

# Методи

№ уроку: 8 Курс: Java Starter

Засоби навчання: Комп'ютер із встановленою IntelliJ IDEA

## Огляд, мета та призначення уроку

Розгляд роботи методів.

## Вивчивши матеріал цього заняття, учень зможе

- Розуміти роботу методів.
- Розуміти роботу рекурсії.
- Розуміти перевантаження методів.
- Використовуйте методи з параметрами.

## Зміст уроку

1. Розгляд прикладу: Перевантаження методів.
2. Розгляд прикладів: Методи з параметрами.
3. Розгляд прикладів: Рекурсивний виклик методу.

## Резюме

- **Метод** – це іменована частина програми, яка може викликатися з інших частин програми стільки разів, скільки потрібно.
- **Сигнатура методу** – частина загального оголошення методу, що дає змогу ідентифікувати функцію серед інших. У Java до сигнатурі методу входить ідентифікатор методу, тип і кількість формальних аргументів.
- **Перевантаження методів** – створення однайменного методу, але з іншим типом та/або кількістю аргументів.
- Мінімізуйте кількість повернень із кожного методу. Важко зрозуміти логіку методу, коли під час аналізу нижніх рядків доводиться пам'ятати про можливі виходи у верхніх рядках.
- **Рекурсія** – виклик функції (процедури) з неї самої, безпосередньо (проста рекурсія) або через інші функції (складна рекурсія). Наприклад, функція A викликає функцію B, а функція B – функцію A. Кількість вкладених викликів функції або процедури називається глибиною рекурсії.
- Одне з ключових завдань рекурсивного методу – запобігання нескінченній рекурсії.
- Реалізація рекурсивних викликів функцій ґрунтуються на механізмі стека викликів – адреса повернення та локальні змінні функції записуються в стек, завдяки чому кожен наступний рекурсивний виклик цієї функції користується своїм набором локальних змінних і завдяки цьому працює коректно.
- На кожний рекурсивний виклик потрібна деяка кількість оперативної пам'яті. За надмірно великої глибини рекурсії може настати переповнення стека викликів. Унаслідок цього рекомендується уникати рекурсивних програм, які призводять (або в деяких умовах можуть призводити) до занадто великої глибини рекурсії.
- Не використовуйте рекурсію для обчислення факторіалів і чисел Фіbonacci.
- Оскільки методи – це конструкції для виконання дій, рекомендується називати їх дієслівними фразами або діесловами.
- Намагайтесь називати методи відповідно до завдань, які вони виконують, а не відповідно до деталей реалізації.
- Для іменування методів у Java рекомендується використовувати угоду camelCasing. Щоби виділити слова в ідентифікаторі, зробіть перші літери кожного слова (окрім первого) великими. Наприклад, `writeln`, `GetType`.
- Мова Java чутлива до регістру (case sensitivity). Наприклад, `GetType` та `getType` – це різні імена.

- Не використовуйте символи підкреслення, дефіси й інші неалфавітно-цифрові символи для розділення слів в ідентифікаторі.
- Описуйте все, що виконує метод.
- Уникайте невиразних і неоднозначних дієслів чи фраз.
- Для іменування методу-функції рекомендується використовувати опис поверненого значення. Наприклад: `currentColor()`
- Метод `main` є точкою входу консольної програми Java. Під час запуску програми метод `main` є першим викликаним методом.
- У програмі Java можлива лише одна точка входу. Якщо є більше одного класу, який має метод `main`, то необхідно скомпілювати програму з параметром компілятора `/main`, щоби вказати, який метод `main` потрібно використовувати як точку входу.
- Метод `main` оголошений усередині класу. Метод `main` має бути статичним і не мати атрибути `public`.
- Main має мати повертальний тип `void`. Метод `main` має бути оголошений із параметром `String[] args`, який містить аргументи командного рядка. Під час використання IntelliJ IDEA для створення програм можна додати параметр вручну або використати клас налаштування IDE для отримання аргументів командного рядка. Параметри зчитуються як аргументи нульового індексу командного рядка.
- Якщо значення, яке повертається методом `main`, не використовується, то вказівка як повернений тип `void` дещо спрощує код. Однак повернення цілого значення дає змогу програмі передавати інформацію про свій стан іншим програмам і скриптам, які викликають файл, що виконується.
- Нульове повернене значення з методу `main` вказує на успішне виконання програми.

## Закріплення матеріалу

- Що таке метод?
- Чим відрізняються функції та процедури?
- Що таке перевантаження методу?
- Що таке рекурсія та які різновиди рекурсії ви знаєте?
- Які правила найменування можна застосовувати до методів?
- Чи можна виконати перевантаження методу `main`?

## Додаткове завдання

Використовуючи IntelliJ IDEA, створіть клас **Calculator**.

Створіть метод з ім'ям `calculate`, який приймає як параметри три ціличислові значення та повертає значення кожного аргументу, поділеного на 5.

## Самостійна діяльність учня

### Завдання 1

Вивчіть основні конструкції та поняття, розглянуті на уроці.

### Завдання 2

Використовуючи IntelliJ IDEA, створіть клас **Bank**.

Уявіть, що ви реалізуєте програму для банку, яка допомагає визначити, погасив клієнт кредит чи ні. Припустимо, щомісячна сума платежу має становити 100 грн. Клієнт має виконати 7 платежів, але може платити рідше великими сумами. Тобто може двома платежами по 300 і 400 грн закрити весь борг.

Створіть метод, який як аргумент прийматиме суму платежу, введену економістом банку. Метод виводить на екран інформацію про стан кредиту (сума заборгованості, сума переплат, повідомлення про відсутність боргу).

### Завдання 3

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 рекурсією. Поясніть, чому не рекомендується використовувати рекурсію для розрахунку факторіала. Вкажіть слабкі місця цього підходу.

#### Завдання 4

Створіть три перевантажені методи для пошуку середнього арифметичного числа з 3, 4, 5 змінних, створіть метод для пошуку мінімального, максимального та середнього значень серед цих трьох методів.

#### Рекомендовані ресурси

Метод main з прикладу програми «Hello World!»

<https://docs.oracle.com/javase/tutorial/getStarted/application/>

Визначення, іменування та навантаження методів у JAVA

<https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html>