

Basic Java №7

В программировании "циклические переменные" обычно относятся к переменным, используемым в циклах для управления их выполнением и отслеживания текущего состояния. В Java (и других языках программирования) циклы позволяют выполнять блок кода многократно, пока выполняется определенное условие.

Типы циклов

1. **Цикл `for`**:

Используется, когда известно количество повторений.

```
```java
public class Main {
 public static void main(String[] args) {
 // Переменная цикла i
 for (int i = 0; i < 10; i++) {
 System.out.println("i = " + i);
 }
 }
}
...`
```

В этом примере `i` является циклической переменной. Цикл `for` выполняется, пока `i` меньше 10, увеличивая значение `i` на 1 после каждой итерации.

#### 2. \*\*Цикл `while`\*\*:

Используется, когда не известно точное количество повторений, и цикл продолжается до выполнения условия.

```
```java
public class Main {
    public static void main(String[] args) {
        int i = 0; // Переменная цикла i
        while (i < 10) {
            System.out.println("i = " + i);
            i++;
        }
    }
}
...`
```

В этом примере `i` является циклической переменной, и цикл продолжается, пока `i` меньше 10.

3. **Цикл `do-while`**:

Похож на `while`, но гарантирует выполнение блока кода хотя бы один раз.

```
```java
public class Main {
 public static void main(String[] args) {
 int i = 0; // Переменная цикла i
 do {
 System.out.println("i = " + i);
 i++;
 } while (i < 10);
 }
}
```
```

В этом примере `i` является циклической переменной, и цикл выполняется хотя бы один раз, даже если условие уже ложное.

Оператор `++` увеличивает значение переменной на единицу, а оператор `--` уменьшает значение переменной на единицу.

Системная команда `System.out.println` в Java используется для вывода текста на консоль и перехода на новую строку.

При одновременном использовании внешнего и внутреннего циклов с помощью оператора `break` можно завершить выполнение цикла и отсеять вывод ненужного результата.

Оператор `continue` используется в циклах для пропуска оставшейся части текущей итерации и перехода к следующей итерации цикла. Он позволяет "перезапустить" цикл без выполнения кода, который следует за ним, в текущей итерации.

Тернарный оператор в Java позволяет присвоить значение переменной на основе условия.

```
int result = (someValue % 2 == 0) ? 1 : 2;
```
```

В этом примере переменная `result` будет равна `1`, если `someValue` делится на 2 без остатка (то есть условие `someValue % 2 == 0` истинно). В противном случае `result` будет равна `2`.

Тернарный оператор сокращает запись условий `if-else`, делая код более компактным.

Локальные переменные исчезают после завершения выполнения метода.