Exercise 8 – Exceptions

Objective

The objective of this exercise is to look at writing, throwing and catching exceptions. By the end of this we should be comfortable with using exceptions and some basic input operations in Java.

A lot of this exercise will be down to you to look up how to do the specific input operations, use the Java API or the oracle tutorials if you get stuck as well as the slides.

Overview

Read in a console input

There are a number of ways to read input using Java. We are going to read input line by line then append input line to StringBuffer object and output the contents from StringBuffer object. Console input and output operations can cause a lot of potential exceptions so we want to be as specific as possible when catching exceptions, don't just use Exception!

1. Create a class called MyConsoleReader with a main method inside it
2. Implement an readAndPrint() method
   a. We want to use both a BufferedReader and an InputStreamReader object in this method. Look them both up in the API and get familiar with some of the options. BufferedReaders allow the program to read in a console input line-by-line.
   b. Create a new StringBuffer object
   c. Create a BufferedReader and InputStreamReader in the method
      i. You will need to create a new InputStreamReader object by supplying System.in as an argument to InputStreamReader constructor
      ii. You will need to create a BufferedReader object by supplying InputStreamReader object created in the above step

      Refer to the slides to see an example of this, or try and work it out using the API methods!

      iii. Prompt and read in each line from the console and append StringBuffer object with the input line, this should continue until input value is “stop”. Once the loop terminates, print the value of StringBuffer object

Hint: You should use a loop

d. IOException need to be caught for the code to work

e. It is not best practice to open resources such as streams without closing them when finished. To ensure that this happens add a finally block to your code which calls the close () method on the buffered reader.

   You will need to handle any exceptions in this code as well! This can lead to very confusing to follow code, which we can deal with better by throwing exceptions out of the method; we’ll have a look at doing that later!

3. Inside the main method - create an object from MyConsoleReader class and invoke readAndPrint( ) method

Glossary of key terms

Exception

It is a runtime error

StringBuffer

It is a mutable class object, means the objects of StringBuffer can be modified after creation

BufferedReader

It is one of the character stream classes from Java API which is can read the data line-by-line

InputStreamReader

It is a bridge class between the byte stream classes and character stream classes

IOException

It is an exception class which is associated with I/O activities