**Лабораторная работа № 8**

**Тема: Процедуры и функции. Рекурсия.**

**(6 часов)**

**Цель работы:** закрепление практических навыков по применению процедур, функций и рекурсии в программах.

**Студент должен:**

знать:

1. понятие процедуры и функции;
2. запись синтаксической диаграммы определения процедуры и функции;
3. связь между фактическими и формальными параметрами;
4. что такое рекурсия.

уметь:

1. правильно оформлять (записывать) тело процедуры и функции (в соответствии с синтаксическими диаграммами);
2. использовать функции в оконном приложении PyQt5.
3. написать программу с использованием рекурсии.

**Контрольные вопросы:**

1. Понятие процедуры.
2. Вид процедуры.
3. Параметры процедуры.
4. Понятие функции.
5. Структура функции.
6. Параметры функции.

**Задание:**

1. Разработка программ в PyQt5, реализующих функции рекурсивно: нахождение наименьшего общего делителя; функция Аккермана; сумма цифр числа, применяя только арифметические операции.
2. Разработка программы в PyQt5, согласно своего варианта.

**Варианты заданий**

**Вариант 1** Даны две последовательности целых чисел. Длину последовательностей задавать с клавиатуры. Числа в последовательности задавать случайным образом из диапазона от 0 до 20. Найти количество четных чисел в первой последовательности и количество нечетных во второй. (Определить функцию, позволяющую распознавать четные числа.)

**Вариант 2** Дана последовательность целых чисел. Длину последовательности задавать с клавиатуры. Числа в последовательности задавать случайным образом из диапазона от 0 до 500. Найти количество чисел, являющихся квадратами. (Определить функцию, позволяющую распознавать квадраты.)

**Вариант 3** Дана последовательность целых чисел. Длину последовательности задавать с клавиатуры. Числа в последовательности задавать случайным образом из диапазона от 0 до 1000. Найти количество чисел, являющихся степенями пятерки, вывести все эти числа. (Определить функцию, позволяющую распознавать степени пятерки.)

**Вариант 4.** Задавать случайным образом время (в часах H, минутах M, секундах S) и целое число T, задающее на сколько секунд увеличить данное время и вывести новые значения H, M, S. (Определить процедуру, которая увеличивает на T секунд время, заданное в часах H, минутах M, секундах S).

**Вариант 5** К целому положительномучислу добавлять справа некоторое количество цифр. Для этого задавать случайным образом количество добавляемых цифр в диапазоне от 1 до 4. Затем случайным образом получать само число в диапазоне от 1 до 999. Сообщать пользователю это число, сколько цифр он должен добавить и давать возможность пользователю добавлять эти цифры с клавиатуры в диапазоне от 0 до 9. (Определить функцию, добавляющую к целому положительному числу справа цифры. С помощью этой процедуры последовательно добавить к числу справа данные цифры, выводя результат каждого добавления).

**Вариант 6** Вводить по четыре числа A, B, C, D с клавиатуры или случайным образом в диапазоне от 0 до 1000. Упорядочить по убыванию эти наборы из четырех чисел (Определить процедуру, меняющую содержимое переменных A, B, C, D таким образом, чтобы их значения оказались упорядоченными по убыванию).

**Вариант 7** Задавать с клавиатуры натуральное шестизначное число или случайным образом в диапазоне от 100000 до 999999. Определить счастливое ли это число. *Счастливым* называют такое шестизначное число, в котором сумма его первых трех цифр равна сумме его последних трех цифр. (Определить функцию для расчета суммы цифр трехзначного числа.) Должна быть возможность бесконечного ввода этих чисел.

**Вариант 8** Даны два натуральных числа, задаваемых с клавиатуры или случайным образом из диапазона от 121 до 10000. Выяснить, является ли оно палиндромом ("перевертышем"), т. е. таким числом, десятичная запись которого читается одинаково слева направо и справа налево. (Определить функцию, позволяющую распознавать числа-палиндромы.)

**Вариант 9** Даны шесть различных чисел, вводимых двумя способами – с клавиатуры или случайным образом в диапазоне от -30 до 30. Определить максимальное из них. (Определить функцию, находящую максимум из двух различных чисел.)

**Вариант 10** Предлагать пользователюзадать величину угла в радианах или градусах. Если пользователь выбрал в градусах, то задавать величину угла в градусах в диапазоне от 0 до 360 с клавиатуры или случайным образом, затем с помощью функции перевести эту величину из градусов в радианы. Воспользоваться следующим соотношением: 180°=π радианов. В качестве значения π использовать 3.14. Если задавать в радианах, то брать диапазон от 0 до 2·π. (Определить функцию, находящую величину угла в радианах, если дана его величина в градусах. И функцию, находящую величину угла в градусах, если дана его величина в радианах).

**Вариант 11** Выдавать случайным образом число в диапазоне от 0 до 10. Вычислять вторую, третью, четвертую и пятую степень этого числа. (Определить процедуру, вычисляющую вторую, третью, четвертую и пятую степень числа).

**Вариант 12** Задавать с клавиатуры два числа (x, y) или случайным образом в диапазоне от -10 до 10. Определять номер координатной четверти, в которой находится точка с этими ненулевыми вещественными координатами (x, y). (Определить функцию, которая находит номер координатной четверти). Если введен 0 для любой координаты, то сообщать об этом пользователю и предлагать ввести координаты заново.

**Вариант 13** Вводить с клавиатуры или случайным образом число в диапазоне от 500 до 10000, являющееся годом. Определять високосный это год или нет и сообщать об этом пользователю. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400. (Определить функцию, которая находит високосный это год или нет).

**Вариант 14** Даны два натуральных числа, задаваемых с клавиатуры или случайным образом в диапазоне от 1 до 9999. Выяснить, в каком из них сумма цифр больше. (Определить процедуру для расчета суммы цифр натурального числа.). Должна быть возможность бесконечного ввода этих чисел.

**Вариант 15** Дана последовательность целых чисел. Длину последовательности задавать с клавиатуры. Числа в последовательности задавать случайным образом из диапазона от 100 до 999. Найти все трехзначные простые числа - число, большее 1, называется простым, если оно не имеет положительных делителей, кроме 1 и самого себя. (Определить функцию, позволяющую распознавать простые числа.)

**Вариант 16** Задавать случайным образом или с клавиатуры время в секундах в диапазоне от 1 до 100000. Определять по времени (в секундах) содержащееся в нем количество часов H, минут M и секунд S (T — входной, H, M и S — выходные параметры целого типа). (Определить процедуру, которая находит количество часов, минут и секунд отрезка времени).

**Вариант 17** Дано число N (> 1) и последовательность из 15 целых положительных чисел. Определить является ли какое-нибудь из этих чисел степенью числа N. Число N вводить с клавиатуры, последовательность задавать случайным образом в диапазоне от 4 до 100 (Описать функцию, определяющую степень числа N в данной последовательности).

**Вариант 18** Вводить с клавиатуры числа или задавать их случайным образом в диапазоне от 1 до 99999. Определять - является ли это число простым, подсчитать их количество. Число, большее 1, называется простым, если оно не имеет положительных делителей, кроме 1 и самого себя. (Определить функцию, находящую простое число.)

**Вариант 19** Вводить по два числа X и Y с клавиатуры или случайным образом в диапазоне от 0 до 1000. Вычислять среднее арифметическое равное (X+Y)/2 и среднее геометрическое равное X\*Y. (Определить процедуру, вычисляющую среднее арифметическое и среднее геометрическое двух положительных чисел).

**Вариант 20** Даны два натуральных числа, задаваемых с клавиатуры или случайным образом в диапазоне от 0 до 10000. Выяснить, в каком из них больше цифр. (Определить функцию для расчета количества цифр натурального числа.)

**Вариант 21** Водить с клавиатуры три числа A, B и C. Найти сумму всех целых чисел от A до B и от B до C включительно (A, B, C — целые). (Определить функцию, считающую сумму чисел от первого заданного числа до второго заданного числа. С помощью этой функции найти суммы чисел от A до B и от B до C если A> B, то функция возвращает 0.)

**Вариант 22** Вводить по четыре числа A, B, C, D с клавиатуры или случайным образом в диапазоне от 0 до 1000. Упорядочить по возрастанию эти наборы из четырех чисел (Определить процедуру, меняющую содержимое переменных A, B, C, D таким образом, чтобы их значения оказались упорядоченными по возрастанию).

**Вариант 23** Вводить с клавиатурылюбое положительное число. Написать рекурсивную функцию нахождения цифрового корня натурального числа. Цифровой корень данного числа получается следующим образом. Если сложить все цифры этого числа, затем все цифры найденной суммы и повторять этот процесс, то в результате будет получено однозначное число (цифра), которая и называется *цифровым корнем* данного числа.

**Вариант 24** Вводить с клавиатурыдва числа (радиус R1 внутренней окружности и радиус R2 внешней окружности, R 1> R2) или задавать их случайным образом в диапазоне от 5 до 20. Рассчитать площадь кольца, заключенного между двумя окружностями с общим центром и радиусами R1 и R2. (Определить функцию, находящую площадь кольца). Воспользоваться формулой площади круга радиуса R: S = π· R2. В качестве значения π использовать 3.14.

**Вариант 25** Вводить с клавиатуры целое положительное число K или задавать его случайным образом в диапазоне от 1 до 99999. Пользователь задает номер N любой позиции из этого числа в диапазоне от 1 до 9. Программа выдает какая цифра стоит на этой позиции. Цифры в числе K нумеруются справа налево. (Определить функцию, определяющую цифру на заданной позиции N. Если количество цифр в числе K меньше N, то функция возвращает –1.)

**Вариант 26** К целому положительномучислу добавлять слева некоторое количество цифр. Для этого задавать случайным образом количество добавляемых цифр в диапазоне от 1 до 4. Затем случайным образом получать само число в диапазоне от 1 до 999. Сообщать пользователю это число, сколько цифр он должен добавить и давать возможность пользователю добавлять эти цифры с клавиатуры в диапазоне от 0 до 9. (Определить функцию, добавляющую к целому положительному числу слева цифры. С помощью этой процедуры последовательно добавить к числу слева данные цифры, выводя результат каждого добавления).

Приложение

***Подпрограмма*** – это часть операторов программы, выделенных в группу, оформленную в виде отдельной конструкции. Выделяют два вида подпрограмм: ***процедуры*** и ***функции***. Различия процедур и функций заключается в назначении и способе их использования.

***Функции*** нужны для вычисления нового значения некоторого выражения, и вызов функции должен быть операндом в выражении.

***Процедуры*** служат для задания последовательности действий и определения новых значений переменных в программе. Вызов процедуры играет роль оператора.

***Процедура***– это независимая именованная часть программы, которую после *однократного* описания можно *многократно* вызывать по *имени* из последующих частей программы для выполнения определенных действий. Если попытаться вывести значение выполнение процедуры – то будет выведено None.

**Процедуры и функции**

Если одна и та же часть программного кода используется в программе несколько раз, то удобно ее оформить в виде подпрограммы.

Процедуры и функции представляют собой программные блоки, которые могут вызываться из разных частей программы множество раз. Процедуры и функции имеют имя и список параметров. И процедура, и функция могут не иметь параметров.

**Процедуры**

Процедура – выполнение определенные действий без возвращаемого значения. Синтаксис создания процедуры:

def *<название процедуры>* (*<список параметров>*):

<команды>

Использование процедур позволяет структурировать код программы, облегчая как её создание, так и её чтение. При использовании как процедур, так и функции следует помнить об областях видимости: глобальная, локальная, нелокальная. Для изменения глобальной переменной в процедуре или функции следует перед названием переменной написать global, таким образом указав, что это глобальная переменная. Для использования переменной с «обёртки» (нелокальной) следует перед названием переменной написать nonlocal.

Также важно знать, какие типы данных являются изменяемыми и неизменяемыми, а также переменные, хранящие ссылку на объект (к примеру, на список), так как их использование напрямую может затронуть значения, используемые в основном коде программы при их обработке.

**Функции**

Функция – выполняет определенные действия и возвращает значение. Синтаксис создания функции:

def *<название функции>* (*<список параметров>*):

*<команды>*

return …

В качестве параметров можно передавать не только фиксированное количество аргументов. Применение \*args и \*\*kwargs используется для передачи множества аргументов.

**Рекурсия**

Суть применении рекурсии – вызов функции внутри функции. Хоть и использование рекурсии приводит к малой производительности и проблемами с памятью, её использование допустимо в решениях задач, где алгоритм реализуется исключительно рекурсивно, либо затраты на реализацию итерационно достаточно велики. Примером использовании рекурсии может служить синтаксический разбор арифметических выражений.  
Примером рекурсии может служить рекурсивная функция Фибоначчи:

def fib(n):

if n<=1:

return 1

return fib(n-1) + fib(n-2)

Если привести графический вид рекурсии данной функции, то для номера 5 дерево рекурсии будет выглядеть так:

fib(5)

/ \

fib(4) fib(3)

/ \ / \

fib(3) fib(2) fib(2) fib(1)

/ \ / \ / \

fib(2) fib(1) fib(1) fib(0) fib(1) fib(0)

/ \

fib(1) fib(0)