

Обработка исключений

№ урока: 8 Курс: Java Essential

Средства обучения: Компьютер с установленной IntelliJ IDEA.

Обзор, цель и назначение урока

Рассмотрение обработки исключений.

Изучив материал данного занятия, учащийся сможет:

- Понимать и использовать техники обработки исключительных ситуаций.
- Понимать работу класса `Exception` и конструкций `try - catch - finally`.

Содержание урока

1. Обработка исключений.
2. Ключевое слово `try`.
3. Ключевое слово `catch`.
4. Ключевое слово `finally`.
5. Ключевое слово `throw`.
6. Класс `Exception`.

Резюме

- Обработка исключительных ситуаций (exception handling) – механизм, предназначенный для описания реакции программы на ошибки времени выполнения и другие возможные проблемы (исключения), которые могут возникнуть при выполнении программы и приводят к невозможности (бессмысленности) дальнейшей отработки программой её базового алгоритма.
- В Java исключения представлены в виде классов. Все классы исключений должны быть производными от встроенного в Java класса `Exception`. Следовательно, все исключения в Java являются подклассами класса `Exception`.
- Главное преимущество обработки исключительных ситуаций заключается в том, что она позволяет автоматизировать получение большей части кода, который раньше приходилось вводить в любую крупную программу вручную для обработки ошибок. Так, если программа написана на языке программирования без обработки исключительных ситуаций, то при неудачном выполнении методов приходится возвращать коды ошибок, которые необходимо проверять вручную при каждом вызове метода. Это не только трудоемкий, но и чреватый ошибками процесс.
- Основу обработки исключительных ситуаций в Java составляет пара ключевых слов `try` и `catch`. Эти ключевые слова действуют совместно и не могут быть использованы порознь.
- Когда исключение генерируется оператором `try`, оно перехватывается составляющим ему пару оператором `catch`, который затем обрабатывает это исключение. В зависимости от типа исключения выполняется и соответствующий оператор `catch`. Так, если типы генерируемого исключения и того, что указывается в операторе `catch`, совпадают, то выполняется именно этот оператор, а все остальные пропускаются.
- Если исключение не генерируется, то блок оператора `try` завершается как обычно, и все его операторы `catch` пропускаются. Выполнение программы возобновляется с первого оператора, следующего после завершающего оператора `catch`. Таким образом, оператор `catch` выполняется лишь в том случае, если генерируется исключение.
- Операторы `catch` выполняются по порядку их следования в программе. Но при этом выполняется только один блок `catch`, в котором тип исключения совпадает с типом генерируемого исключения. А все остальные блоки `catch` пропускаются.
- Блок `finally` будет выполняться не зависимо от того, какие исключения были перехвачены.

Закрепление материала

- Что такое исключение?
- Что такое конструкция **try - catch**?
- Что такое конструкция **try - catch - finally**?
- Как создать пользовательское исключение?
- В каких случаях не срабатывает блок **finally**?

Дополнительное задание

Задание

Создайте проект, используя IntelliJ IDEA.

Создайте класс Calculator.

В теле класса создайте четыре метода для арифметических действий: (add – сложение, sub – вычитание, mul – умножение, div – деление).

Метод деления должен делать проверку деления на ноль, если проверка не проходит, сгенерировать исключение.

Пользователь вводит значения, над которыми хочет произвести операцию и выбрать саму операцию. При возникновении ошибок должны выбрасываться исключения.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2

Создайте проект, используя IntelliJ IDEA.

Требуется:

Описать класс с именем **Worker**, содержащую следующие поля:

- фамилия и инициалы работника;
- название занимаемой должности;
- год поступления на работу.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из пяти элементов типа **Worker** (записи должны быть упорядочены по алфавиту);
- если значение года введено не в соответствующем формате выдает исключение.
- вывод на экран фамилии работника, стаж работы которого превышает введенное значение.

Задание 3

Создайте проект, используя IntelliJ IDEA.

Требуется: Описать класс с именем **Price**, содержащую следующие поля:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в гривнах.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из двух элементов типа **Price** (записи должны быть упорядочены в алфавитном порядке по названиям магазинов);

- вывод на экран информации о товарах, продающихся в магазине, название которого введено с клавиатуры (если такого магазина нет, вывести исключение).

Задание 4

Зайдите на сайт Oracle.

Используя поисковые механизмы Oracle, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

Рекомендуемые ресурсы

Обработка исключений

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

Класс 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>