Java para Cibernética y Computación

Colegio de Ciencias y Humanidades – Plantel Oriente

Instructores:

¡¡¡TODOS LOS PROFESORES INSCRITOS EN EL CURSO!!!

Forma de trabajo

- Asistencia
- Participación
- Propuestas por parte de los profesores asistentes

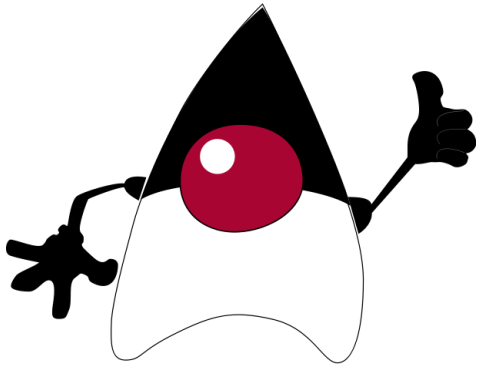
Programación por día

- Día 1 (Lunes 28 de mayo)
  - Repaso de breve historia de Java, sintaxis básica, estructuras de control de flujo y ambientes de trabajo.
- Día 2 (Martes 29 de mayo)
  - Cadenas y arreglos
- Día 3 (Miércoles 30 de mayo)
  - Programación Orientada a Objetos (POO)
- Día 4 (Jueves 31 de mayo)
  - POO e introducción a Java Swing (interfaz gráfica)
- Día 5 (Viernes 01 de junio)
  - Java Swing

Repaso

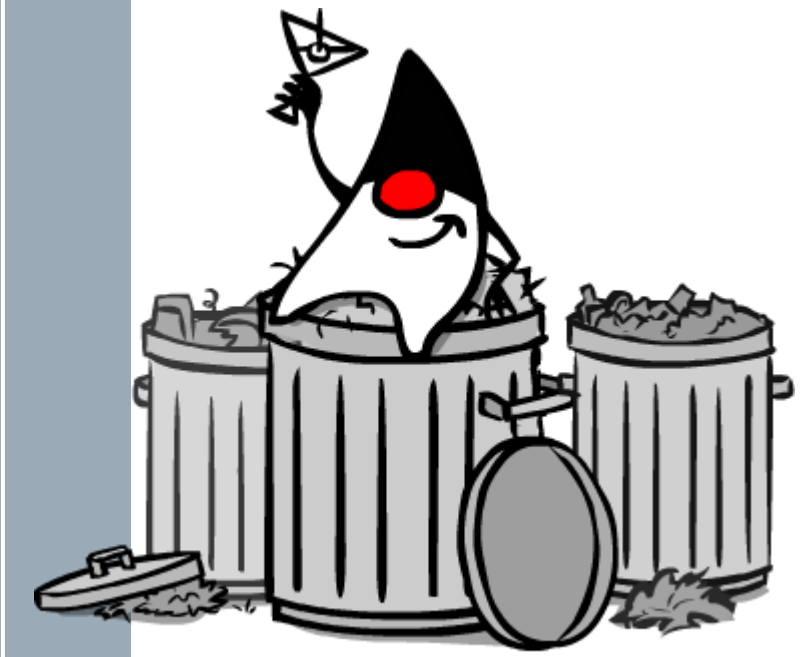
Breve historia

- En 1991 se crea una herramienta de programación en Sun Microsystems.
- James Gosling, Arthur Van Hoff, Andy Bechtolsheim .
- Su objetivo era implementar una máquina virtual y un lenguaje similar a C++.
- En 1994 se reorienta hacia la web.



Duke

Recolector de basura



- Se encarga de liberar la memoria por el usuario.
- Reserva espacios de memoria para su uso.
- Compactar espacios de memoria libres y consecutivos entre sí.
- Llevar cuenta de qué espacios están libres y cuales no.

Java JDK

- JDK (Java Development Kit) – Kit de desarrollo de Java
- Software que provee herramientas de desarrollo para la creación de programas en Java
- javac.exe - **compilador**
- java.exe - **intérprete**

Java JRE

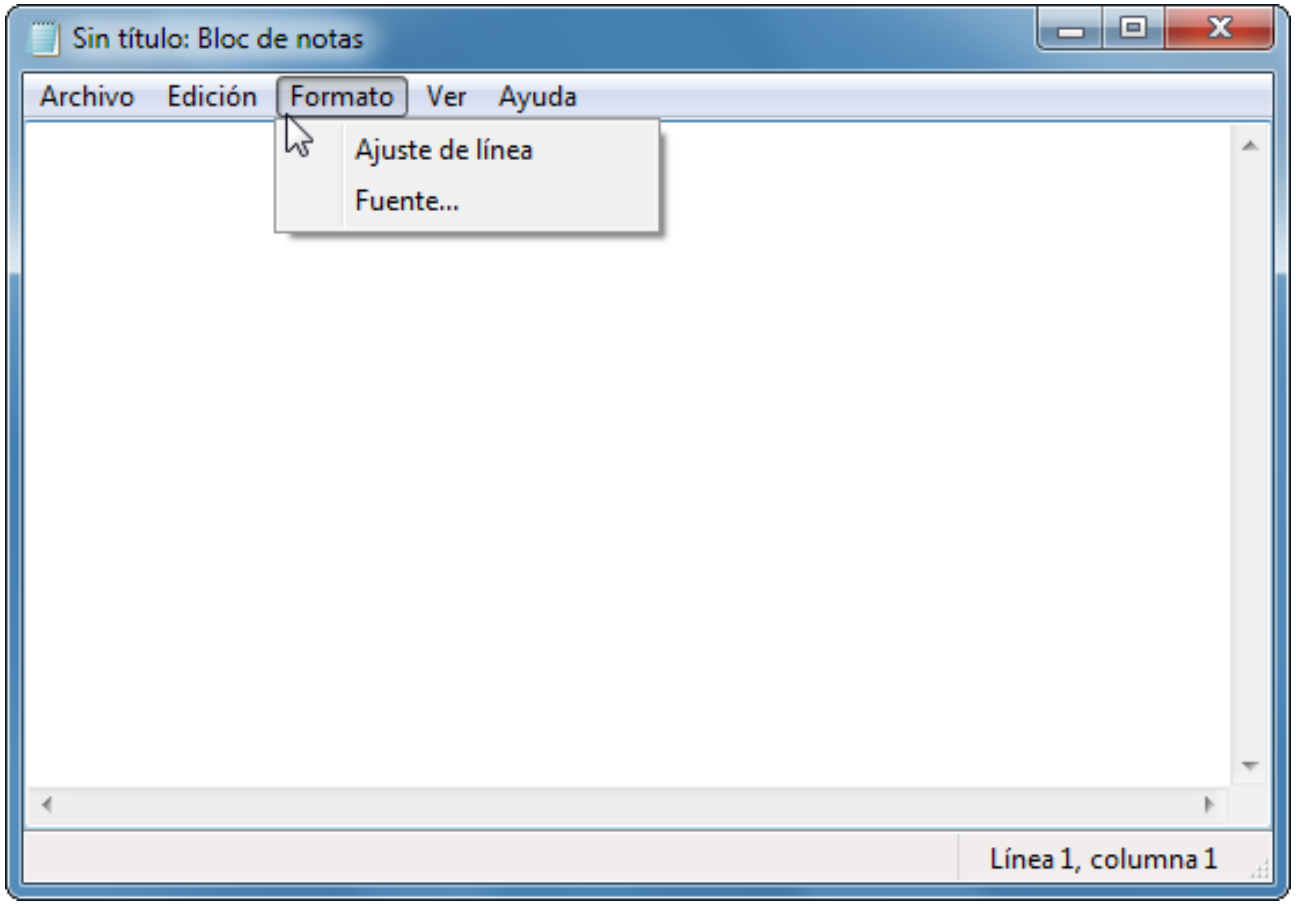
- JRE Java Runtime Enviroment – Entorno de ejecución de Java

- JRE está formado por Java Virtual Machine (Máquina Virtual de Java) y escencialmente sirve para ejecutar programas o aplicaciones creadas en lenguaje Java.

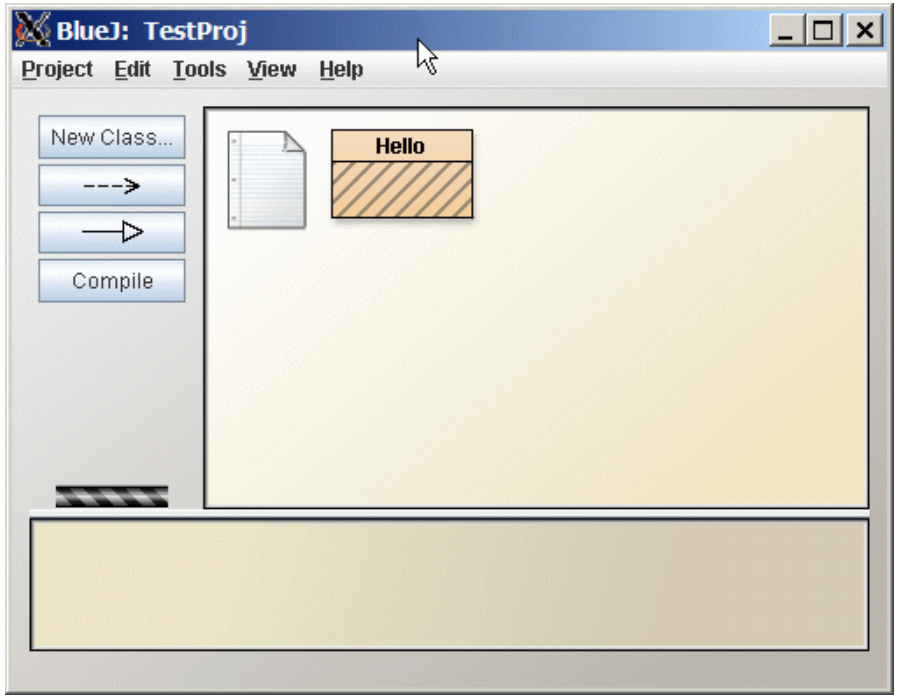
- Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo

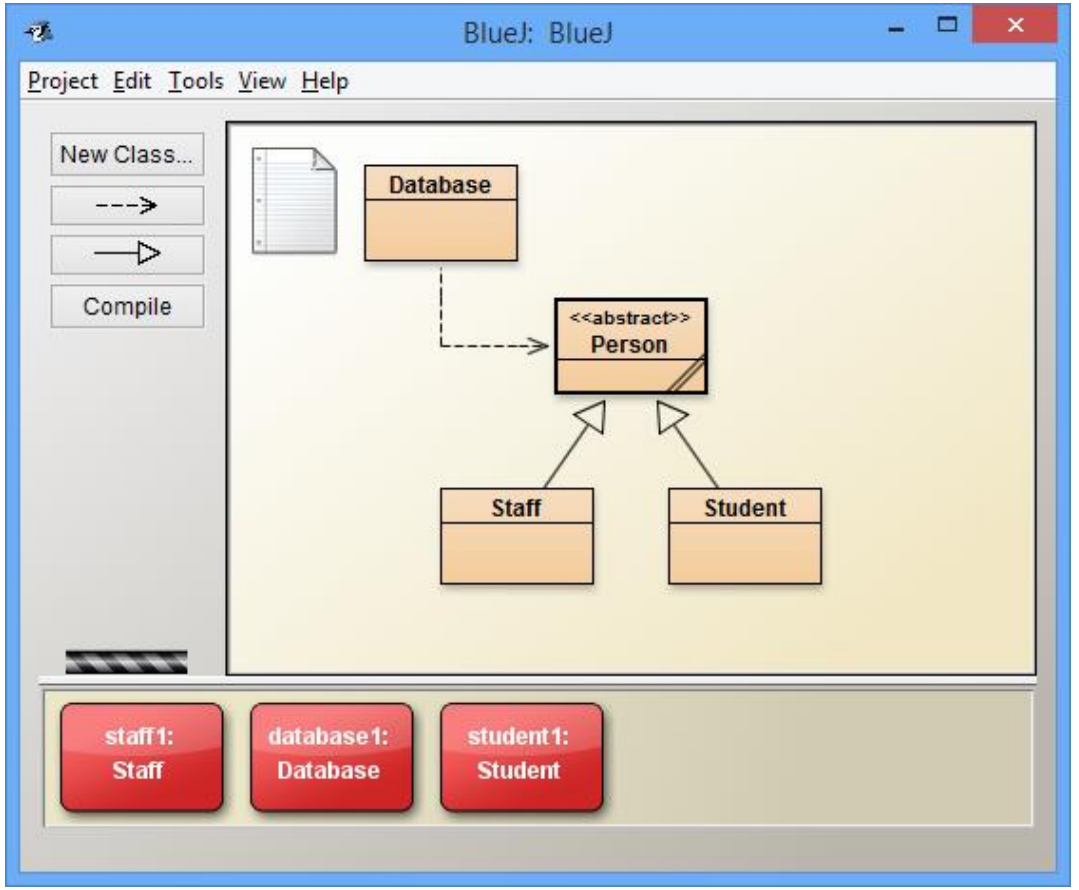
Aplicaciones creadas en Java

Ambientes de trabajo



Ambientes de trabajo





BlueJ

- BlueJ es un entorno integrado de desarrollo muy sencillo de uso, pensado para aprender a programar en Java.

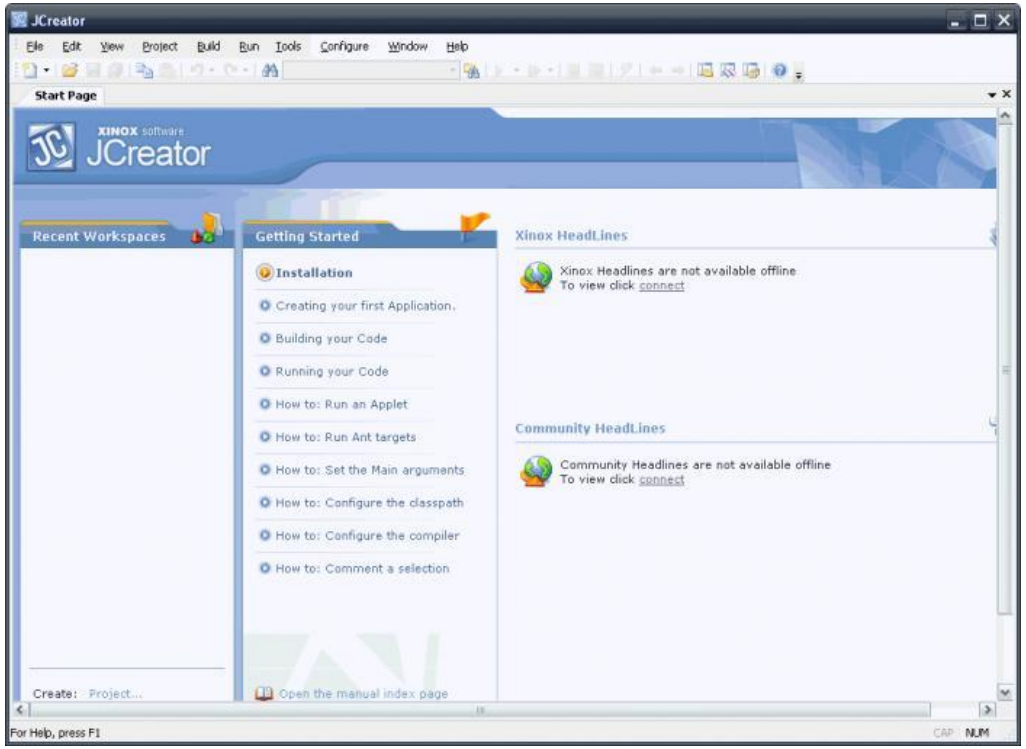
- Es recomendable para estudiantes por el diseño de interfaz gráfica.

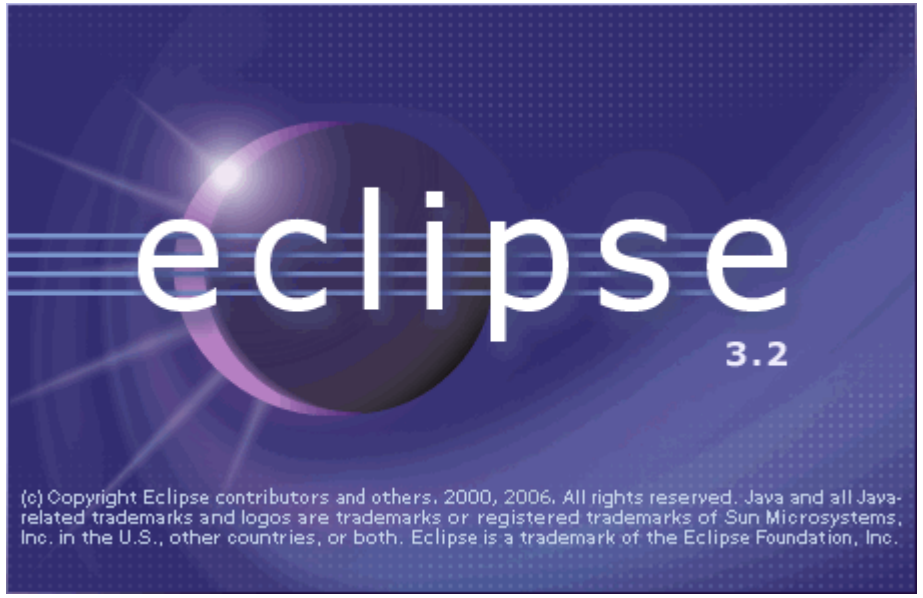
- Las principales ventajas de BlueJ son:
  - Es gratuito
  - Es fácil de usar
  - Es ligero (no requiere una máquina muy potente)
  - Puede ser portable

Instalación

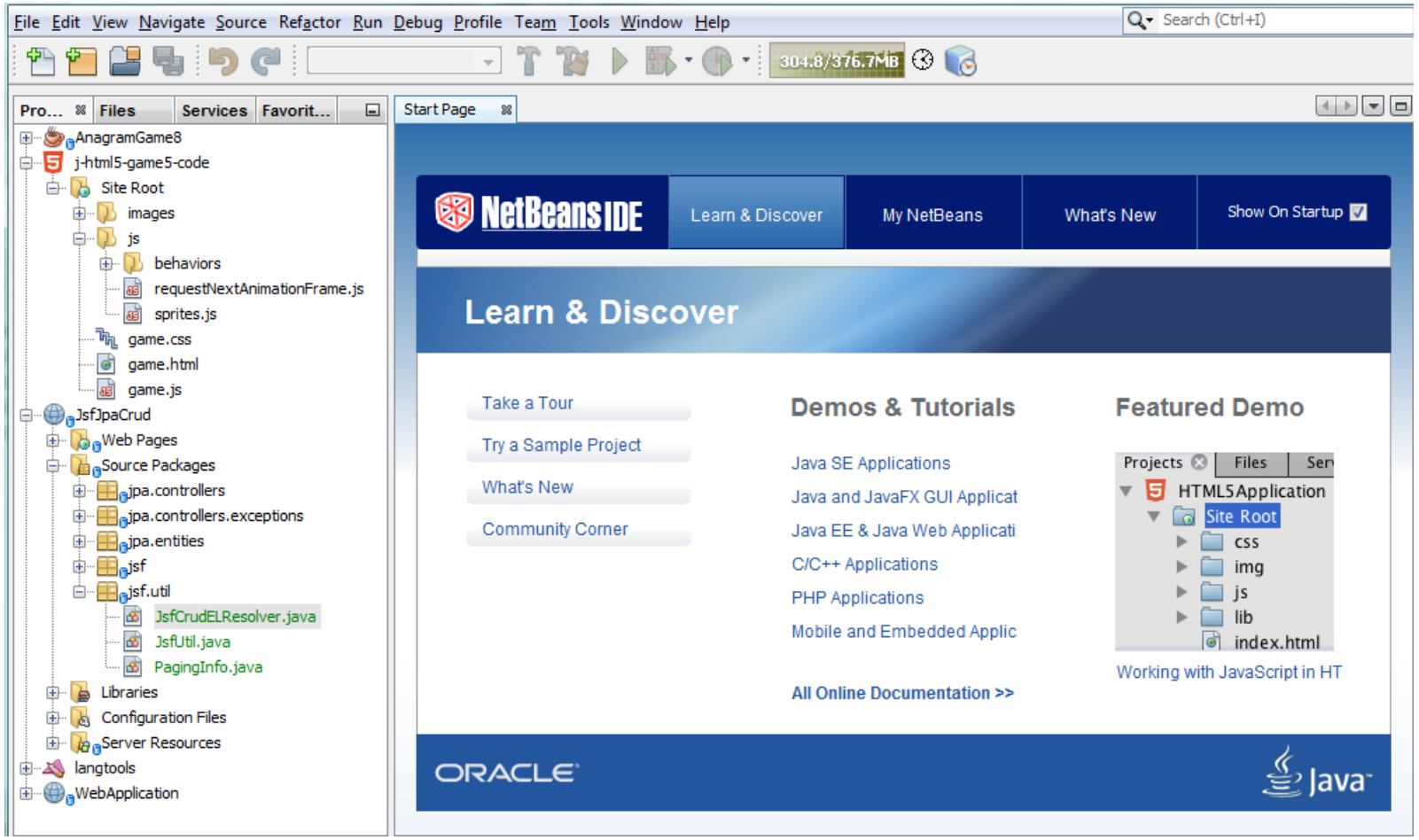
1. Ingresar a http://www.bluej.org/
2. Descargar el ejecutable para Windows
   a. Vaya a la sección "download and install“
   b. Descargue la última versión oficial
3. Ejecute el archivo que ha descargado y siga las instrucciones de instalación
4. Puede encontrar instrucciones detalladas de instalación en http://www.bluej.org/tutorial/tutorial-spanish-201.pdf

Ambientes de trabajo





Ambientes de trabajo



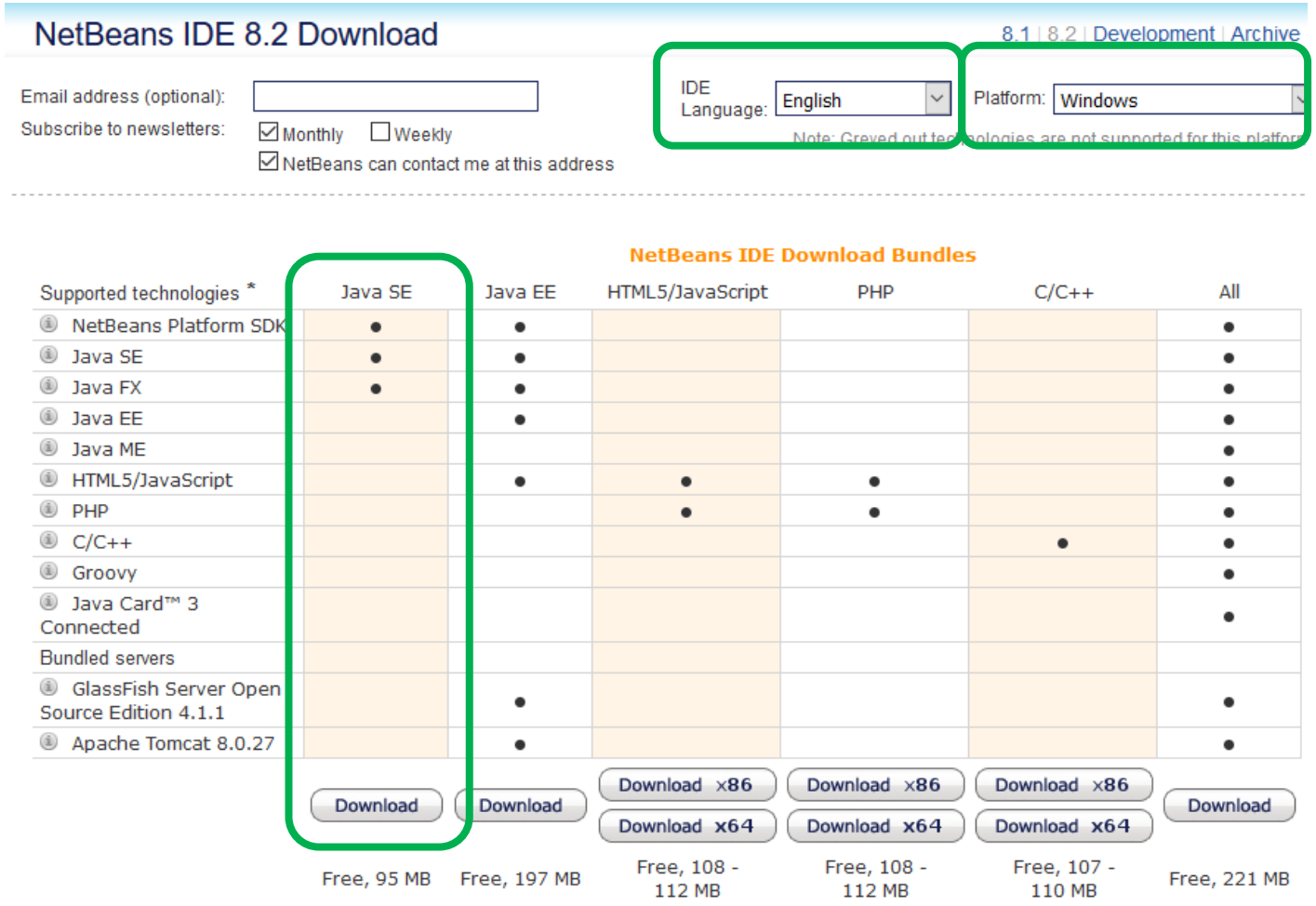
Netbeans

- Es un entorno de desarrollo, es decir, es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. NetBeans es un producto libre y gratuito sin restricciones de uso.

Instalación de Netbeans

- Ingresar a la página : https://netbeans.org/downloads/index.html
- Ejecute el archivo descargado y sigas las instrucciones de instalación.

Opciones de descarga

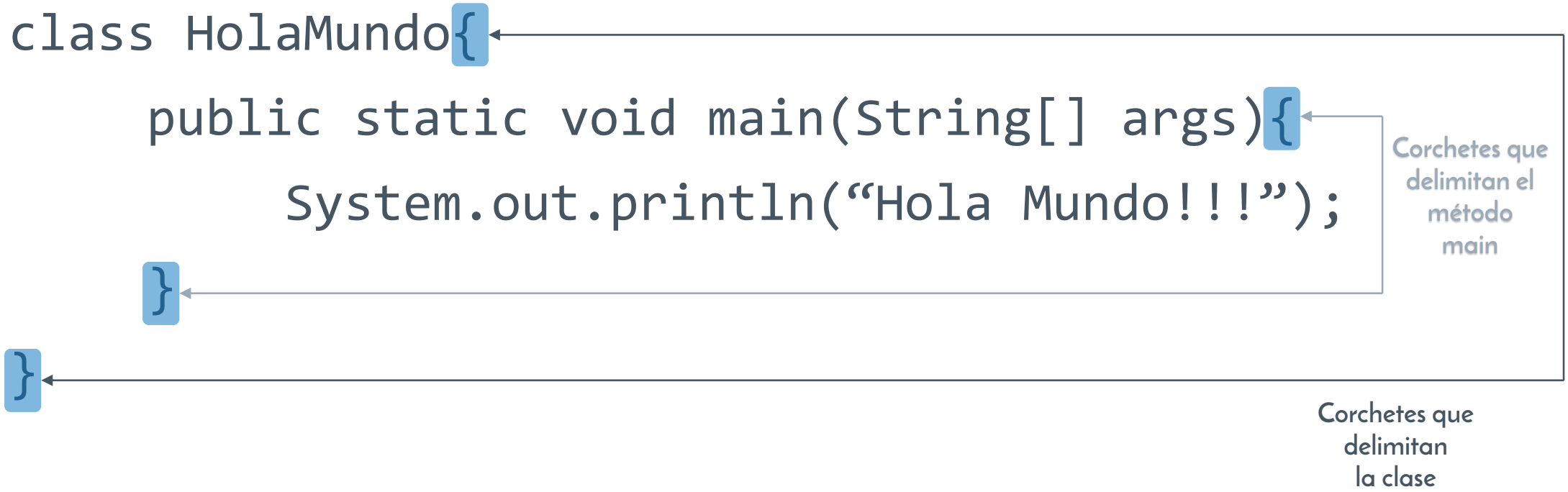


Estructura básica de un programa

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

Después de la palabra **class**, se define el nombre de la clase, el cuál por convención se escribe combinando mayúsculas y minúsculas. Si es una sola palabra se empieza con mayúscula.

La palabra reservada **class** se utiliza para definir la clase.



Los corchetes **{ }** delimitan la extensión, tanto de una clase, como de un método.

El objeto System.out

- El objeto System.out nos sirve tener un flujo de salida hacia el monitor. Este objeto cuenta con dos métodos:
  - System.out.print
  - System.out.println

- El primero imprime en la consola el valor del argumento que le pasamos. El segundo hace lo mismo, pero agrega un salto de línea al final.

La clase Scanner

- La clase Scanner permite leer datos a través del teclado.
- Para hacer uso de esta clase, es necesario importarla desde el paquete: java.util
- Su sintaxis sería la siguiente:

```
import java.util.Scanner;
```


```
import java.util.Scanner;

class PideDatos{
	public static void main(String []args){
		Scanner teclado = new Scanner(System.in);
		System.out.print("Ingresa tu nombre: ");
		String nombre;
		nombre = teclado.nextLine();
		System.out.println("Hola " + nombre);
	}
}
```

Métodos next()

- Los métodos next() sirven para la lectura de datos a través del teclado y puede tener las siguientes formas:
  - **next()** solo lee hasta donde encuentra un espacio (hasta un espacio).
  - **nextLine()** lee todo incluyendo espacios (hasta un enter).
  - **nextInt()** lee un número entero
  - **nextDouble()** lee un número de tipo double
  - **nextFloat()** lee un número de tipo float
  - **next().charAt(0)** lee un caracter

Definición de variables

Podemos definir variables en cualquier parte del código simplemente indicando el tipo de datos y el nombre de la variable (identificador).

- Identificadores válidos son:
  - fecha
  - iFecha
  - fechaNacimiento
  - fecha_nacimiento
  - fecha3
  - _fecha
- Identificadores NO válidos son:
  - 3fecha
  - fecha-nacimiento
  - fecha+nacimiento
  - -fecha

Constantes

En Java, se utiliza la palabra clave *final* para indicar que una variable debe comportarse como si fuese *constante*, significando con esto que no se permite su modificación una vez que haya sido declarada e inicializada.

Por ejemplo:

```
final float PI = 3.14159;
```

Comentarios

En Java hay tres tipos de comentarios:

```
//  comentarios para una sola línea

/* comentarios de una o más líneas   */

/** comentario de documentación, de una o más líneas*/
```

Tipos de datos

Tipo	Valor por default	Longitud
byte	0	8 bits
char	\u000	16 bits
short	0	16 bits
int	0	32 bits
long	0	64 bits
float	0.0	32 bits
double	0.0	64 bits
boolean	false	1 bit
String	null	

Palabras reservadas

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

Operadores aritméticos

Operador	Operación
+	Suma
-	Resta
*	Multiplicación
/	División
%	Resto
=	Asignación

Incremento ‘++’ y Decremento ‘--’

Pre: Se incrementa/decrementa y después se evalúa la expresión.

```
++variable;

--variable;
```

Post: Se evalúa la expresión y luego se incrementa/decrementa.

```
variable++;

variable--;
```

Operadores relacionales

Operador	Significado	Ejemplo
==	Igual a	a == b
!=	No igual a	a != b
>	Mayor que	a > b
<	Menor que	a < b
>=	Mayor o igual que	a >= b
<=	Menor o igual que	a <= b

Operadores lógicos

Operador	Significado
!	Negación
||	O lógica
&&	Y lógica

Comparación de cadenas

› Método **equals()**

Este método compara los contenidos y retorna true o false según estos sean iguales o no.

If

Permite tomar una decisión dependiendo de un resultado (verdadero o falso).

```
if(condición){
     //Código 1;
}else if(condición 2){
     //Código 2;
}else{
     //Código 3;
}
```

Switch

Permite ejecutar una de varias opciones dependiendo del valor que tenga cierta expresión

```
switch (expresión){
        case x:
                //Código para x;
        break;
        case z:
                //Código para z;
        break;
        default:
                //Código para default;
        break;
}
```

Donde x y z son expresiones y si coincide con expresión entra a ese caso.

While

/*Mientras la condición sea verdadera, se ejecutan las instrucciones*/

```
while(condición)
{
    //instrucciones
}
```

Do While

/*Ejecutas una vez, y mientras la condición sea verdadera sigues ejecutando*/

```
do{
      sentencia simple o compuesta;
}while( condición );
```

```
int a = 0;
do{
      System.out.println("la variable a vale : " + a);
      a = a + 1;
}while ( a < 5 );
```

For

/*desde que a vale # si cumple la condición ejecuta e incrementa*/

```
int a;
for(a = #;condición para a; incremento para a ){
      Sentencia simple o compuesta a repetir
}
```