

Практическое занятие №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: .В последовательности на n целых элементов найти количество пар, для которых произведение элементов делится на 3 (элементы пары в последовательности являются соседними).

Тип алгоритма: смешанный

Текст программы:

```
#Вариант 8
```

```
#.В последовательности на n целых элементов найти количество пар,  
для которых
```

```
#произведение элементов делится на 3 (элементы пары в  
последовательности являются соседними).
```

```
from random import randint
```

```
n = int(input("Введите N ")) #ввод данных
```

```
lst = [randint(-100, 100) for i in range(n)] #создаем список со  
случайными значениями
```

```
print(lst)
```

```
func = lambda x, y: x * y % 3 == 0 #проверяем делится ли  
произведение элементов на три без остатка
```

```
counter = 0
```

```
for j in range(n): #заноcим результаты в переменную
    if j == n - 1:
        break
    if func(lst[j], lst[j + 1]) == True:
        counter+=1

print(counter)
```

Протокол работы программы:

Введите N 4

[33, -20, -45, -56]

3

Process finished with exit code 0

Постановка задачи №2: Составить генератор (yield), который преобразует все буквенные символы в заглавные.

Тип алгоритма: смешанный

Текст программы:

```
#Вариант 8
```

```
#Составить генератор (yield), который преобразует все буквенные  
символы в заглавные.
```

```
def uppercase_generator(spisok): #функция переводит каждую букву  
строки в верхний регистр
```

```
    for i in spisok:
```

```
        i = i.upper()
```

```
        yield i
```

```
fraza = input('Введите фразу ') #ввод данных
```

```
fraza = list(fraza.split()) #переводим строку в список из  
символов
```

```
result = []
```

```
generator = uppercase_generator(fraza)
```

```
for k in generator: #заносим результаты в список с результатом
```

```
    result.append(k)
```

```
print(result) #выводим результат
```

Протокол работы программы:

Введите фразу всем привет

['ВСЕМ', 'ПРИВЕТ']

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с использованием списковых включений, итераторов, генераторов. Выполнена разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.