

# Librerías



# Paquetes

Un Paquete en Java es un contenedor de clases que permite agrupar las distintas partes de un programa y que por lo general tiene una funcionalidad y elementos comunes, definiendo la ubicación de dichas clases en un directorio de estructura jerárquica.



# Paquetes - Ejemplo



matematicas



Operacion.java



principales



Principal.java

# Paquetes - Ejemplo

```
package matematicas;  
public class Operacion{  
    public int sumar(int a, int b){  
        return a + b;  
    }  
  
    public int restar(int a, int b){  
        return a - b;  
    }  
}
```



# Paquetes - Ejemplo

```
package principales;
import matematicas.*;
public class Principal{
    public static void main(String[] args){
        Operacion op = new Operacion();
        int x = op.sumar(10,90);
        int y = op.restar(10,90);
        System.out.println(x);
        System.out.println(y);
    }
}
```



# Paquetes - Ejemplo

Se compila empezando por las clases menos dependientes.



```
Símbolo del sistema

C:\LP2>javac matematicas/*.java

C:\LP2>javac principales/*.java

C:\LP2>java principales/Principal
100
-80

C:\LP2>
```

# Librerías



Una librería es un programa cuyos elementos pueden ser utilizados por otros programas.

La forma de crear y utilizar una librería dependen del lenguaje de programación o del sistema operativo.

# Librerías en JAVA

## Definiciones

Están estrechamente relacionadas a los directorios.

Una librería es un directorio con un conjunto de archivos CLASS que forman parte de la librería.

Para encontrar una librería, tanto el compilador como el intérprete utilizan la variable de entorno CLASSPATH



# Librerías en JAVA - Creación

1. Generar los archivos CLASS que conformarán la librería.
2. Empaquetar los archivos CLASS mediante el siguiente comando:

```
jar cvf "nombre del paquete" [clases]
```

Ejemplo:

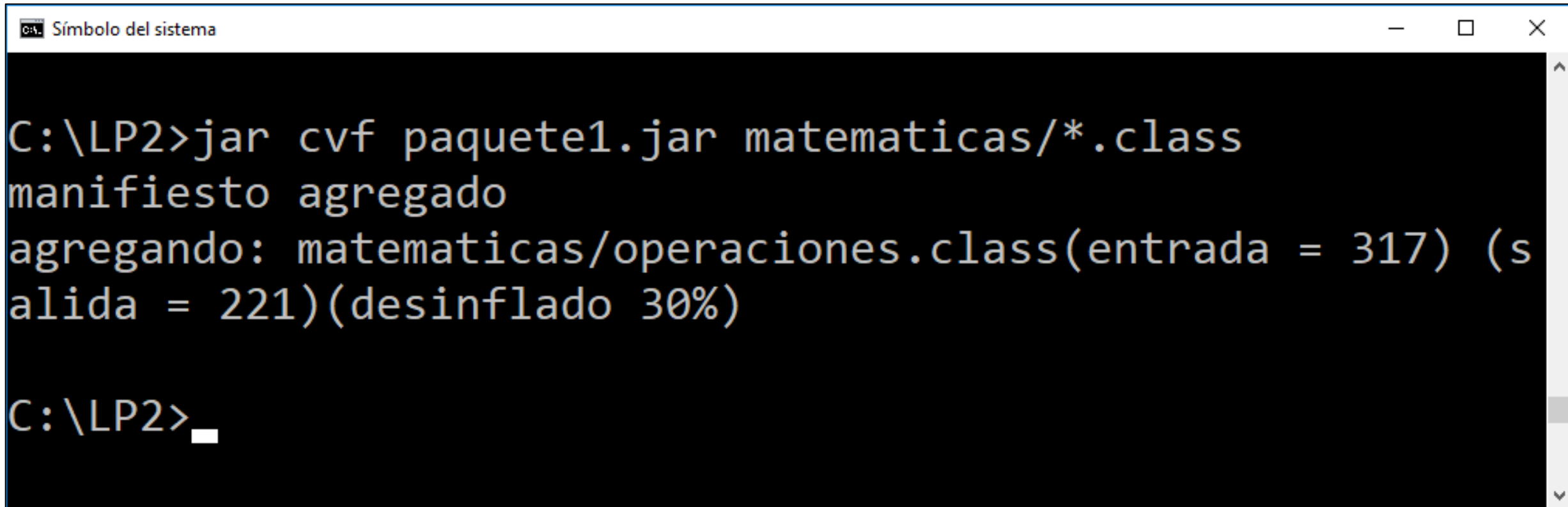
```
jar cvf Paq1.jar Foo.class Bar.class
```

## *Archivos JAR*

1. Comprimen uno o más paquetes conservando la información acerca de los directorios (paquetes) donde se encuentran cada archivo .CLASS.
2. Pueden ser utilizados en CLASSPATH como lugares de búsqueda de archivos .CLASS.
3. Constituyen una mejor forma distribuir un conjunto de archivos .CLASS, como los que forman una aplicación, dado que se tiene la opción de comprimir estos.

# Librerías en JAVA Siguiendo el ejemplo

Generación de una librería del paquete "matematicas".



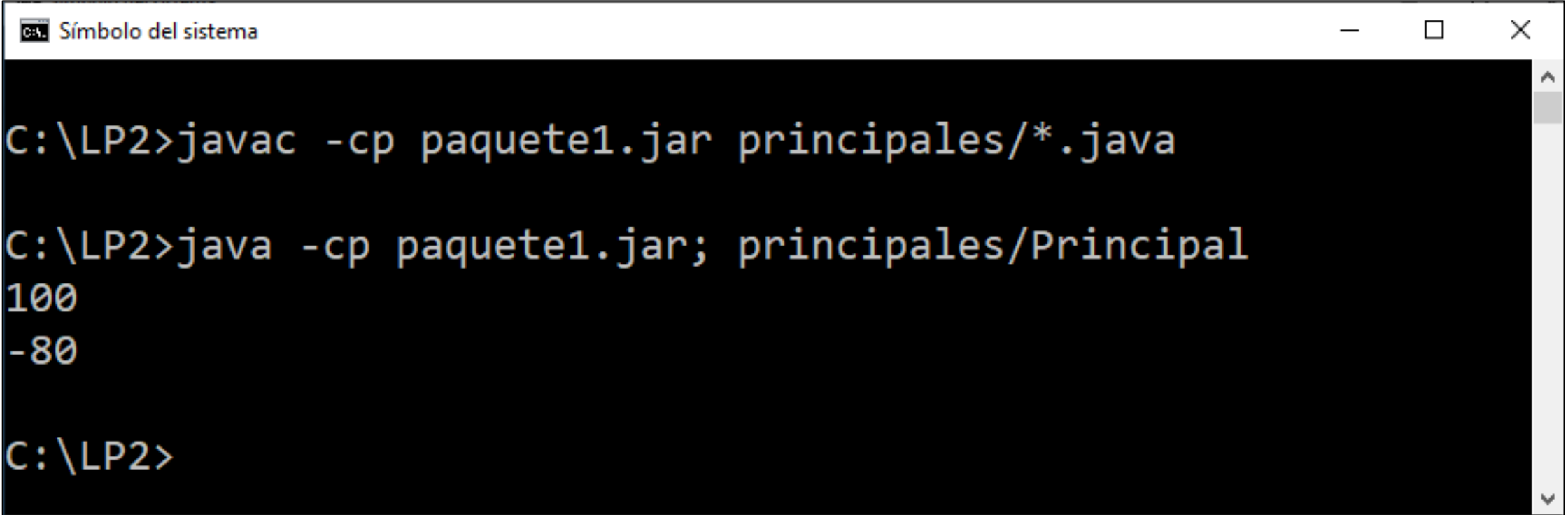
```
C:\LP2>jar cvf paquete1.jar matematicas/*.class
manifiesto agregado
agregando: matematicas/operaciones.class(entrada = 317) (s
alida = 221)(desinflado 30%)

C:\LP2>
```

# Librerías en JAVA Siguiendo el ejemplo

Compilando utilizando la librería "matematicas".

Ejecutando el programa principal utilizando la librería "matematicas".



```
C:\LP2>javac -cp paquete1.jar principales/*.java

C:\LP2>java -cp paquete1.jar; principales/Principal
100
-80

C:\LP2>
```

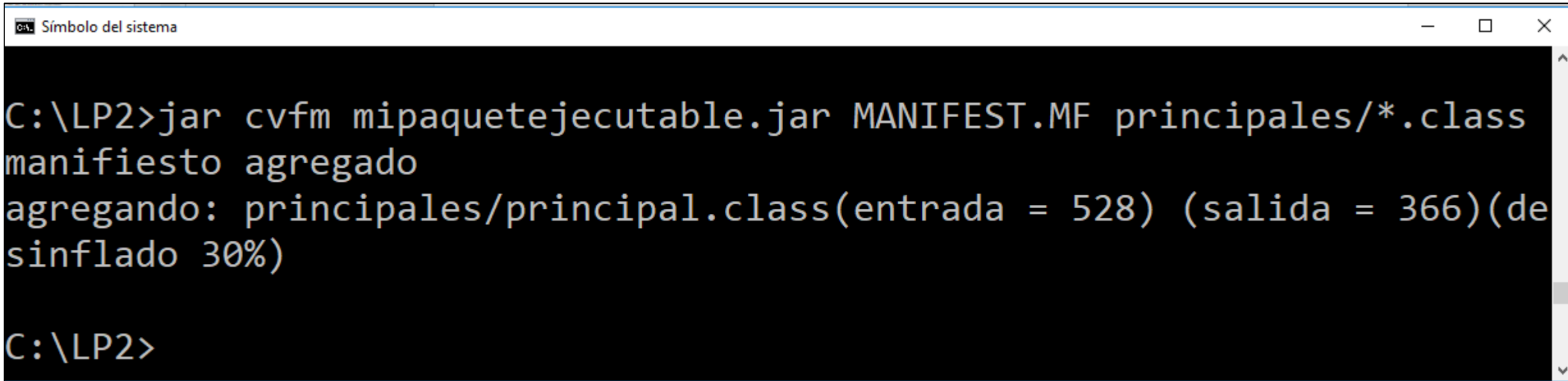
# Librerías en JAVA – JAR Ejecutable

- Es necesario modificar el Manifiesto e indicar en este archivo las librerías que usa el programa en la variable CLASSPATH y la clase principal.

```
1 Manifest-Version: 1.0
2 Created-By: 1.8.0_92 (Oracle Corporation)
3 Class-Path: paquete1.jar
4 Main-Class: principales/Principal
5
6
7
8
9
```

# Librerías en JAVA – JAR Ejecutable

- Creamos el paquete ejecutable.

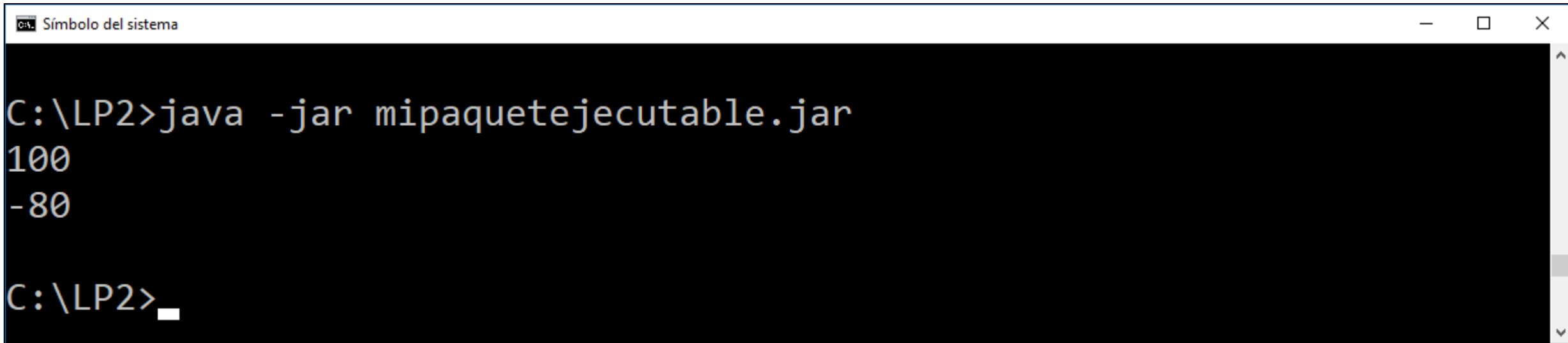


```
C:\LP2>jar cvfm mipaquetejexecutable.jar MANIFEST.MF principales/*.class
manifiesto agregado
agregando: principales/principal.class(entrada = 528) (salida = 366)(de
sinflado 30%)

C:\LP2>
```

# Librerías en JAVA – JAR Ejecutable

- Ejecutamos el paquete.





```
Símbolo del sistema

C:\LP2>java -jar mipaquetejexecutable.jar
100
-80

C:\LP2>
```

# Espacios de Nombres en C#

Proporcionan los medios para agrupar lógicamente las clases relacionadas en sus correspondientes espacios de nombres, haciendo así el sistema más modular.

 matematicas  
 Operacion.cs

 principales  
 Principal.cs



# Espacios de Nombres en C#

```
namespace matematicas{  
    public class Operacion{  
        public int sumar(int a, int b){  
            return a + b;  
        }  
        public int restar(int a, int b){  
            return a - b;  
        }  
    }  
}
```

C#

# Espacios de Nombres en C#

```
using matematicas;
namespace principales{
    public class Principal{
        public static void Main(string[] args){
            Operacion op = new Operacion();
            int x = op.sumar(10,80);
            int y = op.restar(10,80);
            System.Console.WriteLine(x+" "+y);
        }
    }
}
```

C#

# Espacios de Nombres en C#

Se compila empezando por las clases menos dependientes.

```
Símbolo del sistema


C:\LP2>csc matematicas\*.cs principales\*.cs
Microsoft (R) Visual C# Compiler version 4.7.2046.0
for C# 5
Copyright (C) Microsoft Corporation. All rights reserved.

This compiler is provided as part of the Microsoft (R) .NET Framework, but only
y supports language versions up to C# 5, which is no longer the latest version
. For compilers that support newer versions of the C# programming language, se
e http://go.microsoft.com/fwlink/?LinkID=533240

C:\LP2>
```

# Espacios de Nombres en C#

Ejecutamos.



```
Símbolo del sistema

C:\LP2>Principal
90 -70

C:\LP2>
```

# Librerías en C# (DLLs)

## Problemas encontrados en la utilización de dll's:

- Si al desinstalar un programa, este elimina una DLL que es utilizada por otro programa, este último dejará de funcionar correctamente.
- Si un usuario cambia de posición el archivo de la DLL utilizada por un programa, este quizá no lo encuentre, por lo que dejará de funcionar correctamente.

# Librerías en C# (DLLs)

- Ejemplo1: Creación de un archivo .exe

```
csc Foo.cs Bar.cs
```

- Ejemplo2: Creación de un archivo .dll

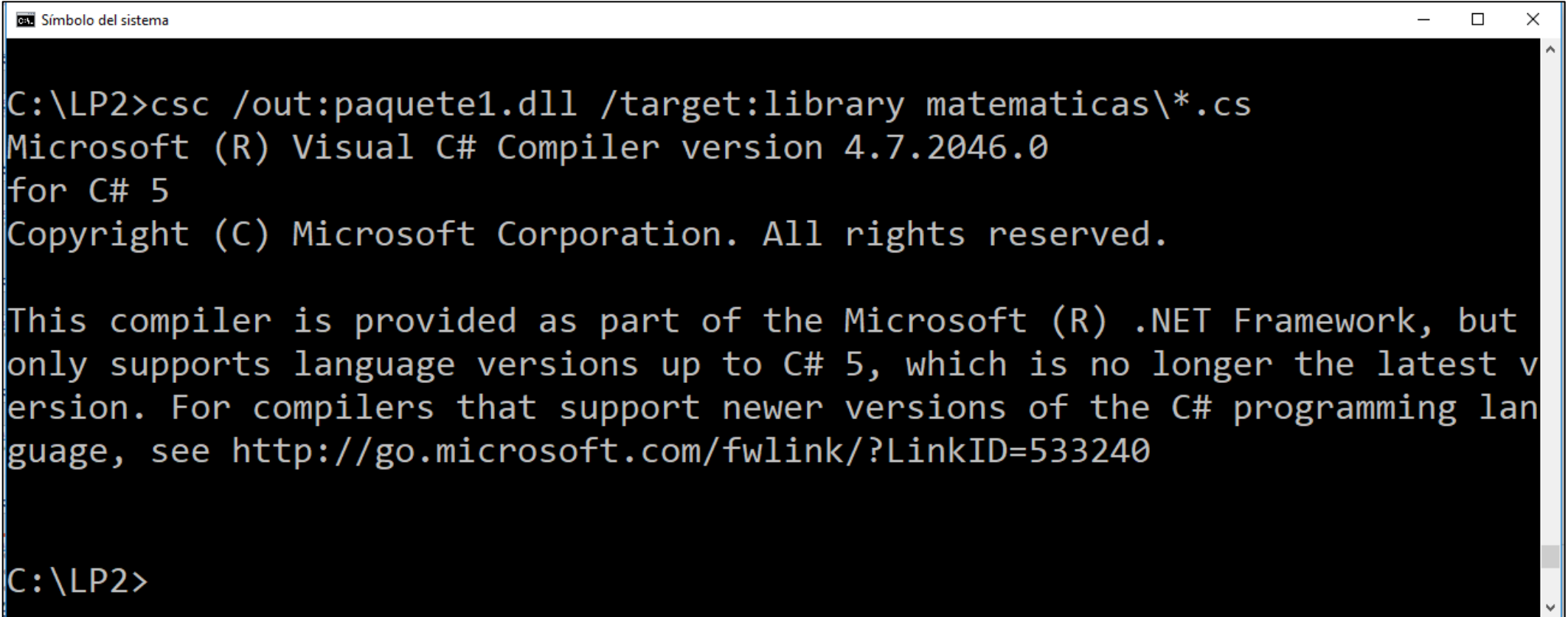
```
csc /target:library Foo.cs Bar.cs
```

- Ejemplo3: Creación de un archivo con nombre .dll

```
csc /out:archivo.dll /target:library  
Foo.cs Bar.cs
```

# Librerías en C# (DLLs)

## Creando una librería del paquete matematicas:



```
C:\LP2>csc /out:paquete1.dll /target:library matematicas\*.cs
Microsoft (R) Visual C# Compiler version 4.7.2046.0
for C# 5
Copyright (C) Microsoft Corporation. All rights reserved.

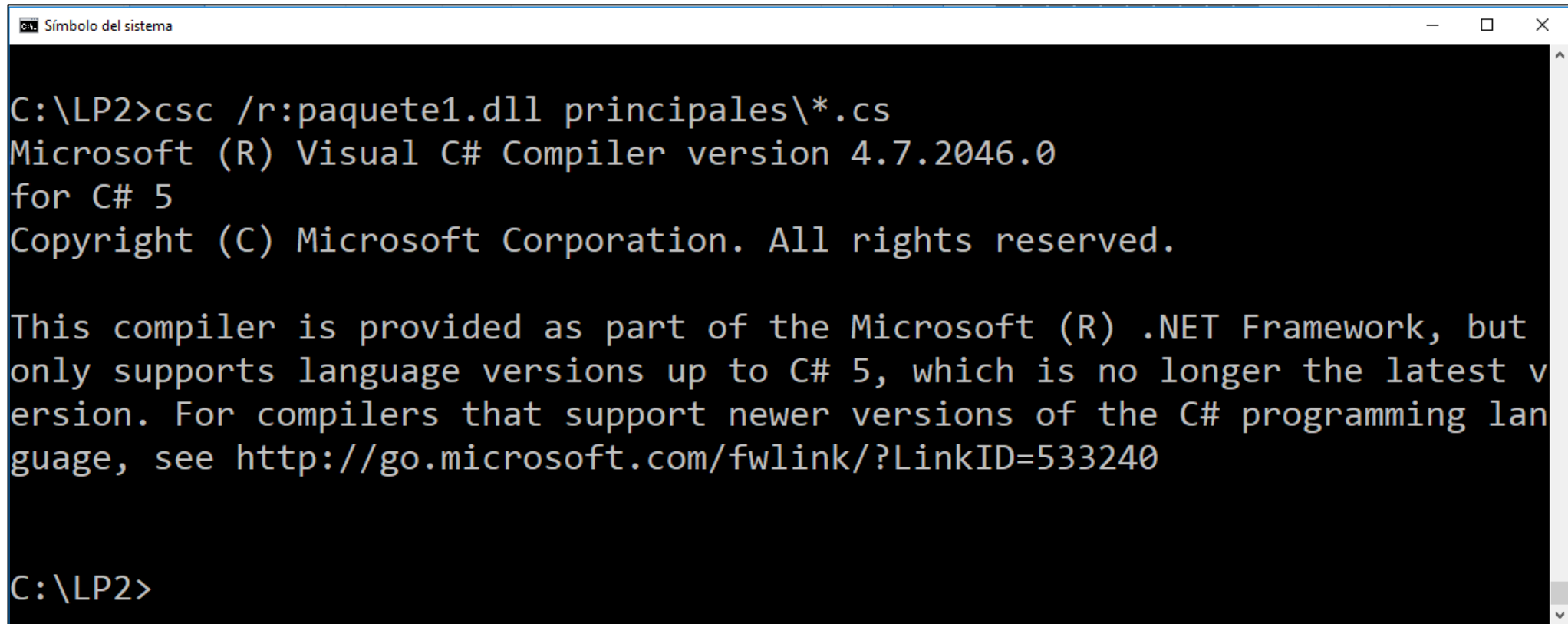
This compiler is provided as part of the Microsoft (R) .NET Framework, but
only supports language versions up to C# 5, which is no longer the latest v
ersion. For compilers that support newer versions of the C# programming lan
guage, see http://go.microsoft.com/fwlink/?LinkID=533240

C:\LP2>
```

# Librerías en C# (DLLs)

Compilando utilizando la librería "matematicas".

Ejecutando el programa principal utilizando la librería "matematicas".



```
C:\LP2>csc /r:paquete1.dll principales\*.cs
Microsoft (R) Visual C# Compiler version 4.7.2046.0
for C# 5
Copyright (C) Microsoft Corporation. All rights reserved.


This compiler is provided as part of the Microsoft (R) .NET Framework, but
only supports language versions up to C# 5, which is no longer the latest v
ersion. For compilers that support newer versions of the C# programming lan
guage, see http://go.microsoft.com/fwlink/?LinkId=533240

C:\LP2>
```



# Librerías en C# (DLLs)

Ejecutamos.



```
Símbolo del sistema

C:\LP2>Principal
90 -70

C:\LP2>
```

# Referencias

- D.J. Barnes y M. Kölling, Programación orientada a objetos con Java. Pearson Educación, 2007
- P. Deitel y Harvey, Visual C# 2012 How to Program, 2013.
- J. Sharp, Microsoft Visual C# 2013 Step by Step. Microsoft Press, 2013
- T. Budd, An introduction to Object-Oriented Programming (Third Edition). Pearson Education, 2001
- E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994
- Agustín Froufe. Java 2. Manual de usuario y tutorial. Ed. Ra-Ma
- J. Sánchez, G. Huecas, B. Fernández y P. Moreno, Iniciación y referencia: Java 2. Osborne McGraw-Hill, 2001.