PÉCSI TUDOMÁNYEGYETEM
TERMÉSZETTUDOMÁNYI KAR

“Methodology of Programming I”

Spring 2025

PTE

Zahra Ahmadipour

Zahra@gamma.ttk.pte.hu

Exceptions

The term exception is shorthand for the phrase "exceptional event."

An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions.

When an exception occurs and it’s not handled, the runtime system and the program will terminate.

try Block:

The first step in constructing an exception handler is to enclose the code that might throw an exception within a try block.

If an exception occurs within the try block, that exception is handled by an exception handler associated with it.

To associate an exception handler with a try block, you must put a catch block after it.

catch Blocks:

You associate exception handlers with a try block, by providing one or more catch blocks directly after the try block.

Each catch block is an exception handler that handles the type of exception indicated by its argument.

The argument type, declares the type of exception that the handler can handle and must be a class that inherits from the Throwable class.

finally Blocks:

The finally block always executes when the try block exits.

This ensures that the finally block is executed even if an unexpected exception occurs.

A try statement does not have to have a catch block if it has a finally block.

try-with-resources statement:

Using a try-with-resources statement instead of a finally block when closing a file is recommended.

The try-with-resources statement automatically releases system resources when no longer needed. It ensures that each resource is closed at the end of the statement.

Any object that implements java.lang.AutoCloseable, can be used as a resource.

throw:

Before you can catch an exception, some code somewhere must throw one.

Exceptions can be thrown by your code, a code from a package written by someone else such as Java runtime environment.

The exception is always thrown with a throw statement.

The throw statement requires a single argument: a throwable object.

Throwable objects are instances of any subclass of the Throwable class.

Exception Classes:

The Java platform provides numerous exception classes. All the classes are descendants of the Throwable class. You can see its hierarchy on the next page.

You can also create your own exception classes to represent problems that can occur within the classes you write.

- java.lang.**Throwable**
  - java.lang.**Error**
    - …
  - java.lang.**Exception**
    - java.lang.**CloneNotSupportedException**
    - java.lang.**InterruptedException**
    - java.lang.**ReflectiveOperationException**
      - java.lang.**ClassNotFoundException**
      - java.lang.**IllegalAccessException**
      - java.lang.**InstantiationException**
      - java.lang.**NoSuchFieldException**
      - java.lang.**NoSuchMethodException**
    - java.lang.**RuntimeException**
      - java.lang.**ArithmeticException**
      - java.lang.**ArrayStoreException**
      - java.lang.**ClassCastException**
      - java.lang.**EnumConstantNotPresentException**
      - java.lang.**IllegalArgumentException**
        - java.lang.**IllegalThreadStateException**
        - java.lang.**NumberFormatException**
      - java.lang.**IllegalMonitorStateException**
      - java.lang.**IllegalStateException**
      - java.lang.**IndexOutOfBoundsException**
        - java.lang.**ArrayIndexOutOfBoundsException**
        - java.lang.**StringIndexOutOfBoundsException**
      - java.lang.**NegativeArraySizeException**
      - java.lang.**NullPointerException**
      - java.lang.**SecurityException**
      - java.lang.**TypeNotPresentException**
      - java.lang.**UnsupportedOperationException**

Overall Process:

After a method throws an exception, the runtime system attempts to find something to handle it.

The set of possible "somethings" to handle the exception, is the ordered list of methods that had been called (call stack) to get to the method where the error occurred.

The runtime system searches the call stack for a method that can handle the exception (exception handler/catch block).

If the type of the exception object thrown, matches the type specified by the handler, the exception handler chosen is said to catch the exception.

Create Exception Classes:

When faced with choosing the type of exception to throw, you can either use one written by someone else or you can write one of your own.

In Java, we can create our own exceptions that are derived classes of the Exception class.

Creating our own Exception is known as custom exception or user-defined exception.

You should write your own exception class in following circumstances:

- An exception type which is not represented by the Java platform is needed.
- Differentiating your exceptions is helpful.
- Your code throws more than one related exception.
- Your package needs to be independent and self-contained.

References:

https://docs.oracle.com/javase/tutorial/tutorialLearningPaths.html

https://www.w3schools.com/

https://www.geeksforgeeks.org/