



Universidad Nacional de San Luis
Área de Programación y Metodologías de Desarrollo de Software

Ingeniería en Informática – Ingeniería en Computación

Programación II

Práctico N° 6

Lenguaje Java: Archivos y Excepciones

Ejercicio 1: Explica y ejemplifica qué son las excepciones.

Ejercicio 2: Explica los distintos tipos de excepciones.

Ejercicio 3: Escriba un programa que maneje una excepción de división por cero.

Ejercicio 4: Escriba un programa que permita generar y capturar una excepción del tipo *ArrayIndexOutOfBoundsException*.

Ejercicio 5: Escriba un programa que muestre un cartel cuando se capture una excepción de puntero null.

Ejercicio 6: Escriba un programa que indique si un archivo ingresado por el usuario existe. En caso afirmativo, el programa debe imprimir por pantalla el contenido del archivo. En caso contrario, el programa debe imprimir un mensaje de error.

Ejercicio 7: Escriba un programa que permita que el usuario ingrese una frase. Luego almacene la frase en el archivo *texto.txt*.

Ejercicio 8: Desarrolle un programa que almacene en el archivo *parrafo.txt* un conjunto de oraciones escritas por el usuario. Las oraciones son solicitadas de a una por vez al usuario hasta que éste ingrese “*”, lo cual indica el fin del ingreso de oraciones. Finalmente, muestre por pantalla el contenido del archivo.

Ejercicio 9: Incorpore al programa del ejercicio anterior las excepciones correspondientes para controlar las siguientes situaciones:

- a) El archivo no puede ser leído.
- b) El archivo no puede ser escrito.

Ejercicio 10: En el archivo *registro.txt* se encuentran los datos de un grupo de personas. Los datos almacenados para cada persona son los siguientes: *Nombre y Apellido, Ocupación, Carrera y e-mail*. Los datos de una persona están separados por “\n”. Teniendo en cuenta la descripción realizada previamente se pide escriba un programa que:

- a) Lea los datos de las personas y los inserte en una lista.
- b) Ordene alfabéticamente la lista.
- c) Permita que el usuario modifique los datos de un usuario específico.
- d) Almacene los datos de la lista en el archivo *registro.txt*.

Ejercicio 11: Defina la clase *ArchivoDeTexto* en la cual se implemente el comportamiento necesario

para el manejo de archivos. La clase debe poseer los siguientes métodos:

- a) Constructor (debe pasar como parámetro el nombre del archivo).
- b) Abrir archivo.
- c) Escribir en el archivo.
- d) Leer del archivo.
- e) Cerrar el archivo.

Ejercicio 12: Escriba un programa que cuente la cantidad de veces que se repite una letra en un archivo. El nombre del archivo es ingresado por el usuario.

Ejemplo: Entrada: “Hola Chicos” Salida: (H,2) (o,2) (l,1) (a,1) (c,2) (s,1)

Nota: realiza este ejercicio modificando la entrada / salida estándar a un archivo.

Ejercicio 13: Implementa un menú iterativo que utilice excepciones para obligar a que el usuario ingrese opciones válidas.

Ejercicio 14: Define la clase *Punto1erCuadrante* que permite crear sólo aquellos puntos que se encuentren en el primer cuadrante. Si se desea crear un punto de otro cuadrante, debe generar una excepción. Para esto, define tu propia excepción.

Ejercicio 15: Escriba un programa en Java que, para cualquier ruta indicada por el usuario, muestre:

- a) Si el archivo existe o no.
- b) En caso de existir, debe mostrar los siguientes datos:
 - a. Nombre.
 - b. Tamaño.
 - c. Permisos de lectura y escritura.

Ejercicio 16: Ejecute y analice los dos manejos de excepciones para un mismo requerimiento que figuran a continuación, plantee los puntos a favor y en contra de cada uno de ellos. Plantee una versión que mejore (en caso de ser posible) los problemas encontrados en dichas versiones.

```
public class Exc_Mod1 {  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int x = 0, y = 0, z = 0;  
        boolean repetir;  
        do {  
            repetir = false;  
            try {  
                System.out.print("Introduce primer número entero: ");  
                x = sc.nextInt();  
  
                System.out.print("Introduce segundo número entero: ");  
                y = sc.nextInt();  
  
                System.out.print("Introduce tercer número entero: ");
```

```
        z = sc.nextInt();

    } catch (InputMismatchException e) {
        System.out.println("Valor no válido" + e.toString());
        sc.nextLine();
        repetir = true;
    }
} while (repetir);
System.out.println("int introducido -> " + x);
System.out.println("int introducido -> " + y);
System.out.println("int introducido -> " + z);
}

}

/*****/
public class Exc_Mod2 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int x = 0, y = 0, z = 0;
        boolean repetir;
        do {
            repetir = false;
            try {
                System.out.print("Introduce primer número entero: ");
                x = sc.nextInt();
            } catch (InputMismatchException e) {
                System.out.println("Valor no válido" + e.toString());
                sc.nextLine();
                repetir = true;
            }
        } while (repetir);
        do {
            repetir = false;
            try {
                System.out.print("Introduce segundo número entero: ");
                y = sc.nextInt();
            } catch (InputMismatchException e) {
                System.out.println("Valor no válido" + e.toString());
                sc.nextLine();
                repetir = true;
            }
        } while (repetir);
        do {
            repetir = false;
            try {
```



```
        System.out.print("Introduce tercer número entero: ");
        z = sc.nextInt();
    } catch (InputMismatchException e) {
        System.out.println("Valor no válido" + e.toString());
        sc.nextLine();
        repetir = true;
    }
} while (repetir);
System.out.println("int introducido -> " + x);
System.out.println("int introducido -> " + y);
System.out.println("int introducido -> " + z);
}
```