

Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería

Plan de Ingeniero en Software y Tecnologías Emergentes



Materia

Verificación y Validación del Software (361)

Meta 3.2

Manejar excepciones

Docente

Claudia Gabriel Tona Castro

Presentado por:

Luis Eduardo Galindo Amaya (1274895)

Sumario

Excepciones.....	3
RuntimeException.....	3
IOException.....	3
NumberFormatException.....	3
FileNotFoundException.....	3
NonMultipleOfSevenException.....	3
Conclusión.....	4
Código.....	5
Main.java.....	5
Student.java.....	8
GradesExporter.java.....	9
GradesImporter.java.....	10
NonMultipleOfSevenException.java.....	12
Fuentes.....	13

Meta 3.2

Manejar excepciones

Excepciones¹

RuntimeException

RuntimeException es la superclase de aquellas excepciones que pueden ser lanzadas durante la operación normal de la Máquina Virtual de Java. RuntimeException y sus subclases son excepciones no comprobadas.

IOException

Señala que ha ocurrido una excepción de E/S de algún tipo. Esta clase es la clase general de excepciones producidas por operaciones de E/S fallidas o interrumpidas.

NumberFormatException

El error indica que la aplicación ha intentado convertir una cadena de texto a uno de los tipos numéricos, pero la cadena no tiene el formato adecuado.

FileNotFoundException

Señales de que ha fallado un intento de abrir el archivo indicado por una ruta de acceso especificada. Esta excepción será lanzada por los constructores FileInputStream, FileOutputStream y RandomAccessFile cuando un archivo con la ruta de acceso especificada no existe. También será lanzada por estos constructores si el archivo existe pero por alguna razón es inaccesible, por ejemplo, cuando se intenta abrir un archivo de solo lectura para escritura.

NonMultipleOfSevenException

Excepción personalizada, las instrucciones indican que cada alumno tiene siete si la cantidad no es un múltiplo entonces no hay una cantidad exacta de alumnos, entonces para evitar problemas se lanza esta excepción.

¹ La traducción se hizo utilizando Chatgpt

Conclusión

Durante esta practica aprendí como manejar excepciones en mis programas, una buena validación de datos disminuye la necesidad de usar excepciones pero las excepciones nos pueden ayudar a controlar que pasa con los fallos mas grandes o con las cosas que no controlamos dentro de la computadora, como el sistema de archivos o el Runtime, hilos, conexiones con sockets, bases de datos etc...

Código

Main.java

```
package Unknown;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.List;

public class Main {
    public static void main(String[] args) throws IOException {
        File entrada = null;
        File salida = new File("salida.txt");

        System.out.println();
        System.out.println("Meta 3.2: Manejar excepciones");
        System.out.println("  1. Flujo Normal");
        System.out.println("  2. NumberFormatException");
        System.out.println("  3. NonMultipleOfSevenException (Propia)");
        System.out.println("  4. FileNotFoundException");
        System.out.println("  0. Salir");
        System.out.print("\nOpción: ");

        switch ((char) System.in.read()) {
            case '1':
                entrada = new File("practica3.2c.csv");
                break;

            case '2':
                entrada = new File("practica3.2b.csv");
                break;

            case '3':
                entrada = new File("practica3.2.csv");
                break;

            case '4':
                entrada = new File("archivo_falso.csv");
                break;

            case '0':
                System.out.println("Adios!");
                System.exit(0);
                break;
        }
    }
}
```

```
        default:
            System.out.println("No existe esa opción.");
            System.exit(0);
            break;
    }

    // cargar datos
    System.out.println("Leyendo Archivos...");
    GradesImporter gradesImporter = null;
    try {
        gradesImporter = new GradesImporter(entrada);
    } catch (FileNotFoundException e) {
        System.out.println("El archivo '" + entrada + "' no existe");
        System.exit(0);
        throw new RuntimeException(e);
    }

    // obtener alumnos
    System.out.println("Importando alumnos...");
    List<Student> alumnos = null;
    try {
        alumnos = gradesImporter.getStudents();
    } catch (NumberFormatException b) {
        System.out.println("El Formato de una de las entradas no es
numerico: ");
        System.out.println(" - " + b);
        System.exit(0);
    } catch (NonMultipleOfSevenException e) {
        System.out.println("El numero de calificaciones no es múltiplo
de 7");
        System.exit(0);
    } catch (IOException e) {
        throw new RuntimeException(e);
    }

    // mostrar calificaciones
    for (var alumno : alumnos) {
        System.out.printf(alumno.prettyPrint());
    }
    System.out.println();

    // guardar los datos del usuario
    System.out.println("Exportando datos...");

    GradesExporter exporter = new GradesExporter(salida, alumnos);
    try {
        exporter.save();
    } catch (IOException e) {
        throw new RuntimeException(e);
    }

    // NumberFormatException
```

```
        System.out.println("Archivo '" + salida + "' creado.");  
    }  
}
```

Student.java

```
package Unknown;

import java.util.Arrays;

public class Student {
    String nombre;
    String matricula;
    Integer[] calificaciones;

    public Student(String nombre, String matricula, Integer[]
calificaciones) {
        this.nombre = nombre;
        this.matricula = matricula;
        this.calificaciones = calificaciones;
    }

    public String getNombre() { return nombre; }

    public String getMatricula() { return matricula; }

    public Integer[] getCalificaciones() {
        return calificaciones;
    }

    public double getPromedio() {
        return Arrays.stream(calificaciones)
            .mapToDouble(a -> a)
            .average()
            .orElse(0);
    }

    public String prettyPrint() {
        return String.format("%n%s%n | Matrícula: %s%n | Calificaciones: %s
%n | Promedio: %.2f %n", this.getNombre(), this.getMatricula(),
Arrays.toString(this.getCalificaciones()), this.getPromedio());
    }

    @Override
    public String toString() {
        return "Alumno{" +
            "nombre='" + nombre + '\'' +
            ", matricula='" + matricula + '\'' +
            ", calificaciones=" + Arrays.toString(calificaciones) +
            ", promedio=" + getPromedio() +
            '}';
    }
}
```


GradesExporter.java

```
package Unknown;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.List;

public class GradesExporter {

    private final File grades;
    private List<Student> students;

    public GradesExporter(File grades, List<Student> students) {
        this.grades = grades;
        this.students = students;
    }

    public void save() throws IOException {
        BufferedWriter br = new BufferedWriter(new FileWriter(this.grades,
false));

        for (Student student : students) {
            br.write(student.toString());
            br.write('\n');
        }

        br.close();
    }
}
```

GradesImporter.java

```
package Unknown;

import java.io.*;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class GradesImporter {

    private static final String COMMA_DELIMITER = ",";
    private final File grades;
    private final List<Student> students;

    public GradesImporter(File grades) throws FileNotFoundException {
        this.grades = grades;
        this.students = new ArrayList<>();

        if (!this.grades.exists()) throw new FileNotFoundException();
    }

    // extrae los datos del csv
    public List<Integer> extractGrades() throws IOException,
    NumberFormatException {
        String[] values = new String[0];
        List<Integer> grades;

        // leer el archivo y separar los valores por coma
        BufferedReader bufferedReader = new BufferedReader(new
        FileReader(this.grades));

        String line;
        while ((line = bufferedReader.readLine()) != null) {
            values = line.split(COMMA_DELIMITER);
        }

        // convertir los valores a números
        grades = Arrays.stream(values)
            .map(Integer::parseInt)
            .toList();

        return grades;
    }

    public List<Student> getStudents() throws IOException,
    NumberFormatException, NonMultipleOfSevenException {
        List<Integer> agrades = this.extractGrades();

        int ngrades = agrades.size();
        // Verificar que el número de elementos sea múltiplo de 7
    }
```

```
    if (ngrades % 7 != 0) throw new NonMultipleOfSevenException();

    // dividir las calificaciones en 'n' grupos de 7 elementos
    for (int i = 0; i < ngrades / 7; i++) {
        int foo = i * 7;
        var a = agrades.subList(foo, foo + 7);

        Integer[] array = new Integer[7];
        a.toArray(array);

        students.add(new Student("User" + i, "127489" + i, array));
    }

    return students;
}
```

NonMultipleOfSevenException.java

```
package Unknown;

public class NonMultipleOfSevenException extends Exception {
    public NonMultipleOfSevenException() {
        super("El numero de elementos en la lista no es múltiplo de
siete.");
    }

    public NonMultipleOfSevenException(String message) {
        super(message);
    }

    public NonMultipleOfSevenException(String message, Throwable cause) {
        super(message, cause);
    }

    public NonMultipleOfSevenException(Throwable cause) {
        super(cause);
    }
}
```

Fuentes

RuntimeException (Java Platform SE 8). (2024, April 4).

<https://docs.oracle.com/javase/8/docs/api/java/lang/RuntimeException.html>

IOException (Java Platform SE 8). (2024, April 4).

<https://docs.oracle.com/javase/8/docs/api/java/io/IOException.html>

NumberFormatException (Java Platform SE 8). (2024, April 4).

<https://docs.oracle.com/javase/8/docs/api/java/lang/NumberFormatException.html>

FileNotFoundException (java platform SE 8). (2024, April 4).

<https://docs.oracle.com/javase/8/docs/api/java/io/FileNotFoundException.html>