Обробка винятків

|  |  |  |  |
| --- | --- | --- | --- |
| № уроку: | 8 | Курс: | Java Essential |
|  | | |  |
| Засоби навчання: | | | Комп'ютер із встановленою IntelliJ IDEA |

# Огляд, мета та призначення уроку

* Розгляд обробки винятків.

**Вивчивши матеріал цього заняття, учень зможе:**

* Розуміти та використовувати техніку обробки виняткових ситуацій.
* Розуміти роботу класу Exception та конструкцій try – catch – finally.

# Зміст уроку

1. Обробка винятків.
2. Ключове слово try.
3. Ключове слово catch.
4. Ключове слово finally.
5. Ключове слово throw.
6. Клас Exception.

# Резюме

* **Обробка виняткових ситуацій (exception handling)** – механізм, який призначений для опису реакції програми на помилки часу виконання та інші можливі проблеми (винятки), які можуть виникнути під час виконання програми та призводять до неможливості (безглуздості) подальшого відпрацювання програмою її базового алгоритму.
* **У Java винятки представлені у вигляді класів**. Усі класи винятків мають бути похідними від вбудованого у Java класу Exception. Отже, усі винятки Java є підкласами класу Exception.
* **Головна перевага обробки виняткових ситуацій** полягає в тому, що вона дає змогу автоматизувати отримання більшої частини коду, який доводилося раніше вводити в будь-яку велику програму вручну для обробки помилок. Якщо програма написана мовою програмування без обробки виняткових ситуацій, то під час невдалого виконання методів доводиться повертати коди помилок, які необхідно перевіряти вручну за кожного виклику методу. Це не тільки трудомісткий, а й процес, який загрожує помилками.
* Основу обробки виняткових ситуацій у Java складає пара ключових слів try та catch. Ці ключові слова діють спільно і не можуть бути використані окремо.
* Коли виняток генерується оператором try, він перехоплюється складовим йому пару оператором catch, який потім обробляє цей виняток. Залежно від типу винятку виконується відповідний оператор catch. Якщо типи винятку, який генерується, і того, що вказується в операторі catch, збігаються, то виконується саме цей оператор, а всі інші пропускаються.
* Якщо виняток не генерується, блок оператора try завершується як завжди, і всі його оператори catch пропускаються. Виконання програми відновлюється з першого оператора, наступного після оператора catch. У такий спосіб оператор catch виконується лише у тому випадку, якщо генерується виняток.
* Оператори catch виконуються за порядком їхнього прямування в програмі. Але водночас виконується тільки один блок catch, у якому тип винятку збігається з типом винятку, що генерується. А решта блоків catch пропускаються.
* Блок finally буде виконуватися незалежно від того, які винятки були перехоплені.

# Закріплення матеріалу

* Що таке виняток?
* Що таке конструкція try – catch?
* Що таке конструкція try – catch – finally?
* Як створити користувацький виняток?
* У яких випадках блок finally не спрацьовує?

# Додаткове завдання

Створіть проєкт за допомогою IntelliJ IDEA.

Створіть клас Calculator.

У тілі класу створіть чотири методи для арифметичних дій: (add – додавання, sub – віднімання, mul – множення, div – ділення).

Метод ділення має перевірити ділення на нуль, якщо перевірка не проходить, згенерувати виняток.

Користувач вводить значення, над якими хоче зробити операцію та вибрати саму операцію. У разі виникнення помилок мають викидатися винятки.

# Самостійна діяльність учня

**Завдання 1**

Вивчіть основні конструкції та поняття, розглянуті на уроці.

**Завдання 2**

Створіть проєкт за допомогою IntelliJ IDEA.

Потрібно:

Описати клас з ім'ям **Worker**, що містить такі поля:

* прізвище та ініціали працівника;
* назва займаної посади;
* рік початку роботи.

Написати програму, яка виконує такі дії:

* введення з клавіатури даних до масиву, що складається з п'яти елементів типу Worker (записи мають бути впорядковані за алфавітом);
* якщо значення року введено не у відповідному форматі, видається виняток.
* виведення на екран прізвища працівника, стаж роботи якого перевищує введене значення.

**Завдання 3**

Створіть проєкт за допомогою IntelliJ IDEA.

Потрібно:

Описати клас з іменем **Price**, що містить такі поля:

* назва товару;
* назва магазину, де продається товар;
* вартість товару у гривнях.

Написати програму, яка виконує такі дії:

* введення з клавіатури даних до масиву, що складається з двох елементів типу **Price** (записи мають бути впорядковані в алфавітному порядку за назвами магазинів);
* виведення на екран інформації про товари, що продаються в магазині, назва якого введена з клавіатури (якщо такого магазину немає, вивести виняток).

**Завдання 4**

Зайдіть на сайт Oracle.

Використовуючи пошукові механізми Oracle, знайдіть самостійно опис теми за кожним прикладом, який було розглянуто на уроці, так, як це наведено нижче, у розділі «Рекомендовані ресурси», описи даного уроку. Збережіть посилання та дайте їм короткий опис.

# Рекомендовані ресурси

Обробка винятків

<https://docs.oracle.com/javase/tutorial/essential/exceptions/>

Клас Exception.

<http://docs.oracle.com/javase/7/docs/api/java/lang/Exception.html>

try – catch

<https://docs.oracle.com/javase/tutorial/essential/exceptions/handling.html>