Циклічні конструкції

№ уроку: 6 **Курс:** Java Starter

Засоби навчання: Комп'ютер із встановленою IntelliJ IDEA

Огляд, мета та призначення уроку

Розгляд циклічних конструкцій.

Вивчивши матеріал цього заняття, учень зможе:

- Розуміти роботу циклічних операторів.
- Розуміти роботу операторів безумовного переходу.
- Застосовувати циклічні конструкції while, do-while, for.
- Працювати із вкладеними циклами.
- Розуміти роботу циклів із гілками, що охороняються.

Зміст уроку

1. Огляд циклічних конструкцій.
2. Розгляд циклу з передумовою while.
3. Розгляд прикладів: Використання циклу з передумовою while.
4. Розгляд циклу з умовою do-while.
5. Розгляд прикладів: Використання циклу з умовою do-while.
6. Розгляд циклу з лічильником for.
7. Розгляд прикладів: Використання циклу з лічильником for.
8. Розгляд оператора дострокового виходу із циклу break.
9. Розгляд оператора пропуску ітерації continue.
10. Розгляд вкладених циклів.
11. Розгляд прикладу: Використання вкладених циклів.
12. Розгляд циклу Дейкстри.
13. Розгляд прикладу: Використання циклу Дейкстри.
14. Розгляд ускладненої форми циклу Дейкстри – циклу «Павук».
15. Розгляд прикладу: Використання циклу «Павук».
16. Розгляд спрощеної форми циклу Дейкстри.
17. Розгляд прикладу: Використання спрощеної форми циклу Дейкстри.
18. Нескінченні цикли.
19. Розгляд прикладу: Нескінченні цикли.

Резюме

- **Цикл** – це керівна конструкція, яка призначена для організації багаторазового виконання набору інструкцій.
  Також циклом може називатися будь-яка багаторазова послідовність інструкцій, яка організована будь-яким способом (наприклад, за допомогою умовного переходу).
- **Ітерація** – це один прохід циклу.
- **Цикл із передумовою** while – це цикл, який виконується до того часу, поки умова задовольняє істинності. Умову перевіряють до виконання тіла циклу. Якщо спочатку умова не задовольняє істинності, то тіло циклу while жодного разу не виконається.
- **Цикл з постумовою** do-while – це цикл, у якому умова перевіряється після виконання тіла циклу. Звідси випливає, що тіло do-while виконується хоча б один раз.
- **Цикл із лічильником** for – це цикл, у якому змінна – лічильник ітерацій циклу з певним кроком змінює значення до заданого кінцевого значення.
- Блок виразів циклу for містить три вирази:

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

t. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

for (початковий вираз; умовний вираз; вираз циклу) { тіло циклу } або в іншій нотації це звучить так:
for (ініціалізація; умова; модифікація) { тіло циклу }

- У тілі циклу for дозволена зміна значення початкового виразу (тобто лічильника ітерацій).
- Цикли з виходом із середини відсутні в Java, але такий цикл можна змоделювати за допомогою будь-якого наявного циклу й оператора дострокового виходу break.
- **Достроковий вихід із циклу.** Команда дострокового виходу з циклу break застосовується тоді, коли необхідно перервати виконання циклу, у якому умова виходу ще досягнута. Варто переривати роботу циклу, якщо, наприклад, під час виконання тіла циклу знайдена помилка, після якої подальше виконання циклу немає сенсу.
- Оператор дострокового виходу з циклу break застосовується тільки для того циклу, у якому він безпосередньо розташований.
- **Пропуск ітерації**. Оператор пропуску ітерації continue застосовується тоді, коли необхідно пропустити всі команди до кінця тіла циклу.
- Неструктурні засоби безумовних переходів: break, continue. З погляду структурного програмування команди дострокового виходу з циклу та продовження ітерації вважаються надлишковими. Потрібно намагатися моделювати їхні дії суто структурними засобами – умовами та циклами.
- З погляду **Едсгера Дейкстри** (це просто його думка) сам факт використання в програмі неструктурних засобів, чи то оператора безумовного переходу goto (не використовується в Java), чи то однієї з його спеціалізованих форм – операторів break і continue, є свідченням недостатньо опрацьованого алгоритму розв'язку завдання.
- Попри свою обмежену корисність і можливість заміни іншими мовними конструкціями, команди пропуску ітерації і, особливо, дострокового виходу з циклу в окремих випадках виявляються корисними, саме тому вони зберігаються в Java та інших сучасних мовах програмування.
- **Спільні цикли** foreach. У мові Java foreach застосовується як спеціальний запис циклу для проходження масивів і списків. Докладно розглядається далі.
- **Вкладені цикли** – це цикли, які організовані в тілі іншого циклу. Вкладений цикл у тіло іншого циклу називається внутрішнім циклом. Цикл, у тілі якого наявний вкладений цикл, називається **зовнішнім**.
- Повна кількість виконань внутрішнього циклу завжди дорівнює добутку числа ітерацій внутрішнього циклу на добуток чисел ітерацій усіх зовнішніх циклів.
- Одна з проблем, яка пов'язана із вкладеними циклами, – це організація дострокового виходу з них. Розв'язків у цієї проблеми кілька. Один із них – використовувати оператор завершення циклу break.
- **Цикли з кількома гілками, що охороняються. Цикл Дейкстри та «Павук».**
- **Цикл Дейкстри** складається з однієї або декількох гілок (виразів, що охороняються). Кожна гілка – це пара з умови й команди, які охороняються.
- **Цикл «Павук»** – це модифікований цикл Дейкстри з явними умовами виходу.
- **Нескінченним циклом** називається цикл, який написаний так, що умова виходу з нього ніколи не виконується.
- Нескінченний цикл while виглядає так: while (true) { тіло циклу }
- Нескінченний цикл do-while виглядає так: do { тіло циклу } while (true)
- Нескінченний цикл for виглядає так: for (; ;) { тіло циклу }
- У написанні програм, які розв'язують реальні завдання, нескінченні цикли, як правило, використовуються дуже рідко та є одним із джерел нестійкої роботи програми. Наприклад, нескінченні цикли варто використовувати в багатопотоковому програмуванні, у потоках, які контролюють роботу інших потоків.

Закріплення матеріалу

- Що таке цикл?
- Перерахуйте відомі Вам циклічні конструкції.
- Де і для чого використовуються циклічні конструкції?
- Значення якого типу можна передавати як параметр while()?

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

t. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Що таке ітерація?
- У чому різниця між циклами while та do-while?
- Навіщо використовуються службові слова continue і break?
- Що таке цикл Дейкстри?
- У чому відмінність циклу «Павук» від циклу Дейкстри?
- Який цикл краще використовувати для розрахунку факторіала?
- Назвіть конструкцію пропуску ітерації.
- Що таке цикл із виходом із середини та як його організувати?

Додаткові завдання

Завдання 1
Використовуючи IntelliJ IDEA, створіть клас **Rectangle**.
Створіть дві цілочислові змінні та задайте їм деякі значення. Застосовуючи техніку вкладених циклів, намалюйте прямокутник із зірочок. Використовуйте значення раніше створених змінних для вказівки висоти та ширини прямокутника.

Завдання 2
Знайдіть послідовність Фібоначчі. Одне стартове число користувач вводить, друге вводить користувач до шуканого.

Самостійна діяльність учня

Завдання 1
Вивчіть основні конструкції та поняття, розглянуті на уроці.

Завдання 2
Використовуючи IntelliJ IDEA, створіть клас **SumMin**. Дано два числа A та B (A <B). Виведіть суму всіх чисел, які розташовані між цими числами на екран. Дано два числа A та B (A <B). Виведіть усі непарні значення, які розташовані між цими числами.

Завдання 3
Використовуючи IntelliJ IDEA, створіть клас **Printing_Shapes**. Використовуючи цикли та метод: `System.out.print("*"), System.out.print(" "), System.out.print("\n")` (для переходу на новий рядок).
Виведіть на екран:
- прямокутник;
- прямокутний трикутник;
- рівносторонній трикутник;
- ромб.

Завдання 4
Є N клієнтів, яким компанія-виробник має доставити товар. Скільки є можливих маршрутів доставлення товару з урахуванням того, що товар доставлятиме одна машина?
Використовуючи IntelliJ IDEA, створіть клас **Delivery**. Напишіть програму, яка розраховуватиме та виводитиме на екран кількість можливих варіантів доставлення товару. Для розв'язку задачі використовуйте факторіал N!, що розраховується за допомогою циклу do-while.

Завдання 5
Дано два цілих числа A і B (A < B). Знайдіть суму всіх цілих чисел від A до B включно.

Завдання 6
Дано ціле число N (> 0). Використовуючи один цикл, знайдіть суму 1 + 1 / (1!) + 1 / (2!) + 1 / (3!) +. . . + 1 / (N!)

Завдання 7

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

t. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Створіть програму таблиці множення для числа 7, використовуючи цикли. Приклад виведення в консоль:
7 * 1 = 7;
7 * 2 = 14;
…;
7 * 10 = 70.

Рекомендовані ресурси

Цикли while та do-while
https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html

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

Оператори переривань break та continue
https://docs.oracle.com/javase/tutorial/java/nutsandbolts/branch.html

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

t. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com