

Object-Oriented Programming

Lab session #7



This week, our lab topic is Exception in Java which is an important topic.

When executing Java code, different errors can occur: coding errors made by the programmer, errors due to wrong input, or other unforeseeable things. When an error occurs, Java will normally stop and generate an error message. The technical term for this is: Java will throw an exception (throw an error).

Therefore, by learning Exceptions, we can enable our Java program to handle different types of exceptions to ensure our programs run smoothly without any interruption.

The good thing is that Exceptions are very easy to learn once you see enough examples.

Please take your time to go through these links to learn more about Exception handling in Java.

Reference:

- https://www.w3schools.com/java/java_try_catch.asp
- https://www.tutorialspoint.com/java/java_exceptions.htm
- <https://www.geeksforgeeks.org/types-of-exception-in-java-with-examples/>
- <https://www.javatpoint.com/exception-handling-in-java>
- <https://codelearn.io/sharing/05-dieu-can-biet-ve-java-exception>
- https://docs.google.com/presentation/d/1RaDI8y_0hCF3Et9h2Inlx8POdwb7zUs9/edit?usp=sharing&oid=114898056082189593797&rtpof=true&sd=true (Full Lecture Slides - Optional Reading)

Alright, let's get the Exception party started by doing these questions:

Question 1: Practice a basic Exception handling problem with Hackerrank:

<https://www.hackerrank.com/challenges/java-exception-handling-try-catch/problem>

In this problem, you need to handle two types of exception: **InputMismatchException** and **ArithmeticException**.

InputMismatchException happens when the user input type is different from the expected input type (like the program asks for a number but the user enters a string)

ArithmeticException happens when a math operation got issue like dividing a number by 0.

Question 2: Throw custom Exceptions

<https://www.hackerrank.com/challenges/java-exception-handling/problem?isFullScreen=true>

In this problem, you need to take care two cases when input numbers are zero or negative. Then you need to throw a custom exception with messages like in the question.

Question 3: Array Out of Bound Exception

In the following example we try to access element 9 of a 7-element array without using exception. It is allowed to ignore the exception, but handling it makes the program user friendlier.

```
1 public class MyClass {  
2  
3     public static void main(String args[]){  
4         int myArray[] = new int[7];  
5         // trying to access element 9  
6         System.out.println(myArray[9]);  
7     }  
8 }
```

If you run the code above, you will receive an error message like this:

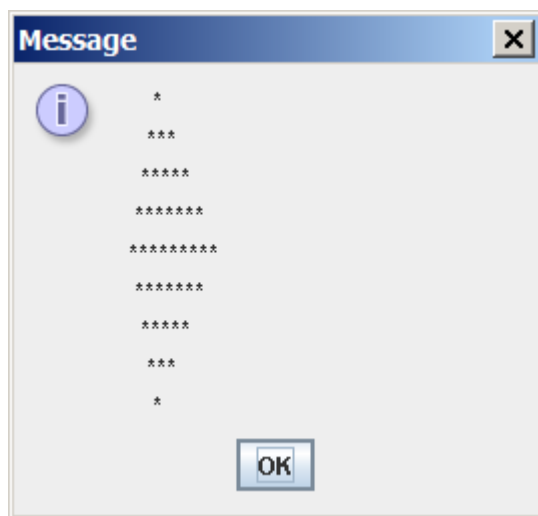
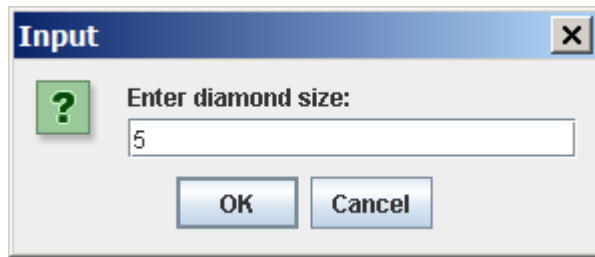
```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 9  
    at _exception_error.ex01.MyClass.main(MyClass.java:8)
```

Could you figure out how to add exception handling to catch that error and print out a more friendly message like this “**The element 9 does not exist!**”.

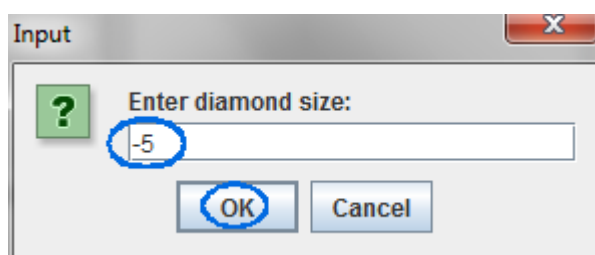
Question 4: Draw diamond shape with UI

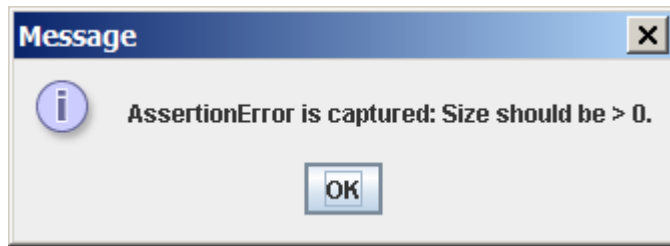
You have learn how to create UI programs with Java Swing so please create a program to ask user for a number to draw a diamond shape, and if the user enters incorrectly (maybe not a number) then display a friendly message to warn them about the error.

If user enters correctly:



If user enters incorrectly (negative/different data type like String, ...):





That's it! Pretty short lab, enjoy! Now, you can work on the final projects with your teammates!

