negative index upscale : ", num_list)
print("Count of chet numbers : ", K)

except ValueError:
print("Input integer number!")
```

Протокол работы программы:

```
Input element 1 (number) : 1
Input element 2 (number) : 2
Input element 3 (number) : 3
```

Input element 4 (number) : 4
Input element 5 (number) : 5
Input element 6 (number) : 6
Input element 7 (number) : 7
Input element 8 (number) : 8
Input element 9 (number) : 9
Input element 10 (number) : 10
Chet numbers in negative index upscale : [10, 8, 6, 4, 2]
Count of chet numbers : 5

Задача 2
Постановка задачи.

Тип алгоритма: циклический с функцией

Текст программы:

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

# введем функцию
def count_segment(lean):
if len(lean) == 0:
return 0
count = -1
increase = False

# введем цикл для подсчета участков
for i in range(1, len(lean)):
if lean[i] > lean[i-1]:
if not increase:
count += 1
increase = True

else:
increase = False

return count

# обработка исключений
try:
N = int(input("Input size : "))
lean = [int(input(f"input element {i + 1} : ")) for i in range(N)]
result = count_segment(lean)
print("Count of increasing segment : ", result)

except ValueError:
print("Input integer data!")
```

Протокол работы программы:
Input size : 5
input element 1 : 1
input element 2 : 2
input element 3 : 3
input element 4 : 4
input element 5 : 5
Count of increasing segment : 1

Задача 3
Постановка задачи.

Тип алгоритма: циклический с функцией

Текст программы:
```
# Дан список размера N. Заменить каждый элемент списка на среднее арифметическое
# этого элемента и его соседей.

# вводим функцию
def replace_el(lean):
if len(lean) == 0:
return lean
# новый список для результатов работы цикла
new_lean = []
n = len(lean)

for i in range(n):
if i == 0:
# для первого значения
element = (lean[i] + lean[i + 1]) / 2
elif i == n - 1:
# для последнего значения
element = (lean[i] + lean[i - 1]) / 2

else:
# для средних значений
element = (lean[i - 1] + lean[i] + lean[i + 1]) / 3

new_lean.append(element)

return new_lean

# обработка исключений
try:
N = int(input("Input size : "))
lean = [int(input(f"Input element {i + 1} : ")) for i in range(N)]

new_lean = replace_el(lean)
```

```
print("New list : ", new_lean)

except ValueError:
print("input integer data!")
```

Протокол работы программы:
Input size : 5
Input element 1 : 1
Input element 2 : 2
Input element 3 : 3
Input element 4 : 4
Input element 5 : 5
New list : [1.5, 2.0, 3.0, 4.0, 4.5]

Вывод: в процессе выполнения практического занятия выработал(а) навыки составления программ циклической структуры с функцией в IDE PyCharm Community. Были использованы языковые конструкции ***def, for, return.*** Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.