

---

---

# Clase 3: Variables y tipos

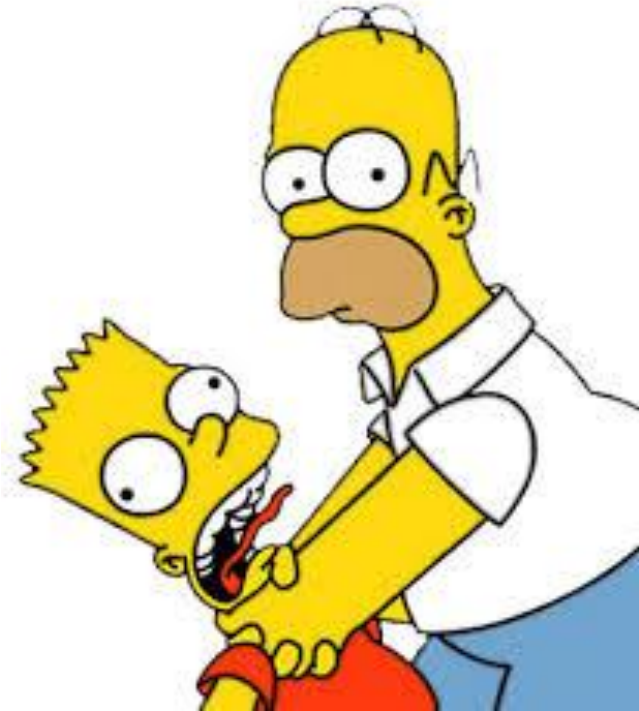
— Programación y Laboratorio III —

---

---

# Agenda

- **Variables**
  - Clasificación
  - Nomenclatura de identificador
- Tipos primitivos
  - Categorización
  - Ejemplos
  - Valores por defecto
- Java Virtual Machine
  - Secciones
  - Tareas principales
  - Garbage Collector
- Tipo objeto
  - Creación
- Diferencias entre variables de tipo objeto y tipo primitivo
- Clases "Wrapper"
  - Ejemplos
  - Unboxing y Autoboxing



# Variables

- Una variable es un identificador que representa una palabra de memoria que contiene información.
- Sintaxis:

`<tipo> <identificador>;`

`<tipo> <identificador> = <valor>;`

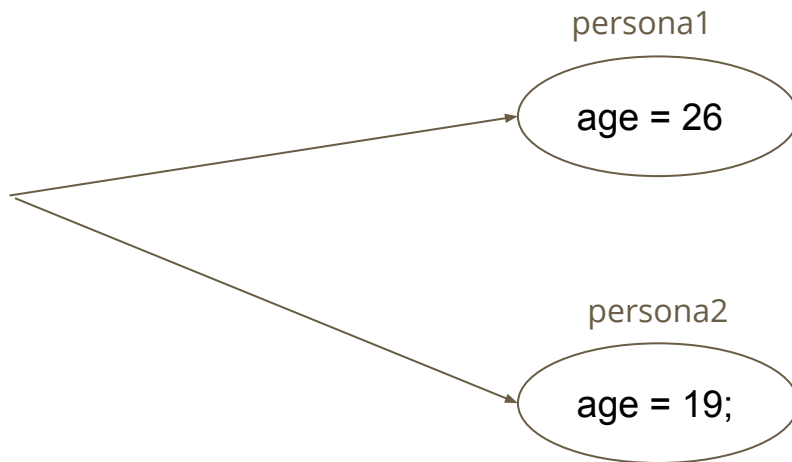


# Variables - Clasificación

- **Variables de instancia:** los valores que pueden tomar son únicos para cada instancia.

Ejemplo:

```
class Person {  
    int age;  
}
```



# Variables - Clasificación

- **Variables de clase:** se declaran con el modificador static para indicarle al compilador que hay exactamente una copia de la variable, y es compartida por todas las instancias.

Ejemplo:

```
class Bicicleta {  
  
    static int cantRuedas = 2;;  
  
}
```



# Variables - Clasificación

- **Variables locales:** la determinación viene desde la ubicación en donde la variable fue creada, es decir, local al método. Sólo es visible al método donde fue declarada y no puede ser accedida desde el resto de las clases.

Ejemplo:

```
public void incrementarContador() {  
  
    int contador = 0;  
  
    contador = contador + 1;  
  
}
```

# Variables - Clasificación

- **Parámetros:** se pasan entre métodos.

Ejemplos:

```
public void incrementarContador(int contador) {  
  
    contador = contador + 1;  
  
}
```



# Variables - Nomenclatura de identificador

Reglas:

- 1) Siempre debe comenzar con una letra.
- 2) Los caracteres subsecuentes pueden ser letras, dígitos, "\$", o "\_"
- 3) Usar palabras descriptivas y no abreviaciones.
- 4) Los nombres no deben contener palabras reservadas en Java.
- 5) Si el nombre contiene más de una palabra, se capitaliza la primer letra de cada palabra subsecuente.



# Variables - Nomenclatura de identificador

Ejemplos:

Identificadores válidos	Identificadores no válidos
<code>customerValidObject</code>	<code>7world</code>
<code>\$rate</code> , <code>fValue</code> , <code>_sine</code>	<code>%value</code>
<code>happy2Help</code> , <code>nullValue</code>	<code>Digital!</code> , <code>books@manning</code>
<code>Constant</code>	<code>null</code> , <code>true</code> , <code>false</code> , <code>goto</code>

# Variables - Nomenclatura de identificadores

<code>abstract</code>	<code>default</code>	<code>goto</code>	<code>package</code>	<code>this</code>
<code>assert</code>	<code>do</code>	<code>if</code>	<code>private</code>	<code>throw</code>
<code>boolean</code>	<code>double</code>	<code>implements</code>	<code>protected</code>	<code>throws</code>
<code>break</code>	<code>else</code>	<code>import</code>	<code>public</code>	<code>transient</code>
<code>byte</code>	<code>enum</code>	<code>instanceof</code>	<code>return</code>	<code>true</code>
<code>case</code>	<code>extends</code>	<code>int</code>	<code>short</code>	<code>try</code>
<code>catch</code>	<code>false</code>	<code>interface</code>	<code>static</code>	<code>void</code>
<code>char</code>	<code>final</code>	<code>long</code>	<code>strictfp</code>	<code>volatile</code>
<code>class</code>	<code>finally</code>	<code>native</code>	<code>super</code>	<code>while</code>
<code>const</code>	<code>float</code>	<code>new</code>	<code>switch</code>	
<code>continue</code>	<code>for</code>	<code>null</code>	<code>synchronized</code>	

# Agenda

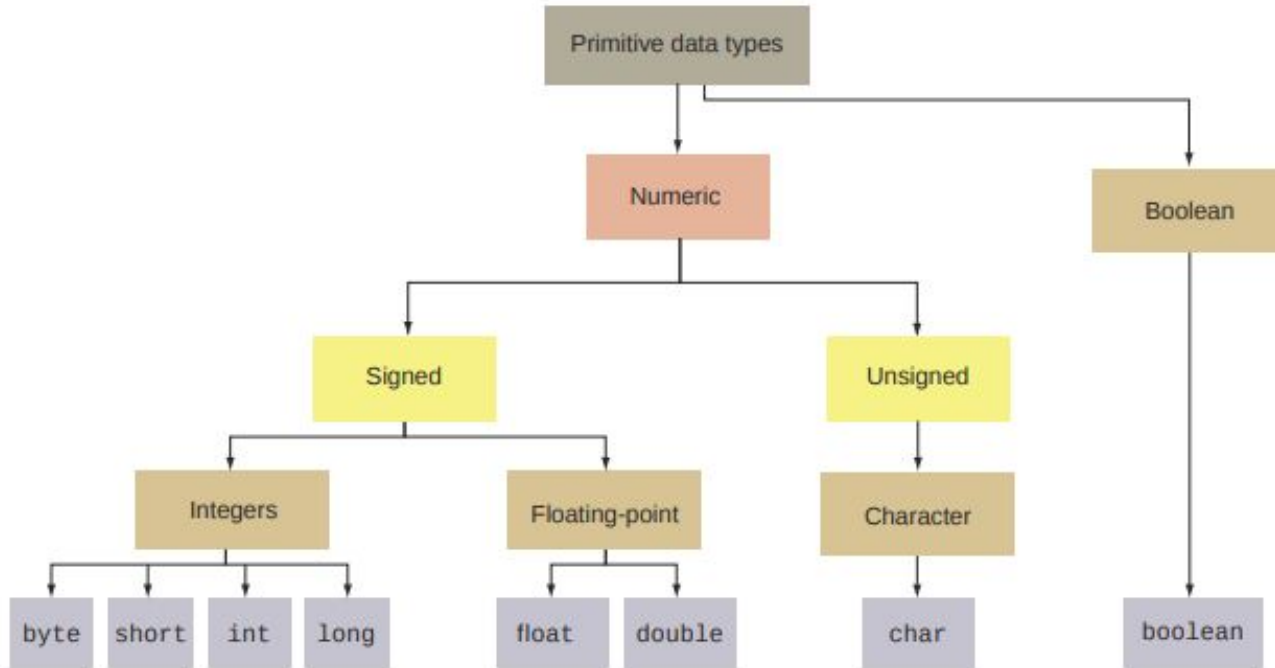
- Variables
  - Clasificación
  - ~~Nomenclatura de identificador~~
- **Tipos primitivos**
  - Categorización
  - Ejemplos
  - Valores por defecto
- Java Virtual Machine
  - Secciones
  - Tareas principales
  - Garbage Collector
- Tipo objeto
  - Creación
- Diferencias entre variables de tipo objeto y tipo primitivo
- Clases “Wrapper”
  - Ejemplos
  - Unboxing y Autoboxing



# Tipo primitivo

- Son el tipo más simple de la programación orientada a objetos.
- En Java ya están predefinidos.
- Los nombres de los tipos primitivos describen el valor que pueden almacenar.
- Ocho tipos de primitivos:
  - `char`
  - `byte`
  - `short`
  - `int`
  - `long`
  - `float`
  - `double`
  - `boolean`

# Tipo primitivo - Categorización



# Tipo primitivo - Ejemplos

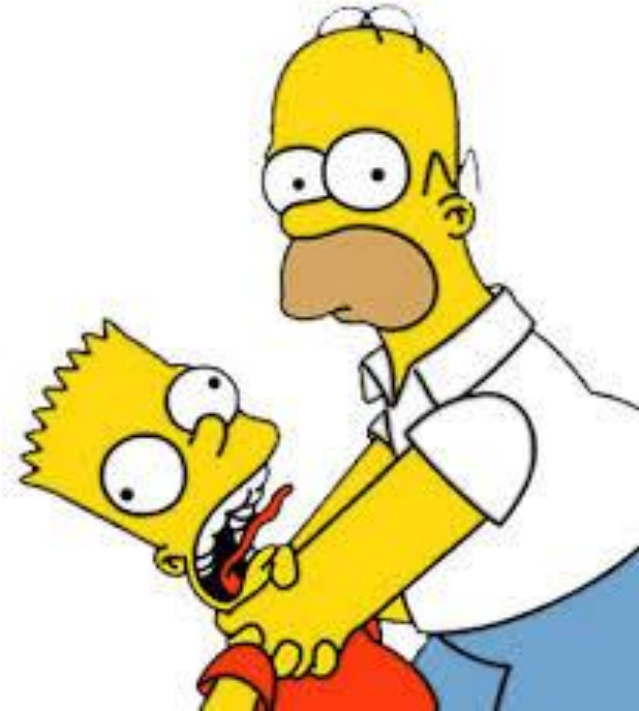
char	byte (8 bits)	short (16 bits)	int (32 bits)	long (64 bits)	float (32 bits)	double (64 bits)	boolean
'a' 'D' '122'	100	1240	48764 0413 0x10B 0b100001011	214748368	20.12F 1765.65f 120.1762	120.176D 120.1762	true false

# Tipo primitivo - Valores por defecto

Tipo	Valor por defecto
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
boolean	false

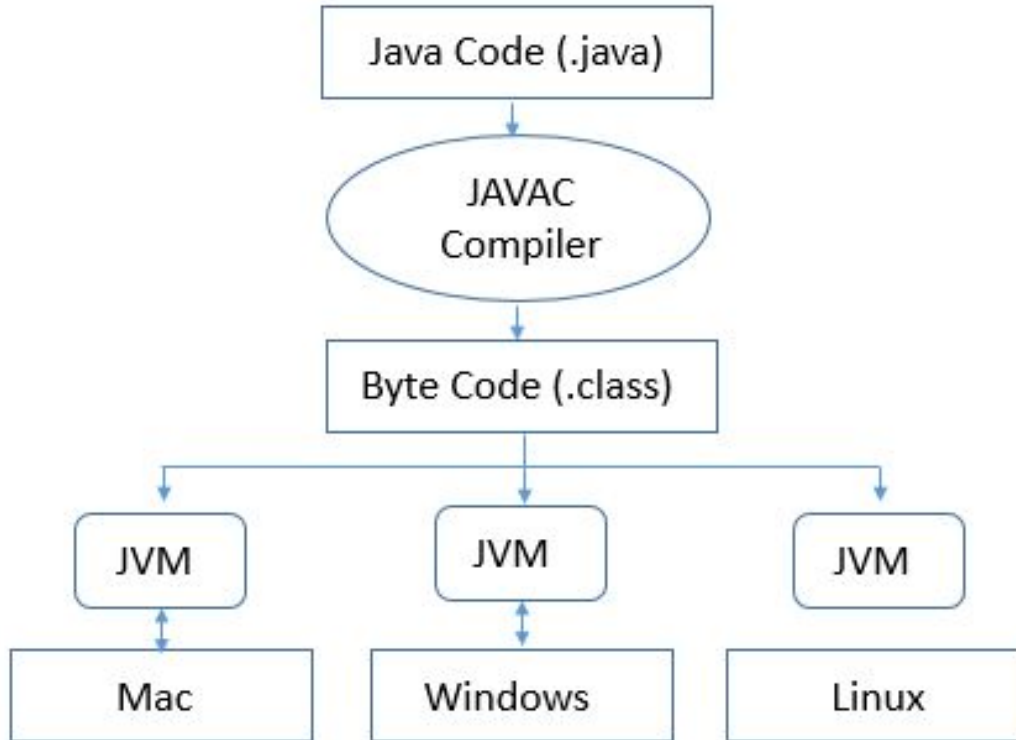
# Agenda

- Variables
  - Clasificación
  - Nomenclatura de identificador
- Tipos primitivos
  - Categorización
  - Ejemplos
  - Valores por defecto
- **Java Virtual Machine**
  - Secciones
  - Tareas principales
  - Garbage Collector
- Tipo objeto
  - Creación
- Diferencias entre variables de tipo objeto y tipo primitivo
- Clases “Wrapper”
  - Ejemplos
  - Unboxing y Autoboxing





# JVM (Java Virtual Machine)



# JVM - Secciones

- **Zona de datos** → donde se almacenan las instrucciones del programa, las clases con sus métodos y constantes. No se puede modificar en tiempo de ejecución.
- **Stack** → El tamaño se define en tiempo de compilación y es estático en tiempo de ejecución. Aquí se almacenan las instancias de los objetos y los datos primitivos (int, float, etc.)
- **Heap** → zona de memoria dinámica. Almacena los objetos que se crean.

# JVM - Tareas principales

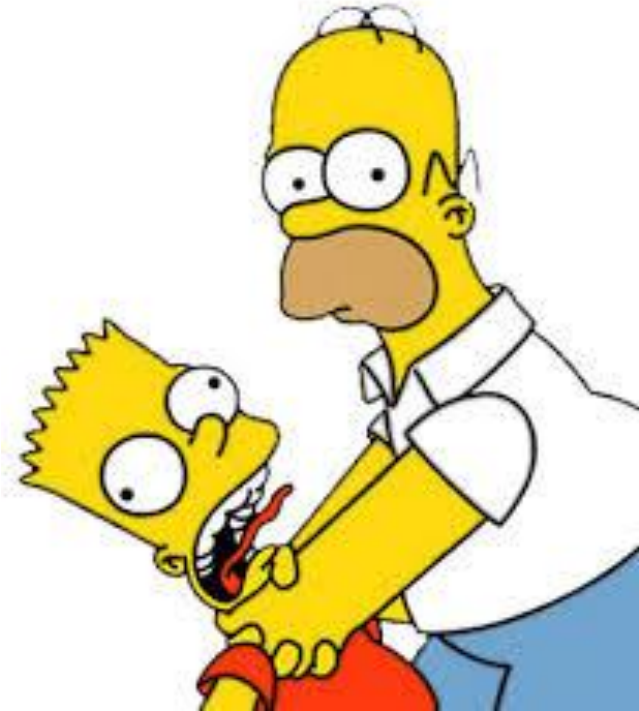
- Reservar espacio en memoria para los objetos creados.
- Liberar la memoria no usada.
- Asignar variables a registros y pilas.
- Llamar al sistema huésped para ciertas funciones, como los accesos a los dispositivos.
- Vigilar el cumplimiento de las normas de seguridad de las aplicaciones Java. Por ejemplo:
  - No se permiten realizar ciertas conversiones (casting) entre distintos tipos de datos.
  - Las referencias a arrays son verificadas en el momento de la ejecución del programa.

# JVM - Garbage Collector

- Es un proceso de baja prioridad que se ejecuta dentro de la JVM.
- Técnica por la cual el ambiente de objetos se encarga de destruir y asignar automáticamente la memoria heap.
- El programador no debe preocuparse por la asignación y liberación de memoria, ya que el entorno la asignará al crear un nuevo objeto y la liberará cuando nadie la esté usando.
- Un objeto podrá ser “limpiado” cuando desde el stack ninguna variable haga referencia al mismo.

# Agenda

- Variables
  - Clasificación
  - Nomenclatura de identificador
- Tipos primitivos
  - Categorización
  - Ejemplos
  - Valores por defecto
- Java Virtual Machine
  - Secciones
  - Tareas principales
  - Garbage Collector
- **Tipo objeto**
  - Creación
- Diferencias entre variables de tipo objeto y tipo primitivo
- Clases “Wrapper”
  - Ejemplos
  - Unboxing y Autoboxing

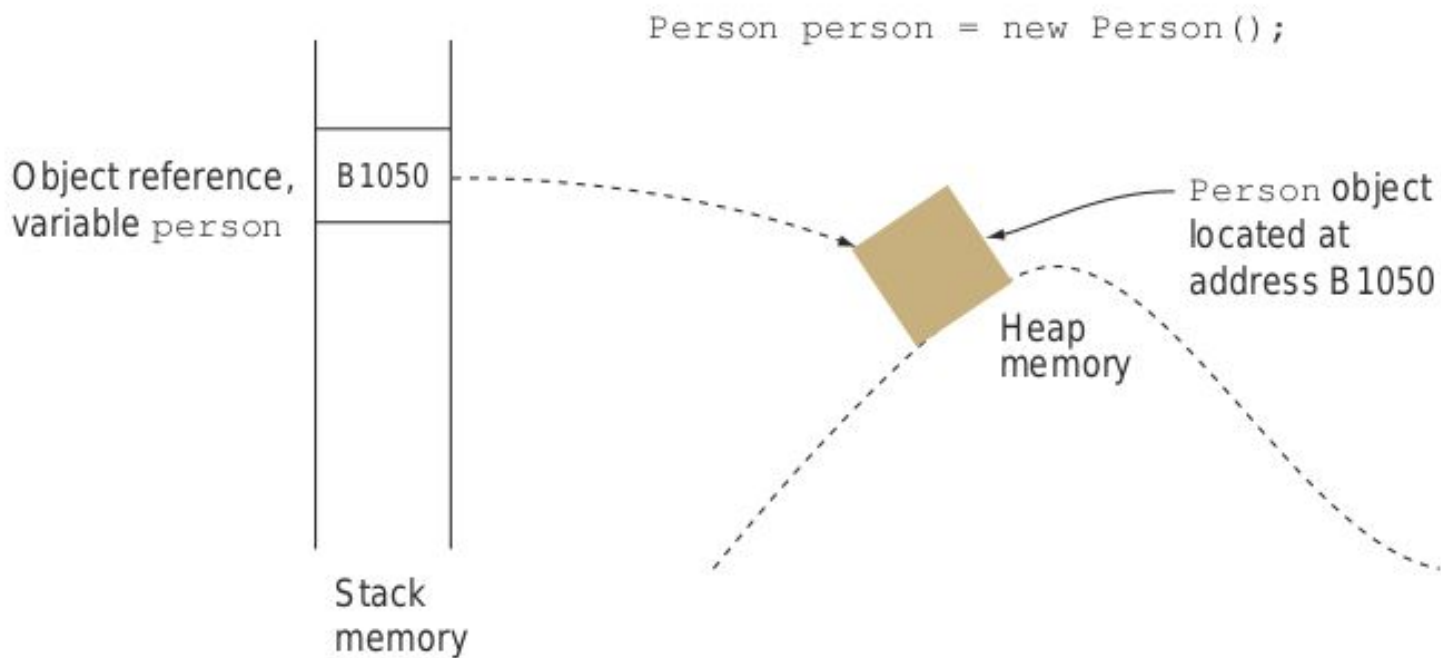


# Tipo objeto

- Son instancias de clases, que pueden estar predefinidas o definidas por el programador.
- Cuando un objeto es instanciado utilizando el operador **new**, se retorna una dirección de memoria.
- La dirección en memoria contiene una referencia a una variable.

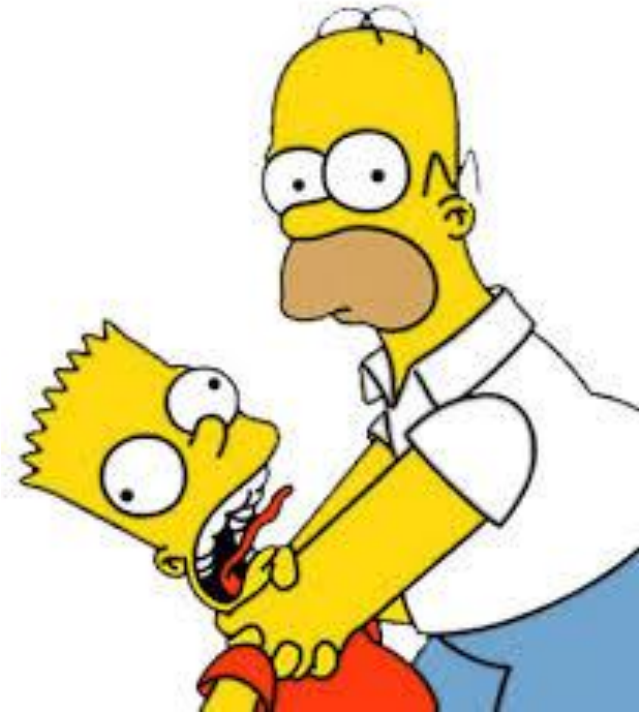


# Tipo objeto - Creación



# Agenda

- Variables
  - Clasificación
  - Nomenclatura de identificador
- Tipos primitivos
  - Categorización
  - Ejemplos
  - Valores por defecto
- Java Virtual Machine
  - Secciones
  - Tareas principales
  - Garbage Collector
- Tipo objeto
  - Creación
- **Diferencias entre variables de tipo objeto y tipo primitivo**
- Clases "Wrapper"
  - Ejemplos
  - Unboxing y Autoboxing

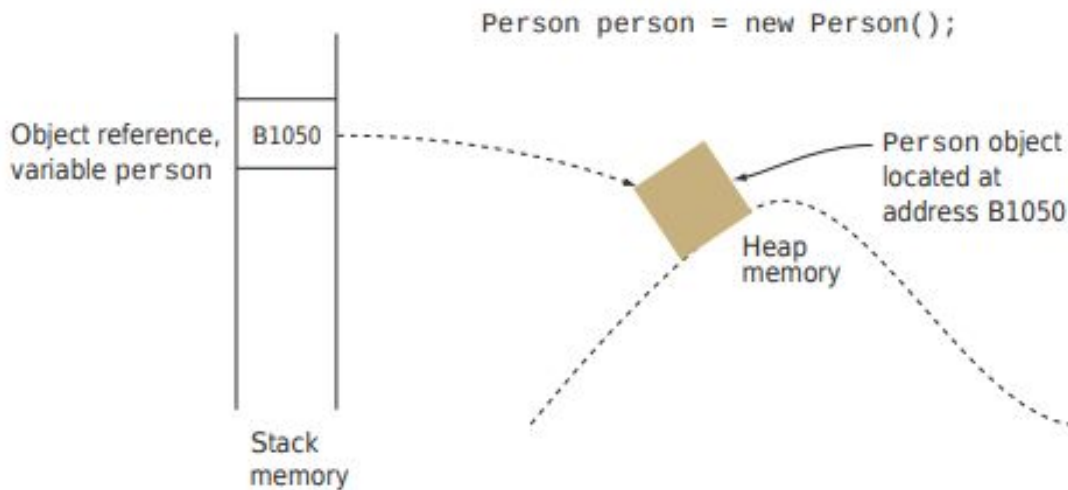
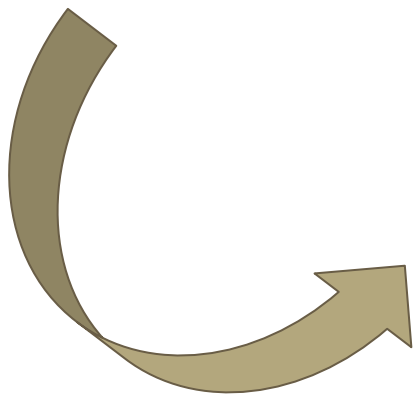




# Diferencias entre variables de tipo objeto y tipo primitivo

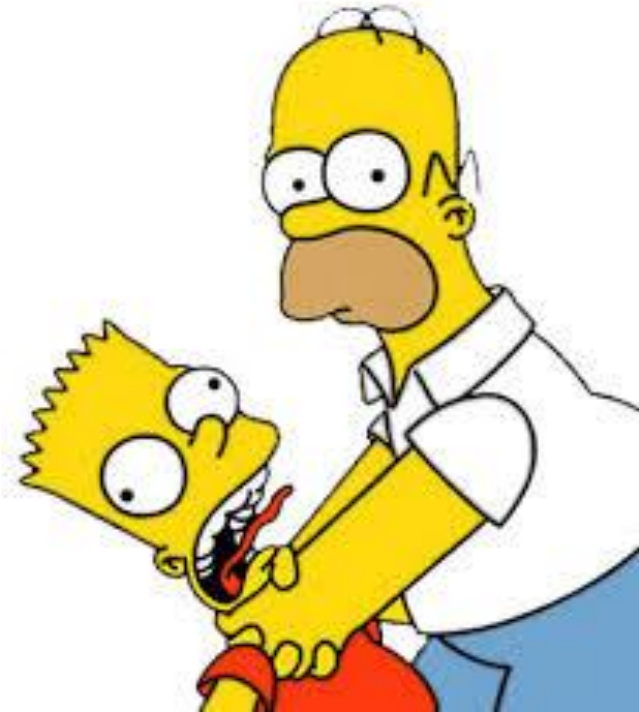
```
int a = 77;
```

```
Persona person = new Person();
```



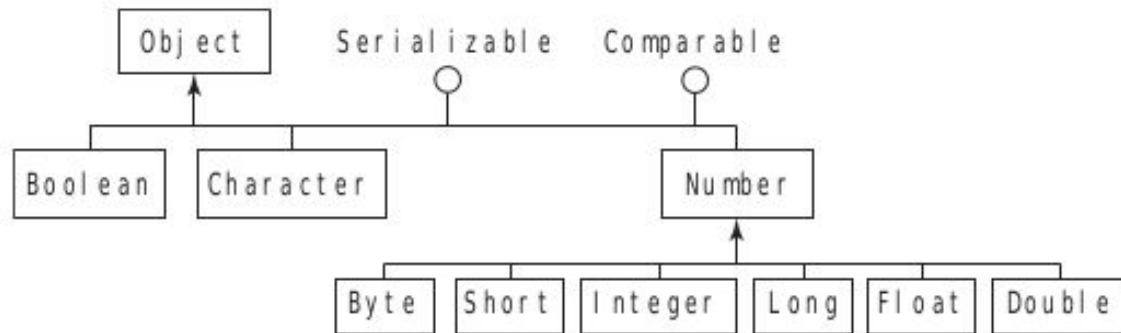
# Agenda

- Variables
  - Clasificación
  - Nomenclatura de identificador
- Tipos primitivos
  - Categorización
  - Ejemplos
  - Valores por defecto
- Java Virtual Machine
  - Secciones
  - Tareas principales
  - Garbage Collector
- Tipo objeto
  - Creación
- Diferencias entre variables de tipo objeto y tipo primitivo
- **Clases “Wrapper”**
  - Ejemplos
  - Unboxing y Autoboxing



# Classes “Wrapper”

- Java define una clase “wrapper” para cada tipo primitivo.



# Clases “Wrapper” - Ejemplos

```
Boolean bool1 = true;
```

```
Character char1 = 'a';
```

```
Double double1 = 10.98;
```

```
Boolean bool2 = new Boolean(true);
```

```
Character char2 = new Character('a');
```

```
Double double2 = new Double(10.98);
```

# Classes “Wrapper” - Unboxing y Autoboxing

