

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №3

з дисципліни «Основи Програмування»

Тема: Exceptions.

Виконали:
Студенти групи ІА-23
Мозоль Владислав,
Курач Владислав,
Лядський Дмитро
Дата здачі: 09.03.2023

Перевірів:
Колеснік В.М

Київ 2023

Хід роботи:

1. Ознайомитись з javadoc для наступних класів:

- Throwable;
- Error;
- Exception;
- RuntimeException;

а також повторити конструкції try-catch-finally, throw, throws.

2. Проаналізувати предметну область з л/р 10-12. Створити свій власний тип checked exception (підклас класу Exception), що описує порушення одного з бізнес-правил предметної області (наприклад, спробу додати студента у групу, в якій вже навчається максимально допустима кількість студентів, або пуста "" назва фільму). Додати throws-декларацію цього exception до відповідних методів, у яких потрібно «викидати» його у разі порушення бізнес-правил. Додати блок try-catch-finally для демонстрації виклику цього метода. Продемонструвати «викидання» та обробку цього exception. Також додати та продемонструвати код для «викидання» кількох стандартних RuntimeException (IllegalArgumentException, NullPointerException, ...).

3. Відповісти на контрольні питання.

```
1      import java.util.*;
2
3      public class Main {
4          public static void main(String[] args) {
5              part1();
6              part2();
7              part3( a: null);
8              part4();
9          }
10
11      @ 2 usages
12      static Set<Good> listOfGoods() {...}
13
14
15
16
17
18
19
20      1 usage
21      static void part1(){
22          Set<Good> list = listOfGoods();
23          try {
24              System.out.println(list);
25              list.add(new Good("Red Dead Redemption 3", -5));
26              System.out.println(list);
27              list.add(new Good( nameOfGood: "Red Dead Redemption 3", recPrice: 14));
28              System.out.println(list);
29              list.add(new Good( nameOfGood: "Mouse", recPrice: 40));
30              System.out.println(list);
31          }
32          catch (IllegalStateException e){
33              System.out.println("Досягнуто максимальної кількості товарів(4)");
34          }
35          catch (IllegalArgumentException e){
36              System.out.println("Ціна товару не може бути від'ємною");
37          }
38      }
39  }
```

```
29         System.out.println(list);
30     }
31     catch (IllegalStateException e){
32         System.out.println("Досягнуто максимальної кількості товарів(4)");
33     }
34     catch (IllegalArgumentException e){
35         System.out.println("Ціна товару не може бути від'ємною");
36     }
37     finally {
38         System.out.println(list);
39     }
40 }
1 usage
41 static void part2(){
42     Set<Good> listOfGoods = listOfGoods();
43     listOfGoods.add(new Good( nameOfGood: null, recPrice: 60));
44     System.out.println(listOfGoods);
45 }
1 usage
46 static void part3(Integer a){
47     List<String> list = new ArrayList<>();
48     list.add("PlayStation 5");
49     list.add("Iphone 45X");
50     list.add("Witcher 3");
51     list.add(null);
52
53     System.out.println(a * 3);
54
55
56 }
57
```

```

57
58      1 usage
    static void part4(){
59          int[] array = new int[5];
60          System.out.println(array[5]);
61      }
62  }
63
64
65      12 usages
    class Good implements Comparable<Good> {
66          4 usages
        String nameOfGood;
67          1 usage
        int recPrice;
68          6 usages
        public Good(String nameOfGood, int recPrice) {
69            if (recPrice < 0){
70                throw new IllegalArgumentException("Ціна не може бути від'ємною");
71            }
72            this.nameOfGood = nameOfGood;
73            this.recPrice = recPrice;
74        }
75
76
77        @Override
78        public int compareTo(Good o) {
79            return this.nameOfGood.compareTo(o.nameOfGood);
80        }
81
82        @Override

```

```

75
76
77     @Override
78     public int compareTo(Good o) {
79         return this.nameOfGood.compareTo(o.nameOfGood);
80     }
81
82     @Override
83     public String toString() {
84         return nameOfGood;
85     }
86 }
87
88 1 usage
89 class MyTreeSet<Good> extends TreeSet<Good>{
90     public boolean add(Good good){
91         if (this.size() >= 4){
92             throw new IllegalStateException("Не можливо додати більше 4 товарів");
93         }
94         return super.add(good);
95     }
96 }

```

```

I:\for programming\java\bin\java.exe -javaagent:I:\for programming\intellij\idea-2022.2.2\lib\idea_rt.jar=50247:I:\for programming\intellij\idea-2022.2.2\bin\java.exe
[iPhone 45X, PlayStation 5, Witcher 3]
[iPhone 45X, PlayStation 5, Red Dead Redemption 3, Witcher 3]
Досягнуто максимальної кількості товарів(4)
[iPhone 45X, PlayStation 5, Red Dead Redemption 3, Witcher 3]
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "String.compareTo(String)" because "this.nameOfGood" is null
    at Good.compareTo(Main.java:79)
    at Good.compareTo(Main.java:65)
    at java.base/java.util.TreeMap.put(TreeMap.java:814)
    at java.base/java.util.TreeMap.put(TreeMap.java:534)
    at java.base/java.util.TreeSet.add(TreeSet.java:255)
    at MyTreeSet.add(Main.java:92)
    at Main.part2(Main.java:43)
    at Main.main(Main.java:6)

Process finished with exit code 1

```

Висновок: виконуючи цю лабораторну роботу ми ознайомилися з класами Throwable, Error, Exception, RuntimeException, а також повторили конструкції try-catch-finally, throw, throws.