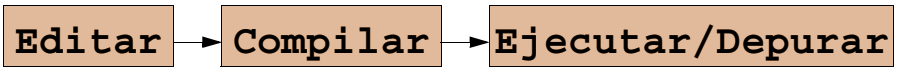
Bloque III. Herramientas

Capítulo 2. Uso de un entorno integrado de desarrollo de programas

- Editor de texto
- El compilador y la ejecución
- Uso del entorno de desarrollo de programas *bluej*
- La depuración
- Generación de documentos
- Empaquetamiento del programa

Proceso de desarrollo del programa

Proceso a seguir:



Para invocar el editor de texto

- en UNIX utilizar el "*emacs*", el "*vi*", o el "*bluej*"
  - para invocar el "*emacs*": `emacs nombre.java`
- en Windows utilizar el *emacs* o el "*bluej*"

Uso del editor emacs:

Moverse por el texto:

- subir, bajar, izquierda, derecha: flechas
- página abajo: `<AvPag>`; página arriba: `<RePag>`

Marcar un bloque:

- marcarlo con el ratón, pulsando el botón izqdo.

Borrar:

- un carácter: tecla `<Supr>`, o `<ctrl>d`
- hasta fin de línea: `<ctrl>k`
- un bloque: marcarlo, y luego elegir el menú "`Edit=>Cut`"

Uso del editor emacs: (cont.)

Recuperar la última línea o bloque borrado: "`Edit=>Paste`"

Copiar un bloque para luego duplicarlo: "`Edit=>Copy`"

Salvar el texto: "`Files=>Save Buffer`"

Finalizar: "`Files => Exit`"

Otras órdenes:

- Buscar un texto: `<ctrl>s texto`
- Volver a buscar el texto anterior: `<ctrl>s`
- Abortar una orden: `<ctrl>g`

El compilador y la ejecución

Utilizaremos el compilador JSDK

Para compilar desde la *shell* o intérprete de órdenes una clase almacenada en un fichero llamado `Nombre.java`:

```
javac Nombre.java
```

La compilación crea la clase ya compilada, en un fichero denominado `Nombre.class`

Para ejecutar una clase llamada `Nombre.class`:

```
java Nombre
```


Ejemplo

```
public class Hola {

   /** este es el método principal */

   public static void main(String[] args) {
      // No hay declaraciones
      System.out.println("Hola, Que tal estas?");
   }
}
```

Uso del entorno de desarrollo de programas Bluej

Un entorno integrado de desarrollo de programas suele presentar las siguientes características:

- interfaz gráfica con menús desplegables, cómoda de usar
- editor de texto orientado al lenguaje
- compilación desde el entorno
- salto automático al lugar donde ocurre un error de compilación
- depuración de alto nivel desde el entorno

Algunos entornos para Java suelen incorporar también herramientas de documentación y de generación de interfaces gráficas ("programación visual")

Gestión de proyectos

Para programar utilizando *Bluej* debe crearse un "*proyecto*" por programa.

El proyecto es un directorio en el disco donde se guarda toda la información que se va generando relacionada con el programa:

- código fuente
- clases compiladas
- documentación
- etc.

Creación de un nuevo proyecto

Elegir `Project=>New Project`, y luego situarse en el directorio deseado y darle un nombre. Ej:

`practica1`

Añadir una clase nueva al proyecto. Para ello, pulsar el botón `New class`,

- luego darle un nombre
- y elegir las opciones deseadas (`Class`, para una clase normal)

Para editar la clase, hacer “doble click” sobre ella

- borrar el código que no necesitemos

Compilar y ejecutar el programa

Para compilar:

- desde el editor: botón `Compile`
- desde el proyecto: con el botón derecho hacer “click” sobre la clase y elegir `Compile`
- los errores de compilación aparecen en la parte inferior de la pantalla
- el puntero del editor se sitúa sobre el error

Para ejecutar desde la ventana del proyecto

- con el botón derecho hacer “click” sobre la clase
- elegir la ejecución del método `main`

Crear y usar objetos

Para crear un objeto de una clase, pulsar el botón derecho sobre ella y elegir la opción **new**

Para ver los atributos de un objeto pulsar el botón derecho sobre él y elegir la opción **inspect**

Para ejecutar un método de un objeto pulsar el botón derecho sobre él y elegir el método deseado

Si el programa se bloquea el ejecutarse, con el botón derecho pulsar sobre la barra de la máquina virtual (parte inferior izquierda de la ventana del proyecto) y elegir "**reset machine**"

Ejemplo

```
public class Notas {
    private int nota1, nota2, nota3;

    /** Pone los valores de las tres notas */
    public void ponNotas (int n1, int n2, int n3) {
        nota1=n1;
        nota2=n2;
        nota3=n3;
    }

    /** Calcula la media real */
    public double media() {
        return (nota1+nota2+nota3)/3.0;
    }
```

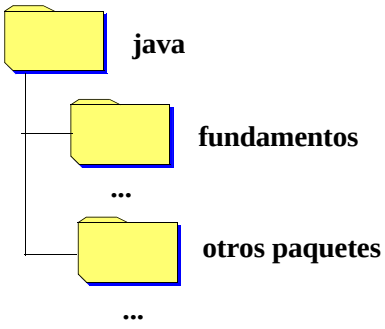
Ejemplo (cont.)

```
    /** Calcula la media entera */
    public int mediaEntera(){
        return (nota1+nota2+nota3)/3;
    }
}
```


Uso de paquetes externos al proyecto

Para poder usar una clase definida en un paquete:

- **Poner todas las clases de un paquete en un directorio con el nombre del paquete**
- **Situar este directorio en un directorio para los paquetes; por ejemplo, en el directorio `java`**



Uso de paquetes externos al proyecto (cont.)

- Añadir el directorio donde están los paquetes (`java` en este caso) a la lista de librerías:
  - `Tools => Preferences`
  - Elegir la pestaña "`Libraries`"
  - Elegir la opción "`Add...`" en "`User Libraries`"
  - Pinchar el directorio de los paquetes (`java`)
  - Pulsar `Abrir`
  - Pulsar `Aceptar`
  - Pulsar `OK`
- Acordarse de importar las clases a usar en el programa:
  - `import fundamentos.Lectura;`
  - `import fundamentos.*;`

Ejemplo de uso del paquete fundamentos

```
import fundamentos.*;
public class Notas {
  ...
  /** Lee de una ventana las tres notas */
  public void leeNotas() {
    Lectura pantalla = new Lectura ("Notas");
    pantalla.creaEntrada("Nota 1º trimestre",nota1);
    pantalla.creaEntrada("Nota 2º trimestre",nota2);
    pantalla.creaEntrada("Nota 3º trimestre",nota3);
    pantalla.esperaYCierra();
    nota1=pantalla.leeInt("Nota 1º trimestre");
    nota2=pantalla.leeInt("Nota 2º trimestre");
    nota3=pantalla.leeInt("Nota 3º trimestre");
  }
}
```

La depuración

Es el proceso de prueba del programa para localizar errores

La depuración se puede realizar:

- manualmente: insertando instrucciones de salida (`println`) que muestren el flujo de control del programa y el valor de las variables de interés
- mediante un depurador de alto nivel

El depurador de alto nivel permite:

- parar el programa en los puntos deseados (*breakpoints*)
- ejecutar paso a paso
- visualizar el contenido de las variables (*watches*)
- visualizar llamadas a procedimientos y sus argumentos

Depurador del Bluej

Es un depurador para programas Java con interfaz gráfica.

Permite introducir puntos de ruptura (breakpoints)

- “click” en la columna a la izquierda del código

También permite controlar la ejecución, y visualizar información en la ventana de depuración.

Depuración: Control de la ejecución

UNIVERSIDAD DE CANTABRIA

La ventana de depuración se abre poniendo un punto de ruptura y ejecutando el programa

Tiene botones para:

- Finalizar la ejecución: `Terminate`
- Continuar la ejecución: `Continue`
- Hacer una pausa en la ejecución `Stop` (útil para lazos largos o infinitos)
- Ejecutar paso a paso (entrando en métodos): `Step Into`
- Ejecutar paso a paso (saltando por encima de métodos): `Step`

Depuración: Visualizar información

UNIVERSIDAD DE CANTABRIA

En la ventana de depuración se puede ver:

- La secuencia de llamadas a métodos
- Atributos estáticos
- Atributos normales
- Variables locales del método actual

Generación de documentos

Para usar una clase, lo único que se necesita conocer de ella es la interfaz pública:

- atributos: sus nombres y tipos, y descripción
- métodos: sus cabeceras y descripción de lo que hacen

Se puede extraer esta información de manera automática, por medio de herramientas de documentación.

Elegir en la ventana del proyecto:

- `Tools => Project documentation`

La primera vez, quitar la opción de documentación JSDK en

- `Tools => Preferences => Miscellaneous`

Empaquetamiento del programa

Hay un formato para guardar en un solo fichero comprimido todas las clases y recursos que necesite un programa

- formato `jar`

Conviene para ejecutarlo que el programa haga toda la entrada/ salida con ventanas (no usar `System.out.println`)

Se genera desde el *bluej* con la opción

- `project => create jar file`

Se ejecuta:

- Linux: `java -jar nombre_fichero.jar`
- Windows: doble click sobre el icono del fichero

Distribución de Software

Se hará accesible en el laboratorio una carpeta con los siguientes contenidos, para poder copiarlos en una memoria USB:

- compilador java de Sun: *jsdk*
- entorno de desarrollo: *Bluej*

Con versiones para *Windows* y *Linux*

Las instrucciones para instalación estarán en esa carpeta en el fichero:

- `instalacion_java.txt`