

# Практическое занятие №5 Разработка программы с использованием цикла for

## 1. Тема работы

Разработка программы с использованием цикла `for`. Организация повторяющихся действий для обработки последовательностей данных.

## 2. Цель работы

Изучить работу цикла `for` в Python, научиться применять его для перебора элементов последовательностей, освоить использование функций `range()`, `enumerate()`, изучить вложенные циклы и генераторы списков.

## 3. Задачи работы

- Научиться использовать цикл `for` для перебора элементов последовательностей
- Освоить функцию `range()` для генерации числовых последовательностей
- Изучить работу с вложенными циклами
- Написать 4 программы с использованием цикла `for` согласно индивидуальному варианту

## 4. Необходимое оборудование и ПО

- Компьютер с ОС Windows 10
- Установленные: VS Code, Python
- Установленные расширения VS Code: Python, Python Debugger

---

## 5. Ход работы

### Часть 1: Повторение и подготовка

1. Откройте вашу папку проекта в VS Code (например, `Иванов_CA-501`)
2. Создайте новую папку `Practice_5` внутри вашей основной папки
3. Создайте файлы для заданий: `for1.py`, `for2.py`, `for3.py`, `for4.py`

### Часть 2: Разбор примера программы

#### Пример 1: Программа "Таблица умножения"

```
# Вывод таблицы умножения для числа N

number = int(input("Введите число: "))

print(f"Таблица умножения для {number}:")
for i in range(1, 11): # от 1 до 10 включительно
    result = number * i
    print(f"{number} × {i} = {result}")
```

## Пример 2: Программа с перебором строки и использованием enumerate

```
# Анализ строки: подсчёт гласных и их позиций

text = input("Введите текст: ")
vowels = "аеёиоуыэюяаеіou" # гласные буквы
vowel_count = 0

print("Позиции гласных букв:")
for index, char in enumerate(text.lower()):
    if char in vowels:
        print(f"Позиция {index}: '{char}'")
        vowel_count += 1

print(f"Общее количество гласных: {vowel_count}")
```

## Пример 3: Вложенные циклы

```
# Вывод прямоугольника из символов

width = int(input("Введите ширину: "))
height = int(input("Введите высоту: "))
symbol = input("Введите символ: ")

for i in range(height):
    for j in range(width):
        print(symbol, end=" ")
    print() # переход на новую строку
```

## Пошаговый разбор:

1. Цикл `for` - перебирает элементы последовательности

2. **Функция** `range()` - создаёт последовательность чисел
3. **Функция** `enumerate()` - возвращает пары (индекс, элемент)
4. **Вложенные циклы** - циклы внутри других циклов
5. **Параметр** `end` - определяет, что выводить после `print()`

### Запуск программы:

- Сохраните файл (Ctrl+S)
- Запустите через Ctrl+F5 или в терминале:

```
python for1.py
```

### Часть 3: Выполнение задания по варианту

1. **Получите задание** у преподавателя согласно вашему варианту
2. **Реализуйте программы** в соответствующих файлах
3. **Протестируйте программы** на разных входных данных
4. **Убедитесь**, что циклы работают корректно

### \*Часть 4: Выполните дополнительные задания

---

## 6. Варианты заданий

### Вариант 1:

1. Напишите программу, которая выводит все чётные числа от 2 до N.
2. Напишите программу, которая вычисляет сумму квадратов чисел от 1 до N.
3. Напишите программу, которая находит наименьшую цифру в введённом числе.
4. \* Напишите программу, которая выводит все простые числа в диапазоне от 2 до N.

### Вариант 2:

1. Напишите программу, которая выводит числа от N до 1 в обратном порядке.
2. Напишите программу, которая вычисляет факториал числа N.
3. Напишите программу, которая находит наибольшую цифру в введённом числе.
4. \* Напишите программу, которая выводит треугольник из чисел (например, для N=4: 1, 12, 123, 1234).

### Вариант 3:

1. Напишите программу, которая выводит все числа кратные 3 в диапазоне от 1 до N.

2. Напишите программу, которая вычисляет среднее арифметическое N введённых чисел.
3. Напишите программу, которая проверяет, является ли строка палиндромом.
4. \* Напишите программу, которая выводит треугольник Паскаля до N-й строки.

#### Вариант 4:

1. Напишите программу, которая выводит таблицу степеней (до N) для заданного числа. Например для числа 3 до степени N=4:  $3^1=3$ ,  $3^2=9$ ,  $3^3=27$ ,  $3^4=81$
2. Напишите программу, которая вычисляет сумму ряда:  $1 - 1/2 + 1/3 - 1/4 + \dots \pm 1/N$ .
3. Напишите программу, которая переворачивает введённую строку.
4. \* Напишите программу, которая преобразует введенный текст следующим образом: буквы выводятся поочередно в верхнем и нижнем регистре. Например для текста `привет, как твои дела?` : `ПрИвЕт, КаК тВоИ дЕлА?`

#### Вариант 5:

1. Напишите программу, которая выводит все делители числа N.
2. Напишите программу, которая вычисляет N-е число Фибоначчи.
3. Напишите программу, которая подсчитывает количество уникальных символов в строке.
4. \* Напишите программу, которая выводит ромб из символов. Необходимо спросить у пользователя сторону ромба и символ.

---

## 7. Отчёт по выполнению работы

Отчёт должен содержать:

1. **Цель работы**
  2. **Задание по варианту** (текст всех 4 заданий)
  3. **Листинги программ** (код всех 4 программ)
  4. **Скриншоты выполнения программ** (результаты работы в терминале)
  5. **Выводы:**
    - С какими трудностями столкнулись при написании программ?
    - В каких задачах целесообразно использовать цикл `for` вместо `while`?
    - Какая программа была самой сложной и почему?
    - Какие новые возможности цикла `for` вы узнали?
-

## 8. Контрольные вопросы

1. Чем цикл `for` отличается от цикла `while` ?
  2. Какие типы данных можно перебирать с помощью цикла `for` ?
  3. Что делает функция `range()` и какие параметры она принимает?
  4. Для чего используется функция `enumerate()` ?
  5. Что такое вложенные циклы и в каких задачах они применяются?
  6. Как работает функция `zip()` ?
  7. Что такое генераторы списков и как они связаны с циклом `for` ?
  8. Можно ли использовать `else` с циклом `for` ? Если да, то когда он выполняется?
- 

## 9. Дополнительные задания

1. Напишите программу, которая находит все совершенные числа в диапазоне до 1000
  2. Реализуйте шифр Цезаря с использованием цикла `for`
- 

## 10. Рекомендации по выполнению

- Используйте осмысленные имена переменных в циклах
- Комментируйте сложные участки кода
- Тестируйте программы на разных входных данных
- Для отладки используйте вывод промежуточных значений
- Обращайте внимание на граничные условия в функциях `range()`