Java

Lenguaje de programación orientado a objetos con el fin de conseguir modularidad en la estructura e implementación.

Principio Wora = escríbelo una vez ejecútalo en cualquier parte

* Diferentes dispositivos
* No recompilar
* El código es portable
* Seguridad
* Evita perdida de memoria

Interrelacion de entidades y conformación de conjuntos –

Aplicaciones cliente

Aplicaciones cliente servidor

Aplicaciones web

Jvm = java virtual machine

C++ = Punteros – espacios de memoria - permiten acceso a los archivos de sistemas

Java no usa punteros

Caracteristicas de java

Es compilada e interpretado – lenguaje pseudo compilado

Seguridad

Portabilidad

Sencilles

Fácil de aprender

Multiproceso

Distribución

TIPOS DE DATOS

Datos primitivos

Datos enteros – numérico entero byte int long short

Decimales o comas decimales doublé float

Bolean = tabla de la verdad

Tipo de dato char = caracteres

LIBRERIAS

clases con metodos y atributos

* Lang\_ clases fundamentales de java
* Útil: coleciones listas arreglos conjuntos
* Io: input out entrada y salida de datos
* math : cálculos matematcos

Variable: espacio en memoria corresponde a un dato que tendrá un valor que puede ser modificado, la variable se le coloca un identificador, la variable debe ser declarada tipo de dato y nombre

Int nombre\_variable

Palabras reservadas:

Return new, default extends import

METODOS: es un subprograma de un programa principal,

Parametros: cantidad y tipo de valores que recibe un método para su ejecución

Característica Strongly typed: cantidad y tipos de datos sean iguales

Metodoi main: meodo principal se plasman ionstrucciones que se necesita resolver en ejecución

estrudctira public static void (método estatico publico y no retoma valor) (string[] args) método llamada cuando la maquina virtual ejecuta el programa y este ejecuta las instrucciones

ESTRUCTURA DE UN PROGRAMA EN JAVA

archivo fuente: base de un programa en java donde se codifica solo se declara una cklas epublica con el mismo nombre del archivo fuente suseptible a mayúsculas.

Archivo fuente extencion.java

Archivo.class (se genera automáticamente como un temp (base de jvm para compilar)

Comando java

Ctr + s = guardar, el titulo esta en nengrita cuando no esta guardado

Ctr + R = modificar una palabra y sus respectivas copias en todo el proyecto

Ctr

Shift + v = pega d eforma ordenado líneas de comando

Ctr + shft + duplica la línea en la que estemos hubicados al final

Ctrl + f3 selecciona todas las palabras con sus duplicados en todo el proyecto F3 va moviéndose entre las palabras

Ctrl + e = borrar línea completa

Ayuda/métodos abreviados de teclado

**SOUR CUT O ATAJOS**

Sour + tab = System.out.print(“”) imprime el parámetro

Alt + ins = menu para generar código y métodos

**EXPRESIONES PARA SALIDA DE DATOS X CONSOLA**

System.out.println(“”) imprime el parámetro + salto de línea

System.err.println(“”) imprime el parámetro en rojo + salto de línea

\ = alt + 92

\” = permite colocar comillas dentro del texto a mostrar

\n = inserta un salto de línea ejmp. System.err.println(“Hola \n Mi nombre es Wilmer”)

\t=coloca un espacio tabulado System.err.println(“\t Hola Mi nombre es Wilmer”)

+ = utilizado para concatenar ej. Int NumeroTelefonico = 6627313

System.out.println(“\t Hola Mi nombre es Wilmer y mi num tel es: ”+NumeroTelefonico)

Sout + tab = escribe automáticamente la línea completa para la salida de datos

**Captura de texto escrito por teclado**

Scanner Teclado = New Scanner(System.in);

String Nombre = Teclado.Nextlinea()

Int Edad = Teclado.Nextint()

Double Promedio = Teclado.NextDoble()

**Entradas y salidas por mensajes**

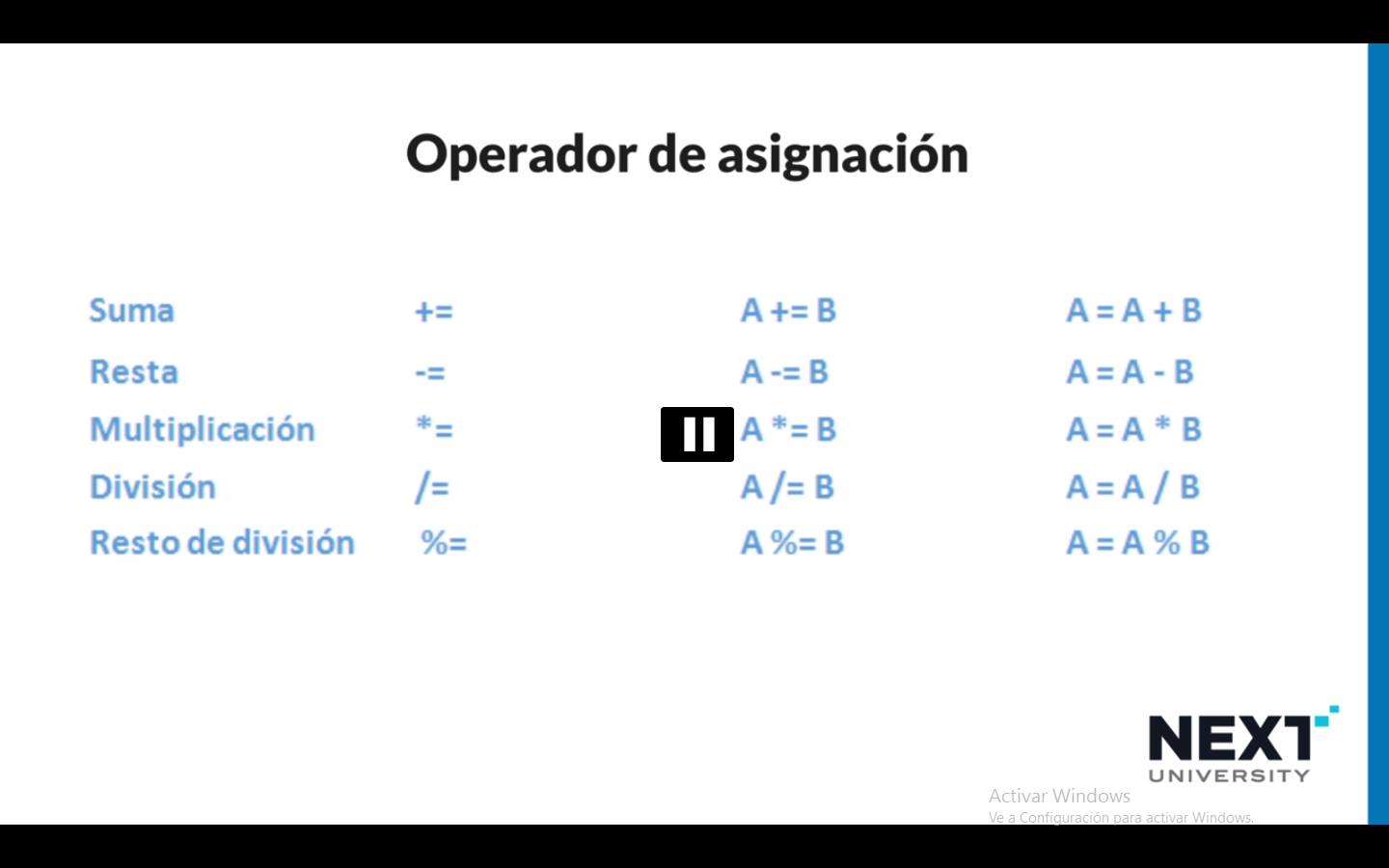
string palabra JOptionpane.showImputDialog(null,””) == entrada de datos

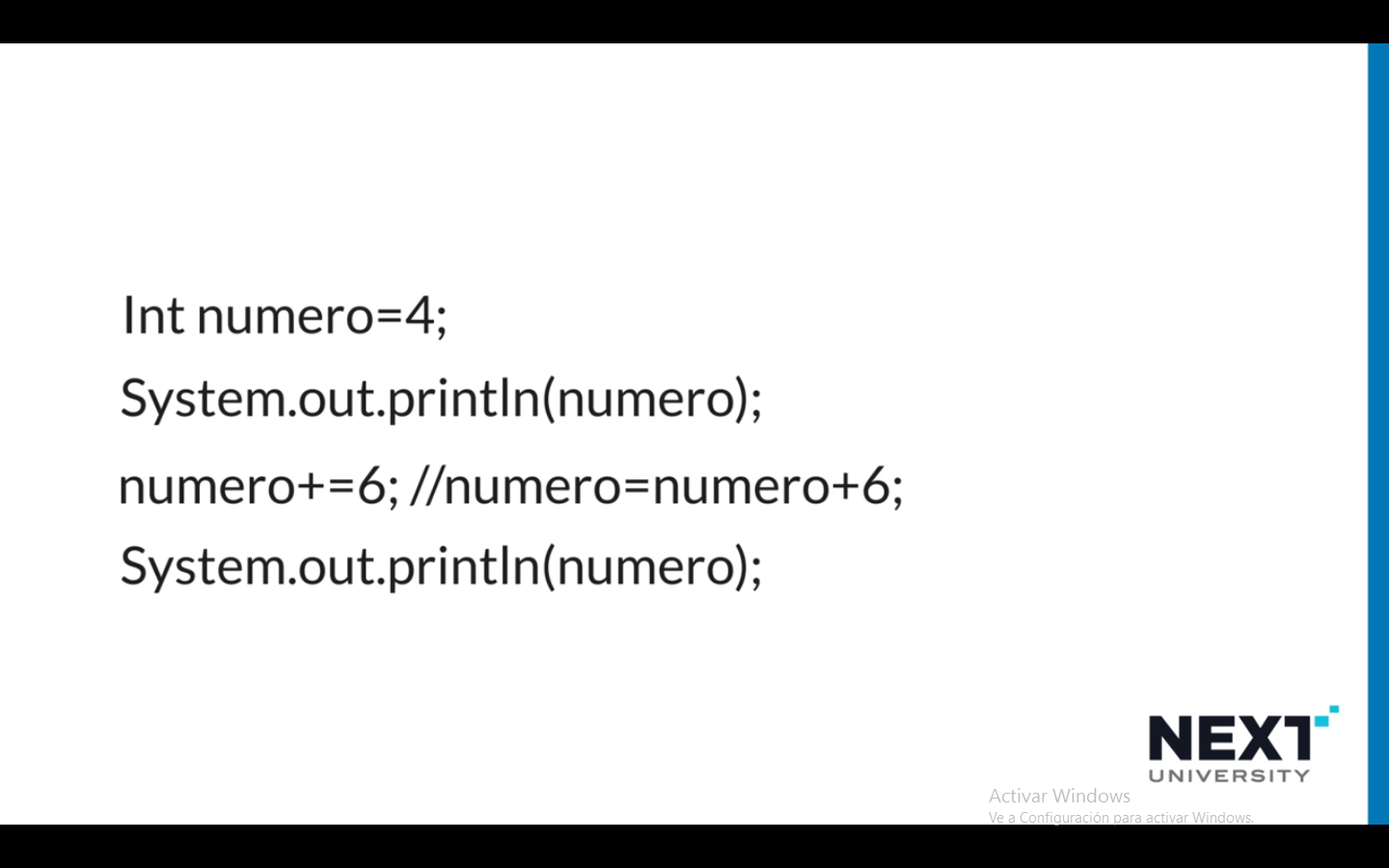
JOptionpane.showMessageDialognull,(null,“La palabra agregada es”+Palabra) == Salida de Datos

Int numero=Integer.ParseInt(JOptionpane.showImputDialog(null,””)) == alamacener datos números en variables

OPERADORES

Modulo % devuelve el sobrante de una operación ejemplo: (11%5) el resultado sera 1





CONDICIONES

If(a<b){

JOptionPane.ShowMessageDialog(null, “a es menor que b”);

}else if(a=b){

JOptionPane.ShowMessageDialog(Null, “a es es igual a b”);

}else{

JOptionPane.ShowMessageDialog(null, “a es mayor que b”);

SWITCH estructura condicional

Int opción=Integer.ParseInt(JOptionPane.ShowInputDialog(null, “Menu de Opciones\n1.Opcion 1\n2.Opcion 2\n3. Opcion 3”));

Swith(opción){

Case 1:

JOptionPane.ShowMessageDialog(null, “Seleccionaste la Opcion 1”);

break;

Case 2:

JOptionPane.ShowMessageDialog(null, “Seleccionaste la Opcion 2”);

break;

Case 2:

JOptionPane.ShowMessageDialog(null, “Seleccionaste la Opcion 3”);

break;

Default:

JOptionPane.ShowMessageDialog(null, “La opción no es valida”);

Break;

}

CICLOS FOR

For(int i=1;i<5;i++){

JOptionPane.ShowDialog(null, “se repetirá 5 veces”

}

CICLOS WHILE – DO WHILE

INT A = 0

WHILE(A<5){

A++

System.Out,println(“Hola”);

}

D0{

A=teclado.nextline()

}while(a<5);

Sistema.out.println(“Terminado”);

EXEPCIONES PARA PREVENIR ERRORES

TRY{

System.out.println(“Ingresa un numero”);

Int num=teclado.nextLine();

}catch(excepción e){

System.out.println(“numero incorrecto”);

}finally{

System.out.println(“fin del programa”);

}

EXEPCIONES ESPECIFICAS

}catch(NullPointerException e){

System.out.println(“No digitó Datos”);

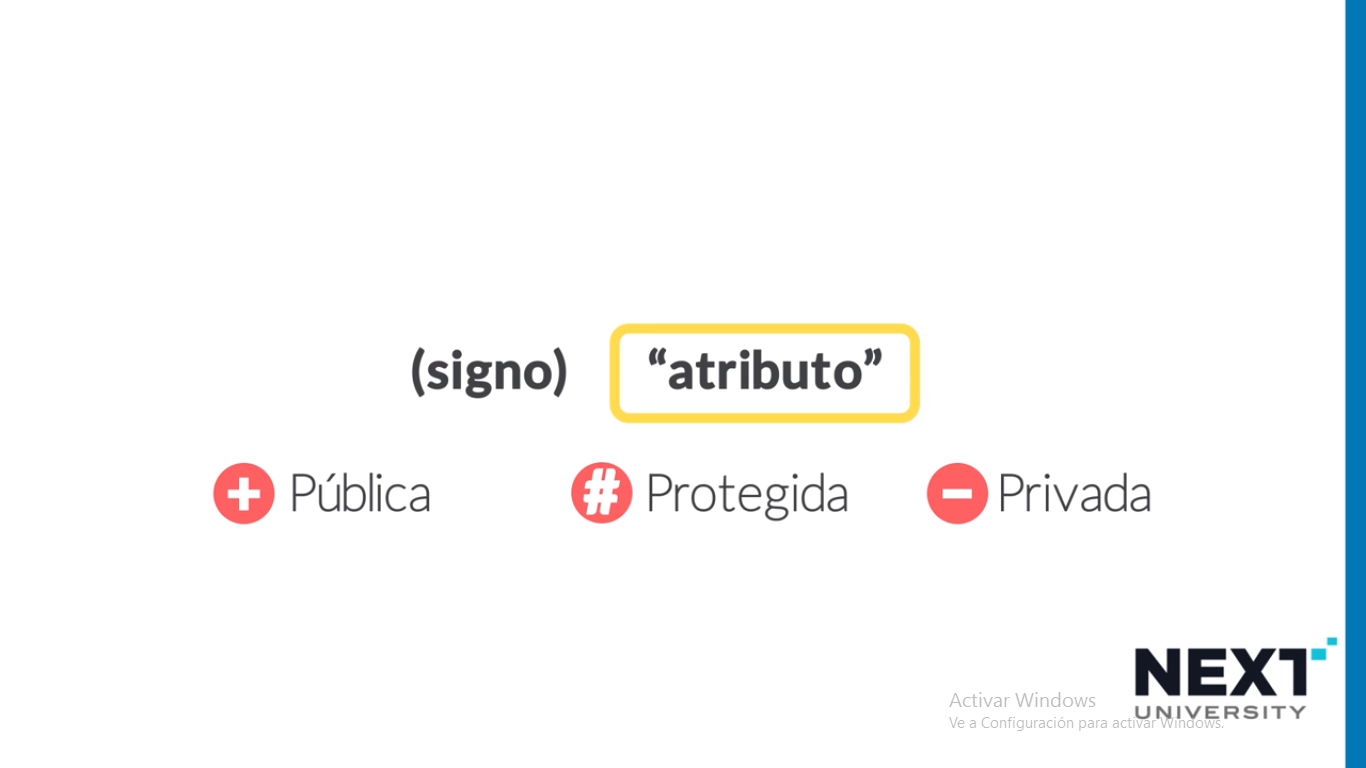
}

CLASES: es el nombre que se le da a la clasificación de Un grupo de objetos con características comunes, atributos, operaciones y sus relaciones con otros

OBJETO: elemento de la clase

ATRIBUTO: características que describen el estado de un objeto o una clase

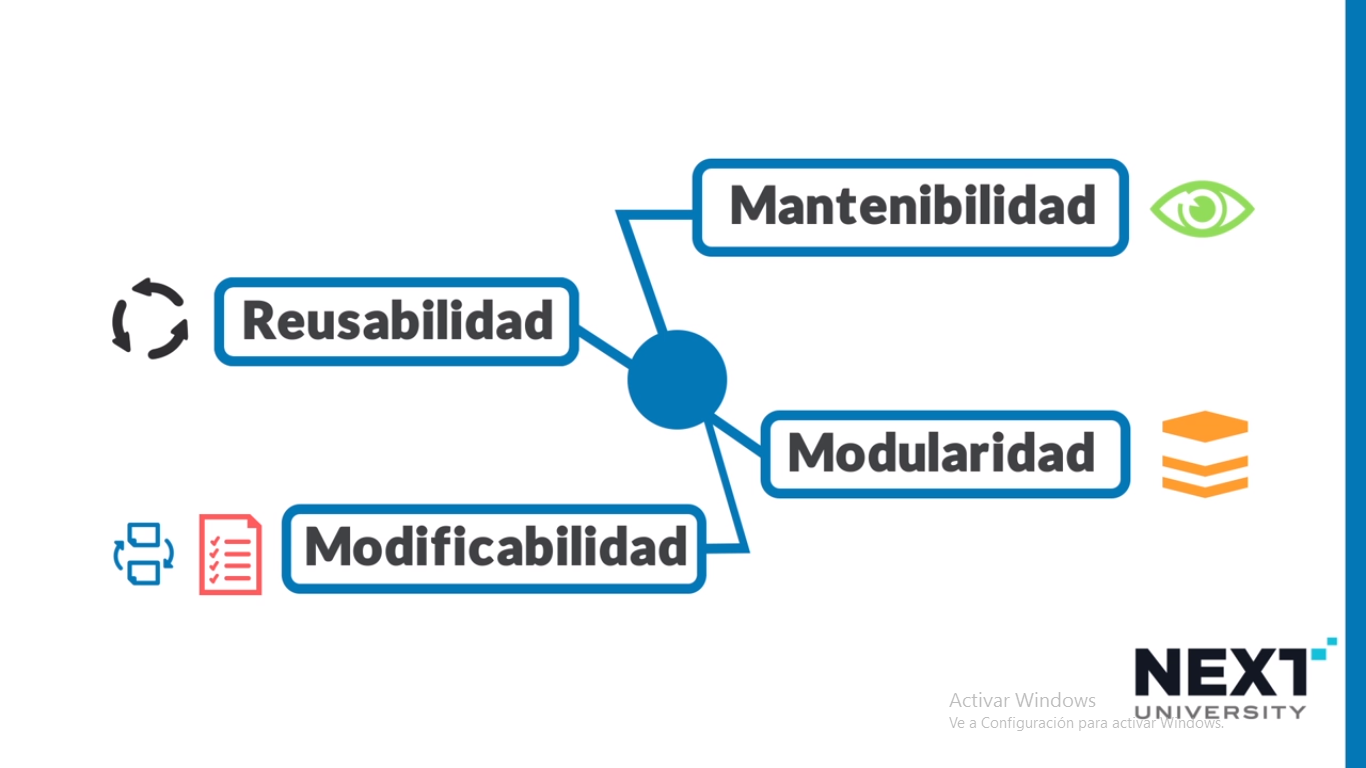
Visibilidad de atributos



DATOS PRIMITIVOS: son aquellos que no se pueden descomponer

GRANULARIDAD: complejidad de detalles de un proyecto

VENTAJAS DE LA PROGRAMACION ORIENTADA A OBJETOS



Reusabilidad: reutilizar las clases en todo el programa y proyectos acelera produccion de proyectos

Mantebilidad: sencillos de leer y comporender; ocultan detalles de implementación dejan visibles los de mayor relebancia

Modificabilidad: añadir suprimir y modificar objetos de forma mas sencilla

Modularidad: permite probar e identificar y corregir los errores de forma mas fácil al estar dividido el problema en partes

VENTAJAS DE LA PROGRAMACION ORIENTADA A OBJETOS

Dificultad de aprendizaje por el paradigma nuevo

Uso de recursos robisto

Tamaño de los programas grandes piesas de proyectos que no se usaran

CREAR CLASES (click derecho sobre el paquete / nuevo / java class)

Carpeta / paquete / clases

La estructura del código de una clase nueva es:

Package Animales;

Public class Caballo { }

CREAR METODOS (se crean automáticamente al crear un proyecto si esta activa la opción)

Public = lo puede usar otra clase

Static = define las propiedades propias de la clase

Void = no devuelve parámetros ya que se le asignan los tipo de parámetros que devuelven

Main = nombre del método principal de una clase ( comparado con index.html)

**si el método va a recoger parámetro se debe crear un vector**

string[]args{ } Vector que recogerá los argumentos pasados en la ejecucuion del programa

**ejemplo**: Public Static Void Main string[]args{ }

**PARA IMPLEMENTARLA SE CREA UNA CLASE EXTERNA Y LUEGO SE IMPLEMENTA**

**Crear Los Atributos De Una Clase**

Se declaran los atributos como se declaran las variables especificando que son tipo private o privadas es decir que solo podrán ser usadas por esta clase.

private int NumDePatas;

private String Tipo;

Cada atributo tiene un método set que es un modificador y un get retorna valor

Para crear los métodos nos ubicamos debajo de las declaraciones de los atributos y usamos alt + ins o click derecho y la opción insertar código para que salga el menú de los generadores automáticos y se selecciona getter and setter

**Imprementar una método en una clase hacer Instancia**

1. Se coloca el nombre de la clase
2. Se coloca el nombre de un objeto a crear
3. Se le asigna con = el atributo new + el nombre de la clase sin darle valor
4. Se atiende la sugerencia de netbeans y se importa la clase que se la esta haciendo instancia

Import Animales.Caballo;

Caballo caballo=new Caballo ();