

8.C. Actividades propuestas.

Sitio: [Formación Profesional a Distancia](#)

Curso: Programación

Libro: 8.C. Actividades propuestas.

Imprimido por: Iván Jiménez Utiel

Día: lunes, 10 de febrero de 2020, 16:00

Tabla de contenidos

[1. ActividadUT08-1: Calculadora.](#)

[2. ActividadUT08-2: Calculadora2.](#)

[3. ActividadUT08-3: Calculadora3.](#)

[4. ActividadUT08-4: Throw ArithmeticException.](#)

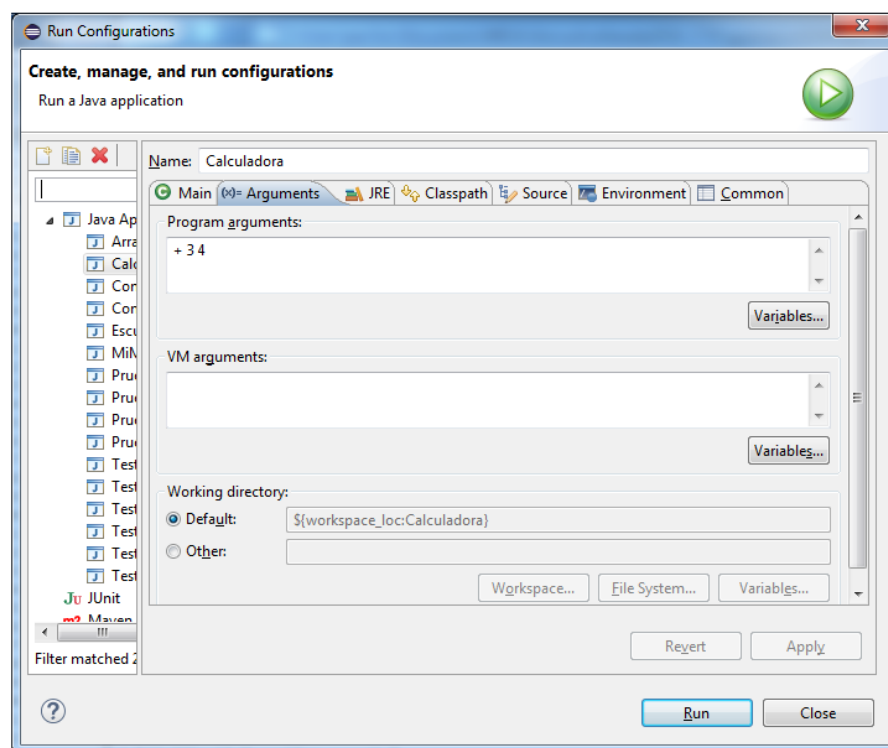
[5. ActividadUT08-5: Excepción de usuario NotaMal.](#)

1. ActividadUT08-1: Calculadora.

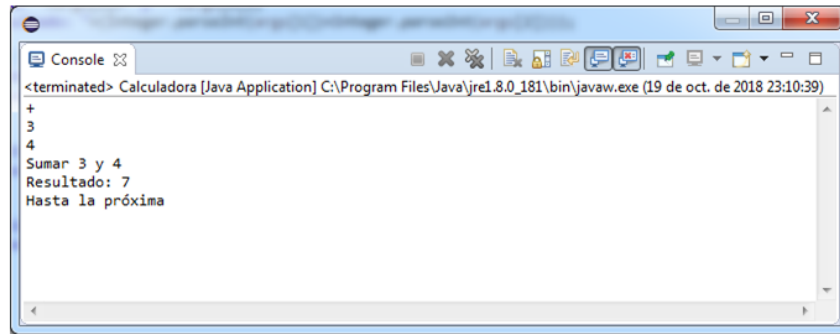
Para hacer esta actividad propuesta, es necesario que conozcas la manera de incorporar parámetros de entrada al **método main()**. Para ver como se hace, te sugiero que eches un vistazo al siguiente vídeo:

Una vez tengas claro la manera de incorporar parámetros, debes hacer una calculadora simple que utilice parámetros del **main()**, concretamente el operador (Será "+", "-", "*" o "/"), el primer operando (número entero) y el segundo operando (número entero).

Por ejemplo, si incorporamos los argumentos del **main()** de la forma:



Al dar a "Run", aparecerá:



Teniendo en cuenta lo anterior, debemos contemplar en nuestro programa un control de excepciones que considere los casos de:

- Número de parámetros incorrecto (debe ser tres).
- División por 0.
- Que alguno de los operandos no sea un número entero.

Además, haremos que se produzca excepción o no, siempre se despida, tal y como se aprecia en la imagen anterior.

2. ActividadUT08-2: Calculadora2.

Rehacer la actividad anterior, pero lanzando, en caso de [excepción](#), la instanciación de unas clases definidas por usuario. Concretamente, el usuario definirá una [clase](#) **ErrorNumeroArgumentos** (que se usará cuando el número de argumentos no sea exactamente 3). El constructor de dicha [clase](#) recibirá como parámetro el mensaje relativo al número incorrecto de parámetros. Además, crearemos una [excepción](#) de usuario que contemple en caso en que para la división, en dividendo y el divisor sean el mismo número. En realidad, dicho caso no es una [excepción](#) aritmética, sin embargo, nosotros lo trataremos como si lo fuera, mostrando el mensaje "El resultado de dividir es la unidad", en lugar del "1" que correspondería al cálculo a realizar.

3. ActividadUT08-3: Calculadora3.

Por último, modificar el ejercicio anterior no considerando la [excepción](#) **NumberFormatException** (podemos comentarla), poniendo en su lugar la [excepción](#) general **Exception**, y comprobando como el orden en que se sitúe con respecto a las otras excepciones más particulares, es importante.

4. ActividadUT08-4: Throw ArithmeticException.

Realizar una [clase](#) dirigida a calcular el cociente de dos números de tipo double. Los datos serán leídos a través de la [clase](#) **BufferedReader**, con el [método](#) **readLine()**, el cuál devolverá una cadena que parsearemos a double.

La forma de instanciar el [objeto](#) **BufferedReader** será:

```
BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
```

Cuando se produzca una **ArithmeticException** debido a la división por 0, lanzaremos explícitamente la **ArithmeticException** mediante la palabra [clave](#) **Throw**.

5. ActividadUT08-5: Excepción de usuario NotaMal.

Crear una aplicación que permita leer una serie de notas y hacer su media. Se incorporarán notas indefinidamente hasta escribir una "z". Tras ello se calculará la nota media de las notas incorporadas que hayan sido validadas.

Habrá que considerar las excepciones de tipo **NumberFormatException**, y además, crearemos una [excepción](#) de usuario que contemple los casos en que metamos una nota que no esté comprendida entre 0 y 10. En este caso, mostrará un mensaje indicando el rango válido de notas.

Nota: usaremos para leer las notas el [método](#) **readLine()** de la [clase](#) **BufferedReader** (la cual devuelve una cadena, por lo que tendremos que aplicar un cast para convertir a número) de la forma:

```
BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
```

```
notaCad = entrada.readLine()
```