Конспект лекції № 17-18

Тема № 9. ОБРОБКА ВИКЛЮЧЕНЬ

Міжпредметні зв'язки: Зв'язок із елементами знань і умінь таких навчальних дисциплін як „Алгоритмізація та програмування", „Інформатика", „Вступ до фаху", „Технології Інтернет", „Web-дизайн та Web-програмування", „Технології проектування та адміністрування БД і СД ".

Мета лекції: розглянути особливості об'єктно-орієнтованого підходу до програмування; отримати базові знання щодо обробки виключень мовою Java; формувати вміння аналізувати, узагальнювати, порівнювати, абстрагуватися, синтезувати знання, отримані при вивченні даної теми.

План лекції:

1. Виключення (exceptions). Виключна ситуація.
2. Ієрархія виключень.
3. Ключові слова try, catch, finally.
4. Оператори throw, throws.
5. Класи Java для обробки виключних ситуацій з пакету java.lang. Основні типи виключень в Java.
6. Методи класу Throwable.
7. Клас Exception.
8. Створення власних класів виключень.

Опорні поняття: виключення, ієрархія виключень, помилка, перехоплення, обробник виключення, генерування виключення, власний клас виключення.

Інформаційні джерела:

Основна та допоміжна література:

1. Алгоритми і структура даних: Навчальний посібник / В.М.Ткачук. - Івано-Франківськ : Видавництво Прикарпатського національного університету імені Василя Стефаника, 2016. − 286 с.
2. Блинов И.Н., Романчик В. С. Java. Методы программирования : уч.-мет. пособие / И. Н. Блинов, В. С. Романчик. − Минск : издательство "Четыре четверти", 2013. − 896 с.
3. Васильев А. Н. Java. Объектно-ориентированное программирование: Учебное пособие. − СПб.: Питер, 2011. − 400 с.
4. Грязнова В. О., Єфіменко С. В. Основи методології програмування. - К.: ВПЦ "Київський університет", 2010.
5. Кингсли-Хьюджес Э., Кингсли-Хьюджес К. Справочник программиста. - М.: ООО "ИД Вильямс", 2007.
6. Хорстманн, К. С., Корнелл, Г. Библиотека профессионала. Java 2 : Том 1. Основы. − 8-е изд. − М. : Вильямс, 2013. − 816 с.
7. Шилдт Г. Полный справочник по Java SE 6. М.: Вильямс, 2010. − 1040 с.

8. Эккель Б. Философия Java. − 4-е изд. − СПб. : Питер, 2011. − 640 с.

Інтернет ресурси:

1. Java - Учебник для начинающих программистов : [Електрон. ресурс]. - Режим доступу: http://proglang.su/java
2. Программирование на Java: [Електрон. ресурс]. - Режим доступу: https://www.intuit.ru/studies/courses/16/16/info
3. Топ-5 полезных видеокурсов по Java : [Електрон. ресурс]. - Режим доступу: https://javarush.ru/groups/posts/528-top-5-poleznihkh-videokursov-po-java-chastjh-1--
4. Уроки по основам языка программирования JAVA для начинающих : [Електрон. ресурс]. - Режим доступу: https://www.fandroid.info/tutorial-po-osnovam-yazyka-programmirovaniya-java-dlya-nachinayushhih/
5. Язык программирования Java и среда NetBeans : [Електрон. ресурс]. - Режим доступу: https://www.intuit.ru/studies/courses/569/425/info
6. Классы исключений : [Електрон. ресурс]. - Режим доступу: https://www.examclouds.com/ru/java/java-core-russian/exception-hierarchy
7. Java : [Електрон. ресурс]. - Режим доступу: https://www.bestprog.net/uk/sitemap_ua/4547-2/

Навчальне обладнання, ТЗН, презентація тощо: ноутбук, проектор, мультимедійна презентація.

ВИКЛАД МАТЕРІАЛУ ЛЕКЦІЇ

Питання 1. Виключення (exceptions). Виключна ситуація.

Виключна ситуація – це програмна помилка, яка виникає під час виконання послідовності програмного коду. Програмна помилка може бути логічною помилкою програміста під час розробки програми.

Наприклад, виключна ситуація може виникати у випадку:

- спроби ділення на нуль
- спроби звернення до елементу масиву, номер індексу якого виходить за межі оголошеного;
- спроби взяти корінь з від'ємного числа;
- спроби відкрити файл за іменем, якого немає на диску;
- інші випадки.

Виділяють три типи виключень.

Контрольовані виключення - представляють собою вид виключення, яке відбувається на стадії компіляції, їх також називають виключеннями періоду компіляції. Окреслені виключення не слід ігнорувати в ході компіляції, вони вимагають належного управління (дозволу) з боку програміста.

Неконтрольовані виключення - є виключеннями, які відбуваються під час виконання. Вони також носять назву виключення на етапі виконання. Дана категорія може включати помилки програмування, такі як логічні помилки або

невірний спосіб використання API. Виключення на етапі виконання ігноруються в ході компіляції.

Помилки - не є виключеннями, проте представляють проблеми, які виникають незалежно від користувача або програміста. Помилки в коді зазвичай ігноруються тому, що в рідкісних випадках їх обробка виявиться результативною. Наприклад, помилка виникне при переповненні стеку. На етапі компіляції вони також ігноруються.

Правильна обробка виключень є важливим елементом написання професійних програм на Java.

Як правило, ***кожна виключна ситуація має мати свій код помилки*** та обробку цього коду (виведення відповідних повідомлень, тощо). У мові програмування Java розроблено механізм обробки виключних ситуацій.

У мові програмування Java, виключення – це спеціальний об'єкт, який описує виключну ситуацію, що виникла в деякій частині програмного коду. Об'єкт, що представляє виключення, генерується в момент виникнення виключної ситуації. Після виникнення критичної (виключної) ситуації виключення перехоплюється та обробляється. Таким чином, виникає поняття генерування виключення. Генерування виключення – процес створення об'єкту, що описує дане виключення.

У мові Java виключення можуть генеруватись одним з двох способів:

- ***автоматично***. У цьому випадку виключення генеруються виконавчою системою Java. До таких виключень входять помилки, що порушують правила мови Java чи обмеження, що накладаються системою;
- ***вручну***. У цьому випадку виключення генеруються в програмному коді, який розробляється. Ручні виключення програмуються для повідомлення викликаючому коду про можливі помилки в методах розроблюваної програми.

Стандартний обробник виключень Java викликається, коли програма:

- не використовує блок try…catch для обробки та перехоплення виключної ситуації взагалі;
- містить блок try…catch, однак у цьому блоці даний тип виключення не перехоплюється.

Якщо програма містить власний код try…catch для обробки даної виключної ситуації, тоді стандартний обробник виключень не викликається.

Для обробки виключень в Java використовуються наступні ключові слова:

- try;
- catch;
- throw;
- throws;
- finally.

Висновок. Виключення - якась виняткова, незапланована ситуація, яка сталася під час роботи програми.

Продовжити перегляд

Загальний висновок за темою лекції

1. Виключення - це проблема (помилка), яка виникає під час виконання програми. Виключення можуть виникати у багатьох випадках, наприклад: Користувач ввів некоректні дані; Файл, до якого звертається програма, не знайдений; Мережеве з'єднання з сервером було втрачено під час передачі даних. Всі виключення в Java є об'єктами. Тому вони можуть породжуватися не тільки автоматично при виникненні виняткової ситуації, але і створюватися самим розробником.
2. Виключення діляться на кілька класів, але всі вони мають спільного предка - клас Throwable. Його нащадками є підкласи Exception і Error. Виключення (Exceptions) є результатом проблем в програмі, які в принципі можуть бути вирішені і передбачувані. Наприклад, відбулося ділення на нуль в цілих числах. Помилки (Errors) представляють собою більш серйозні проблеми, які, відповідно до специфікації Java, не слід намагатися обробляти у власній програмі, оскільки вони пов'язані з проблемами рівня JVM. Наприклад, виключення такого роду виникають, якщо закінчилася пам'ять, доступна віртуальній машині. Програма додаткову пам'ять все одно не зможе забезпечити для JVM.
3. У Java всі виключення діляться на два типи: контрольовані виключення (checked) і неконтрольовані виключення (unchecked), до яких відносяться помилки (Errors) і виключення часу виконання (RuntimeExceptions, нащадок класу Exception). Контрольовані виключення є помилки, які можна і потрібно обробляти в програмі, до цього типу належать усі нащадки класу Exception (але не RuntimeException).
4. У Java є п'ять ключових слів для роботи з виключеннями: try - це ключове слово використовується для позначення початку блоку коду, який потенційно може привести до помилки; catch - ключове слово для позначення початку блоку коду, призначеного для перехоплення і обробки виключень; finally - ключове слово для позначення початку блоку коду, який є додатковим і в який поміщається після останнього блоку 'catch' (управління зазвичай передається в блок 'finally' в будь-якому випадку); throw - служить для створення виключень; throws - ключове слово, яке прописується в сигнатурі методу, і що означає що метод потенційно може викинути виключення із зазначеним типом.

Питання і завдання студентам для контролю знань

1. Що таке парадигма "виключення"?
2. Яка ієрархія виключень?
3. Можна / чи потрібно обробляти помилки jvm?
4. Які існують способи обробки виключень?
5. Про що говорить ключове слово throws?
6. У чому особливість блоку finally? Чи завжди він виконується?
7. Чи може не бути жодного блоку catch при відлові виключень?

8. При якій ситуації блок finally не буде виконано?
9. Чи може один блок catch відловлювати кілька виключень (з одного і різних гілок наслідувань)?
10. Що таке оброблювані і необроблювані (checked / unchecked) виключення?
11. У чому особливість RuntimeException?
12. Як написати власне ("призначене для користувача") виключення?
13. Який оператор дозволяє примусово викинути виключення?
14. Чи є додаткові умови до методу, який потенційно може викинути виключення?
15. Якщо оператор return міститься і в блоці catch, і в finally, який з них "головніший"?
16. Яке призначення OutOfMemoryError?
17. Яке призначення SQLException? До якого типу checked або unchecked воно відноситься, чому?
18. Що таке Error? В якому випадку використовується Error?
19. Яка конструкція використовується в Java для обробки виключень?
20. Припустимо, є блок try-finally. У блоці try виникло виключення і виконання перемістилося в блок finally. У блоці finally теж виникло виключення. Яке з двох виключень "випаде" з блоку try-finally? Що трапиться з другим виключенням?
21. Припустимо, є метод, який може викинути IOException і FileNotFoundException в якій послідовності повинні йти блоки catch? Скільки блоків catch буде виконано?
22. Якими способами можуть генеруватись виключення?
23. В яких випадках викликається стандартний обробник виключень Java?
24. Які ключові слова використовуються для обробки виключень в Java?
25. Яке призначення конструкції try… catch…finally?
26. Яке призначення ключового слова throw?
27. Які типи заборонено використовувати для генерування виключень оператором throw?
28. Яким чином створити власний клас виключення?

Укладач: _____ Шевчук І.Б., доцент, д.е.н., доцент
(підпис) (ПІБ, посада, науковий ступінь, вчене звання)