**P5: Excepciones y herencia**

**La actividad es individual, pero os podéis ayudar sin copiar.**

Esta entrega se realizará mediante la plataforma Moodle y con el control de versiones en

[GitHub.](https://github.com/) En la entrega del Moodle, habrá que subir un **archivo .zip** con el siguiente nombre: ***DAW\_ POO\_Apellido1\_Apellido2\_Nombre.zip***. En el GitHub de la asignatura, añade los ejercicios de esta práctica.

Dentro del archivo comprimido, se deben tener las siguientes carpetas y archivos[[1]](#footnote-1):

* README.md
* EX\_01 o App.java
* EX\_03 o App.java
* EX\_04 o App.java
* EX\_05

|  |  |  |
| --- | --- | --- |
|  | **Curso**: CFGS Desarrollo de aplicaciones multiplataforma  **Módulo**: M03 Programación  **Nombre y apellidos: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Cualquier tipo de plagio o copia será penalizado siguiendo la normativa de la escuela.** | **Fecha**: 09/11/2022  **Tipo**: Actividad práctica    **P5: Excepciones y herencia** |

o README.md (con el enunciado y la solución) o App.java (programa, si es un ejercicio práctico)

El archivo README.md tiene que seguir el formato Markdown para explicar cuál ha sido la solución del ejercicio, las dificultades encontradas y una sección con las funcionalidades no conseguidas. Además, en esta práctica hace falta **responder a los ejercicios teóricos en el archivo README.md**.

1. Define una clase “NegativeNumberException” que herede de “Exception” y que contenga un constructor sin parámetros y un constructor que reciba como parámetro un String, de tal modo que ambos invoquen a los constructores de la clase “Exception” correspondientes.

**.**

1. ¿Qué tiene de especial la excepción del tipo “RuntimeException”?

**.**

|  |  |  |
| --- | --- | --- |
|  | **Curso**: CFGS Desarrollo de aplicaciones multiplataforma  **Módulo**: M03 Programación  **Nombre y apellidos: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Cualquier tipo de plagio o copia será penalizado siguiendo la normativa de la escuela.** | **Fecha**: 09/11/2022  **Tipo**: Actividad práctica    **P5: Excepciones y herencia** |

1. Escribe un código que genere y capture una excepción al sobrepasar el límite de un array. Añada el código a la tabla

|  |
| --- |
| **import** java.util.Scanner;  **public** **class** ArrayException {  **public** **static** **void** main(String[] args) {  Scanner sc = **new** Scanner(System.***in***);    **try** {  **int** Array[]= **new** **int**[5];    **for**(**int** i=0; i<6;i++) {    System.***out***.println("Numero:");    Array[i]= sc.nextInt();  }    }**catch**(ArrayIndexOutOfBoundsException e) {    System.***out***.println("No puedes insertar más datos");  }  }  } |
|  |
|  |

1. Crea un programa **con un bucle while** que pida al usuario un número entero y muestre el resultado del cuadrado. En caso de introducir un carácter o valor que no sea un número, captura la excepción y vuelva a pedirle el número hasta que sea correcto. Añada el código a la tabla

|  |
| --- |
| **import java.util.InputMismatchException;**  **import java.util.Scanner;**  **public class Ejercicio4Excepciones {**  **public static void main(String[] args) throws InputMismatchException{**  **int num=0;**  **Scanner sc= new Scanner(System.in);**  **boolean entradaOk=false;**  **while (entradaOk==false) {**  **try {**    **System.out.println("Introduce un numero:");**    **num= sc.nextInt();**    **entradaOk=true;**  **}catch(InputMismatchException e) {**  **System.out.println("No puedes poner caracteres especiales");**    **sc.next();**      **}**  **}**    **System.out.println("Resultado del cuadrado:" + (num\*num));**      **}** |

1. Propón y soluciona un ejercicio que trabaje las excepciones.

1. Se pueden añadir más archivos en caso de que se considere necesario. [↑](#footnote-ref-1)