

JAVA

Lenguaje de programación

Relator : Jaime Tobar
jaime.tobar@gmail.com

Diapositivas basadas en la versión original de
Mohamed Mohamed
mmohamed@dcc.uchile.cl

Contenido

- Introducción a Java
- El lenguaje Java
- Arreglos en Java
- Programación OO con Java
- Manejo de Errores
- Streams de Entrada/Salida
- Threads de Control
- Red
- Swing

Contenido

- Acceso a Base de Datos
- Internacionalización
- JAR

Introducción a Java

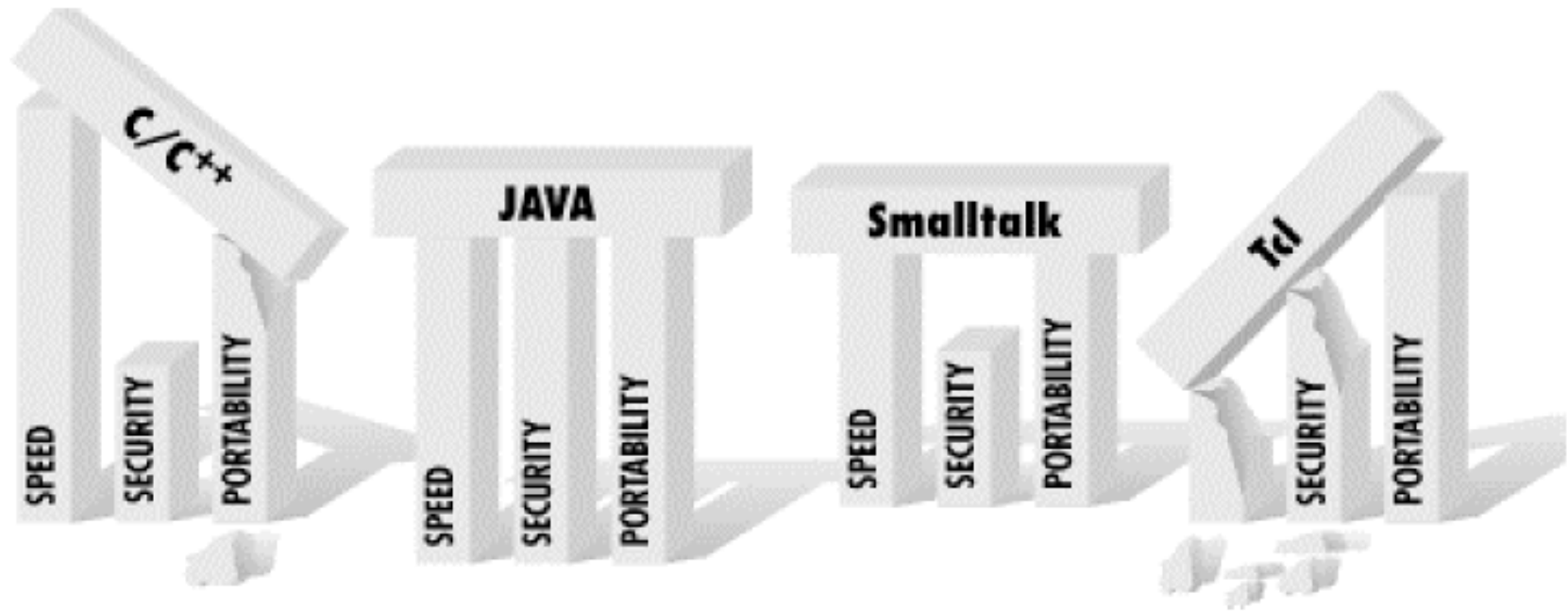
- Origen del lenguaje
 - Desarrollado por SUN <http://java.sun.com>
 - Enfocado a los electrodomésticos (FirstPerson)
- Características de Java
 - Portabilidad (medio compilado y medio interpretado)
 - Robusto (manejo de excepciones)
 - Orientado a Objeto
 - Seguro (no hay punteros)
 - “Simple” (lo más simple tomando en cuenta la OO)

Introducción a Java

- Herramientas de desarrollo
 - JDK de SUN Microsystem
- Tipo de programas
 - Aplicaciones
 - Applet
 - Servlets
 - etc..

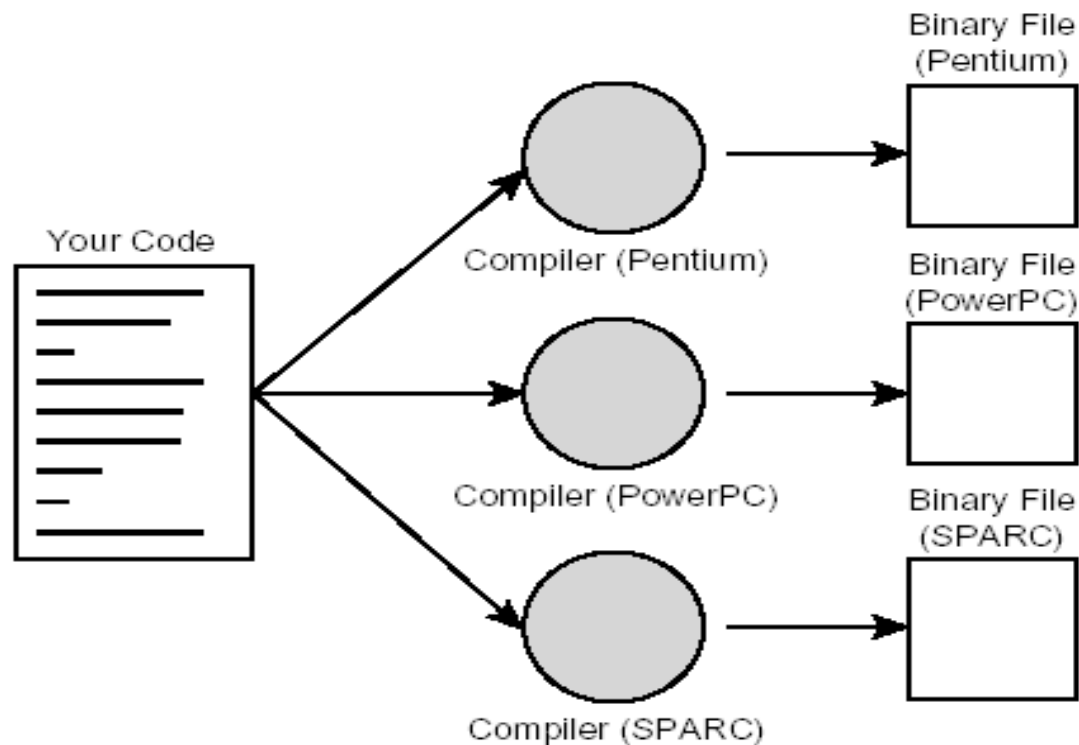
Introducción a Java

- Comparación



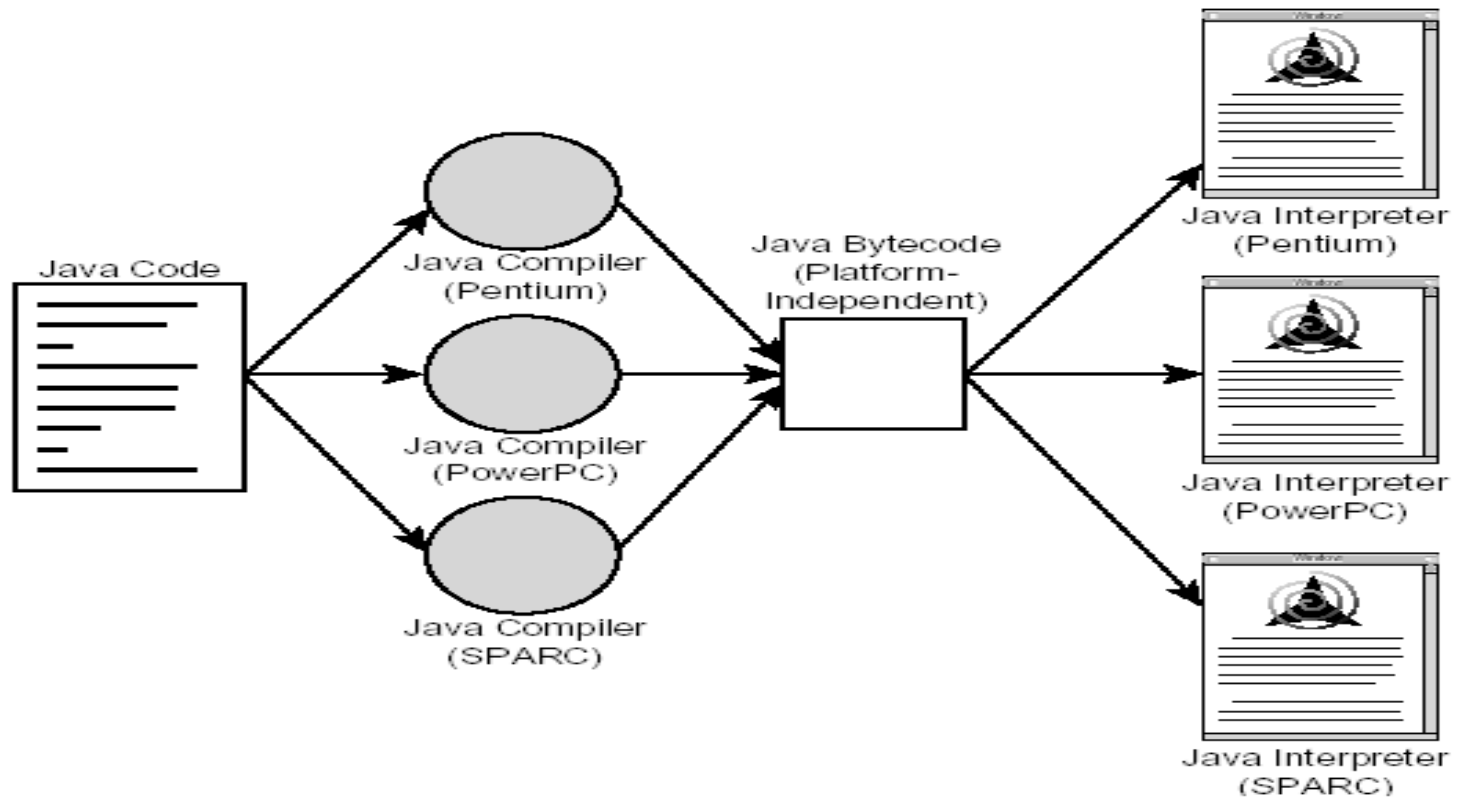
Introducción a Java

- Java Virtual Machine (JVM)
 - Compilación tradicional



Introducción a Java

- Java Virtual Machine (JVM)
 - Compilación JVM



El lenguaje Java

- **Nuestro primer programa en Java**

- `public class HolaMundo {`
- `public static void main(String[] args) {`
- `System.out.println("Hola Mundo!!!");`
- `System.out.print("...en java");`
- `System.out.println(" en el 2005");`
- `}`
- `}`

- El archivo debe llamarse `HolaMundo.java`
 - Se compila con: `javac HolaMundo.java`
 - Se ejecuta con: `java HolaMundo`

- `ver:HolaMundo.java`

El lenguaje Java

- El compilador de JAVA necesita que todas las clases ejecutables contengan el método **main**, para poderlo distinguir de otros:
- `public static void main(String[] args)`

Tipos de datos básicos

• Tipo	Descripción	Tamaño	Clase
• boolean	Valor lógico	1 bit	Boolean
• char	Carácter	16 bit	Character
• byte	Entero muy pequeño	8 bit	
• short	Entero pequeño	16 bit	
• int	Entero normal	32 bit	Integer
• long	Entero grande	64 bit	Long
• float	Precisión simple	32 bit	Float
• double	Doble precisión	64 bit	Double
• void	Tipo vacío		

El lenguaje Java

- Comentarios
 - // Aquí empieza un comentario
 - /* Soy un comentario de
 - de varias líneas */
- Identificadores (nombres de variables)
 - Java distingue entre minúsculas y MAYUSCULAS
 - el nombre de las variables comienza con una letra con un _ o con el signo \$
 - ejemplos:
 - ! Contador
 - ! _camino
 - ! \$dolar

El lenguaje Java

- Operadores

– Descripción	Símbolo	Expresión	Resultado
– Multiplicación	*	2*4	8
– Resto	%	5%2	1
– Suma	+	2+2	4
– Resta	-	7-2	5
– Incremento	++	2++	3
– Decremento	--	--2	1
– Menos unario	-	-(2+4)	-6

- NOTA: Hay diferencias entre ++j y j++

- ver:IncDec.java

El lenguaje Java

- Separadores

- **Separador Descripción**

- () Contienen listas de parámetros. También se utilizan para modificar la precedencia en una expresión, contener expresiones para control de flujo y realizar conversiones de tipo.
- { } Se utilizan para definir bloques de código.
- [] Se utiliza tanto para declarar arreglos como para referenciar valores dentro de los mismos.
- ; Separa sentencias.
- , Separa identificadores consecutivos en la declaración de variables y en las listas de parámetros.
- . Separa un nombre de propiedad o método de una variable de referencia. También separa nombre de paquete de los de un subpaquete o una clase.

El lenguaje Java

- **Palabras claves**

- `abstract` `continue` `for` `new` `switch`
- `boolean` `default` `goto` `null` `synchronized`
- `break` `do` `if` `package` `this`
- `byte` `double` `implements` `private` `threadsafe`
- `byvalue` `else` `import` `protected` `throw`
- `case` `extends` `instanceof` `public` `transient`
- `catch` `false` `int` `return` `true`
- `char` `final` `interface` `short` `try`
- `class` `finally` `long` `static` `void`
- `const` `float` `native` `super` `while`

El lenguaje Java

- Expresiones

- `int i = 45;`
- `double j = 1.5;`
- `String nombre = "María";`
- `String nombre = null;`
- `nombre = new String("María");`
- `boolean flag = true;`
- `Float f = new Float(1.0);`

El lenguaje Java

- Ejemplos:

- `String mensaje = "Hola "+"Mundo";`
- `int i = 1;`
- `int m = i++; // m = ?`
- `int k = i; // k = ?`
- `int j = --k; // j = ?`
- `boolean b = false;`
- `boolean v = !b; // v es true o false?`

El lenguaje Java

- Clase String
 - Esta clase almacena una cadena de caracteres.
 - Existen varios métodos que nos facilitan el trabajo.
 - `String cadena = new String("Hello World");`
 - `cadena.length()` // devuelve el largo del String
 - `cadena.toLowerCase()` // convierte el String a minúsculas
 - `cadena.toUpperCase()` // a MAYUSCULAS
 - `cadena.trim()` // elimina los espacios en blanco al comienzo y al final del String

El lenguaje Java

- Control de flujo
 - if
 - sentencia;
 - if(expresión booleana)
 - sentencia;
 - if - else
 - if(expresión booleana)
 - sentencia;
 - else
 - sentencia;

El lenguaje Java

- while
 - while (expresión booleana)
 - sentencia;
 - do – while
 - do
 - sentencia;
 - while (expresión booleana)
- for
 - for (inicialización; expresión booleana; paso)
 - sentencia;

El lenguaje Java

- switch
- switch (expresión) {
- case valor1: sentencia; break;
- case valor2: sentencia; break;
- case valor3: sentencia; break;
- // ...
- default: sentencia;
- }

El lenguaje Java

- Ejemplos:

- `if(m == 1) {`
 - `m++;`
 - `flag = true;`
 - `} else {`
 - `m--;`
 - `flag = false;`
 - `}`
- `for(int i=0,k=1; i<10 && k < 50; i++,k +=2) {`
- `sentencias;`
- `}`

El lenguaje Java

➡ Ejercicio:

Haga un programa que recorra los 100 primeros numeros, los imprima en pantalla y diga a su lado si es impar o par

[terminal] El numero 25 es impar

ver:Numeros.java

Arreglos

- Un arreglo (array) es una manera de agrupar valores de un mismo tipo bajo un mismo nombre.
- Existen dos formas de declarar arreglos:
 - `char [] c;`
 - `char c [];`
- Crear matrices, es sencillamente agregar corchetes.
 - `String [][] s;`

Arreglos

- Ejemplos

- `String [] nombres = new String[3];`
 - `nombres[0] = “María”;`
 - `nombres[1] = “José”;`
 - `nombres[2] = “Carmen”;`
- o bien
- `String [] paises = { “Chile”;”Argentina”;”Francia” };`
- `if(nombres.length == paises.length)`
- `sentencia; // Se ejecuta?`

Arreglos

- archivo Arreglos.java

- ```
public class Arreglos {
- static String[] paises = { "España",
 "Holanda", "Brasil" };
- public static void main(String[] args) {
- System.out.println(paises[1]);
- System.out.println(paises.length);
- }
- }
```

- Modificar el archivo para que imprima los nombres de todos los paises y su largo.

- ver: Arreglos.java y Arreglos\_mod.java

# Interactuando con nuestro programa

- Hay varias formas de entregarle información a nuestro programa, pero las más básicas que usaremos en el comienzo son a través de la línea de comando:

- `c:\java Aplicacion` estos son parametros

- estos se recuperan dentro del programa vía el arreglo `args[]`, con `args[0]="estos"` y así sucesivamente.

# Interactuando con nuestro programa

La otra forma es que nuestro programa pregunte por ella:

```
c : \ java Aplicacion
```

```
Ingresa una linea(termina con enter)
```

lo ingresado se recupera via un lector de teclado:

```
import java.io.*;
public class
public static void main(String[] args) throws
 IOException{
 BufferedReader input ;
 input = new BufferedReader(new
 InputStreamReader(System.in));
```

Donde `System.in` es el teclado e `input` la variable que nos permitira recoger lo que se escriba.

# Interactuando con nuestro programa

- Esta última manera acarrea ciertas rigideces que se apreciarán en el ejemplo:
- Se debe poner al principio del archivo .java:
- `import java.io.*;`
- Se debe agregar la clausula throws IOException al main
- `public static void main (String args[]) throws IOException`

# Ejercicios

- Ejercicios

- Inicializar un arreglo con los primeros 100 números e imprimirlo en pantalla (el arreglo).
  - Inicializar un arreglo con los primeros 100 números pares (0,2,4,6,..) e imprimir los primeros 10 números.
  - Recibir su nombre vía línea de comandos e imprimirlo en pantalla.
  - Que su programa pregunte por su nombre, lo lea del teclado y lo imprima en pantalla.
- 
- **ver: Arreglos100.java, ArreglosPar.java, leer\_stdin.java y leer\_args.java**