

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Информатика»
Тема: Основные управляющие конструкции

Студент гр. 8303

Гришин К. И.

Преподаватель

Берленко Т. А.

Санкт-Петербург

2018

Цель работы

Изучить основные управляющие конструкции языка Python на примере программы «Функция-Меню», которая в зависимости от введенной опции выполняет те или иные действия со списком.

Задание

На вход программе подается список длины 10, элементы которого разделены символом перевода строки, и число, которое означает номер операции. Реализуйте функцию-меню, которая должна в зависимости от *номера операции*, выводить следующее:

- 0: Только числа исходного списка.
- 1: Сумму всех четных чисел исходного списка.
- 2: Строку, полученную путем конкатенации всех элементов списка, длина которых меньше или равна трем.
- 3: Произведение всех нечетных чисел исходного массива.
- 4: Каждый пятый символ каждой строки (нумерация элементов строки начинается с нуля). При этом полученные строки становятся элементами нового списка. Строкой считается элемент списка, который нельзя привести к целому числу.
- 5: Индекс каждого элемента и сам элемент.

Любой другой символ: Исходный список.

Каждый пункт меню должен быть реализован как отдельная функция.

Вызовите функцию-меню.

Основные теоретические положения

Основным типом данных, с которым будет работать программа — список.

Используя условный оператор «If» будет создана функция-меню, которая в зависимости от введенного номера будет выполнять определенные действия с списком.

Проверка элементов списка в каждой функции проводится с помощью цикла «For» и условного оператора «If».

Ход выполнения работы

Программа получает на вход десять элементов списка, разделенных знаком переноса строки, а также номер функции-меню.

В зависимости от номера, к списку применяются 6 различных функций.

При вводе «0», вызывается функция «zero», которая с помощью конструкции try - except проверяет каждый элемент посредством цикла «for» на то, можно ли преобразовать строку в число.

При вводе «1», вызывается функция «first», которая также используя конструкцию try - except, проверяет является ли текущий элемент числом и посредством условного оператора «If» находит все четные элементы и складывает их.

При вводе «2», вызывается функция «second», которая сравнивает длину каждого элемента списка с числом 3 и производит конкатенацию отобранных элементов.

При вводе «3», вызывается функция «third», которая с помощью конструкции tryexcept определяет, является ли элемент числом, а затем выводит произведение всех нечетных элементов

При вводе «4», вызывается функция «fourth», которая преобразует каждый элемент списка таким образом, что остаются только каждые пятые символы. Затем с помощью конструкции tryexcept проверяется возможно ли преобразование полученного элемента в число и если это невозможно, то элемент становится частью новой строки, которая идет на вывод.

При вводе «5», вызывается функция «fifth», которая выводит отдельно каждый элемент списка, а также индекс.

Вывод

В ходе написания работы были изучены основные управляющие конструкции языка Python, в частности, цикл «for», условный оператор «if», типы данных «список» и «исключение», а также различные методы работы со списком и строкой, описанные в ходе выполнения работы.

ПРИЛОЖЕНИЕ

КОД ПРОГРАММЫ

```
def zero(list):
    sort = ''
    for i in list:
        try:
            buf = int(i)
            sort += str(buf) + ' '
        except Exception:
            continue
    print(sort)

def first(list):
    res = 0
    for i in list:
        try:
            buf = int(i)
            if buf % 2 == 0:
                res += buf
        except Exception:
            continue
    print(res)

def second(list):
    sort = ''
    for i in list:
        if len(i) <= 3:
            sort += i
    print(sort)

def third(list):
    res = 1
    for i in list:
        try:
            buf = int(i)
            if buf % 2 == 1:
                res *= buf
        except Exception:
            continue
    print(res)

def fourth(list):
    sort = ''
    for i in list:
        buf = i[5::5]
        try:
            buf = int(buf)
            continue
        except Exception:
            if buf != '':
                sort += buf + ' '
    print(sort)

def fifth(list):
    for i in range(0, len(enter), 1):
        print(i, enter[i])

def menu(enter, op):
    if op == '0':
        zero(enter)
    elif op == '1':
        first(enter)
    elif op == '2':
        second(enter)
    elif op == '3':
        third(enter)
    elif op == '4':
        fourth(enter)
    elif op == '5':
        fifth(enter)
    else:
        print(' '.join(enter))

enter = []
for i in range(0, 10, 1):
    enter.append(input())
op = input()
menu(enter, op)
```