

EDA II – Diseño del programa principal con argumentos

Estructuras de datos y Algoritmos II (EDA II). 2º Grado en Ingeniería Informática. Universidad de Almería

Version curso.2022

Objetivos

- Describir en Java el diseño del programa principal (**main**) con argumentos en la llamada
- Mostrar ejemplos de llamada al programa principal en la Práctica 4

Método **main** con argumentos

Para la ejecución de los algoritmos, y con el objetivo de facilitar la ejecución de los distintos algoritmos implementados sobre los distintos archivos de datos **sin tener que modificar el código fuente**, se muestra cómo diseñar el método principal (**main**) para que permita recibir argumentos, separados por espacios en blanco. Para ello, se usará el parámetro **args** en la llamada a **main**, como en el siguiente ejemplo:

```
package org.ualeda2.mainWithArguments;

public class MainWithArguments {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("The number of arguments is: "+args.length);
        System.out.println("Your first argument is: "+args[0]);
        System.out.println("Your second argument is: "+args[1]);
    }

}
```

Ejecución con Eclipse

Para ejecutar en Eclipse, como siembre, sobre la clase Java, botón derecho, *Run as...*, *Java Application*:

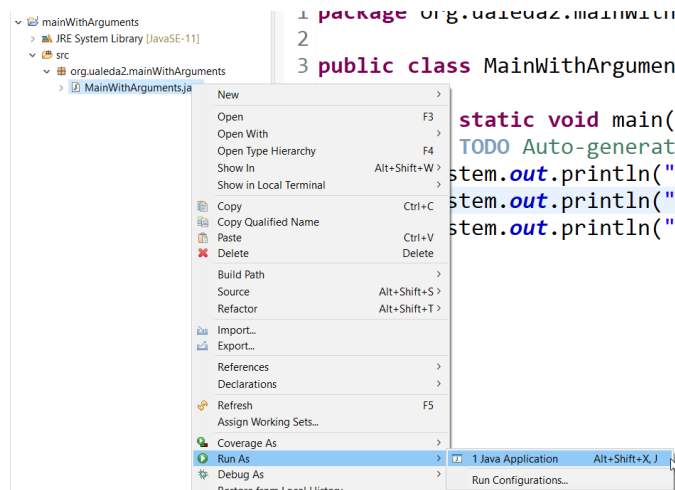


Figura 1. Run as... Java Application

La salida por consola mostrará la excepción `ArrayIndexOutOfBoundsException` cuando intenta acceder a `args[0]`, porque no le hemos puesto argumentos.

```
The number of arguments is: 0
<strong>Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 0
out of bounds for length 0
    at
    org.ualeda2.mainWithArguments.MainWithArguments.main(MainWithArguments.java:8)</strong>
>
```

Para añadir argumentos, menú *Run*, *Run Configurations*. A continuación, buscamos la configuración de ejecución de `MainWithArguments`, y editamos la pestaña *Arguments*:

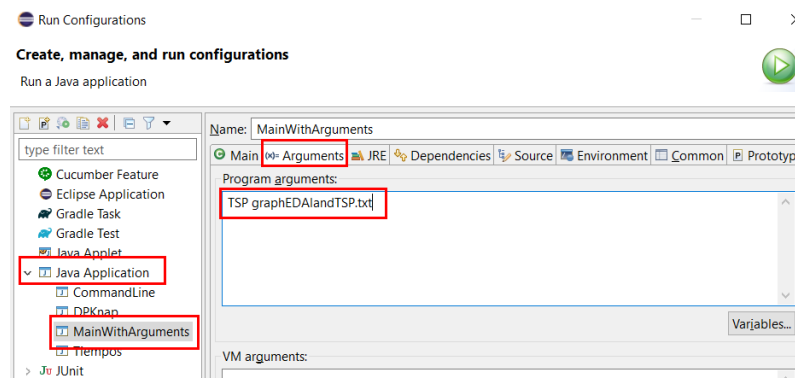


Figura 2. Run Configurations with Arguments

Volvemos a ejecutar, y ahora la ejecución será correcta.



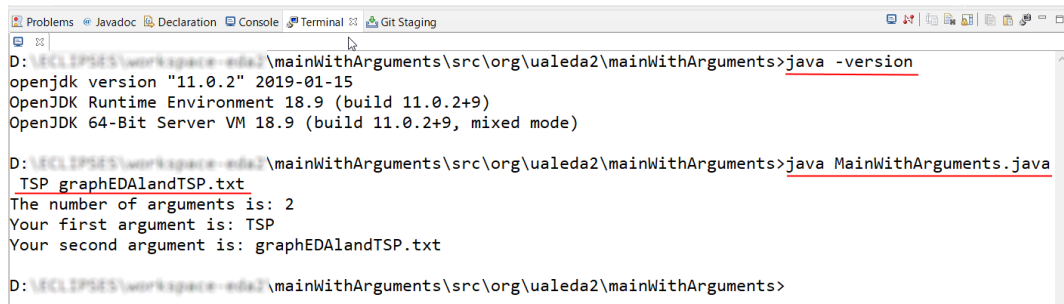
Figura 3. Salida por consola correcta

Ejecución desde el terminal

Otra alternativa para ejecutar nuestra aplicación con argumentos es llamarla desde el terminal. Abre el terminal como una vista dentro de Eclipse: sobre la clase Java, botón derecho, *Show in local terminal*, *Terminal*.

Y en el terminal llama a tu programa con:

```
java MainWithArguments.java TSP graphEDALandTSP.txt
```



```
D:\ECLIPSES\workspace-idea\mainWithArguments\src\org\ualeda2\mainWithArguments>java -version
openjdk version "11.0.2" 2019-01-15
OpenJDK Runtime Environment 18.9 (build 11.0.2+9)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.2+9, mixed mode)

D:\ECLIPSES\workspace-idea\mainWithArguments\src\org\ualeda2\mainWithArguments>java MainWithArguments.java
TSP_graphEDALandTSP.txt
The number of arguments is: 2
Your first argument is: TSP
Your second argument is: graphEDALandTSP.txt
D:\ECLIPSES\workspace-idea\mainWithArguments\src\org\ualeda2\mainWithArguments>
```

Figura 4. Llamada al programa Java desde consola con argumentos

Esta es la opción recomendada, ya que te permite escribir los argumentos facilmente, y usar las flechas arriba y abajo para repetir comandos previos.

Argumentos en Práctica 4

Los argumentos que debe permitir en la llamada al programa principal son:

1. Primer argumento: **algoritmo** a ejecutar (ejemplos: **TSP_distance**, **TSP_fuel**, **HamiltonianCycles**, **Backtraking**, **BandB**, ¿otros?...)
2. Segundo argumento: **carpeta/archivoDeDatos.txt** (es recomendable que acepte *rutas relativas, no absolutas*)
3. Tercer argumento (*opcional*): **vérticeDePartida**

Ejemplos de llamadas en la práctica 4

1. **Caso 1** del Trabajo a desarrollar: todos los posibles circuitos partiendo desde Almería, que recorren cada ciudad de la nueva red reducida de carreteras de EDALand. Argumentos:

```
HamiltonianCycles dataset/graphEDALandTSP.txt Almeria
```

2. **Caso 2** del Trabajo a desarrollar: Circuito que, partiendo desde Almería, visite cada ciudad de la nueva red reducida de carreteras de EDALand exactamente una vez, regresando a Almería y habiendo recorrido en total la menor distancia posible. Argumentos:

```
TSP_distance dataset/graphEDALandTSP.txt Almeria
```

3. **Caso 3** del Trabajo a desarrollar: Circuito que, partiendo de Almería, recorra cada ciudad exactamente una vez, regrese a Almería y cuyo gasto en combustible sea el mínimo posible, en la nueva red reducida de carreteras de EDALand. Para este caso, cada grupo ha debido generar un nuevo archivo de datos `graphEDALandTSPFuels.txt` con los valores aleratorios (reales) tal y como se indica en el enunciado de la práctica. Argumentos:

```
TSP_fuel dataset/graphEDALandTSPFuels.txt Almeria
```

4. **Caso 4** del Trabajo a desarrollar: Circuito en la red nacional de carreteras completa, partiendo de un núcleo urbano cualquiera, que visite cada población exactamente una vez, regrese al núcleo de partida y tenga la menor distancia posible. Argumentos (ejemplo):

```
algoritmo dataset/graphEDALandLarge.txt numVertice
```