

El refactoring de Eclipse Parameter Object puede ser uno de los refactorings más prácticos que tenemos a mano en el entorno de desarrollo . Aún así es poco conocido por parte de los programadores. vamos a ver un ejemplo sencillo de como usarlo . Supongamos que tenemos una clase Main que dispone de una Compra y esa compra hay que entregarla en alguna Coordenada especifica . Lo más sencillo es construir una clase Compra y definir las Coordenadas X, Y que son direcciones de entrega. Veámoslo en código en una primera aproximación.,

```
package com.arquitecturajava;
```

```
public class Compra {

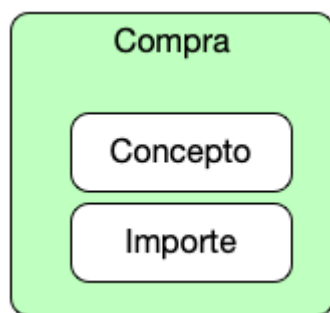
    private String concepto;
    private double importe;
    public String getConcepto() {
        return concepto;
    }
    public void setConcepto(String concepto) {
        this.concepto = concepto;
    }
    public double getImporte() {
        return importe;
    }
    public void setImporte(double importe) {
        this.importe = importe;
    }
    public Compra(String concepto, double importe) {
        super();
        this.concepto = concepto;
        this.importe = importe;
    }
}
```

```

        public Compra() {
            super();
        }
    }
}

```

Tenemos la clase Compra con sus campos



Nos falta la clase principal que es la encargada de entregar la Compra geolocalizada

```

package com.arquitecturajava;

```

```

public class Principal {

    public static void main(String[] args) {
        Compra c= new Compra("ordenador",200);
        entregarCompra(c, 1.8999,2.3454);
    }
    public static void entregarCompra(Compra c , double x ,
double y) {
        System.out.println("la compra " + c.getConcepto() +"se
ha entregado");
        System.out.println("el importe es :" +
c.getImporte());
        System.out.println("la coordenada x es " + x );
        System.out.println("la coordenada y es " + y );
    }
}

```

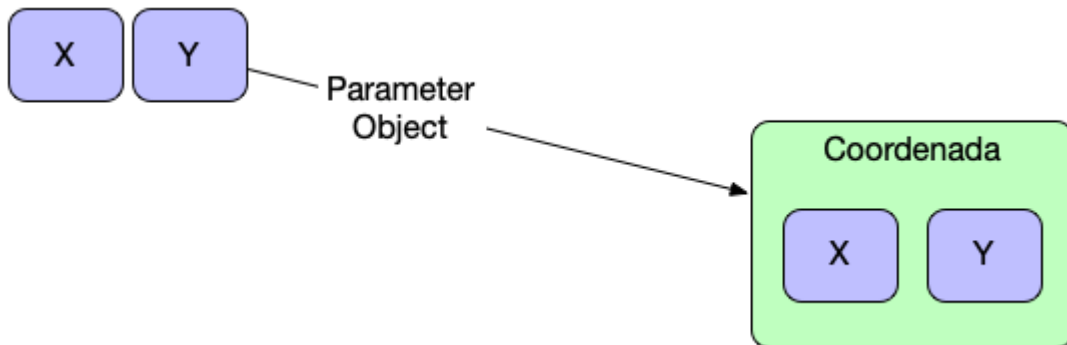
```
}
```

```
}
```

El resultado lo podemos ver por la consola:

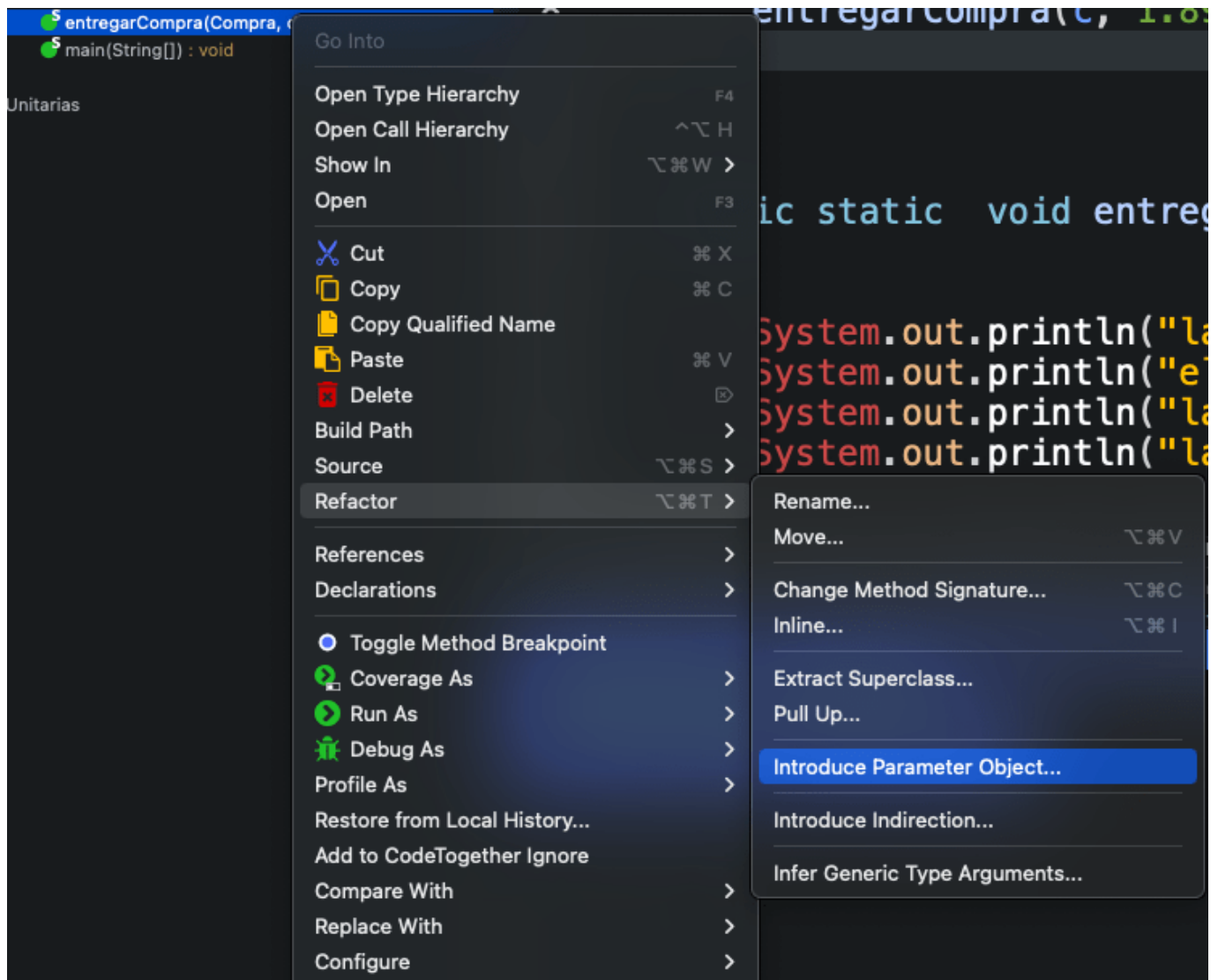
```
la compra ordenador se ha entregado  
el importe es :200.0  
la coordenada x es 1.8999  
la coordenada y es 2.3454
```

Si revisamos el código nos daremos cuenta que probablemente no hemos definido el concepto de Coordenada y tenemos dos variables doubles x, y .

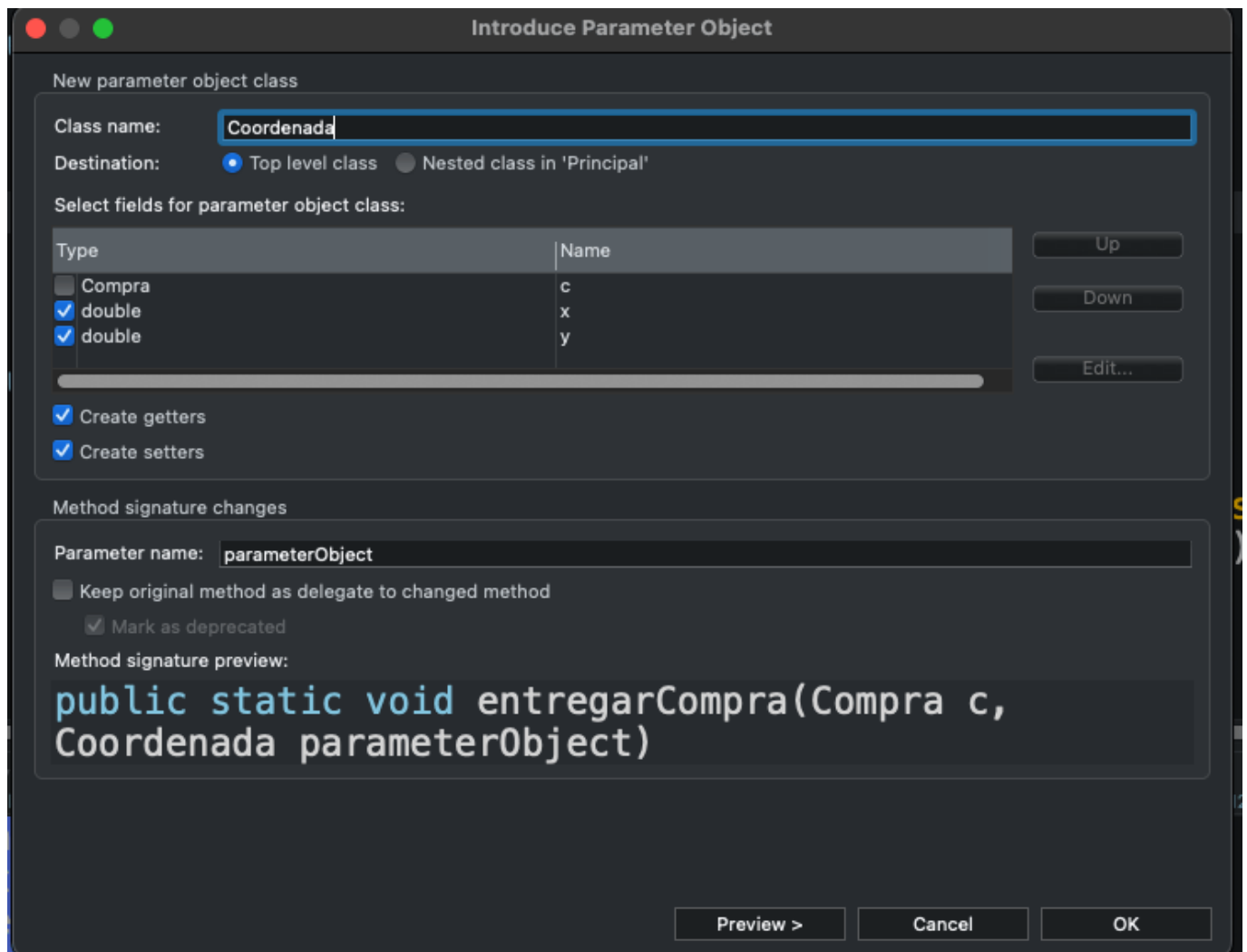


Eclipse Parameter Object

Eclipse Parameter Object es el refactoring que se encarga de solventar este problema y generar una nueva clase a partir de la selección de parámetros que tengamos.



Una vez seleccionado el refactoring el siguiente paso es definir el nombre de la clase y solicitar que añada los métodos set/get con estas dos cosas pulsamos ok y nos generará el nuevo código



Eclipse Parameter Object y Clase Coordenada

Hemos generado una clase adicional que se denomina Coordenada y el método main ha sido refactorizado completamente para usarla de forma natural con sus métodos get/set y constructor

```
package com.arquitecturajava;
```

```
public class Coordenada {  
    private double x;  
    private double y;
```

```

    public Coordenada(double x, double y) {
        this.x = x;
        this.y = y;
    }

    public double getX() {
        return x;
    }

    public double getY() {
        return y;
    }
}

```

El programa main queda de la siguiente forma:

```

package com.arquitecturajava;

public class Principal {

    public static void main(String[] args) {
        Compra c= new Compra("ordenador",200);
        entregarCompra(c, new Coordenada(1.8999, 2.3454));
    }

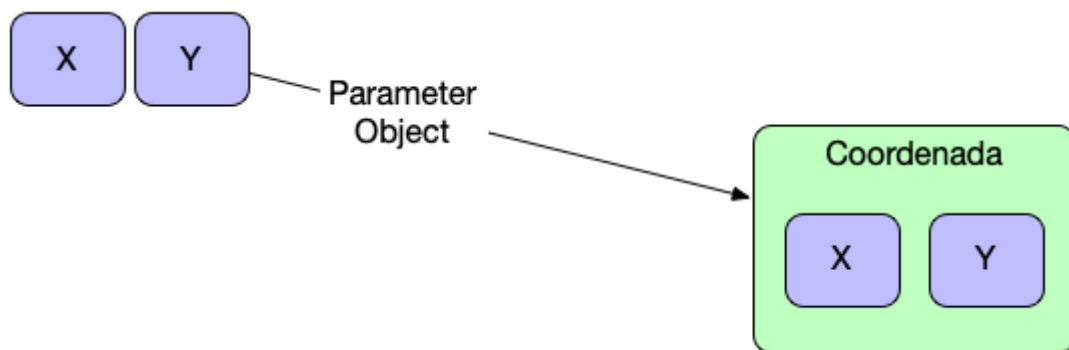
    public static void entregarCompra(Compra c , Coordenada
parameterObject) {
        System.out.println("la compra " + c.getConcepto() +"se
ha entregado");
        System.out.println("el importe es :" +
c.getImporte());
        System.out.println("la coordenada x es " +

```

```
parameterObject.getX() );  
        System.out.println("la coordenada y es " +  
parameterObject.getY() );  
    }  
  
}
```

Conclusiones

Hemos usado Eclipse Parameter Object y convertido dos variables sencillas x,y en una nueva clase que encaja en el modeloy contiene los métodos set/get y el constructor.



Otros artículos relacionados

- [Eclipse recuperar archivos borrados](#)
- [Eclipse static import y refactorings](#)
- [Eclipse refactor move y como utilizarlo](#)
- [Java toString overriding y Eclipse](#)
- [Eclipse Refactorings](#)