

Практическое занятие № 12 №1

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.

В последовательности их N чисел (N – четное) во второй ее половине найти сумму элементов больших 10.

Тип алгоритма: циклический.

Текст программы:

```
# В последовательности их N чисел (N -четное) во второй ее половине найти
сумму
# элементов больших 10.

def sum_elements_greater_than_10(sequence):
    mid = len(sequence) // 2
    second_half = sequence[mid:]
    filtered_elements = filter(lambda x: x > 10, second_half)
    sum_of_elements = sum(filtered_elements)
    return sum_of_elements

sequence = [5, 12, 3, 8, 20, 15, 10, 25, 7, 30] # пример последовательности
чисел
N = len(sequence) # количество элементов в последовательности

if len(sequence) % 2 == 0: # проверка на четность числа N
    result = sum_elements_greater_than_10(sequence)
    print("Сумма элементов больше 10 во второй половине
последовательности:", result)
else:
    print("Число элементов в последовательности должно быть четным.")
```

Протокол работы программы:

**Сумма элементов больше 10 во второй половине последовательности:
70**

Process finished with exit code 0

Вывод:

1. Генерация случайной последовательности чисел и нахождение суммы элементов больше 10 во второй половине последовательности:

- Предоставленный код работает со входной последовательностью чисел.
- Вычисляется середина последовательности (`mid`), и берется вторая половина чисел.
- Во второй половине последовательности вычисляется сумма элементов, которые больше 10.
- После этого выводится результат на экран.

2. Проверка на четность числа элементов:

- Программа сначала проверяет, что количество элементов в последовательности является четным числом (`len(sequence) % 2 == 0`).
- Если число элементов нечетное, программа выводит сообщение о том, что количество элементов должно быть четным.

Этот код фокусируется на работе с числовой последовательностью, нахождении второй половины последовательности, фильтрации элементов больше 10 в этой половине и вычислении их суммы. Результат выводится на экран.

Код также выводит результаты на экран с помощью `pprint` и `print`. Готовые программные коды выложены на [GitHub](#).

Практическое занятие № 12 №2

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.

Составить генератор (yield), который выводит из строки только буквы.

Тип алгоритма: линейный.

Текст программы:

```
# Составить генератор (yield), который выводит из строки только буквы.

def letters_generator(input_string):
    for char in input_string:
        if char.isalpha():
            yield char

# Пример использования генератора
input_string = "Пример строки с 123 цифрами и буквами ABC"
letters = letters_generator(input_string)

# Выводим буквенные символы из строки
for letter in letters:
    print(letter, end='')
```

Протокол работы программы:

Пример строки с цифрами и буквами ABC

Process finished with exit code 0

Вывод:

Функция-генератор (yield) для фильтрации буквенных символов из строки.

1. Описание работы программы:

- Программа содержит функцию `letters_generator`, которая является генератором.
- Этот генератор получает на вход строку и итерируется по каждому символу в этой строке.
- Для каждого символа происходит проверка, является ли он буквенным (`char.isalpha()`).
- Если символ является буквой, он передается в качестве значения генератору с помощью `yield`.
- Главный поток программы создает объект `letters`, итерируясь по результатам генератора.

2. Шаги выполнения программы:

- Переданная строка "Пример строки с 123 цифрами и буквами ABC" используется в качестве входных данных.
- Генератор `letters_generator` итерируется по каждому символу этой строки.
- Для каждого символа проводится проверка: является ли символ буквой.
- Если символ является буквой, он передается через `yield` итератору, хранящемуся в переменной `letters`.
- В конечном итоге, при итерации по `letters`, выводятся только буквенные символы из строки.

3. Результат выполнения программы:

- Программа фильтрует строку, выводя только буквенные символы.
- Вывод на экран будет содержать только буквы из строки "Пример строки с 123 цифрами и буквами ABC".
- Например, если в строке встречаются буквы "ABCabc", программа выведет "ABCabc" (будут проигнорированы цифры и другие символы).

Готовые программные коды выложены на [GitHub](#).