4. Семинар: обработка исключений

4.1. Инструментарий

— Презентация для преподавателя, ведущего семинар;
— Фон GeekBrains для проведения семинара в Zoom;
— JDK любая 11 версии и выше;
— IntelliJ IDEA Community Edition для практики и примеров используется IDEA.

4.2. Цели семинара

— Закрепить полученные на лекции знания об исключениях;
— Попрактиковаться в написании собственных классов исключений;
— Создание, выбрасывание и обработка исключений;
— Обработка исключений в стиле «до захвата ресурса»;
— Проброс исключений выше по стеку.

4.3. План-содержание

Что происходит	Время	Слайды	Описание
Организационный момент	5	1-5	Преподаватель ожидает студентов, поддерживает активность и коммуникацию в чате, озвучиает цели и планы на семинар. Важно упомянуть, что выполнение домашних заданий с лекции является, фактически, подготовкой к семинару
Quiz	10	6-18	Преподаватель задаёт вопросы викторины, через 30 секунд демонстрирует слайд-подсказку и ожидает ответов (6 вопросов, по минуте на ответ)
Рассмотрение ДЗ лекции	10	19-23	Преподаватель демонстрирует свой вариант решения домашнего задания с лекции, возможно, по предварительному опросу, демонстрирует и разбирает вариант решения одного из студентов
Вопросы и ответы	10	24	Преподаватель ожидает вопросов по теме прошедшей лекции, викторины и продемонстрированной работы
Задание 1	30	25-28	Сквозное задание, состоящее из объяснений в 6 пунктах, обязательных к исполнению. Всё задание выдаётся сразу целиком и является неделимым.

Что происходит	Время	Слайды	Описание
Перерыв (если нужен)	5	33	Преподаватель предлагает студентам перерыв на 5 минут (студенты голосуют)
Задание 2	30	29-32	Сквозное задание, состоящее из объяснений в 6 пунктах, обязательных к исполнению. Всё задание выдаётся сразу целиком и является неделимым.
Домашнее задание	5	45-46	Объясните домашнее задание, подведите итоги урока
Рефлексия	10	47-48	Преподаватель запрашивает обратную связь
Длительность	115		

4.4. Подробности

4.4.1. Организационный момент

- **Цель этапа:** Позитивно начать урок, создать комфортную среду для обучения.
- **Тайминг:** 3-5 минут.
- **Действия преподавателя:**
  - Запрашивает активность от аудитории в чате;
  - Презентует цели курса и семинара;
  - Презентует краткий план семинара и что студент научится делать.

4.4.2. Quiz

- **Цель этапа:** Вовлечение аудитории в обратную связь.
- **Тайминг:** 5-7 минут (4 вопроса, по минуте на ответ).
- **Действия преподавателя:**
  - Преподаватель задаёт вопросы викторины, представленные на слайдах презентации;
  - через 30 секунд демонстрирует слайд-подсказку и ожидает ответов.
- **Вопросы и ответы:**
  1. (a)
     (b)
     (c)
  2. (a)
     (b)
     (c)

4.4.3. Рассмотрение ДЗ

- **Цель этапа:** Пояснить не очевидные моменты в формулировке ДЗ с лекции, синхронизировать прочитанный на лекции материал к началу семинара.
- **Тайминг:** 15-20 минут.
- **Действия преподавателя:**

- Преподаватель демонстрирует свой вариант решения домашнего задания из лекции;
- возможно, по предварительному опросу, демонстрирует и разбирает вариант решения одного из студентов.

— **Домашнее задание из лекции:**
- Напишите два наследника класса `Exception`: ошибка преобразования строки и ошибка преобразования столбца.
  Вариант решения

Листинг 1: Конструктор с вариативностью

```
private static final class ColumnMismatchException extends RuntimeException {
   ColumnMismatchException(String message) {
      super("Columns exception: " + message);
   }
}

private static final class NumberIsNotNumberException extends RuntimeException {
   NumberIsNotNumberException(String message) {
      super("Not a number found: " + message);
   }
}
```

- Разработайте исключения-наследники так, чтобы они информировали пользователя в формате ожидание/реальность.

Листинг 2: Конструктор с вариативностью

```
private static final class RowMismatchException extends RuntimeException {
   RowMismatchException(int expected, int current, String value) {
      super(String.format("Rows exception: expected %d rows. Received %d rows in '%s' string",
            expected, current, value));
   }
}
```

- для проверки напишите программу, преобразующую квадратный массив целых чисел 5x5 в сумму чисел в этом массиве, при этом, программа должна выбросить исключение, если строк или столбцов в исходном массиве окажется не 5.
  Вариант решения представлен в приложении А

4.4.4. Вопросы и ответы

— **Ценность этапа** Задать задание для самостоятельного выполнения между занятиями.
— **Тайминг** 5-15 минут
— **Действия преподавателя**
- Преподаватель ожидает вопросов по теме прошедшей лекции, викторины и продемонстрированной работы;
- Если преподаватель затрудняется с ответом, необходимо мягко предложить студенту ответить на его вопрос на следующем семинаре (и не забыть найти ответ на вопрос студента!);
- Предложить и показать пути самостоятельного поиска студентом ответа на заданный вопрос;

- Посоветовать литературу на тему заданного вопроса;
- Дополнительно указать на то, что все сведения для выполнения домашнего задания, прохождения викторины и работы на семинаре были рассмотрены в методическом материале к этому или предыдущим урокам.

4.4.5. Задание 1

— **Ценность этапа** Написание почти полноценной механики по краткому ТЗ.
— **Тайминг** 25-30 минут.
— **Действия преподавателя**
  - Выдать задание студентам;
  - Подробно объяснить, что именно требуется от студентов, избегая упоминания конкретных языковых конструкций.
— **Задание**: Класс «Проверка логина и пароля».
  1. Создать статический метод который принимает на вход три параметра: `login`, `password` и `confirmPassword`.
  2. Длина `login` должна быть **меньше** 20 символов. Если `login` не соответствует этому требованию, необходимо выбросить `WrongLoginException`.
  3. Длина `password` должна быть **не меньше** 20 символов. Также `password` и `confirmPassw` должны быть равны. Если `password` не соответствует этим требованиям, необходимо выбросить `WrongPasswordException`.
  4. `WrongPasswordException` и `WrongLoginException` – пользовательские классы исключения с двумя конструкторами –- один по умолчанию, второй принимает параметры исключения (неверные данные) и возвращает пользователю в виде «ожидалось/фактически».
  5. В основном классе программы необходимо по-разному обработать исключения.
  6. Метод возвращает `true`, если значения верны или `false` в противном случае.

  Вариант исполнения класса в приложении ??

Листинг 3: Создание объекта класса

4.4.6. Задание 2

— **Ценность этапа** Написание наброска пет-проекта, повторение информации об ООП, работа с исключениями.
— **Тайминг** 25-30 минут.
— **Действия преподавателя**
  - Выдать задание студентам;
  - Подробно объяснить, что именно требуется от студентов, избегая упоминания конкретных языковых конструкций.
  - обратить внимание на порядок обработки исключений, повторная попытка купить товар может производиться только тогда, когда остальные параметры покупки уже проверены.
— **Задание**: Класс «Эмуляция интернет-магазина».

1. Написать классы покупатель (ФИО, возраст, телефон), товар (название, цена) и заказ (объект покупатель, объект товар, целочисленное количество).
2. Создать массив покупателей (инициализировать 2 элемента), массив товаров (инициализировать 5 элементов) и массив заказов (пустой на 5 элементов).
3. Создать статический метод «совершить покупку» со строковыми параметрами, соответствующими полям объекта заказа. Метод должен вернуть объект заказа.
4. Если в метод передан несуществующий покупатель – метод должен выбросить исключение `CustomerException`, если передан несуществующий товар, метод должен выбросить исключение `ProductException`, если было передано отрицательное или слишком больше значение количества (например, 100), метод должен выбросить исключение `AmountException`.
5. Вызвать метод совершения покупки несколько раз таким образом, чтобы заполнить массив покупок возвращаемыми значениями. Обработать исключения следующим образом (в заданном порядке):
   - если был передан неверный товар – вывести в консоль сообщение об ошибке, не совершать данную покупку;
   - если было передано неверное количество – купить товар в количестве 1;
   - если был передан неверный пользователь – завершить работу приложения с исключением.
6. Вывести в консоль итоговое количество совершённых покупок после выполнения основного кода приложения.

Вариант исполнения класса в приложении ??

Листинг 4: Создание объекта класса

4.4.7. Домашнее задание

— **Ценность этапа** Задать задание для самостоятельного выполнения между занятиями.
— **Тайминг** 5-10 минут.
— **Действия преподавателя**
  - Пояснить студентам в каком виде выполнять и сдавать задания
  - Уточнить кто будет проверять работы (преподаватель или ревьювер)
  - Объяснить к кому обращаться за помощью и где искать подсказки
  - Объяснить где взять проект заготовки для дз
— **Задания**

5-25 мин Решить все задания (в том числе «со звёздочкой»), если они не были решены на семинаре, без ограничений по времени;

Все варианты решения приведены в тексте семинара выше

15 мин 1. В класс покупателя добавить перечисление с гендерами, добавить в сотрудника свойство «пол» со значением созданного перечисления. Добавить геттеры, сеттеры.

20-25 мин 2. Добавить в основную программу перечисление с праздниками (Новый Год, 8 марта, 23 февраля), написать метод, принимающий массив сотрудников, поздравляющий всех сотрудников с Новым Годом, женщин с 8 марта, а мужчин с 23 февраля.

4.4.8. Рефлексия и завершение семинара

- **Цель этапа:** Привести урок к логическому завершению, посмотреть что студентам удалось, что было сложно и над чем нужно еще поработать
- **Тайминг:** 5-10 минут
- **Действия преподавателя:**
  - Запросить обратную связь от студентов.
  - Подчеркните то, чему студенты научились на занятии.
  - Дайте рекомендации по решению заданий, если в этом есть необходимость
  - Дайте краткую обратную связь студентам.
  - Поделитесь ощущением от семинара.
  - Поблагодарите за проделанную работу.

Приложения

A. Домашнее задание 3

Листинг 5: Основная программа

```
package ru.gb.jcore;

import java.util.Arrays;

public class Exceptional {
   private static final class ColumnMismatchException extends RuntimeException {
      ColumnMismatchException(String message) {
         super("Columns exception: " + message);
      }
   }

   private static final class NumberIsNotNumberException extends RuntimeException {
      NumberIsNotNumberException(String message) {
         super("Not a number found: " + message);
      }
   }

   private static final class RowMismatchException extends RuntimeException {
      RowMismatchException(int expected, int current, String value) {
         super(String.format("Rows exception: expected %d rows. Received %d rows in '%s' string",
               expected, current, value));
      }
   }

   private static final String CORRECT_STRING= "1 3 1 2\n2 3 2 2\n5 6 7 1\n3 3 1 0";
   private static final String EXTRA_ROW_STRING= "1 3 1 2\n2 3 2 2\n5 6 7 1\n3 3 1 0\n1 2 3 4";
   private static final String EXTRA_COL_STRING= "1 3 1 2 1\n2 3 2 2 1\n5 6 7 1 1\n3 3 1 0 1";
   private static final String NO_ROW_STRING= "1 3 1 2\n2 3 2 2\n5 6 7 1";
   private static final String NO_COL_STRING= "1 3 1 2\n2 3 2 2\n5 6 7 1\n3 3 1";
   private static final String HAS_CHAR_STRING= "1 3 1 2\n2 3 2 2\n5 6 7 1\n3 3 1 A";

   private static final int MATRIX_ROWS= 4;
   private static final int MATRIX_COLS= 4;

   private static String[][] stringToMatrix(String value) {
      String[] rows = value.split("\n");
      if (rows.length !=MATRIX_ROWS)
         throw new RowMismatchException(MATRIX_ROWS, rows.length, value);

      String[][] result = new String[MATRIX_ROWS][];
      for (int i = 0; i < result.length; i++) {
         result[i] = rows[i].split(" ");
         if (result[i].length !=MATRIX_COLS)
            throw new ColumnMismatchException(result[i].length + ":\n" + value);
      }
      return result;
   }

   private static float calcMatrix(String[][] matrix) {
      float result = 0;
      int len = 0;
      for (int i = 0; i < matrix.length; i++) {
         for (int j = 0; j < matrix[i].length; j++) {
            try {
               result += Integer.parseInt(matrix[i][j]);
               ++len;
            } catch (NumberFormatException e) {
               throw new NumberIsNotNumberException(matrix[i][j]);
```