UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

IS2104 - Rapid Application Development 2020

Activity 12

In this tutorial, we are going to learn about Exception Handling in Java.

Instructions

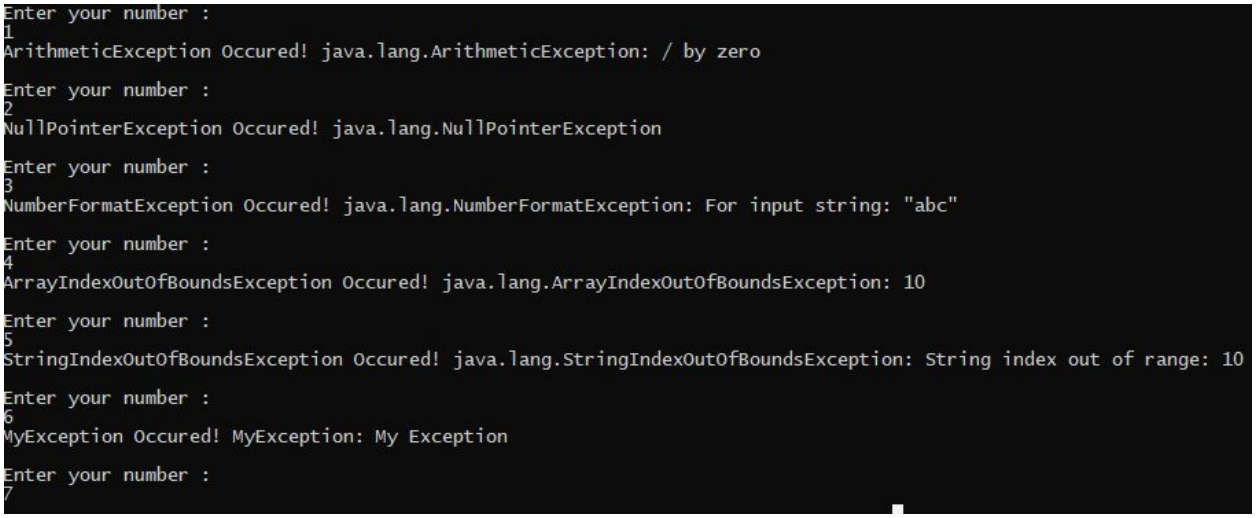
- This assignment contains **4** tasks..
- You should submit
    - "**.java**" and "**.class**" files for all 4 tasks
    - A report with screenshots of contents of "**.java**" file of each question
    - Output of running each ".**class**" file with given test cases.
- Report must be in **PDF** format.
- Report name should be <Index Number>.pdf
    - eg – 18000000.pdf
- Zip **all files** (".java" files, ".class" files and report), name it <Index Number >.zip and upload it to the submission link.
    - eg – 18000000.zip
- Use appropriate text editor (eg – Notepad , Notepad ++)
- You are not allowed to use IDEs
- Any form of plagiarism or collusion is not allowed.

Assignment 12

1. Create a simple program to read a number from the command line and run a code segment. Each code segment should cause a specified exception. You have to handle the exception and print the exception. Then you have to wait for the next user input. (ExceptionHandling.java)

User Input	Action
1	Generate ArithmeticException
2	Generate NullPointerException
3	Generate NumberFormatException
4	Generate ArrayIndexOutOfBoundsException
5	Generate StringIndexOutOfBoundsException
6	Generate CustomException (MyException)
7	End Program

Output:



2. Create a simple calculator. It should take 3 inputs. 1st and 2nd input is two numbers and 3rd input is the operator. (Calc.java)
    a. It should generate an exception if the either of first two inputs is NOT an integer.
    b. It should generate an exception if the 3rd input is NOT a operator (operators ={+,-,*,/})
    c. It should generate a custom exception if the first number is smaller than the second number and the operator is subtraction.
    d. It should be able to handle divide by zero exception.

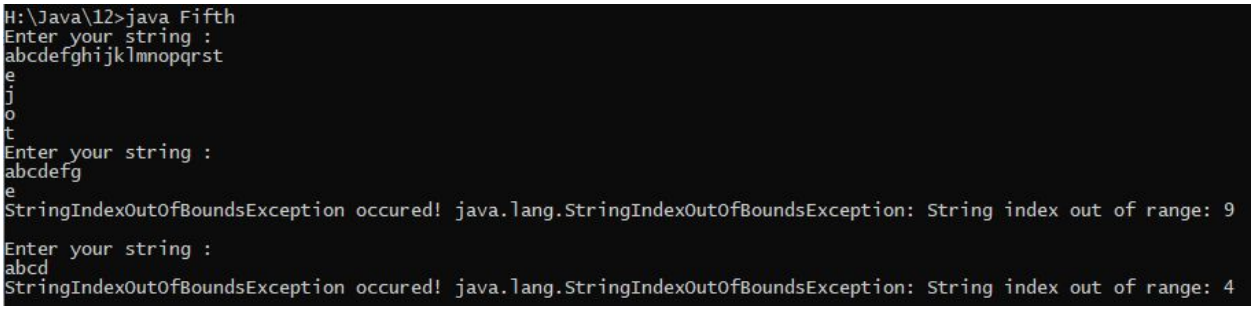
Example Output :

```
H:\Java\12>java Calc
Enter number 1, number 2 and operator :
a 10 *
Number Format Exception Occured! :java.lang.NumberFormatException: For input string: "a"

Enter number 1, number 2 and operator :
10 5 ?
Invalid Operator Exception Occured! :invalid_operator: Invalid Operator!

Enter number 1, number 2 and operator :
5 10 -
Negative Result Exception Occured! :negative_result: Negative Results!

Enter number 1, number 2 and operator :
10 0 /
Arithmetic Exception Occured! :java.lang.ArithmeticException: / by zero

Enter number 1, number 2 and operator :
10 5 +
15
Enter number 1, number 2 and operator :
10 5 -
5
Enter number 1, number 2 and operator :
10 5 *
50
Enter number 1, number 2 and operator :
10 5 /
2
```

- **Test Cases:**
    - b 10 +
    - 10 5 @
    - 10 20 -
    - 20 0 /
    - 20 5 +
    - 20 5 -
    - 20 5 *
    - 20 5 /

3. Create a program to print every 5th letter of the given string. Assume String length is less than or equal to 20. Don't use loops for iterating through String. (Fifth.java)
    a. It should demonstrate handling of StringIndexOutOfBoundsException.

Example output :



- **Test Cases**
    - thequickbrownfoxjump
    - thequick
    - thequickbrown
    - thequickbrownfox

4. Create a simple inventory management program. It should take 3 inputs. 1st input is the action (action ={add,remove}) , 2nd input is the index of the item (0-9) , 3rd input is the quantity. This inventory is stored in an array with 10 elements. (Inventory.java)
    a. It should generate a custom exception if the 1st input is not an action.
    b. It should generate an ArrayIndexOutOfBoundsException if the 2nd input is incorrect.
    c. It should generate an exception if the quantity will be negative, if the action is done. (eg - Assume the index 0 of the array has a value of 5. If we give the command “remove 0 10”, the program will try to subtract 10 from 5, which will result -5. But an item cannot have a negative quantity. Therefore you have to raise a custom exception.)

Example Output :

```
H:\Java\12>java Inventory
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Enter action, index and quantity :
add 0 10
[10, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Enter action, index and quantity :
add 1 20
[10, 20, 0, 0, 0, 0, 0, 0, 0, 0]
Enter action, index and quantity :
wrong 0 10
Invalid Action Exception Occured! :invalid_action: Invalid Action !

[10, 20, 0, 0, 0, 0, 0, 0, 0, 0]
Enter action, index and quantity :
add 20 10
ArrayIndexOutOfBoundsException Occured! :java.lang.ArrayIndexOutOfBoundsException: 20

[10, 20, 0, 0, 0, 0, 0, 0, 0, 0]
Enter action, index and quantity :
remove 0 5
[5, 20, 0, 0, 0, 0, 0, 0, 0, 0]
Enter action, index and quantity :
remove 1 40
Negative Quantity Exception Occured! :negative_quantity: Negative Quantity at 1!

[5, 20, 0, 0, 0, 0, 0, 0, 0, 0]
Enter action, index and quantity :
```

- **Test Cases**
    - add 0 20
    - add 1 40
    - test 1 40
    - add 50 50
    - remove 0 10
    - remove 1 100