Parallel Programming

Exceptions --- Examples

Thomas Gross

Spring Semester 2010

Mar 4, 2010

Martin Lanter
Javimka im Inforum

lanterm@student.ethz.ch
www.lantersoft.ch

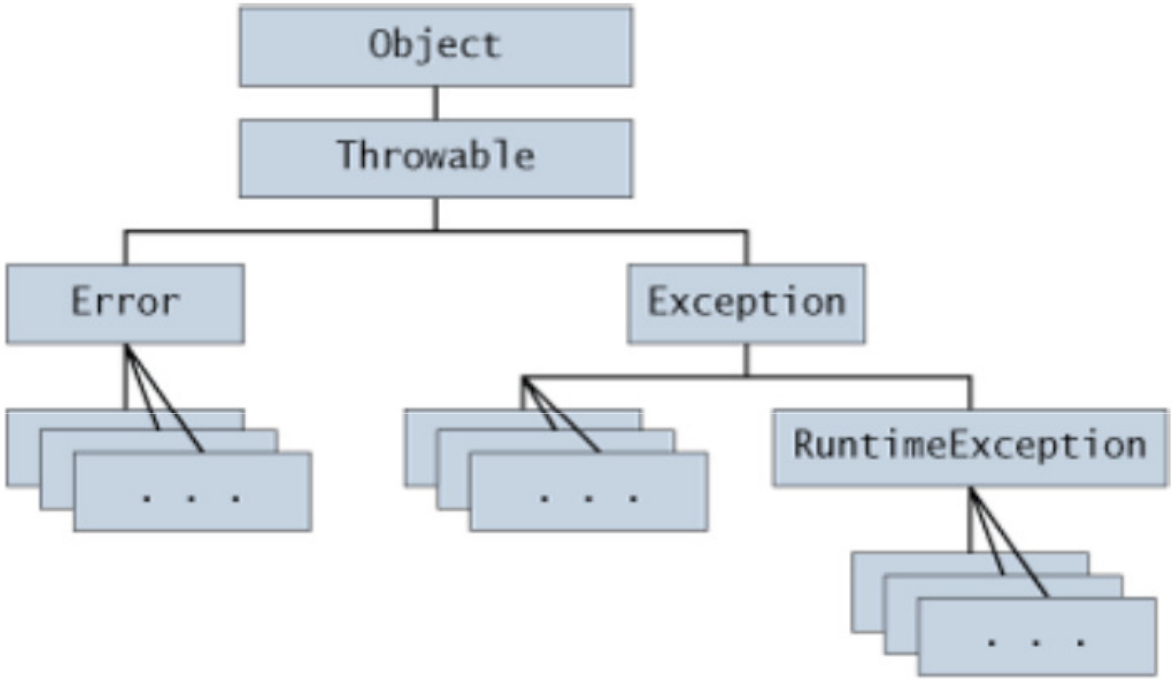
www.java-forum.org

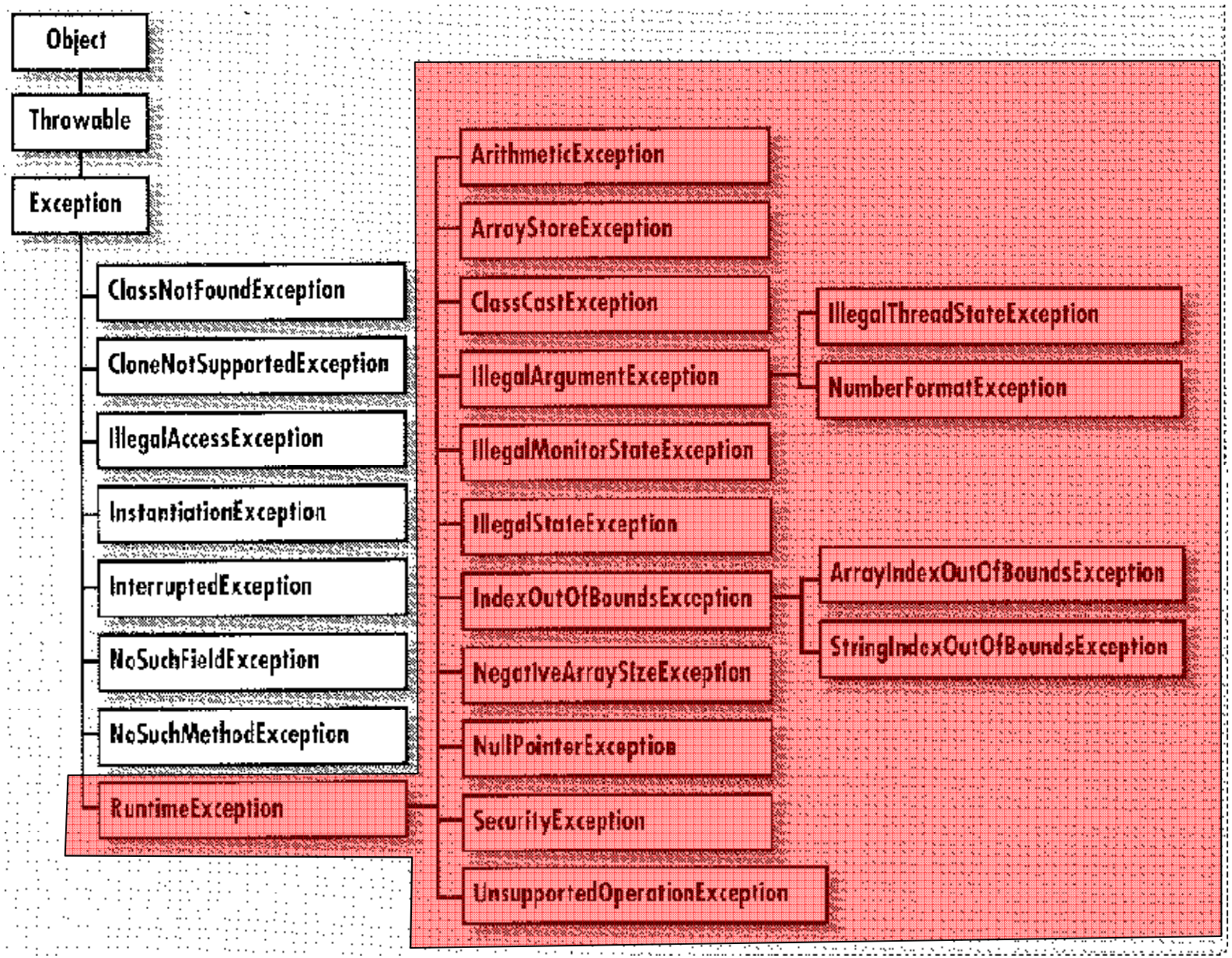
Solution of assignment 1

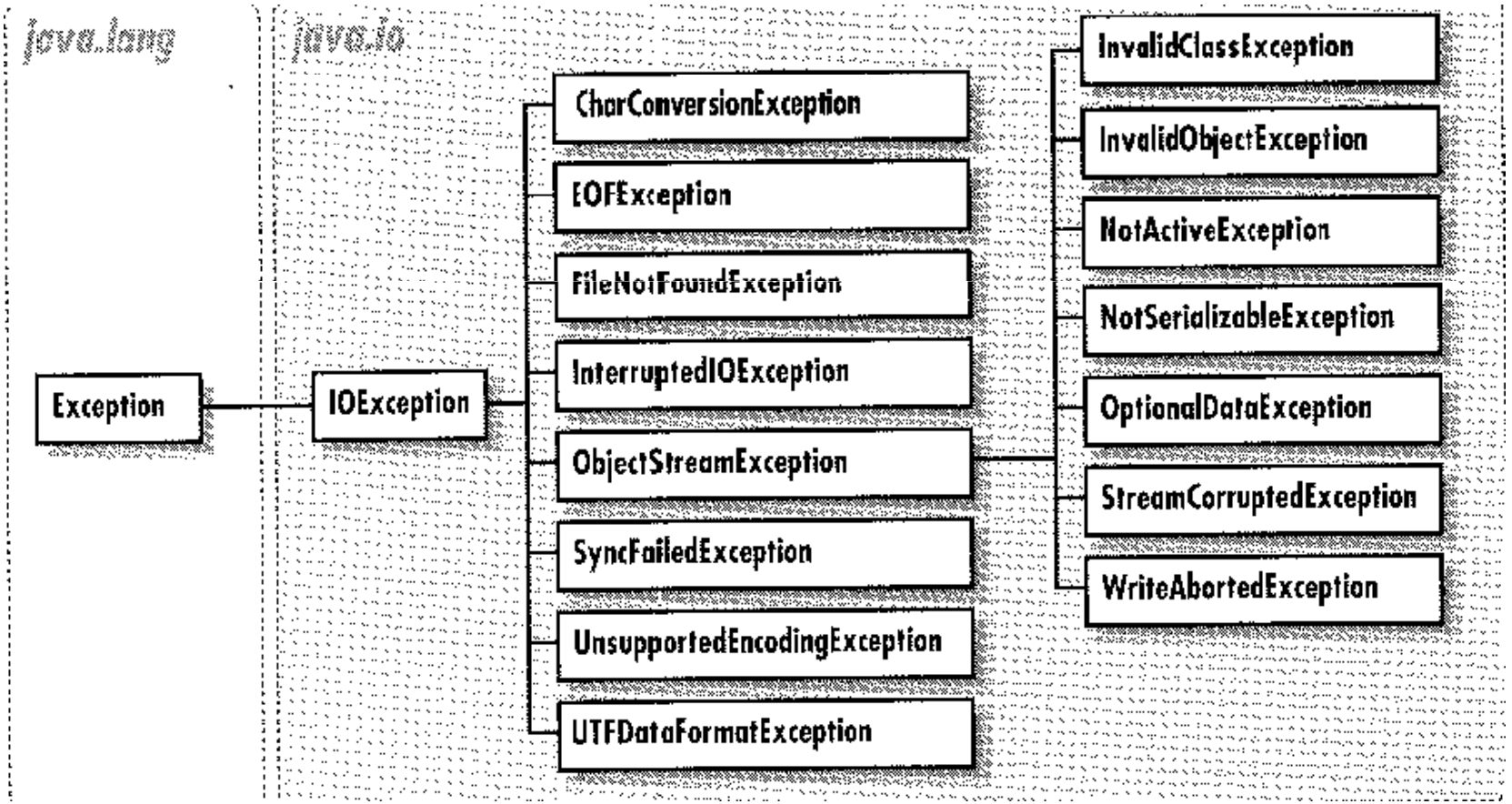
```
class Example1 {
 /* baseline */
  public static void main(String[] args) {
    int i;
    int tmp;
    /* iterate through the argument vector */
    for (i = 0; i < args.length; i++) {
      tmp = Integer.parseInt(args[i]) + 1;
      System.out.println(tmp); /* print out the result */
    }
  }
}
```

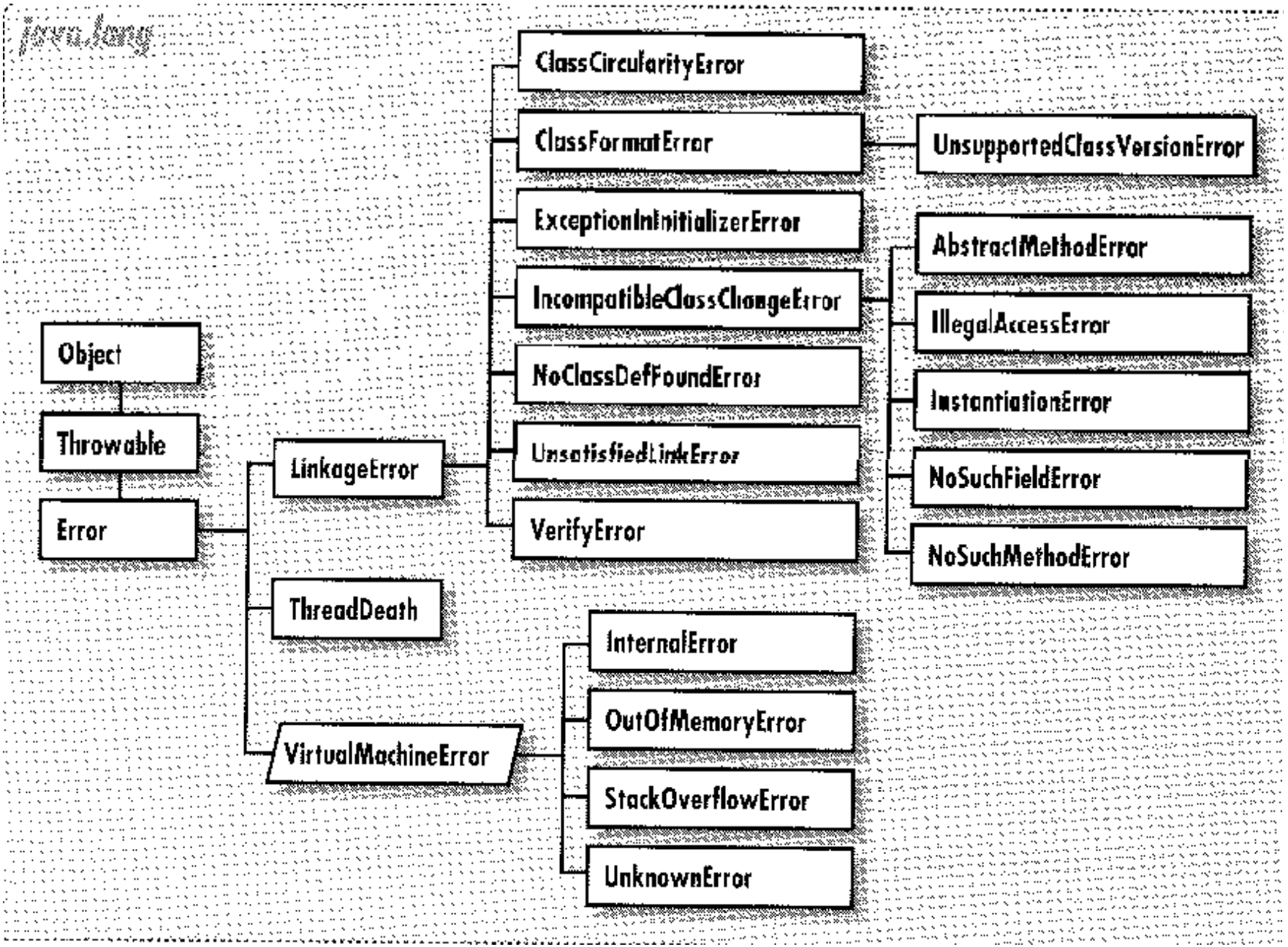
```
/*
This program takes any number of integer arguments and increases
each of them by 1. It then outputs them line by line.

This source code is intellectual property of ***.
    All attempts to copy, modify and/or redistribute this file will be
    prosecuted. Trespassers will be shot, survivors will be shot again.
    Don't f*** with the source or the source will f*** you.
    Polymorphic code rulez the world. You will never know.
*/
```

```
public class Assignment1 {

    public static void main(String[] args) {

        for (int i = 0; i < args.length; i++) {

          System.out.println(Integer.parseInt(args[i])+1);
        }

    }
}
```


Object
Throwable
Error
Exception
. . .
. . .
RuntimeException
. . .


Object
Throwable
Exception
ClassNotFoundException
CloneNotSupportedException
IllegalAccessException
InstantiationException
InterruptedException
NoSuchFieldException
NoSuchMethodException
RuntimeException
ArithmeticException
ArrayStoreException
ClassCastException
IllegalArgumentException
IllegalMonitorStateException
IllegalStateException
IndexOutOfBoundsException
NegativeArraySizeException
NullPointerException
SecurityException
UnsupportedOperationException
IllegalThreadStateException
NumberFormatException
ArrayIndexOutOfBoundsException
StringIndexOutOfBoundsException


java.lang
java.io
Exception
IOException
CharConversionException
EOFException
FileNotFoundException
InterruptedIOException
ObjectStreamException
SyncFailedException
UnsupportedEncodingException
UTFDataFormatException
InvalidClassException
InvalidObjectException
NotActiveException
NotSerializableException
OptionalDataException
StreamCorruptedException
WriteAbortedException


java.lang
Object
Throwable
Error
LinkageError
ThreadDeath
VirtualMachineError
ClassCircularityError
ClassFormatError
ExceptionInInitializerError
IncompatibleClassChangeError
NoClassDefFoundError
UnsatisfiedLinkError
VerifyError
InternalError
OutOfMemoryError
StackOverflowError
UnknownError
UnsupportedClassVersionError
AbstractMethodError
IllegalAccessError
InstantiationError
NoSuchFieldError
NoSuchMethodError

Example 2

```
class Example2 {
 public static void main(String[] args) {
   int i;
   int tmp;
   /* iterate through the argument vector */
   for (i = 0; i < args.length; i++) {
     try {
       tmp = Integer.parseInt(args[i]);
       if (tmp < 0) {
         throw(new MyException("less than zero"));
       }
       System.out.println(tmp);
     }
```

Example 2

```
      catch (MyException e) {
      System.out.println("Exception caught ...");
      System.out.println(e.getMessage());
    }
  }
  System.out.println(“done.”)
  }
}
class MyException extends Exception {
  MyException(String text) {
    super(text);
  }
}
```

Example2

Input: 8, 3, 11, -4, 6, -7, 2, 0
Output:

```
8
3
11
Exception caught ...
less than zero
6
Exception caught ...
less than zero
2
0
done.
```

Example 5

```
class Example5 {

  public static void main(String[] args) {

    int i;
    int tmp;
    int count = 0;

    try {
      /* iterate through the argument vector */
      for (i = 0; i < args.length; i++) {
```

Example 5

```
try {
    tmp = Integer.parseInt(args[i]);
    if (tmp < 0) {
       throw(new MyException1("less than zero"));
    }
    System.out.println(tmp);
} catch (MyException1 e) {
    System.out.println("Exception1 caught ...");
    System.out.println(e.getMessage());
    count++;
    if (count >= 3) {
       throw(new MyException2 ());
    };
}
```

Example 5

```
      finally {
       System.out.println("Processed one integer");
      }
    }
   } catch (MyException2 e) {
      System.out.println("Exception2 caught");
      System.out.println("Enough.");
    }
    finally {
      System.out.println("done.");
    }
   }
}
```

Example 5

```
class MyException1 extends Exception {
  MyException1(String text) {
    super(text);
  }
}

class MyException2 extends Exception {}
```

Example5

Input: 8, 3, -11, -4, 5, -2, 6

Output:

```
8
Processed one integer
3
Processed one integer
Exception1 caught ...
less than zero
Processed one integer
Exception1 caught ...
less than zero
Processed one integer
5
Processed one integer
Exception1 caught ...
less than zero
Processed one integer
Exception2 caught
Enough.
done.
```