Kotlin vs Java es una pregunta que cada día es mas habitual hacerse. Kotlin es uno de los nuevos lenguajes de programación que empieza a tener peso en la plataforma Java por su sencillez a nivel de sintaxis. Vamos a ver un ejemplo de manejo sencillo de clases que nos permita darnos cuenta del nivel de ahorro de código usando la clase Persona:

```
package ejemplo0;

public class Persona {
    private String nombre;
    private String apellidos;
    private int edad;

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getApellidos() {
        return apellidos;
    }

    public void setApellidos(String apellidos) {
        this.apellidos = apellidos;
    }

    public int getEdad() {
        return edad;
    }
```

```
    public void setEdad(int edad) {
        this.edad = edad;
    }

    public Persona(String nombre, String apellidos, int edad) {
        this.nombre = nombre;
        this.apellidos = apellidos;
        this.edad = edad;
    }
}
```

Este es el código de la clase Persona , si escribimos algo similar con Kotlin nos encontraremos con algo de este estilo:

```
package ejemplo1

class Persona {
    var nombre:String
    var apellidos:String
    var edad:Int

    constructor( nombre:String, apellidos:String,edad:Int ) {

        this.nombre=nombre;
        this.apellidos=apellidos;
        this.edad=edad;
    }
}
```

Es un ahorro significativo ya que declaramos las variables y generamos el constructor no

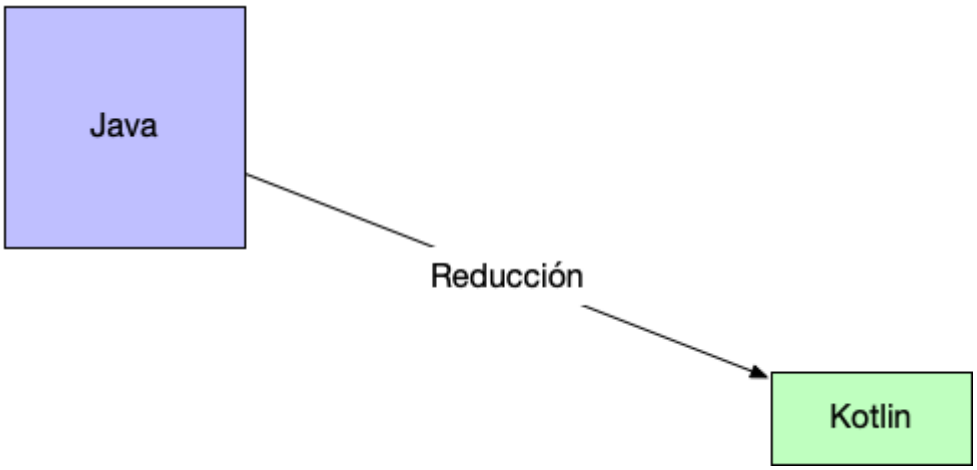
hace falta añadir visibilidad a las variables ya que estas son públicas por defecto. En principio la simplificación es interesante pero Kotlin permite simplificaciones mucho más fuertes . Ya que dispone de constructores que declaran a la vez las variables :

```
package ejemplo2

class Persona  constructor(var nombre:String ,var apellidos:String , var edad:Int){}
```

Kotlin vs Java y funciones

En este caso la reducción de código es dramática y nos permite escribir mucho menos.



A esto le podemos añadir en cualquier momento una función que por ejemplo nos diga si la Persona esta o no jubilada.

```
class Persona   constructor(var nombre:String ,var apellidos:String , var edad:Int){
```

```
    fun estaJubilado():Boolean = edad>65

}
```

Como vemos también la sintaxis del método es muy muy compacta a la hora de declarar la función . Una vez construido el código solo necesitamos un programa Main que nos permita invocarlo :

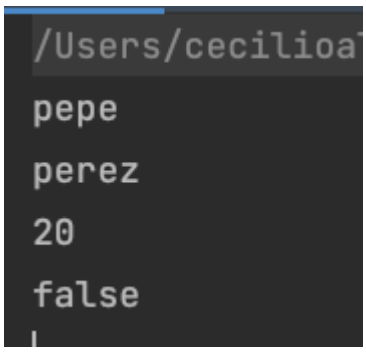
```
import ejemplo2.Persona

fun main(args: Array<String>) {

var persona1 = Persona("pepe", "perez", 20)
println(persona1.nombre)
println(persona1.apellidos)
println(persona1.edad)
println(persona1.estaJubilado())

}
```

Si ejecutamos el código nos encontraremos con :



Acabamos de ver como de compacto es usar Kotlin a nivel de sintaxis vs Java tengámoslo cada día más en cuenta como uno de los lenguajes de referencia:

Otros artículos relacionados

1. Kotlin Destructuring
2. Kotlin Ranges y sentencia de Control
3. Kotlin Delegación buenas prácticas.
4. Kotlin Clases