Мобильная разработка

Урок 3

**Циклы**

Для того чтобы «заставить» компьютер повторять последовательность действий, нужно использовать циклы. Цикл в программе организовать просто. По структуре он очень похож на if.

В Java выделяют 4 вида циклов:

1. Цикл while
2. Цикл do…while
3. Цикл for
4. Цикл for each**.** Его пока рассматривать не будем, рассмотрим не следующем уроке.

**Цикл while**

Пример в коде:

*int x = 0;*

*while (x < 5){*

*System.оut.print(x + " ");*

*x++;*

*}*

Слово while в переводе с английского значит «до тех пор», «пока». Программа будет выполнять команды до тех пор, пока условие истинно. Проверяется условие, если оно истинно, выполняются команды, и программа переходит опять на проверку условия.

Блок команд, расположенных после while, называют телом цикла. Т

Запомните: суть оператора while: он повторяет действие(-я) до тех пор, пока условие истинно.

Условия, как и в if, могут быть абсолютно любыми логическими выражениями, например, содержать логические операции && и ||.

Пример бесконечного цикла

*while (true) {*

*System.out.print(i + " ");*

*i++;*

*}*

**Цикл do...while**

Пример в коде:

*int x = 0;*

*do {*

*оut.print(x + " ");*

*x++;*

*} while (x < 5);*

цикл while сначала проверяет условие, потом выполняет код (тело цикла), а do..while сначала выполняет тело цикла, а потом проверяет условие, поэтому тело цикла выполнится, как минимум, 1 раз.

### **Цикл for**

Цикл for еще называют цикл "со счетчиком". Когда Вы заранее знаете сколько раз повторится одно и то же действие, необходимо использовать цикл for.

Пример в коде:

*for (int i = 0; i < 10; i++){*

*out.println(i);*

*}*

Первая часть объявления цикла - int i = 0 создает и инициализирует счетчик i. Счетчик необязательно должен представлять тип int. Это может быть и любой другой числовой тип, например, float. Перед выполнением цикла значение счетчика будет равно 1. В данном случае это то же самое, что и объявление переменной.

Вторая часть - условие, при котором будет выполняться цикл. В данном случае цикл будет выполняться, пока i не достигнет 9.

И третья часть - приращение счетчика на единицу. Опять же нам необязательно увеличивать на единицу. Можно уменьшать: i--.

Пример цикла без инициализации и счестчика:

Пример в коде:

*int i = 1;*

*for (; i<9;){*

*System.out.printf("Квадрат числа %d равен %d \n", i, i \* i);    i++; }*

Пример бесконечного цикла:

*int i = 1;*

*for (; ;){*

*System.out.printf("Квадрат числа %d равен %d \n", i, i \* i);*

*}*

Пример цикла с несколькими переменными

*int n = 10;*

*for(int i=0, j = n - 1; i < j; i++, j--){*

*System.out.println(i \* j);*

*}*

### **Операторы continue и break**

Оператор break позволяет выйти из цикла в любой его момент, даже если цикл не закончил свою работу. Оператор break не проектировался в качестве нормального средства завершения цикла. Эту цель обслуживает условное выражение заголовка цикла. Оператор break следует использовать для прерывания цикла только тогда, когда возникают некоторые специальные ситуации.

Пример кода:

*for (int i = 0; i < 10; i++){*

*if (i == 5)*

*break;*

*System.out.println(i);*

*}*

Оператор continue позволяет перейти к следующей итерации цикла в любой момент текущей итерации:

Пример кода:

*for (int i = 0; i < 10; i++){*

*if (i == 5)*

*continue;*

*System.out.println(i);*

*}*

Иногда возникают ситуации, когда требуется выход из многократно вложенных циклов.

Для решения подобных задач в Java применяется расширенная форма инструкции break, используя которую можно выйти за пределы одного или нескольких блоков кода.

Пример кода:

*for (int i = 1; i < 10; i++) {*

*for (int j = 1; j < 10; j++) {*

*if (j >= 5) {*

*break out;*

*}*

*System.out.printf("%2d ", i \* j);*

*}*

*System.out.println();*

*}*

Выход из вложенных циклов с помощью break с отметкой:

Пример кода:

*public static void main(String[] args) {*

*outer:*

*for (int i = 0; i < 3; i++) {*

*System.out.print("Итерация " + i + ": ");*

*for (int j = 0; j < 100; j++) {*

*if (j == 10) {*

*break outer;*

*}*

*System.out.print(j + " ");*

*}*

*System.out.println("Эта строка никогда не будет выведена");*

*}*

*System.out.println("Цикл закончен");*

*} }*

**Вложенные циклы**

Циклы можно вкладывать друг в друга. С помощью вложенных циклов решаются самые разные задачи.

Для наглядности, давайте выведем таблицу умножения.

Пример кода:

*public static void main(String[] args) {*

*for (int i = 1; i < 10; i++) {*

*for (int j = 1; j < 10; j++) {*

*System.out.printf("%2d ", i \* j);*

*}*

*System.out.println();*

*}*

*}*

Область видимости переменной цикла

Пример кода:

*int x = 0;  
do {*

*int t;  
 System.out.print(x + " ");*

*x++;  
   
} while (x < 5);  
t = 6; // ОШИБКА! Переменная t здесь не существует*

Пример кода:

*for (int i = 10; i > 0; i--){*

*out.println(i);*

*}*

*out.println(i); // ОШИБКА! Переменная i здесь не существует*

**Практическое занятие**

Android-практикум. Вопросы по прохождению тестов.