

## *Ejercicios de introducción*

### *Programación en Java para la asignatura Algoritmos y estructuras de datos (AyEdD)*

*Gorka Guardiola Múzquiz*

*1 de Febrero de 2022*

Este documento contiene el primer ejercicio obligatorio de la asignatura.

Este ejercicio es, en parte para que configures e instales el sistema de entrega y prueba de prácticas.

En la introducción se describen aspectos importantes que deberás tener en cuenta en todos los ejercicios de la asignatura.

## *Introducción*

CUANDO REALICES TUS PROGRAMAS PUEDES AÑADIR MENSAJES DE DEPURACIÓN O TRAZAS. Estos mensajes pueden ayudarte a entender mejor tu programa y a depurarlo hasta eliminar todos sus errores. Cuando consigas que tu programa tenga un comportamiento correcto, tras eliminar todos sus errores, **debes eliminar de tu programa los mensajes de depuración o traza.**

DEBES MOSTRAR MENSAJES DE ERROR cuando compruebes alguna condición que no debería producirse.

Tanto los mensajes de depuración como los mensajes de error deben escribirse **por la salida de error y no por la salida estándar.** Por la salida estándar sólo se deben mostrar los mensajes que se espera que genere un programa correcto que respete la especificación del mismo.

Para escribir por la salida de error se puede utilizar `System.err.println()`, por ejemplo:

```
System.err.println("Error: el valor de x no puede ser negativo");
System.err.println("Taza: contenido de x: " + x);
```

ES IMPORTANTE QUE LOS PROGRAMAS QUE ESCRIBAS MANEJEN BIEN LOS ERRORES. Cuando un programa llega a la última sentencia (la que hay justo antes de la llave que termina el método `main()`), el programa **termina devolviendo al sistema operativo un estado correcto.**

Si un programa detecta cualquier situación errónea que le impide concluir su ejecución devolviendo un estado correcto, debe escribir un mensaje por la salida de error, y **terminar devolviendo un estado de fallo al sistema operativo.**

Para hacer que un programa termine devolviendo un estado de fallo al sistema operativo puedes usar:

```
System.exit(1);
```

CUANDO EJECUTES UN PROGRAMA, LA SALIDA DE ERROR Y LA SALIDA ESTÁNDAR ESTÁN AMBAS ASOCIADAS AL TERMINAL. Tanto una como otra pueden redirigirse a un fichero. Para entenderlo, estudia el código de este ejemplo:

```
import edu.princeton.cs.algs4.*;

public class PruebaError {
    public static void main(String[] args){
        int x = -3;
        StdOut.println("Programa PruebaError");
        System.err.println("Traza: contenido de x: " + x);
        if (x < 0) {
            System.err.println("Error: el valor de x no puede ser negativo");
            System.exit(1);
        }
    }
}
```

Compíllalo y ejecútalo de estas dos maneras:

1. % java -cp ../algs4.jar PruebaError

En este caso tanto la salida estándar como la de error están asociadas al terminal, por lo que en él se muestra tanto el texto de la sentencia `StdOut.println()` como el de `System.err.println()`:

```
Programa PruebaError
Traza: contenido de x: -3
Error: el valor de x no puede ser negativo
```

2. % java -cp ../algs4.jar PruebaError 2>/tmp/errores

En este otro caso se ha redirigido la salida de error (2) al fichero `/tmp/errores`, por lo que en la salida estándar asociada al terminal no sale el mensaje de depuración:

```
Programa PruebaError
```

Si muestras el contenido del fichero `/tmp/errores` verás que en él están el mensaje de depuración o traza y el de error.

```
% cat /tmp/errores
Traza: contenido de x: -3
Error: el valor de x no puede ser negativo
```

COMPILA SIEMPRE TUS PROGRAMAS CON `-encoding utf-8`, por ejemplo:

```
% javac -cp .:algs4.jar -encoding utf-8 SumaEnterosArray.java
```

Esto hará que si tu editor utiliza una codificación de caracteres extraña el compilador la detecte y no compile tu programa.

### *Ejercicio 1*

**MyEcho** Escribe un programa en Java llamado `MyEcho.java`. El programa escribe sus argumentos, separados por un tabulador, por pantalla. Si no recibe argumentos, escribe un error por la salida de error y sale con error.

```
%
% java -cp .:algs4.jar MyEcho hola
hola
% java -cp .:algs4.jar MyEcho hola adios
hola adios
% java -cp .:algs4.jar MyEcho
usage: myecho arg...
% java -cp .:algs4.jar MyEcho > /tmp/salida
usage: myecho arg...
```

Ojo, el programa debe escribir las cosas exactamente como se explican en el enunciado.

### **Pruebas:**

Antes de probar tu código tienes que haber clonado el repositorio del sistema de pruebas y entregas de ejercicios de la asignatura en un directorio llamado `practicas-ayedd-2022`. En clase te hemos enseñado cómo hacerlo.

Las pruebas te permitirán comprobar si tu código compila, y si al ejecutarlo produce los resultados esperados según la especificación del ejercicio.

Podrás probar tu código usando el programa `prueba.sh` y para probarlo y entregarlo en el servidor usando `rprueba.sh`. Al probarlo en el servidor se suben al servidor de la asignatura las diferentes versiones que has ido probando.

*No tienes que hacer una sola entrega. Te conviene hacer múltiples entregas de tu código ejecutando para ello `rprueba.sh` pues así se subirán versiones de tu código al servidor para que los profesores puedan ir evaluando tu progreso y ayudarte a resolver dudas.* Para realizar las pruebas tienes que seguir los siguientes pasos:

1. Copia el fichero `MyEcho.java` en el subdirectorio `src/main/java/ayedd/e1` del sistema de pruebas y entregas.

2. Ve al directorio directorio practicas-ayedd-2022 del sistema de pruebas y entregas que has clonado, y ejecuta el siguiente comando cada vez que quieras probar tu código:

```
./prueba.sh "ayedd.e1.TestMyEcho" "COMENTARIO"
```

En COMENTARIO pon un texto que identifique los cambios que has realizado al código que quieres probar.

Este comando compila tu código y lo prueba con los tests diseñados para evaluar tu ejercicio.

En la salida estándar podrás ver el resultado de la ejecución de dos tipos de tests: los que prueban métodos de tu código y los que ejecutan el programa y comprueban que la salida obtenida en el descriptor de fichero de la salida estándar es la correcta.

Todas las versiones de los ejercicios que vayas probando se irán guardando para ser evaluadas por los profesores de la asignatura.

**Entrega antes del 28/02/2021**

La entrega del ejercicio tiene dos pasos:

1. Para subir todas las versiones que has ido probando del código de este ejercicio tienes que ejecutar el siguiente comando desde el directorio raíz del sistema de pruebas y entregas:

```
./prueba.sh "ayedd.e1.TestMyEcho" "COMENTARIO"
```

Pon en COMENTARIO algún texto descriptivo sobre el estado del ejercicio que estás entregando. Ejemplo: "Versión que funciona para todos los casos salvo cuando hay acarreo"

Puedes realizar tantas pruebas en el servidor como quieras, si bien sólo se corregirán las versiones probadas antes de la fecha límite de entrega.

Tras ejecutar `rprueba.sh` ve a tu proyecto en el servidor <https://gitlab.etsit.urjc.es> para asegurarte de que tu código se ha subido al servidor: entra en la rama `rama-alumno` pulsa en History para ver si la última versión de tu código se ha subido al servidor correctamente.

2. Además tendrás que subir los ficheros que has subido con `rprueba.sh` en el sistema de pruebas y entregas a una tarea de entrega en Aula Virtual.