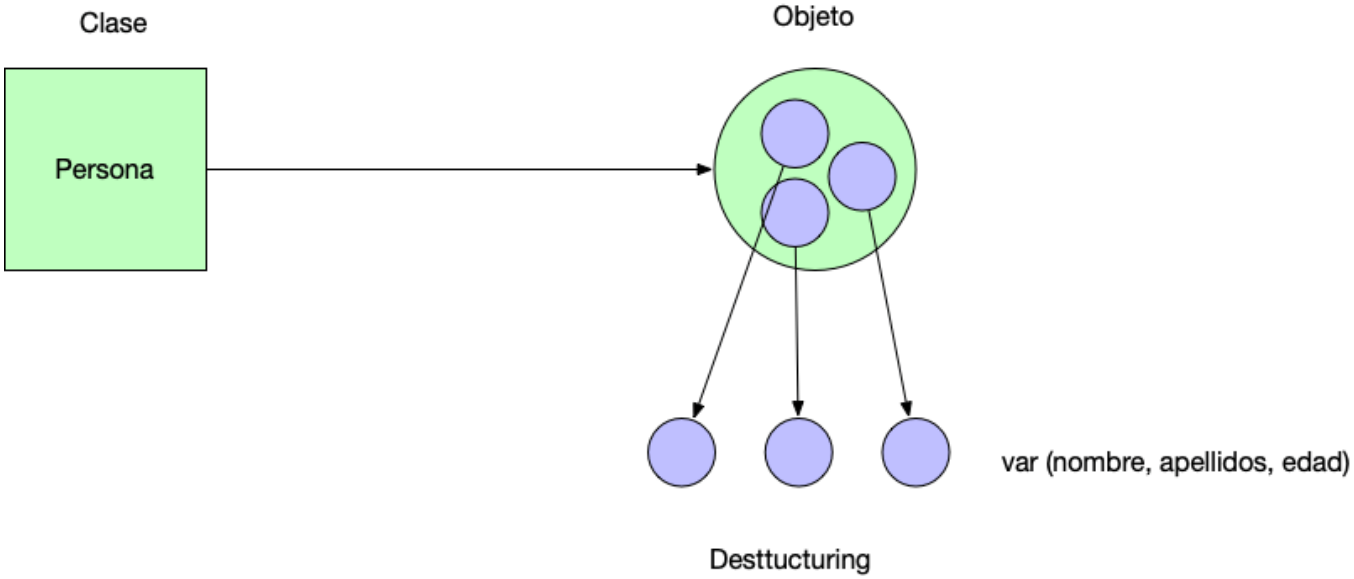
Vamos a ver el uso de Kotlin Destructuring a la hora de trabajar de una forma más sencilla con Kotlin y variables . El concepto de destructuring es uno de los conceptos mas comunes en los nuevos lenguajes de programación. Este concepto hace referencia a como podemos descomponer una clase con sus variables en variables independientes para un manejo más sencillo de estas .



Kotlin nos permite hacer estas operaciones de forma bastante natural . Para ello lo primero que tendremos que definir una clase . En este caso usaremos un Data Class con las propiedades de Persona.

```
data class Persona (val nombre:String,val apellidos:String,val
edad:Int) {

}
```

Una vez definida la clase con sus propiedades podemos imprimir los datos de la clase usando una función Main sencilla:

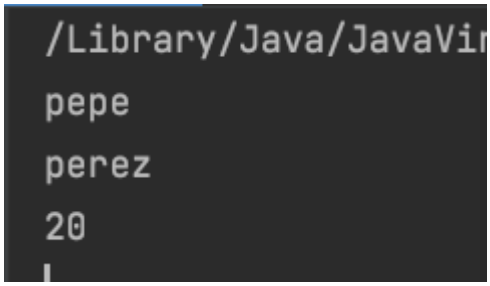
```
fun main(args: Array<String>) {
```

```
   val persona= Persona ("pepe","perez",20)
    println(persona.nombre)
    println(persona.apellidos)
    println(persona.edad)




}
```

El resultado es :



Podemos realizar la misma operación realizando un destructuring . En este caso cada propiedad se convertirá en una variable que pasaremos a manejar.

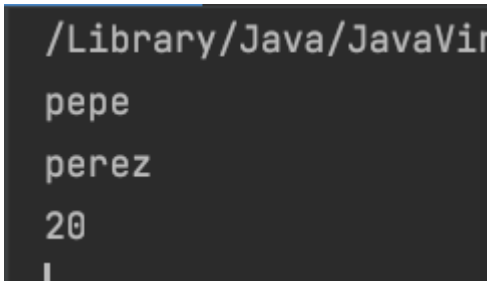
```
fun main(args: Array<String>) {

   val persona= Persona ("pepe","perez",20)


    val (nombre,apellidos,edad)= persona
    println (nombre)
    println(apellidos)
    println(edad)
```

```
}
```

El resultado es el mismo



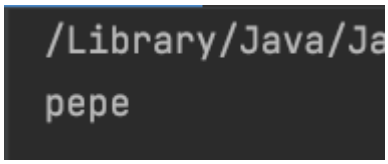
Podemos realizar operaciones diferentes a la hora de realizar un destructuring por ejemplo podemos obtener únicamente una de las variables utilizando guiones bajos en el resto de las propiedades.

```
fun main(args: Array<String>) {

   val persona= Persona ("pepe","perez",20)
   val (nombre,_,_)= persona
   println (nombre)



}
```



Kotlin Destructuring y Colecciones

Las operaciones de destructuring también pueden afectar al uso de Colecciones como por ejemplo si disponemos de un Mapa con nombres y edades :

```
fun main() {


    val map = mapOf("pedro" to 10, "gema" to 30, "ana" to 50)

    for ((clave , valor) in map) {
        println("$clave = $valor")
    }

}
```

En este caso hemos usado destructuring a nivel de la estructura for ya que nos encontramos con un Mapa que almacena claves y valores . Este tipo de almacenamiento permite un destructuring sencillo con Kotlin.

Otros artículos relacionados

- Kotlin Delegación buenas prácticas.
- Hello World Kotlin utilizando Eclipse
- Java forEach y sus opciones
- ¿Que es un Java Stream?
- Kotlin Mapas