

package **initialpackage**;  
import **java.io.**\*;  
import **java.util.**\*;  
  
public class **LanzadorDeComandos** {  
 public static void lanzador(int n1, int n2) {  
 try {  
 **String** javaHome = **System**.*getProperty*("java.home");  
 **String** javaBin = javaHome + **File**.*separator* + "bin" + **File**.*separator* + "java";  
 **String** classpath = **System**.*getProperty*("java.class.path");  
 **String** clase = "initialpackage.Sumador";  
 **List**<**String**> command = new ArrayList<>();  
 command.add(javaBin);  
 command.add("-cp");  
 command.add(classpath);  
 command.add(clase);  
 command.add(**String**.*valueOf*(n1));  
 command.add(**String**.*valueOf*(n2));  
  
 **ProcessBuilder** builder = new ProcessBuilder(command);  
 //builder.start();  
 builder.inheritIO().start();  
 } catch (**Exception** e) {  
 e.printStackTrace();  
 }  
  
 }  
 public static void main (**String**[] args) {  
 *lanzador*(1, 50);  
 *lanzador*(51, 100);  
 }  
}

DADO ESTE FRAGMENTO DE CÓDIGO, DESGLOSAMOS LO QUE HACE CADA INSTRUCCIÓN, LÍNEA POR LÍNEA.

Para empezar, este fragmento de código, hace una llamada a la clase Sumador, la cual requiere dos parámetros, es por eso, que a su vez el método public static void lanzador(int n1, int n2)también requiere de 2 parámetros, que posteriormente, al declarar el ProcessBuilder e iniciarlo, va a llamar a la clase Sumador.

CONTENIDO DE LA CLASE SUMADOR

package **initialpackage**;  
  
public class **Sumador** {  
 public static int sumar(int n1, int n2) {  
 **System**.*out*.println("Sumando de "+n1+" hasta "+n2);  
 int resultado=0;  
 for (int i=n1; i<=n2;i++) {  
 resultado+=i;  
 }  
 return resultado;  
 }  
  
 public static void main (**String**[] args) {  
 int n1 = **Integer**.*parseInt*(args[0]);  
 int n2 = **Integer**.*parseInt*(args[1]);  
 int resultado=*sumar*(n1,n2);  
  
 **System**.*out*.println("La suma del intervalo de estos dos números es: "+resultado);  
  
 }  
}

volviendo a la clase LanzadorDeComandos, explicamos lo que hace cada instrucción:

public static void lanzador(int n1, int n2) {  
 try {  
 **String** javaHome = **System**.*getProperty*("java.home");  
 **String** javaBin = javaHome + **File**.*separator* + "bin" + **File**.*separator* + "java";  
 **String** classpath = **System**.*getProperty*("java.class.path");  
 **String** clase = "initialpackage.Sumador";  
 **List**<**String**> command = new ArrayList<>();  
 command.add(javaBin);  
 command.add("-cp");  
 command.add(classpath);  
 command.add(clase);  
 command.add(**String**.*valueOf*(n1));  
 command.add(**String**.*valueOf*(n2));  
  
 **ProcessBuilder** builder = new ProcessBuilder(command);  
 //builder.start();  
 builder.inheritIO().start();  
 } catch (**Exception** e) {  
 e.printStackTrace();  
 }  
  
}

String javaHome 🡪 obtiene el directorio donde esta instalado java.

String javaBin 🡪 construye la ruta completa al ejecutable java



Aquí podemos ver lo que contiene cada variable. Cabe recalcar que javaBin es la suma de javaHome + bin + java.

File.separator 🡪 añade una contrabarra (\).



Classpath 🡪 obtiene la ruta donde java va a buscar las clases para ejecutar

Clase 🡪 indica la clase en la que se va a ejecutar el nuevo proceso Java (Sumador dentro del paquete initialpackage).