Циклические конструкции

|  |  |  |  |
| --- | --- | --- | --- |
| № урока: | 6 | Курс: | JAVA Starter |
|  | | |  |
| Средства обучения: | | | Компьютер с установленной IntelliJ IDEA |

# Обзор, цель и назначение урока

Рассмотрение циклических конструкций.

**Изучив материал данного занятия, учащийся сможет:**

* Понимать работу циклических операторов.
* Понимать работу операторов безусловного перехода.
* Применять циклические конструкции while, do-while, for.
* Работать с вложенными циклами.
* Понимать работу циклов с охраняемыми ветвями.

# Содержание урока

1. Обзор циклических конструкций.
2. Рассмотрение цикла с предусловием while.
3. Рассмотрение примеров: Использование цикла с предусловием while.
4. Рассмотрение цикла с постусловием do-while.
5. Рассмотрение примеров: Использование цикла с постусловием do-while.
6. Рассмотрение цикла со счетчиком for.
7. Рассмотрение примеров: Использование цикла со счетчиком for.
8. Рассмотрение оператора досрочного выхода из цикла break.
9. Рассмотрение оператора пропуска итерации continue.
10. Рассмотрение вложенных циклов.
11. Рассмотрение примера: Использование вложенных циклов.
12. Рассмотрение цикла Дейкстры.
13. Рассмотрение примера: Использование цикла Дейкстры.
14. Рассмотрение усложненной формы цикла Дейкстры – цикла «Паук».
15. Рассмотрение примера: Использование цикла «Паук».
16. Рассмотрение упрощенной формы цикла Дейкстры.
17. Рассмотрение примера: Использование упрощенной формы цикла Дейкстры.
18. Бесконечные циклы.
19. Рассмотрение примера: Бесконечные циклы.

# Резюме

* Цикл — это управляющая конструкция, предназначенная для организации многократного исполнения набора инструкций.
* Также циклом может называться любая многократно исполняемая последовательность инструкций, организованная любым способом (например, с помощью условного перехода).
* Итерация – это один проход цикла.
* Цикл, с предусловием while – это цикл, который выполняется до тех пор, пока условие удовлетворяет истинности. Условие проверяется до выполнения тела цикла. Если изначально условие не удовлетворяет истинности, то тело цикла while ни разу не выполнится.
* Цикл, с постусловием do-while – это цикл, в котором условие проверяется после выполнения тела цикла. Отсюда следует, что тело do-while выполняется хотя бы один раз.
* Цикл со счетчиком for – это цикл, в котором переменная – счетчик итераций цикла, с определенным шагом, изменяет свое значение до заданного конечного значения.
* Блок выражений цикла for, содержит три выражения:

for ( начальное-выражение; условное-выражение; выражение-цикла ) { тело цикла } или в другой нотации это звучит так:

for ( инициализация; условие; модификация ) { тело цикла }

* В теле цикла for, разрешено изменение значения начального выражения (т.е. счетчика итераций).
* Циклы с выходом из середины, отсутствуют в JAVA, но такой цикл можно смоделировать при помощи любого существующего цикла и оператора досрочного выхода break.
* Досрочный выход из цикла. Команда досрочного выхода из цикла break, применяется тогда, когда необходимо прервать выполнение цикла, в котором условие выхода еще не достигнуто. Работу цикла есть смысл прерывать, если например, при выполнении тела цикла, обнаруживается ошибка, после которой дальнейшее выполнение цикла не имеет смысла.
* Оператор досрочного выхода из цикла break, применяется только для того цикла в котором он непосредственно находится.
* Пропуск итерации. Оператор пропуска итерации continue, применяется, когда необходимо пропустить все команды до конца тела цикла.
* Неструктурные средства безусловных переходов: break, continue. С точки зрения структурного программирования, команды досрочного выхода из цикла и продолжения итерации считаются избыточными. Нужно стараться моделировать их действия чисто структурными средствами – условиями и циклами.
* С точки зрения Эдсгера Дейкстры (это просто его мнение), сам факт использования в программе неструктурных средств, будь то оператор безусловного перехода goto (не используется в Java) или одной из его специализированных форм – операторов break и continue, является свидетельством недостаточно проработанного алгоритма решения задачи.
* Несмотря на свою ограниченную полезность и возможность замены на другие языковые конструкции, команды пропуска итерации и, особенно, досрочного выхода из цикла в отдельных случаях оказываются полезны, именно поэтому они сохраняются в JAVA и других современных языках программирования.
* Совместные циклы - foreach. В языке Java применяется как специальная запись цикла for для прохождения массивов и списков. Детально рассматривается в дальнейших курсах.
* ***Вложенные циклы*** – это циклы, организованные в теле другого цикла. Вложенный цикл в тело, другого цикла, называется внутренним циклом. Цикл, в теле которого существует вложенный цикл, называется ***внешним***.
* Полное число исполнений внутреннего цикла, всегда равно произведению числа итераций внутреннего цикла на произведение чисел итераций всех внешних циклов.
* Одна из проблем, связанных с вложенными циклами – это организация досрочного выхода из них. Решений у этой проблемы несколько, одна из них - использовать оператор завершения цикла break.
* Циклы с несколькими охраняемыми ветвями. Цикл Дейкстры и «Паук».
* Цикл Дейкстры состоит из одной или нескольких ветвей (охраняемых выражений),

каждая, ветвь представляет собой пару из охраняющего условия и охраняемой команды.

* Цикл «Паук» - это модифицированный цикл Дейкстры с явными условиями выхода.
* Бесконечным циклом называется цикл, написанный таким образом, что условие выхода из него никогда не выполняется.
* Бесконечный цикл while выглядит так: while (true) { тело цикла }
* Бесконечный цикл do-while выглядит так: do { тело цикла } while (true)
* Бесконечный цикл for выглядит так: for ( ; ; ) { тело цикла }
* В написании программ, решающих реальные задачи, бесконечные циклы, как правило, используются очень редко и являются одним из источников неустойчивой работы программы. Например, бесконечные циклы есть смысл использовать в многопоточном программировании, в потоках контролирующих работу других потоков.

# Закрепление материала

* Что такое цикл?
* Перечислите известные Вам циклические конструкции.
* Где и для чего используются циклические конструкции?
* Значения, какого типа можно передавать в качестве параметра while()?
* Что такое итерация?
* В чем разница между циклом while и do-while?
* Для чего используются служебные слова continue и break?
* Что такое цикл Дейкстры?
* В чем отличие цикла «Паук» от цикла Дейкстры?
* Какой цикл предпочтительней использовать для расчета факториала?
* Назовите конструкцию пропуска итерации.
* Что такое цикл с выходом из середины и как его организовать?

# Дополнительное задание

Задание

Используя IntelliJ IDEA, создайте класс **Rectangle**.

Создайте две целочисленные переменные и задайте им некоторые значения. Применяя технику вложенных циклов, нарисуйте прямоугольник из звездочек. Используйте значения ранее созданных переменных для указания высоты и ширины прямоугольника.

# Самостоятельная деятельность учащегося

Задание 1

Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2

Используя IntelliJ IDEA, создайте класс **SumMin**.

Дано два числа A и B (A<B) выведите суму всех чисел расположенных между данными числами на экран.

Дано два числа A и B (A<B) выведите все нечетные значения, расположенные между данными числами.

Задание 3

Используя IntelliJ IDEA, создайте класс.

Используя циклы и метод:

System.out.print("\*"), System.out.print (" "), System.out.print ("\n") (для перехода на новую строку).

Выведите на экран:

* прямоугольник
* прямоугольный треугольник
* равносторонний треугольник
* ромб

Задание 4

Имеется N клиентов, которым компания производитель должна доставить товар. Сколько существует возможных маршрутов доставки товара, с учетом того, что товар будет доставлять одна машина?

Используя IntelliJ IDEA, создайте класс **Delivery**.

Напишите программу, которая будет рассчитывать, и выводить на экран количество возможных вариантов доставки товара. Для решения задачи, используйте факториал N!, рассчитываемый с помощью цикла do-while.

# Рекомендуемые ресурсы

Циклы while и do-while

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html>

Цикл for

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html>

Операторы прерываний break и continue

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/branch.html>