

## Практическое занятие №12

**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 из N случайных целых чисел. Из исходной последовательности организовать первую последовательность, содержащую числа кратные трем, и вторую – для всех остальных. Найти количество элементов в полученных последовательностях.

**Тип алгоритма:** циклический

**Текст программы:**

```
"""
Организовать и вывести последовательность из N случайных
целых чисел. Из
исходной последовательности организовать первую
последовательность, содержащую
числа кратные трем, и вторую – для всех остальных. Найти
количество элементов в
полученных последовательностях.
"""

import random

def generate_sequences(N):
    # Генерируем исходную последовательность из N случайных
    чисел
    original_sequence = [random.randint(0, 100) for _ in
range(N)]
```

```
# Разделяем на две последовательности

multiple_of_three = [num for num in original_sequence
if num % 3 == 0]

other_numbers = [num for num in original_sequence if
num % 3 != 0]

return original_sequence, multiple_of_three,
other_numbers

# Пример использования функции
N = random.randint(5, 15)

original_sequence, multiple_of_three, other_numbers =
generate_sequences(N)

print("Исходная последовательность:", original_sequence)
print("Числа, кратные трем:", multiple_of_three)
print("Остальные числа:", other_numbers)
print("Количество элементов, кратных трем:",
len(multiple_of_three))
print("Количество остальных элементов:",
len(other_numbers))
```

Протокол работы программы:

Исходная последовательность: [24, 14, 47, 32, 85]

Числа, кратные трем: [24]

Остальные числа: [14, 47, 32, 85]

Количество элементов, кратных трем: 1

Количество остальных элементов: 4

Process finished with exit code 0

**Постановка задачи №2:** Составить генератор (yield), который выводит из строки только цифры.

**Тип алгоритма:** циклический

**Текст программы:**

```
"""
Составить генератор (yield), который выводит из строки
только цифры.
"""

def digit_generator(string):
    for char in string:
        if char.isdigit():
            yield char

# Пример использования генератора
string = "Пример строки с цифрами 123 и символами 456"
digits = ''.join(digit_generator(string))
print("Цифры в строке:", digits)
```

**Протокол программы:**

Цифры в строке: 123456

Process finished with exit code 0

**Вывод:**

В процессе выполнения практического занятия №12, я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с использованием списковых включений, итераторов, генераторов в IDE PyCharm Community.