

Introducción a Java

Curso de Programación en Java

Jesús Montes Sánchez
jmontes@fi.upm.es

¿Qué es...?

- ⊗ Ordenador
- ⊗ Arquitectura del ordenador
- ⊗ Software
- ⊗ Programar
- ⊗ Lenguaje de programación
- ⊗ C, C++, Java
- ⊗ Compilador
- ⊗ Máquina virtual
- ⊗ Programación estructurada
- ⊗ Programación orientada a objetos
- ⊗ Aplicación Web
- ⊗ Terminal (símbolo de sistema)
- ⊗ IDE

Introducción a Java - Jesús Montes Sánchez (jmontes@fi.upm.es)

Contenidos

- ⊗ Origen, características y evolución de Java
- ⊗ Temario del curso
- ⊗ Paradigmas de programación
- ⊗ Entorno de trabajo

Introducción a Java - Jesús Montes Sánchez (jmontes@fi.upm.es)

Origen, características y evolución de Java

Estructura de un ordenador

- ⊗ ¿Qué sabe hacer un ordenador?
- ⊗ Muy poco, muy rápido.
- ⊗ ¿Qué partes tiene un ordenador?
- ⊗ Componentes físicos: Hardware
- ⊗ Programas: Software



Introducción a Java - Jesús Montes Sánchez (jmontes@fi.upm.es)

Estructura de un ordenador

- ⊗ Dispositivos de Entrada/ Salida
- ⊗ Sirven para la interacción hombre-máquina
- ⊗ Entrada: Teclado, ratón, escáner...
- ⊗ Salida: Monitor, impresora...



Introducción a Java - Jesús Montes Sánchez (jmontes@fi.upm.es)

Estructura de un ordenador

- ⊗ Memoria principal
 - ⊗ Conjunto de celdas en una tabla
 - ⊗ Almacena **datos** (entrada, salida y procesados) e **instrucciones** (programas)
 - ⊗ De acceso rápido
 - ⊗ Dos tipos: RAM y ROM



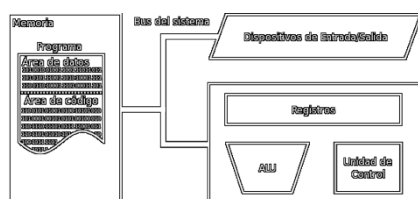
Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Estructura de un ordenador

- ⊗ Unidad central de procesamiento (CPU)
 - ⊗ Unidad aritmético-lógica (ALU): Realiza operaciones
 - ⊗ Aritméticas (suma, resta...)
 - ⊗ Lógicas (AND, OR...)
 - ⊗ Banco de registros: Almacenamiento temporal de información, de muy rápido acceso.
 - ⊗ Unidad de control (UC): Controla la ejecución del programa. Da las órdenes para acceder a los datos en memoria.

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Estructura de un ordenador



Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Ciclo de vida del software



Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Ciclo de vida del software

- ⊗ Especificación
 - ⊗ ¿Qué quiere exactamente el usuario final?
 - ⊗ Contrato
- ⊗ Análisis
 - ⊗ ¿Cómo resolver el problema?
 - ⊗ Descomposición
- ⊗ Diseño
 - ⊗ Cada una de las partes y su interconexión

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Ciclo de vida del software

- ⊗ Programación
 - ⊗ Validación
 - ⊗ Calidad de la integración de cada módulo
 - ⊗ Cumplimiento de las especificaciones
 - ⊗ Implantación
 - ⊗ Correcto funcionamiento y aceptación

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Ciclo de programación

- ⊗ Codifica
 - ⊗ Implementa con precisión cada elemento
- ⊗ Comenta el código fuente
 - ⊗ Ayuda a su lectura y comprensión
- ⊗ Verifica
 - ⊗ Casos de prueba
- ⊗ Depura
 - ⊗ Localiza y corrige errores
- ⊗ Documenta

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Lenguajes de programación

- ⊗ Lenguaje máquina
 - ⊗ Código binario
 - ⊗ Fuertemente relacionado con el hardware (i386, PowerPC...)
 - ⊗ **El único lenguaje que entiende el ordenador**
 - ⊗ Comprensible para la máquina...
 - ⊗ ...pero imposible para una persona

00111101010111000101110001110001

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Lenguajes de programación

- ⊗ Lenguaje ensamblador
 - ⊗ Juego de instrucciones básico del ordenador
 - ⊗ De muy bajo nivel (traducción directa a lenguaje máquina)
 - ⊗ Muy rápido y eficiente

Suma de dos números:

```
MOV R1, 10
MOV R2, 20
ADD R1, R2
ST #0014, R1
```

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Lenguajes de programación

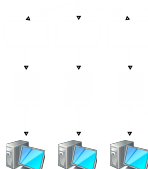
- ⊗ Lenguaje de alto nivel
 - ⊗ Comúnmente conocido como lenguaje de programación
 - ⊗ Cercano al lenguaje humano
 - ⊗ Abstrae las características específicas del hardware
 - ⊗ Mediante la compilación y el enlazado se traduce a lenguaje máquina
 - ⊗ C, C++, Java, Python...

r1 = 10 + 20

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Codificación, compilación, enlazado y ejecución

- ⊗ Codificación
 - ⊗ Escritura del programa
- ⊗ Compilación y enlazado
 - ⊗ Traducción del programa a lenguaje máquina ejecutable
 - ⊗ Depende de la máquina
- ⊗ Ejecución
 - ⊗ Puesta en funcionamiento del programa resultante



Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Lenguajes de alto nivel

- ⊗ C
 - ⊗ Creado de 1972 por Dennis M. Ritchie
 - ⊗ De alto nivel, pero muy relacionado con las funcionalidades básicas de la máquina (de forma abstracta)
 - ⊗ Ideal para la programación de sistemas operativos, sistemas empujados, drivers...
 - ⊗ Muy eficiente
 - ⊗ En teoría, portable siempre que se disponga de un compilador de C para la arquitectura destino

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Lenguajes de alto nivel

- ⊗ C++
 - ⊗ Evolución de C
 - ⊗ Creado por Bjarne Stroustrup en 1979
 - ⊗ Orientado al desarrollo de aplicaciones de mas alto nivel
 - ⊗ Incorpora mecanismos para realizar programación **orientada a objetos**
 - ⊗ Depende del compilador para garantizar la portabilidad entre arquitecturas

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Los orígenes de Java

- ⊗ Surge en la década de los 90 de la mano de Patrick Naughton y James Gosling
- ⊗ Inicialmente llamado Oak
- ⊗ Desarrollado por Sun Microsystems para facilitar la portabilidad de software entre dispositivos electrónicos
- ⊗ Motivaciones:
 - ⊗ Problemas en la depuración de programas C/C++
 - ⊗ Movilidad de software al actualizar los componentes hardware

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Java

- ⊗ Filosofía *write once, run anywhere* (se escribe una vez y se ejecuta en todas partes)
- ⊗ Programación orientada a objetos
- ⊗ Uso fácil, robusto y eficiente en entornos distribuidos (aplicaciones web...)

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Java

- ⊗ Lenguaje simple y sencillo, basado en C++
- ⊗ De alto nivel
- ⊗ Incorpora un amplio conjunto de bibliotecas
- ⊗ Facilita la gestión de errores (gestión automática de memoria, control de definición de tipos...)
- ⊗ Controla aspectos de seguridad (acceso ilegal a memoria, acceso restringido a ficheros...)

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Java

- ⊗ Se ejecuta sobre una aplicación llamada Máquina Virtual (JVM)
 - ⊗ Abstrae las características específicas del hardware
 - ⊗ Facilita la portabilidad de código
- ⊗ Portable a cualquier arquitectura en la que exista la JVM
- ⊗ Hoy en día, muy orientado al mundo web
- ⊗ Potencialmente menos eficiente que C/C++
- ⊗ Software libre y abierto

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Java

```
// Esto es un comentario de una línea
/*
  Esto es un comentario de
  multiples líneas
*/
// Clase holaMundo
public class holaMundo {
    /* El metodo main es el punto de comienzo del programa */
    public static void main(String[] args) {
        System.out.println("Hola mundo");
    }
}
```

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Java

- ⊗ Componentes
 - ⊗ **javac**, el compilador de Java
 - ⊗ Una JVM específica para cada plataforma
- ⊗ Los archivos *.java* contienen el código fuente
- ⊗ Los archivos *.class* contienen el lenguaje de JVM (bytecode)



Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Evolución de Java

- ⊗ Enero de 1996: aparece java con JDK 1.0
- ⊗ Enero de 1997: JDK 1.1 con:
 - ⊗ Restructuración de AWT (GUI)
 - ⊗ Clases internas
 - ⊗ JavaBeans
 - ⊗ JDBC (Java Database Connectivity)
 - ⊗ RMI (Remote Method Invocation)

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Evolución de Java

- ⊗ Diciembre de 1998: J2SE 1.2 – **Java 2**
 - ⊗ Mecanismos de reflexión. Dinamismo del código
 - ⊗ Incorporación de Swing
 - ⊗ Se incorpora un compilador JIT (Just in Time)
 - ⊗ Aparición de Java Plug-ins: **Giro hacia el mundo de las aplicaciones web**
 - ⊗ Java IDL (Interface Description Language) para interoperar con CORBA

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Evolución de Java

- ⊗ Mayo de 2000: J2SE 1.3
 - ⊗ Nuevo RMI, basado en CORBA
 - ⊗ JavaSound
 - ⊗ JNDI (Java Naming and Directory Interface)
 - ⊗ JPDA (Java Platform Debugger Architecture)
- ⊗ Febrero 2002: J2SE 1.4
 - ⊗ Expresiones regulares
 - ⊗ XML parser
 - ⊗ Extensiones criptográficas
 - ⊗ Java Web Start

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Evolución de Java

- ⊗ Septiembre de 2004: J2SE 1.5
 - ⊗ Plantillas
 - ⊗ Metaprogramación
 - ⊗ Varargs (argumentos variables)
- ⊗ Diciembre de 2006: J2SE 1.6 – **Java 6**
 - ⊗ Combinación con otros lenguajes dinámicos (PHP, Python...)
 - ⊗ Cliente de Servicios Web
 - ⊗ Mejorar en GUI y rendimiento de la JVM
- ⊗ Julio de 2011: Java 7

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Temario del curso

Temario

- ⊗ **Módulo 1: Introducción a Java (5 horas)**
- ⊗ **Módulo 2: Programación Java (35 horas)**
 - ⊗ Introducción a la programación. Secuencia de ejecución de un programa
 - ⊗ Lógicas
 - ⊗ Programación de métodos, definición y llamada de los mismos
 - ⊗ Estructuras de control, bucles y evaluación de condiciones
 - ⊗ Tipos de datos básicos, uso de la clase String y otras clases sencillas
 - ⊗ Otros aspectos de la sintaxis de Java

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Temario

- ⊗ **Módulo 3: Programación orientada a objetos en Java (35 horas)**
 - ⊗ Nociones básicas
 - ⊗ Declaración de clases y concepto de herencia. Clases e interfaces
 - ⊗ Elementos del lenguaje orientados a la gestión de clases e interfaces
 - ⊗ Empaquetado de clases y organización de un proyecto completo
- ⊗ **Módulo Extra: Repaso de los fundamentos de java (15 horas)**
 - ⊗ Programación básica
 - ⊗ Programación orientada a objetos

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Temario

- ⊗ **Módulo 4: Interfaces Gráficas de Usuario (75 horas)**
 - ⊗ Programación basada en eventos
 - ⊗ Componentes básicos de los interfaces de usuario, ventanas, campos de texto y cuadros de diálogo. Java AWT
 - ⊗ Manejadores de eventos y acciones
 - ⊗ Despliegue de componentes, organizadores de elementos gráficos
 - ⊗ Componentes gráficos extendidos, clases Java Swing
 - ⊗ Modelo Vista-Controlador
 - ⊗ Entornos gráficos de desarrollo. Desarrollo de aplicaciones profesionales

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Temario

- ⊗ **Módulo 5: Paquete de Utilidades Java (27.5 horas)**
 - ⊗ Visión general del paquete java.util. Reutilización de código y uso de tipos abstractos
 - ⊗ Tipos de datos para almacenamiento de otros elementos. Clase Object
 - ⊗ Clases de utilidad para la gestión del tiempo
 - ⊗ Clases de utilidad para la organización de aplicaciones complejas

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Temario

- ⊗ **Módulo 6: Gestión de Ficheros (32.5 horas)**
 - ⊗ Entrada/Salida en Java. Concepto de *stream* o flujo de entrada/salida
 - ⊗ Clases Java asociadas a la manipulación de directorios y ficheros
 - ⊗ Gestión de errores por medio de excepciones
- ⊗ **Módulo 7: Desarrollo de Aplicaciones Web en Java (15 horas)**
 - ⊗ Conceptos generales del desarrollo de aplicaciones sobre Internet
 - ⊗ Programación de *applets*. Características y restricciones de seguridad

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Temario

- ⊗ **Módulo 8: Aspectos Avanzados de Java (30 horas)**
 - ⊗ Comunicación de aplicaciones Java vía RMI
 - ⊗ Extensiones multimedia de Java. Manipulación de imagen vídeo y audio
 - ⊗ Presentación de las tecnologías Java avanzadas: Jini, JNI, EJB, JDBC...
- ⊗ **Módulo 9: Sensibilización medioambiental (10 horas)**

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Paradigmas de programación

¿Cómo se debe programar?

- ⊗ El desarrollo *artesano* de software derivó el **software**
 - ⊗ Baja productividad
 - ⊗ Muy difícil mantenimiento
 - ⊗ Redundancia de código
- ⊗ Para conseguir software de calidad se desarrollan los **paradigmas de programación**
 - ⊗ Técnicas de programación
 - ⊗ Estructuras de programación
 - ⊗ Reglas de estilo de codificación

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Software de calidad

- ⊗ Correcto
- ⊗ Eficiente
- ⊗ Reutiliza y es reutilizable
- ⊗ Portable
- ⊗ Estándar
- ⊗ Robusto
- ⊗ Legible
- ⊗ Mantenible

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Software de calidad

- ⊗ Correcto
 - ⊗ Hace exactamente lo que dice
- ⊗ Eficiente
 - ⊗ Lo hace de la forma menos costosa
- ⊗ Reutilización
 - ⊗ No reinventa la rueda
 - ⊗ Utiliza las bibliotecas
 - ⊗ Permite reaprovechar el código

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Software de calidad

- ⊗ Portable
 - ⊗ Evita dependencias (arquitectura, sistema,...)
- ⊗ Estándar
 - ⊗ El comportamiento esperado
 - ⊗ Evita originalidades innecesarias
- ⊗ Robusto
 - ⊗ Tolerar los errores de otros y no errar él

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Software de calidad

- ⊗ Legible
 - ⊗ El código debe facilitar su comprensión a otros programadores
- ⊗ Mantenible
 - ⊗ Destinado a evolucionar, a ser corregido y mejorado
 - ⊗ Diseño y estructura deben facilitar el mantenimiento

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Paradigma de programación

- ⊗ Conjunto de reglas que nos ayudan a desarrollar software de calidad
- ⊗ Programación estructurada
- ⊗ **Programación orientada a objetos**

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Programación estructurada

⊗ Algoritmo

- ⊗ Método para resolver un problema sin ambigüedades en un número finito de pasos
- ⊗ Ha de ser la solución general a todos los problemas del mismo tipo
- ⊗ Deben considerarse todas las posibles situaciones



⊗ Datos

- ⊗ Información que el programa recibe, maneja y devuelve
- ⊗ Las estructuras de datos facilitan su manejo

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Programación estructurada

- ⊗ Situaciones cotidianas vistas como programas
 - ⊗ ¿Cómo ordeno las cartas de una baraja española?
 - ⊗ ¿Qué secuencia de pasos tengo que seguir para programar el despertador?
 - ⊗ ¿cómo preparo una tortilla?
- ⊗ En estos ejemplos, identificar:
 - ⊗ Datos de entrada
 - ⊗ Datos de salida
 - ⊗ Algoritmo

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Programación estructurada

⊗ Programa

- ⊗ Expresión de un algoritmo en un lenguaje
- ⊗ Incluye la descripción de los datos
- ⊗ Su ejecución producirá el resultado deseado

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Estructuración de un programa

- ⊗ Razonamiento descendente
 - ⊗ Si la magnitud del problema no permite visualizarlo en toda su extensión
 - ⊗ Identificar etapas o fases
 - ⊗ Dividir el problema sub-problemas
 - ⊗ De lo general a lo específico
 - ⊗ Razonando en **términos abstractos**
 - ⊗ Ir refinando una solución

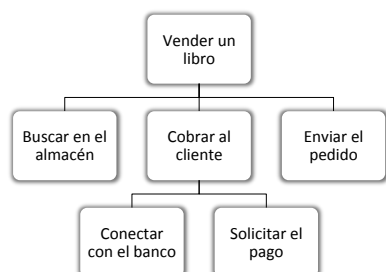
Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Estructuración de un programa

- ⊗ División en módulos funcionales
 - ⊗ Perfilar más y más hasta delimitar cada **módulo** de nuestro programa
 - ⊗ ¿Es suficientemente **genérico**?
 - ⊗ ¿Está suficientemente **delimitado**?
 - ⊗ ¿Merece ser una pieza **independiente**?
 - ⊗ ¿Puede ser **reutilizable**?

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Estructuración de un programa



Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Programación orientada a objetos

Objetos: Elementos que interactúan entre sí para conseguir un fin

- Autónomos
- Con una funcionalidad concreta definida
- Instancias de una clase



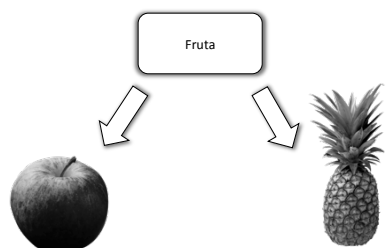
Clases

- Especificación de un conjunto de elementos

Todo objeto pertenece a una clase

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Objetos y clases



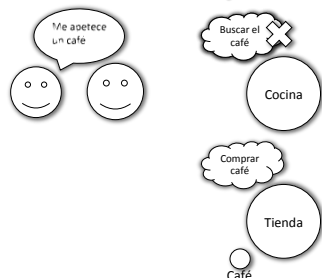
Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Objetos y clases

- Los objetos tienen propiedades que los distinguen
- Comunes a todos los de su clase
- Propias de cada uno
- Los objetos interactúan entre sí por medio de mensajes
- Los objetos representan los conceptos fundamentales del programa, y de su interacción surge la funcionalidad

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

¿Cómo funciona un programa orientado a objetos?



Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Lenguajes y paradigmas

- En teoría, el paradigma de programación escogido es independiente del lenguaje que se utilice
- No obstante, la mayoría de lenguajes de programación están pensados para un paradigma concreto:
 - C: Estructurado
 - C++: Estructurado y/o orientado a objetos
 - Java: Orientado a objetos
 - Python: Orientado a objetos
 - ...

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Entorno de trabajo

Entorno de trabajo

- ⊗ Conjunto de herramientas que usamos para desarrollar software
 - ⊗ Herramientas de diseño asistido
 - ⊗ Editores
 - ⊗ Compiladores
 - ⊗ Depuradores
 - ⊗ ...
- ⊗ Facilitan nuestra labor y proporcionan funcionalidades añadidas

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Java SE (Standard Edition)

- ⊗ JRE (*Java Runtime Environment*)
 - ⊗ Incluye los elementos necesarios para ejecutar aplicaciones Java (JVM, Java Web Start...)
- ⊗ JDK (*Java Development Kit*)
 - ⊗ Incluye JRE
 - ⊗ Incorpora además herramientas de desarrollo (compiladores...)

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Variables PATH y CLASSPATH

- ⊗ PATH
 - ⊗ Variable de entorno que indica donde se ubican los programas
 - ⊗ Podemos ejecutar programas mediante la línea de mandatos
- ⊗ CLASSPATH
 - ⊗ Variable de entorno que usa Java
 - ⊗ Indica donde se almacenan las bibliotecas y ficheros .class de Java

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Entorno básico

- ⊗ Instalar JDK
 - ⊗ www.oracle.com/technetwork/java/javase/downloads/index.html
- ⊗ Incluir en el PATH los ejecutables del JDK
 - ⊗ Propiedades de Mi PC > Opciones Avanzadas > Variables de Entorno
- ⊗ Copiar el programa de ejemplo
- ⊗ Usando la terminal de Windows (símbolo de sistema), compilar y ejecutar el programa

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Programa de ejemplo

```
// Esto es un comentario de una línea
/*
  Esto es un comentario de
  múltiples líneas
*/
// Clase holaMundo
public class holaMundo {
    /* El metodo main es el punto de comienzo del programa */
    public static void main(String[] args) {
        System.out.println("Hola mundo");
    }
}
```

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

NetBeans

- ⊗ Es un IDE (*Integrated Development Environment*)
- ⊗ Plataforma para el desarrollo de aplicaciones
- ⊗ Soporta múltiples lenguajes (Java, C++, PHP...)
- ⊗ Dispone de multitud de plug-ins útiles (OpenSwing...)
- ⊗ Libre y gratuito (Licencia GPL)
- ⊗ netbeans.org

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)

Uso de NetBeans

- ⊗ Crear un nuevo proyecto
- ⊗ Gestión de las ventanas del entorno
 - ⊗ Área de proyectos, vista del sistema de ficheros
 - ⊗ Área de edición
 - ⊗ Consola de mensajes
 - ⊗ Propiedades del proyecto (CLASSPATH...)
- ⊗ Codificación, compilación y ejecución de un proyecto de prueba (holaMundo)

Introducción a Java - Jesús Montes Sánchez (jmontes@fl.upm.es)