

Практическая работа №13.

Тема: составление программ в функциональном стиле в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием списковых включений, итераторов, генераторов в IDE PyCharm Community.

Постановка задачи:

1. Организовать и вывести последовательность A из n чисел. Из последовательности A получить две последовательности B и C: в последовательности B – четные элементы A, в C – нечетные элементы A. Произвести суммирование соответствующих элементов последовательностей B и C. Найти минимальный элемент полученной последовательности.
2. Из списка: ['Валентин', 'Петр', 'Анна', 'Евгений', 'Константин', 'Валерия', 'Юлия'] получить новый список, в котором длина слов не превышает 5 символов.

Тип алгоритма: циклический и ветвление.

Текст программы:

Задача 1:

```
# Организовать и вывести последовательность A из n чисел. Из
# последовательности A получить две последовательности B и C: в
# последовательности B –
# четные элементы A, в C – нечетные элементы A. Произвести суммирование
# соответствующих элементов последовательностей B и C. Найти минимальный
# элемент
# полученной последовательности.

from random import randint

# Генерируем список случайных чисел.
main_list = [randint(-999, 999) for i in range(int(input(f"Введите предел
последовательности A: ")))]
print('Первоначальная последовательность: \n', main_list)

# Создаём новые списки на основе главного с проверкой на чётность
```

```

even_list = list(filter(lambda n: n % 2 == 0, main_list))
nteven_list = list(filter(lambda m: m % 2 != 0, main_list))

# Список-результат суммирования элементов двух ранее полученных списков.
new_list = [i + j for i, j in zip(even_list, nteven_list)]

print("Список чётных чисел: \n", even_list, end='\n\n') # Выводы
результатов.
print("Список нечётных чисел: \n", nteven_list, end='\n\n')
print("Список-результат суммирования чётного и нечётного списков: \n",
new_list, end='\n\n')
print("Минимальный элемент конечного списка: ", min(new_list))

```

Задача 2:

```

# Из списка: ['Валентин', 'Петр', 'Анна', 'Евгений', 'Константин',
'Валерия', 'Юлия']
# получить новый список, в котором длина слов не превышает 5 символов.

name_list = ['Валентин', 'Петр', 'Анна', 'Евгений', 'Константин',
'Валерия', 'Юлия']

new_list = [i for i in name_list if len(i) <= 5] # Генерация списка с
применением условия.
print('Список имён, длина которых не превышает 5 символов: ', new_list)
# Вывод результата.

```

Протокол работы программы:

Задача 1:

Введите предел последовательности A: 20

Первоначальная последовательность:

[834, 567, -72, 934, -699, 786, -113, 874, 628, -28, 858, -899, -552, -525, 471, 36, -900, -589, -777, -73]

Список чётных чисел:

[834, -72, 934, 786, 874, 628, -28, 858, -552, 36, -900]

Список нечётных чисел:

[567, -699, -113, -899, -525, 471, -589, -777, -73]

Список-результат суммирования чётного и нечётного списков:

[1401, -771, 821, -113, 349, 1099, -617, 81, -625]

Минимальный элемент конечного списка: -771

Process finished with exit code 0

Задача 2:

Список имён, длина которых не превышает 5 символов: ['Петр', 'Анна', 'Юлия']

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ с использованием списковых включений, итераторов, генераторов в IDE PyCharm Community.