

## EJERCICIO U2\_B5\_E1:

```
class Circulo{
    int coordenadaX;
    int coordenadaY;
    int radio;
    Circulo elMayor(Circulo c){
        return this.radio>c.radio?this:c;
    }
}

class Unidad2{
    public static void main(String[] args) {
        Circulo c1=new Circulo();
        Circulo c2=new Circulo();
        c1.coordenadaX=3;
        c1.coordenadaY=3;
        c1.radio=10;
        c2.coordenadaX=50;
        c2.coordenadaY=45;
        c2.radio=5;

        Circulo circuloGrande=c1.elMayor(c2);
        System.out.println("el círculo mayor: ");
        System.out.println("\t coordenadaX: "+circuloGrande.coordenadaX);
        System.out.println("\t coordenadaY: "+circuloGrande.coordenadaY);
        System.out.println("\t radio: "+circuloGrande.radio);
    }
}
```

## EJERCICIO U2\_B5\_E2:

```
class Circulo{
    int coordenadaX;
    int coordenadaY;
    int radio;
    Circulo(int coordenadaX, int coordenadaY, int radio){
        this.coordenadaX=coordenadaX;
        this.coordenadaY=coordenadaY;
        this.radio=radio;
    }
    Circulo elMayor(Circulo c){
        if(this.radio>c.radio){ //o también if(this.radio>c.radio)
            return this;
        }else{
            return c;
        }
    }
}

public class Unidad2 {
    public static void main(String[] args) {
        Circulo c1=new Circulo(3,3,10);
        Circulo c2=new Circulo(50,45,5);

        Circulo circuloGrande=c1.elMayor(c2);
        System.out.println("el círculo mayor: ");
        System.out.println("\t coordenadaX: "+circuloGrande.coordenadaX);
        System.out.println("\t coordenaday: "+circuloGrande.coordenadaY);
    }
}
```

```

        System.out.println("\t radio: "+circuloGrande.radio);
    }
}

```

## EJERCICIO U2\_B5\_E3:

```

class Circulo{
    int coordenadaX;
    int coordenadaY;
    int radio;
    Circulo(int coordenadaX, int coordenadaY, int radio){
        this.coordenadaX=coordenadaX;
        this.coordenadaY=coordenadaY;
        this.radio=radio;
    }
    Circulo elMayor(Circulo c){
        Circulo copiaThis= new Circulo(this.coordenadaX,this.coordenadaY,this.radio);
        Circulo copiaC=new Circulo(c.coordenadaX,c.coordenadaY,c.radio);
        return this.radio>c.radio? copiaThis:copiaC;
    }
}

class Unidad2{
    public static void main(String[] args) {
        Circulo c1=new Circulo(3,3,10);
        Circulo c2=new Circulo(50,45,5);

        Circulo circuloGrande=c1.elMayor(c2);
        System.out.println("el círculo mayor: ");
        System.out.println("\t coordenadaX: "+circuloGrande.coordenadaX);
        System.out.println("\t coordenadaY: "+circuloGrande.coordenadaY);
        System.out.println("\t radio: "+circuloGrande.radio);
        System.out.println("demostramos que aunque circuloGrande inicialmente es copia de c1 son objetos diferentes");
        System.out.println("cambiamos el radio de circulo grande pero no cambia el de c1");
        circuloGrande.radio=-99;
        System.out.println(circuloGrande.radio+", "+c1.radio);
    }
}

```

Podríamos haber creado los círculos directamente en la expresión del operador condicional así sólo creamos el círculo que nos interesa, pero entonces sale una un return monstruoso.

```

Circulo elMayor(Circulo c){
    return this.radio>c.radio? new Circulo(this.coordenadaX,this.coordenadaY,this.radio):new Circulo(c.coordenadaX,c.coordenadaY,c.radio);
}

```

Por eso en este caso, es “mejor” escribir el código anterior con if que hace el código más legible.

```

Circulo elMayor(Circulo c){
    if(this.radio>c.radio){
        return new Circulo(this.coordenadaX,this.coordenadaY,this.radio);
    }else{
        return new Circulo(c.coordenadaX,c.coordenadaY,c.radio);
    }
}

```

## EJERCICIO U2\_B5\_E4:

```
class Coche{
    int pasajeros;
    int deposito;
    int kpl;

    void setPasajeros(int pasajeros){
        this.pasajeros = pasajeros;
    }
    void setDeposito(int deposito){
        this.deposito = deposito;
    }

    void setKpl(int kpl){
        this.kpl=kpl;
    }

    //aunque no es necesario también uso this en el resto de métodos
    int getPasajeros(){
        return this.pasajeros;
    }

    int getDeposito(){
        return this.deposito;
    }
    int getKpl(){
        return this.kpl;
    }
}

class Unidad2 {
    public static void main(String[] args) {
        Coche citroenC1= new Coche();
        citroenC1.setPasajeros(4);
        citroenC1.setDeposito(50);
        citroenC1.setKpl(25);
        System.out.println("un citroen C1 permite sólo " + citroenC1.getPasajeros() + " pasajeros");
        System.out.println("un citroen C1 tiene una autonomía de " + citroenC1.autonomia() + "
kilómetros");
    }
}
```

## EJERCICIO U2\_B5\_E5:

```
class Coche{
    int pasajeros;
    int deposito;
    int kpl;
    int calcularAutonomia(){
        return deposito*kpl;
    }

    boolean mayorAutonomia(Coche c){
        return this.calcularAutonomia()>c.calcularAutonomia();
    }
}

class Unidad2{
```

```

public static void main(String[] args) {
    Coche coche1= new Coche();
    coche1.pasajeros=5;
    coche1.deposito=60;
    coche1.kpl=20;

    Coche coche2= new Coche();
    coche2.pasajeros=7;
    coche2.deposito=1000;
    coche2.kpl=30;
    System.out.println("Tiene coche1 más autonomía que coche2? "+ coche1.mayorAutonomia(coche2));
}
}

```

## EJERCICIO U2\_B5\_6:

RECUERDA: Se puede escribir de muchas formas, unas más compactas que otras ¿Cuál es la mejor? Sin duda alguna: escríbelo como te sientas más seguro, como lo entiendas mejor.

Aquí vemos 3 formas:

```

Coche mayorAutonomia(Coche c){
    Coche mayor;
    if (this.calcularAutonomia()>=c.calcularAutonomia()){
        mayor=this;
    }else{
        mayor=c;
    }
    return mayor;
}

```

```

Coche mayorAutonomia(Coche c){
    if (this.calcularAutonomia()>=c.calcularAutonomia()){
        return this;
    }else{
        return c;
    }
}

```

```

Coche mayorAutonomia(Coche c){
    return this.calcularAutonomia()>c.calcularAutonomia()?this:c;
}

```

ESTA SOLUCIÓN NO ES MEJOR NI PEOR QUE LA DE CREAR UN NUEVO COCHE CON LOS DATOS COPIADOS DE MAYOR. SIMPLEMENTE SON EFECTOS DIFERENTES. MÁS ADELANTE VOLVEREMOS A COMPARAR AMBOS ENFOQUES Y ANALIZAREMOS QUE NORMALMENTE SE PREFIERE EL ENFOQUE DE CREAR UN OBJETO COPIA PARA EVITAR ACCESO AL OBJETO ORIGINAL(por cuestiones de seguridad).

## EJERCICIO U2\_B5\_E7:

```

class Cuenta {
    private String numeroCuenta;
}

```

```

private String titular;
private double saldo;

Cuenta(String numeroCuenta, String titular, double saldo) {
    this.numeroCuenta = numeroCuenta;
    this.titular = titular;
    this.saldo = saldo;
}

Cuenta() {
    this("sin numero", "sin titular", 0.0);
}

void setSaldo(double saldo) {
    this.saldo = saldo;
}

double getSaldo() {
    return saldo;
}

public String toString() {
    return "("+numeroCuenta + ", " + titular + ", " + saldo+")";
}
}

```

## EJERCICIO U2\_B5\_E8:

```

class Circulo{
    int coordenadaX;
    int coordenadaY;
    int radio;
    Circulo(int coordenadaX, int coordenadaY, int radio){
        this.coordenadaX=coordenadaX;
        this.coordenadaY=coordenadaY;
        this.radio=radio;
    }
    Circulo(Circulo c){
        this(c.coordenadaX,c.coordenadaY,c.radio);
    }

    Circulo elMayor(Circulo c){
        return this.radio>c.radio? new Circulo(this): new Circulo(c);
    }
}

class Unidad2{
    public static void main(String[] args) {
        Circulo c1=new Circulo(3,3,10);
        Circulo c2=new Circulo(50,45,5);

        Circulo circuloGrande=c1.elMayor(c2);
        System.out.println("el círculo mayor: ");
        System.out.println("\t coordenadaX: "+circuloGrande.coordenadaX);
        System.out.println("\t coordenadaY: "+circuloGrande.coordenadaY);
        System.out.println("\t radio: "+circuloGrande.radio);
        System.out.println("demostramos que aunque circuloGrande inicialmente es copia de c1 son objetos diferentes");
        System.out.println("cambiamos el radio de circulo grande pero no cambia el de c1");
        circuloGrande.radio=-99;
        System.out.println(circuloGrande.radio+" ", "+c1.radio);
    }
}

```