UD2Tarea8 - Try Catch

Andrés Pérez Guardiola - 1st DAW

1. Supón que un programa lanza una excepción tipo ArrayIndexOutOfBoundsException. Usando la jerarquía de excepciones del curso, determina cuál de las siguientes cláusulas catch podría manejar esa excepción.

a. catch (RunTimeException e)

También podría manejar este tipo de excepciones.

b. catch (StringIndexOutOfBoundsException e)

No, esta excepción solo se lanzaría cuando es un String y no un Array el que se ha salido el índice fuera de los límites.

c. catch (IndexOutOfBoundsException e)

Es el padre inmediatamente superior a ArrayIndexOutOfBoundsException

d. catch (Exception e)

ArrayIndexOutOfBoundsException hereda de Exception a través de sus padres: IndexOutOfBoundsException y RuntimeException.

e. catch (ArrayStoreException e)

No, el error es de Index no de Store. Esta excepción la tendríamos si intentáramos guardar algo dentro de un array de un tipo distinto.

2. ¿Qué tipo de excepción se lanzaría para las siguientes declaraciones?

a. Integer.parseInt("26.2");

Integer intenta parsear un String de un número que contiene decimales. Como resultado dará una excepción del tipo NumberFormatException.

b. String s; s.indexOf(’a’);

No lanza ninguna excepción, sino un error. El string s no ha sido inicializado.

c. String s = "hello"; s.charAt(5);

StringIndexOutOfBoundsException. El motivo es que la posición de los caracteres dentro del string empiezan en 0 y este string tiene desde posición 0 a posición 4. El carácter 5 no existe y dará esa excepción.

3. En el programa siguiente, supón que la primera vez que se llama a Math.random() devuelve 0,98, y la segunda vez que se llama devuelve 0,44. ¿Qué resultado imprimirá el programa?

```
try {
    double d = Math.random();
    if (d > 0.95)
        throw new ArithmeticException(d + " está fuera de rango") ;
    System.out.println("El número es " + d);

    double j = Math.random();
    if (j > 0.5)
        throw new ArithmeticException(j + " está fuera de rango") ;
    System.out.println("El número es " + j);
} catch (ArithmeticException e) {
    System.out.println(e.getMessage());
}
```

En la primera ejecución, el programa obtuvo un valor de 0.98 lo cual provocó que de forma manual se lance una excepción del tipo ArithmeticException con un mensaje personalizado diciendo “0.98 está fuera de rango”.
Este excepción es capturada por el catch y pintaría en pantalla el mensaje personalizado.
Una vez se produce la excepción, el segundo Math.random() no puede devolver nada ya que no se ejecuta. El programa continuaría después del catch.

4. Para los valores devueltos por Math.random() en el ejercicio anterior, muestra qué se generaría si se llamara a printStackTrace() además de imprimir un mensaje de error.

```
public class Main
{
   public static void main(String[] args)
   {

       try{
           double d = Math.random();
           if (d > 0.95d)
               throw new ArithmeticException(d +
                                             " está fuera de rango");
           System.out.println("El número es " + d);

           double j = Math.random();
           if (j > 0.5d)
               throw new ArithmeticException(j +
                                            " está fuera de rango");
           System.out.println("El número es " + j);
       } catch (ArithmeticException e)
       {
           System.out.println(e.getMessage());
           e.printStackTrace();
       }
   }
}
```

Resultado

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\I
El número es 0.12200579091393138
0.6438660661499271 está fuera de rango
java.lang.ArithmeticException Create breakpoint : 0.6438660661499271 está fuera de rango
    at Main.main(Main.java:14)

Process finished with exit code 0
```

En este caso hemos lanzado la segunda excepción. El primer número no ha cumplido la condición y se ha pintado el mensaje “El número es ….”.

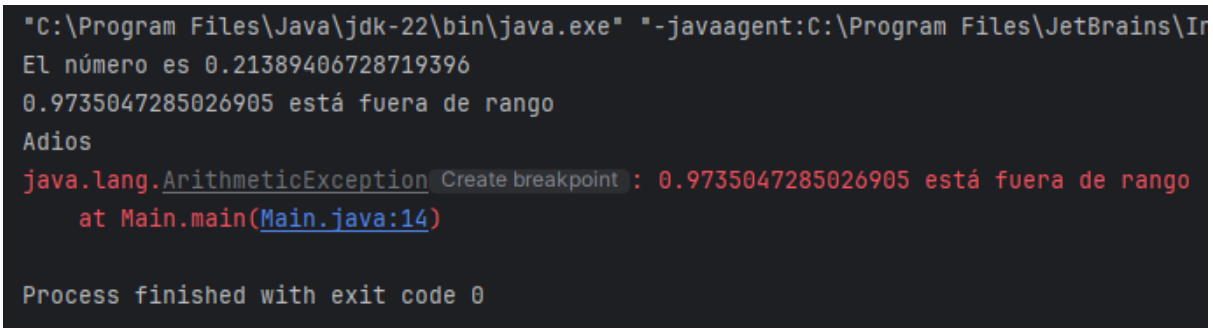
e.printStackTrace() nos devuelve, a parte del mensaje, una traza de donde ha se ha producido la excepción. En este caso, solo sucede en la línea 14, justo donde hacemos throw de la excepción por segunda vez.

5. Supón que la primera vez que se llama a Math.random() devuelve 0,44, y la segunda vez que se llama devuelve 0,98. ¿Qué resultado imprimirá el programa?

La primera vez que se ejecute el método random() devuelve un valor de 0.44 que no supera la condición justo abajo, por tanto seguiría el programa pintando el mensaje:
“El número es 0.44”

La segunda vez obtenemos el valor 0.98 que si cumple la condición del if que tenemos abajo, ejecutándose la generación de la excepción. El try se detendría en ese punto y el programa seguiría por el catch.

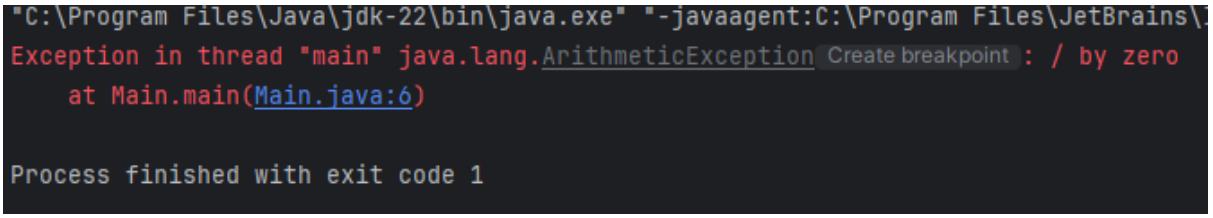
El catch recibiría el objeto de excepción y lo asignaría a e. En el primer paso pintaría el mensaje personalizado que le hemos escrito y luego nos mostraría la traza. Cuando lo he probado, si pongo un mensaje después del try-catch el programa continúa con normalidad pero la traza no la pinta en el programa hasta que este termina. Es decir, si pongo un mensaje de despedida en el programa, primero pintaría el mensaje de despedida y cuando el programa termina es cuando pintaría la traza.



6. Busca el error de división por cero en el siguiente programa y luego muestra qué trazo de pila imprimirá el programa:

```
for (int k = 0; k < 5; k++)
    System.out.println(100 / k);
```

El programa fallará en la primera iteración ya que empieza con k = 0.



La excepción que lanza es ArithmeticException por división por cero.

7. Modifica el programa anterior para que maneje la excepción de división por cero en sí mismo, en lugar de dejar que Java la maneje. Haga que imprima un mensaje de error y un seguimiento de la pila.

```
public class Main
{
   public static void main(String[] args) {

        try{
            for(int k = 0; k < 5; k++)
            {
                System.out.println(100 / k);
            }
        } catch (ArithmeticException e)
        {
            System.out.println(e.getMessage());
            e.printStackTrace();
        }

   }
}
```

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:
/ by zero
java.lang.ArithmeticException Create breakpoint : / by zero
    at Main.main(Main.java:8)

Process finished with exit code 0
```

8. ¿Qué imprimiría el siguiente segmento de código si valor es igual a 1000?

```
int m = valor;
try {
    System.out.println("Entrando en el try");
    if (m > 100)
        throw new Exception(m + " es muy grande");

    System.out.println("saliendo del bloque try");
}catch (Exception e) {
    System.out.println("ERROR: " + e.getMessage());
}
```

Entraría en el bloque Try y pintaría primero el mensaje “Entrando en el try”
Cuando se evalúe la condición en el if será verdadera y esto lanzará una excepción.
Saldría del bloque try una vez se lanza la excepción y esta sería capturada por el catch.
Dentro del bloque catch se lanza el mensaje “ERROR: 1000 es muy grande”
El programa continuaría después de catch pero como no hay más sentencias terminaría con normalidad.

9. ¿Qué se imprimiría en el programa anterior si valor es igual a 50?

Empezamos el bloque try y pintaría el mensaje de “Entrando en el try”.
La condición del if se evaluaría como falsa, por lo que no se lanzaría la excepción.
Luego se pintaría el mensaje “Saliendo del bloque try” y como no se ejecuta ninguna excepción pues no se iniciaría el bloque catch y continuaría el programa con normalidad.

10. Escribe un programa con un bloque try / catch que arroje una excepción si el valor de la variable x es menor que cero. La excepción debe ser una instancia de Exception y, cuando se detecta, el mensaje devuelto por getMessage() debe ser "ERROR: valor negativo en la coordenada X".

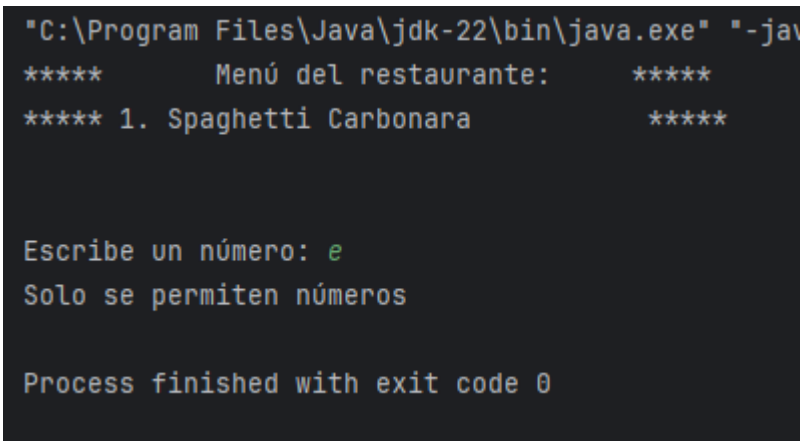
```
import java.util.Scanner;

public class Main
{
   public static void main(String[] args) {
       Scanner sc  = new Scanner(System.in);

       try {
           System.out.print("Dame un número: ");
           int x = sc.nextInt();

           if (x < 0)
           {
               throw new ArithmeticException("ERROR: valor
negativo en la coordenada X");
           }
       } catch (ArithmeticException e)
       {
           System.out.println(e.getMessage());
       }

   }
}
```

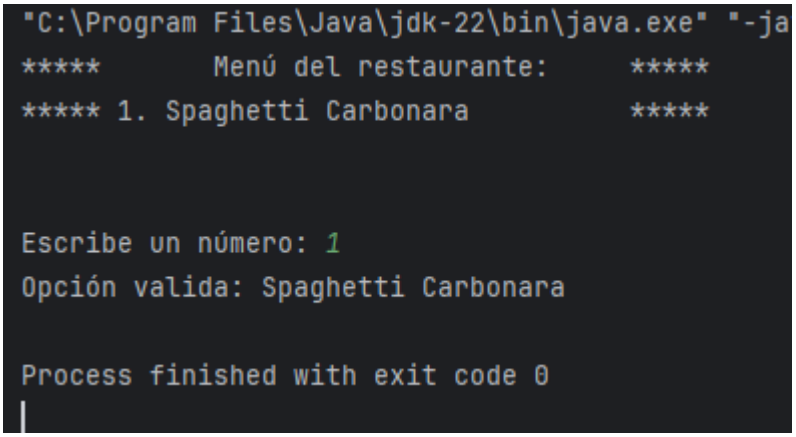
```
"C:\Program Files\Java\jdk-22\bin\java.exe
Dame un número: -4
ERROR: valor negativo en la coordenada X

Process finished with exit code 0
```

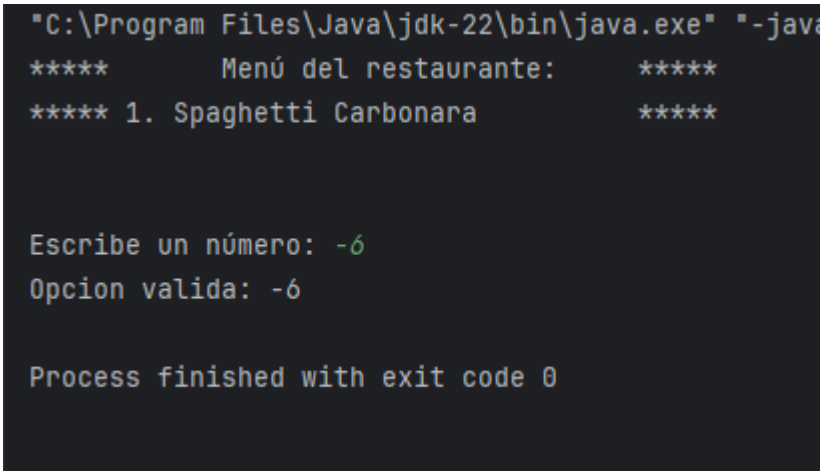
Solo capturo en el caso de que la excepción sea del tipo ArithmeticException. En este caso, la que se ha producido es del tipo InputMismatchException. Esta excepción no viene de java.lang, viene de java.util.

11. Escribe un programa que muestre un menú iterativamente con opciones para dar elegir al usuario. Maneja con un try catch si el usuario inserta una opción que no es numérica, es decir letras, mostrando el mensaje: “Solo se permiten números”. Si el usuario inserta una opción correcta acaba el programa con el mensaje: “Opción válida”, si la opción no está entre las ofrecidas vuelve a mostrar el menú.

```
import java.util.InputMismatchException;
import java.util.Scanner;

public class Main
{
   public static void main(String[] args) {

       Scanner sc = new Scanner(System.in);

       System.out.println("*****       Menú del restaurante:
*****");
       System.out.println("***** 1. Spaghetti Carbonara
*****");
       System.out.print("\n\nEscribe un número: ");

       try {
           int opcion = sc.nextInt();

           switch (opcion){
               case 1:
                   System.out.println("Opción valida: Spaghetti
Carbonara");
                   break;
               default:
                   System.out.println("Opcion valida: " + opcion);
           }

       } catch (InputMismatchException e)
       {
           System.out.println("Solo se permiten números");
       }
   }
}
```


"C:\Program Files\Java\jdk-22\bin\java.exe" "-jav
***** Menú del restaurante: *****
***** 1. Spaghetti Carbonara *****
Escribe un número: e
Solo se permiten números
Process finished with exit code 0


"C:\Program Files\Java\jdk-22\bin\java.exe" "-ja
***** Menú del restaurante: *****
***** 1. Spaghetti Carbonara *****
Escribe un número: 1
Opción valida: Spaghetti Carbonara
Process finished with exit code 0


"C:\Program Files\Java\jdk-22\bin\java.exe" "-java
***** Menú del restaurante: *****
***** 1. Spaghetti Carbonara *****
Escribe un número: -6
Opcion valida: -6
Process finished with exit code 0