

---

HAGA CLIC PARA MODIFICAR EL ESTILO DE TÍTULO DEL PATRÓN

# Bytecodes y Compilación y Ejecución de Programas desde línea de Comandos.

## BYTECODES

El compilador de Java traduce el código fuente de Java a Bytecodes. La máquina virtual de Java ejecuta bytecodes.

Las instrucciones bytecode son independientes de la plataforma. Las mismas instrucciones pueden ejecutarse en cualquier plataforma que contenga una máquina virtual de java que “entienda” la versión de Java en la cual los bytecodes fueron compilados.

Los archivos .class generados por al compilarse el archivo .java contienen los bytecodes que serán ejecutados por el compilador.

# COMPILACIÓN Y EJECUCIÓN DE PROGRAMAS

- Los programas en java pueden compilarse desde línea de comandos en Windows, para ejemplo se creará el siguiente programa. Ejecútelo en Netbeans.

```
package javaapplication20;

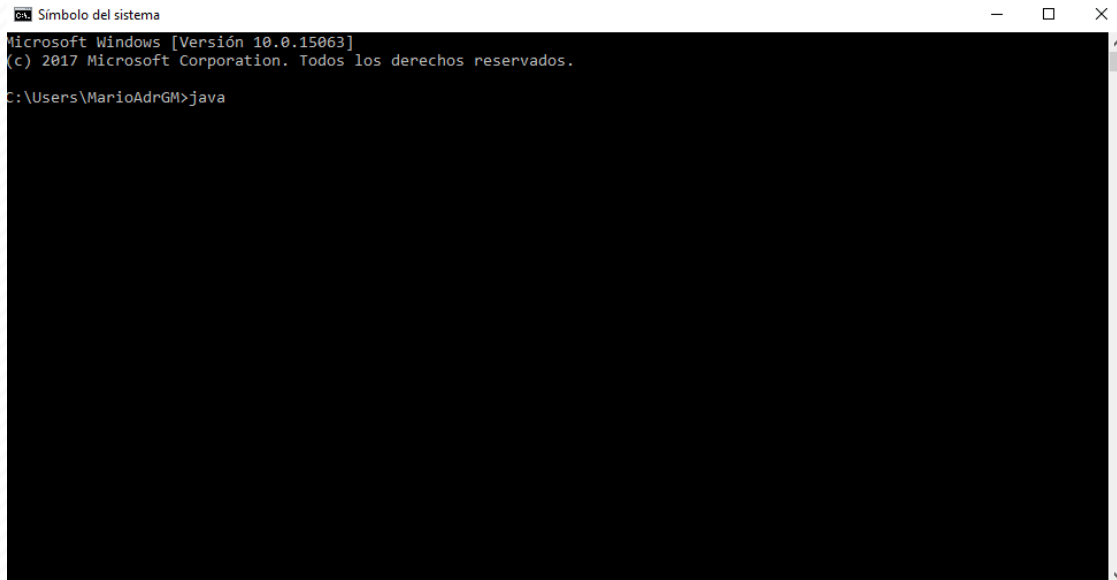
public class JavaApplication20 {

    public static void main(String[] args) {
        // TODO code application logic here
        for (int i = 0; i < 65; i++){
            System.out.println(i);
        }
    }

}
```

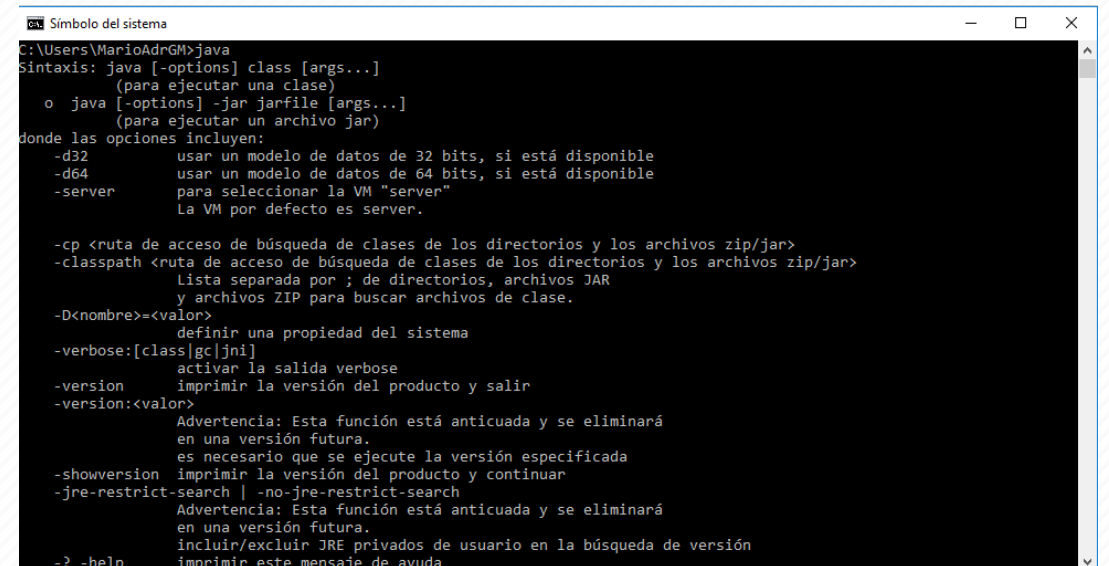
# COMPILACIÓN Y EJECUCIÓN DE PROGRAMAS

Abra la terminal, clic en el símbolo de Windows y teclee cmd



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.15063]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.
C:\Users\MarioAdrGM>java
```

Para probar escriba java en la terminal y presione enter, si muestra un resultado parecido a la imagen de la derecha se puede continuar. En caso contrario es posible que sea necesario configurar las variables de entorno.

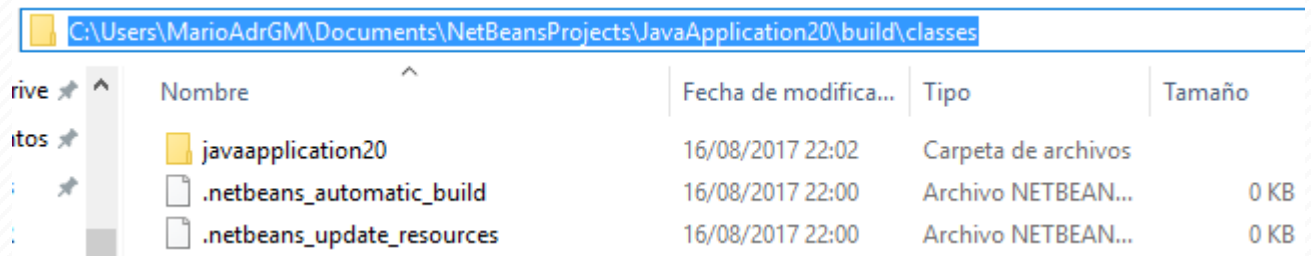


```
Símbolo del sistema
C:\Users\MarioAdrGM>java
Síntaxis: java [-options] class [args...]
           (para ejecutar una clase)
    o  java [-options] -jar jarfile [args...]
       (para ejecutar un archivo jar)
donde las opciones incluyen:
  -d32      usar un modelo de datos de 32 bits, si está disponible
  -d64      usar un modelo de datos de 64 bits, si está disponible
  -server   para seleccionar la VM "server"
            La VM por defecto es server.

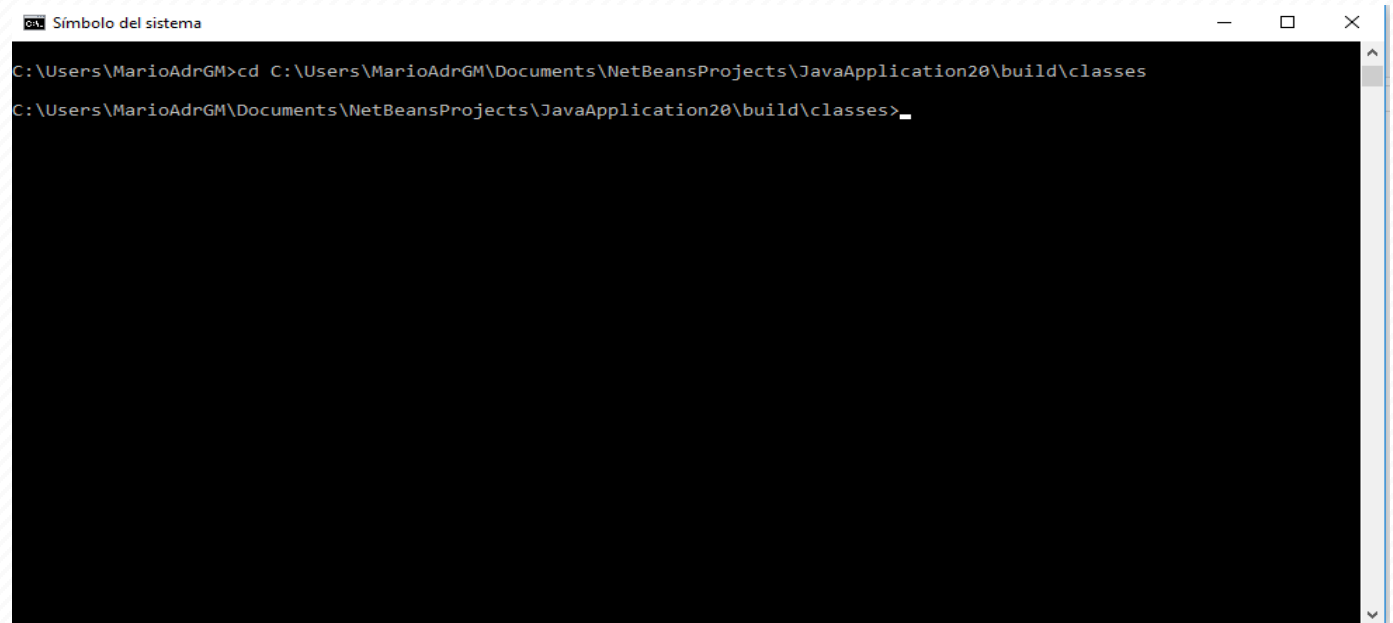
  -cp <ruta de acceso de búsqueda de clases de los directorios y los archivos zip/jar>
  -classpath <ruta de acceso de búsqueda de clases de los directorios y los archivos zip/jar>
            Lista separada por ; de directorios, archivos JAR
            y archivos ZIP para buscar archivos de clase.
  -D<nombre>=<valor>
            definir una propiedad del sistema
  -verbose:[class|gc|jni]
            activar la salida verbose
  -version   imprimir la versión del producto y salir
  -version:<valor>
            Advertencia: Esta función está anticuada y se eliminará
            en una versión futura.
            es necesario que se ejecute la versión especificada
  -showversion
            imprimir la versión del producto y continuar
  -jre-restrict-search | -no-jre-restrict-search
            Advertencia: Esta función está anticuada y se eliminará
            en una versión futura.
            incluir/excluir JRE privados de usuario en la búsqueda de versión
  -? -help  imprimir este mensaje de ayuda
```

# COMPILACIÓN Y EJECUCIÓN DE PROGRAMAS

Busca la ruta de las clases del proyecto, ve a la carpeta del proyecto→Build→classes y copia la ruta al portapapeles.

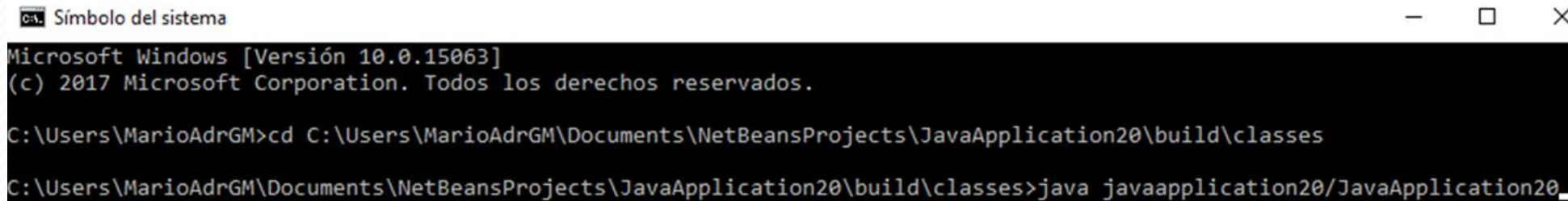


Dirígete a la ruta copiada con  
cd "Ruta a donde ir"



# COMPILACIÓN Y EJECUCIÓN DE PROGRAMAS

Para correr el programa ejecute la línea `java package/archivo.class`, en este caso sería `java javaapplication20/JavaApplication20`



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.15063]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\MarioAdrGM>cd C:\Users\MarioAdrGM\Documents\NetBeansProjects\JavaApplication20\build\classes

C:\Users\MarioAdrGM\Documents\NetBeansProjects\JavaApplication20\build\classes>java javaapplication20/JavaApplication20_
```

Salida del programa.

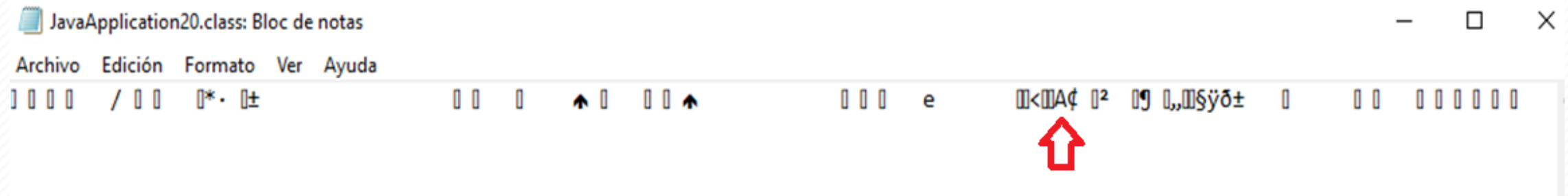


```
Símbolo del sistema
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64

C:\Users\MarioAdrGM\Documents\NetBeansProjects\JavaApplication20\build\classes>
```

# COMPILACIÓN Y EJECUCIÓN DE PROGRAMAS

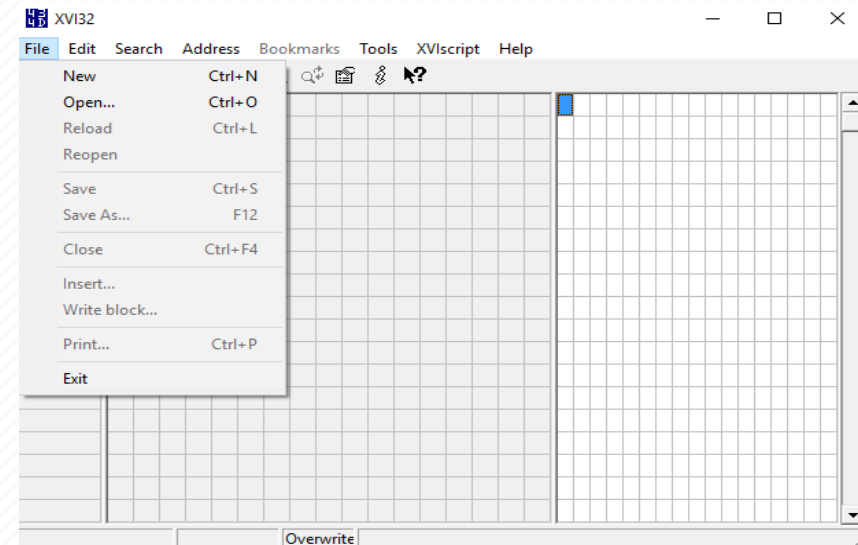
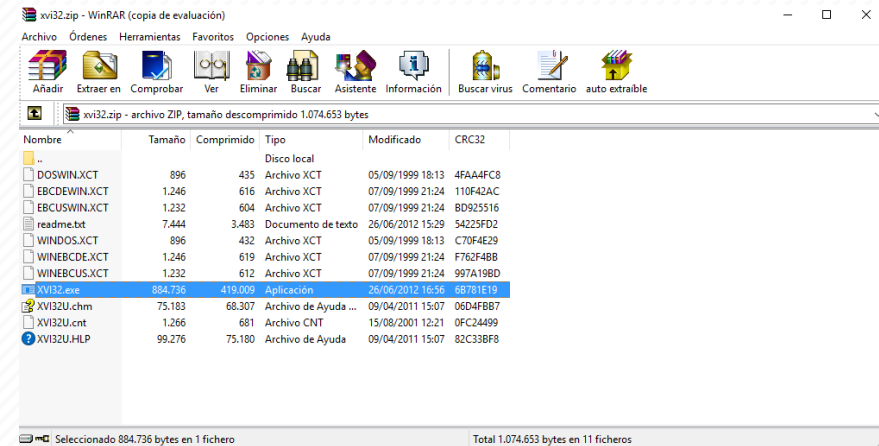
Localice el archivo class en su proyecto y ábralo con el bloc de notas, observe que es lo que muestra el archivo, en el caso del programa donde está localizada una A es donde se encuentra el 65 (el código ASCII de la letra A es 65).



# VISUALIZACIÓN DE LOS BYTECODES

Descargue de Internet el xvi32 hex editor y ejecute el archivo exe. (Es un editor hexadecimal de archivos binarios).

Interfaz del programa. De clic en File→Open y localice el archivo de la clase (.class) y ábralo.

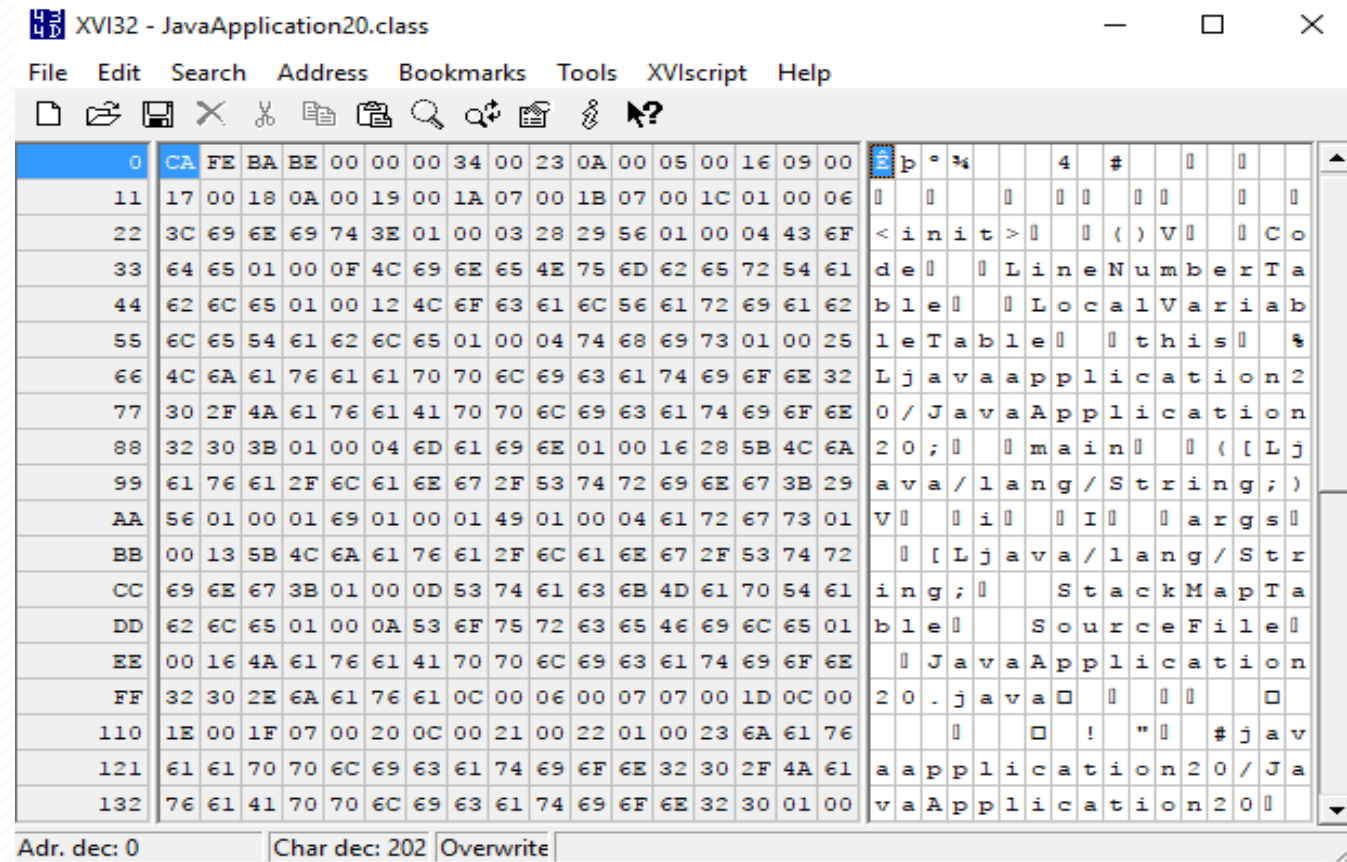
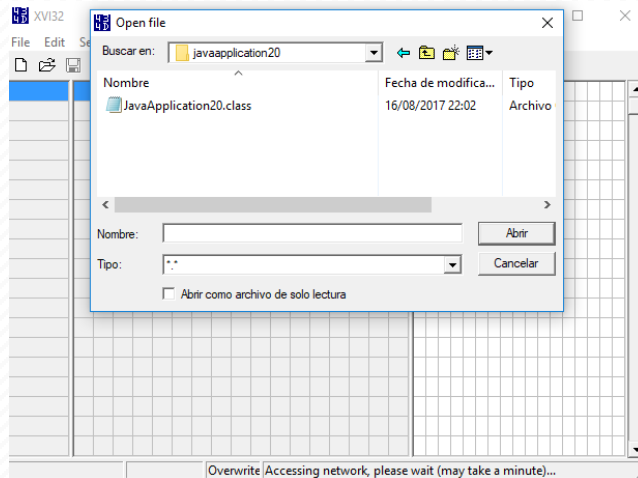




# VISUALIZACIÓN DE LOS BYTECODES

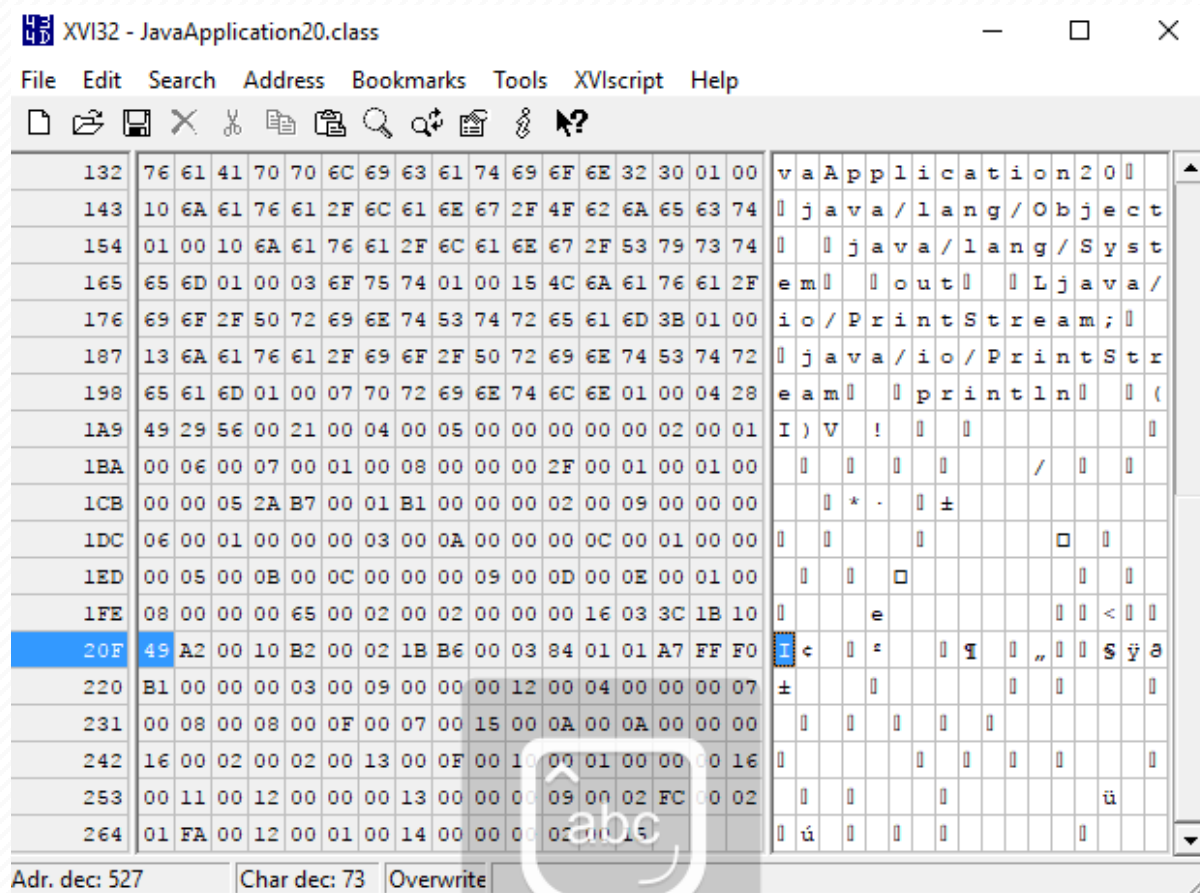
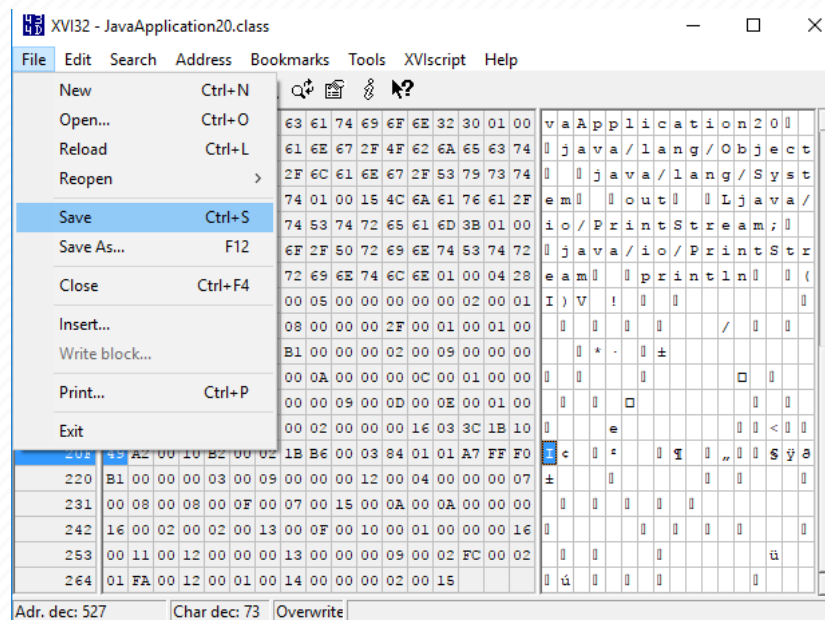
Así se visualizan los archivos en el editor.

Abrir archivo



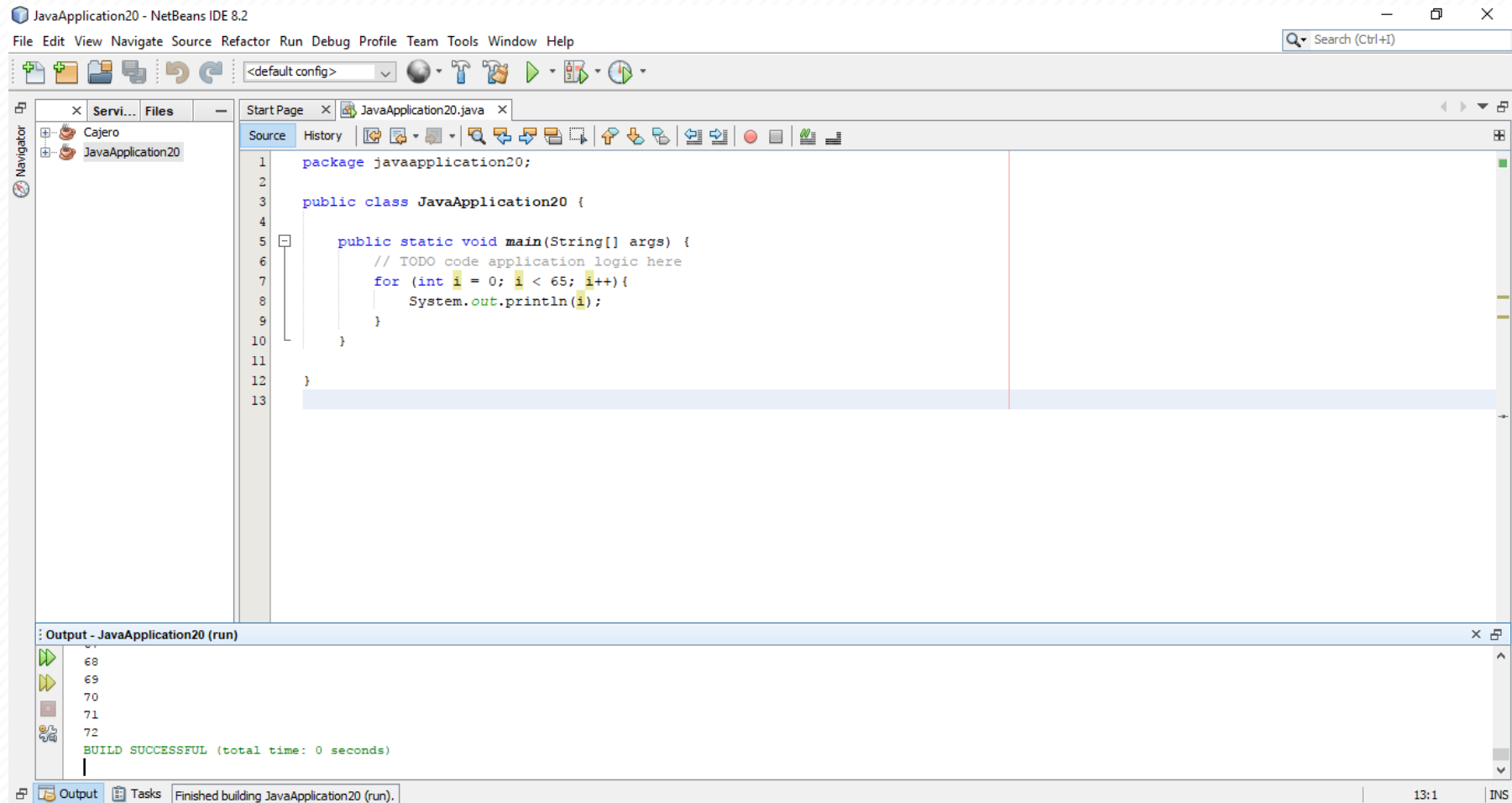
# VISUALIZACIÓN DE LOS BYTECODES

Localice la letra A en el programa y cámbiela por una I o por alguna otra letra mayúscula, guarde al archivo en el menú File→Save.



# VISUALIZACIÓN DE LOS BYTECODES

Vuelva e ejecutar el programa en Netbeans sin cambios y observe lo que sucede.



## OTRO EJERCICIO

Haga el siguiente programa y ejecútelo:

```
public class JavaApplication2 {  
    public static void main(String[] args) {  
        // TODO code application logic here  
        int a, b, c;  
        a = 3;  
        b = 4;  
        c = a*b;  
        System.out.println(c);  
    }  
}
```



## OTRO EJERCICIO

Modifique el programa para que cambie de :

```
C= a * b;    a  
C = a + b;
```

Abra con el editor hexadecimal el archivo .class del proyecto, verá que el 68 cambio a un 60, cámbielo a 64, guárdelo, ejecute nuevamente y vea que ocurre.

Como puede observar los bytecodes almacenan de forma binaria instrucciones, operaciones y valores.

## PARA ENTREGAR

Para la entrega de esta sesión se requieren las capturas de pantalla y una breve explicación de como modificar los bytecodes.