

PROGRAMACIÓN ORIENTADA A OBJETOS

1. HERENCIA

La herencia se refiere a crear una clase que extienda la información de otra.

La forma de heredar es sencilla. Si tenemos ya una clase, por ejemplo, Bicicleta en este proyecto, y creamos otra que queremos que sea hereditaria de esta, lo que debemos hacer es utilizar “extends” al exportar la nueva clase.

Ejemplo:

```
Export class Montaña extends Bicicleta {}
```

La clase hereditaria puede utilizar tanto los campos de la superclase como los métodos declarados en ella.

Se utiliza “protected” al declarar los campos de la superclase para que este sea privado, pero a su vez pueda ser utilizado en la subclase.

Para llamar al constructor de la superclase desde una subclase empleamos “super()”.

En caso de que queramos referirnos a un método de la superclase se utiliza “super.”

2. POLIMORFISMO

El polimorfismo se refiere a que, al invocar un método de un objeto, pueden obtenerse distintos resultados según la clase.

Un método puede ser sobrecargado cuando existen funciones con el mismo nombre y con funcionalidad similar, pero pertenecientes a clases independientes.