

### 3 ЛЕКЦИЯ: УПРАВЛЕНИЕ ПОТОКОМ ВЫПОЛНЕНИЯ ПРОГРАММЫ В PYTHON

Управление потоком выполнения программы — это важный аспект программирования, который позволяет программам принимать решения и повторять действия. В Python для этого используются конструкции: условия (`if`, `else`), циклы (`for`, `while`) и обработка исключений (`try`, `except`).

#### 1. Условия (`if`, `else`)

Условия позволяют выполнять различные блоки кода в зависимости от выполнения определенных условий. Основные конструкции:

-`if`: проверяет условие.

-`elif`: (сокращение от "else if") проверяет дополнительные условия, если предыдущее не выполнено.

-`else`: выполняется, если ни одно из предыдущих условий не выполнено.

Пример:

```
x = 10
```

```
if x > 0:
```

```
    print("Положительное число")
```

```
elif x < 0:
```

```
    print("Отрицательное число")
```

```
else:
```

```
    print("Ноль")
```

#### 2. Циклы

Циклы позволяют повторять блоки кода несколько раз, пока выполняется определенное условие.

-Цикл `for`: используется для итерации по последовательностям (например, спискам, строкам).

Пример:

```
for i in range(5):
```

```
    print(i) # Выведет числа от 0 до 4
```

-Цикл `while`: выполняет блок кода, пока условие истинно.

Пример:

```
count = 0
```

```
while count < 5:
```

```
    print(count)
```

```
    count += 1 # Увеличиваем count на 1
```

#### 3. Обработка исключений (`try`, `except`)

Обработка исключений позволяет управлять ошибками, которые могут возникать во время выполнения программы. Это помогает избежать аварийного завершения программы и позволяет контролировать, что делать в случае ошибки.

-`try`: блок кода, в котором может произойти ошибка.

-`except`: блок кода, который выполняется, если в `try` произошла ошибка.

Пример:

try:

```
result = 10 / 0 # Деление на ноль
```

except ZeroDivisionError:

```
print("Ошибка: Деление на ноль!")
```

### **Заключение**

Используя конструкции управления потоком выполнения, такие как условия, циклы и обработка исключений, вы можете создавать более сложные и гибкие программы. Эти инструменты позволяют обрабатывать различные сценарии и ошибки, делая ваш код более надежным и удобным для пользователя.

### **Рекомендации**

- Практикуйтесь с условиями и циклами на небольших примерах.
- Изучайте различные типы исключений и способы их обработки.
- Пишите код с учетом возможных ошибок и обеспечивайте его читаемость.

Это основные аспекты управления потоком выполнения в Python, которые вы должны знать для создания эффективных программ.