

Базові конструкції в програмуванні

№ урока: 2 **Курс:** Basis for Java

Засоби навчання: Інтегроване середовище розробки (IntelliJ Idea, Eclipse...), консоль.

Огляд, ціль та призначення уроку

Познайомитися з семантикою, синтаксисом та типами даних мови програмування Java. Познайомитися з масивами, як з інструментом зберігання даних. Познайомитися з умовними та циклічними конструкціями та навчитися створювати прості алгоритми обробки даних на їх основі.

Вивчивши матеріал даного заняття, студент зможе:

- Розбиратися в семантиці та синтаксисі мови програмування Java.
- Розрізняти типи даних та застосовувати потрібні для конкретних задач.
- Застосовувати масиви для зберігання даних.
- Виконувати різноманітні дії з даними.
- Застосовувати умовні конструкції.
- Застосовувати циклічні конструкції.
- Створювати прості алгоритми для вирішення задач.

Зміст уроку

1. Поняття синтаксису, семантики та типізації
2. Змінні, типи даних
3. Масиви
4. Операції над даними
5. Умовні конструкції
6. Циклічні конструкції
7. Алгоритми

Резюме

- Синтаксис – визначає набір правил, що визначає комбінації символів в тексті програми.
- Семантика — значення операторів, основних конструкцій мови, тощо.
- Мова програмування Java є чітко типізованою (strongly typed language). Це означає, що будь-яка змінна та будь-який вираз повинні мати відомий тип ще на момент компіляції.
- Типи даних в Java: byte, char, int, long, float, double, Boolean, String...
- Масив – це змінна, яка може містити не одне, а відразу кілька значень, на відміну від звичайної змінної, що містить єдине значення.
- Операції над даними в Java: арифметичні, логічні, операції відношення, операції з рядками...
- Умовні конструкції – це конструкції прийняття рішень. Найбільш розповсюджена умовна конструкція є конструкція if/else. Після if може слідувати вираз else if ... else. Дану конструкцію використовується для вибору одного з декількох блоків коду для виконання.
- Оператор switch також дозволяє будувати умовні конструкції. Після switch іде набір секцій case, а в кінці - default.
- Цикл – це конструкція, призначена для організації багаторазового виконання набору інструкцій (команд). Існує декілька циклічних конструкцій в мові програмування Java.

- Цикл for – управляюча конструкція повторення, застосовується при необхідності виконати інструкції певну кількість разів. Цикл корисний, коли заздалегідь відомо яка повинна бути кількість повторень.
- Цикл while – це цикл з передумовою, тіло циклу (тобто інструкція або блок інструкцій) виконується, якщо умова істинна. Заздалегідь невідомо скільки разів виконається код всередині циклу. Якщо умова хибна з самого початку, то цикл не виконається жодного разу.
- Алгоритм – набір інструкцій, які описують порядок дій виконавця, щоб досягти результату розв'язання задачі за скінченну кількість дій; система правил виконання дискретного процесу, яка досягає поставленої мети за скінченний час. Для візуалізації алгоритмів часто використовують блок-схеми.

Закріплення матеріалу

- Що таке синтаксис та семантика мови програмування?
- Що таке типізація?
- Мова Java є чітко типізованою чи м'яко типізованою?
- Які типи даних Ви знаєте?
- Що таке масив?
- Для чого використовують умовні конструкції?
- Які умовні конструкції Ви знаєте?
- Для чого використовують цикли?
- Які цикли Ви знаєте?
- Що таке алгоритм?

Додаткове завдання

Завдання

Написати цикл, в якому ваше будь-яке число буде множитись саме на себе певну кількість разів. Якщо добуток перевищить 1000, перервати цикл. Порахувати кількість ітерацій і вивести цю кількість на екран.

Самостійна діяльність студента

Завдання 1

Використовуючи свої знання, здобуті на лекції, написати самостійно програму в якому буде масив цілочисельних чисел, який заповнить різними числами (десяти чисел буде досить). Використовуючи цикл та умовну конструкцію для перевірки числа на парність виведіть на екран фразу «Число » + саме число + «парне/непарне», використавши метод System.out.println().

Завдання 2

Завантажити програму Расман та проаналізувати її код, відзначити знайомі з лекцій елементи. Запустити програму, але обов'язково з консолі, а не в інтегрованому середовищі розробки, та, слідуючи інструкції, пройти рівень міні гри. Метод керування трішечки відрізняється від методу у програмі з попереднього уроку MazeRunner.

Завдання 3

Це завдання підвищеної складності. Завантажити програму РасманAuto та проаналізувати її код. Ця програма дещо відрізняється від попередньої. Тут не треба самому курувати Расман-ом, він вміє рухатись самостійно. Майже вміє. В програмі реалізований весь функціонал для руху по лабіринту, але не вистачає алгоритму. Саме алгоритм Вам доведеться реалізувати. Расман рухається покроково, тому Вам необхідно буде застосувати цикл. Наперед невідомо, яка кількість кроків буде необхідна, тому визначайтеся з яким циклом тут треба працювати. Умовою

завершення програми вважається момент, коли Растман з'їсть усі крапочки. Існує багато алгоритмів проходження по лабіринту, наприклад «триматися рукою за ліву/праву стінку». Вам необхідно використати методи `turnLeft()` та `turnRight()`, які повертають Растман-а обличчям у відповідному напрямку, метод `step()` робить крок на одну «клітинку» вперед, а метод `isWall()` перевіряє, чи є зараз перед обличчям у Растман-а стіна, та повертає `true` або `false`. Код необхідно вставити в методі `main()`, у визначеному місці. Запускайте програму у консолі. Мій алгоритм зайняв усього 6 рядочків коду. Успіхів.

Рекомендовані ресурси

Список зарезервованих слів в Java.

<http://javaphp.ptnгу.com/javalessons/lesson2>

Операції над даними.

<https://www.bestprog.net/uk/2017/01/24/%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D1%97-%D0%B0%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%BD%D1%96-%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D1%97/>

Конструкції в Java.

<https://uk.myservername.com/agile-vs-waterfall-which-is-best-methodology>

Масиви.

<https://vertex-academy.com/tutorials/uk/masivi-v-java/>