

1. Crear una clase publica: **Repaso**
2. Crear el método **main** dentro de la clase Repaso
3. En el main, declarar una variable **mensaje** inicializada a: “Hola, ¿cómo estás?”
4. En el main, declarar una variable **haceSol** inicializada a true.
5. En el main, declarar una variable **días** inicializada a 7.
6. Imprimir por pantalla la variable mensaje
7. Crear una clase pública: **Caballo**
8. En la clase Caballo, crear 4 atributos **publicos**: nombre, color, edad y carrerasGanadas
9. En el main de la clase Repaso:
 - Crear una variable **caballo1** que sea un objeto de la clase **Caballo**.
 - Escribir por pantalla el **color** del **caballo1**.
 - Ponerle el valor ‘marrón’ al **color** del **caballo1**
 - Escribir por pantalla el **color** del **caballo1**.
10. En la clase Caballo, cambiar la visibilidad de los atributos a privado. ¿Qué ocurre en el main?
11. Comentar lo que falla del apartado 9 en el main.
12. Crear un método **público** en la clase Caballo: **imprimeCarrerasGanadas()** que escriba por pantalla lo siguiente: “El número de carreras ganadas es: ”
carrerasGanadas.
13. Comentar el método anterior **imprimeCarrerasGanadas**
14. En el main de la clase Repaso llamar al método **imprimeCarrerasGanadas** del objeto **caballo1**.
15. Crear un método **público** en la clase Caballo: **anyoNacimiento()** que devuelve el año del nacimiento del caballo.
16. Comentar el método anterior
17. Desde el main, invocar al método **anyoNacimiento** y mostrar por pantalla el resultado.
18. Crear un método público **incrementarCarrerasGanadas** que incremente en una unidad el numero de carrerasGanadas.
19. Desde el main, invocar el método **incrementarCarrerasGanadas**.
20. Desde el main, invocar el método **imprimeCarrerasGanadas**
21. Escribir un método en la clase Caballo: **relincha()** que escriba por pantalla: “Hiiiiiiiiiiiiiiiiiiii”.
22. Desde el main, invocar al método **relincha** del objeto **caballo1**.
23. Desde el main, imprimir por pantalla el nombre del caballo. ¿Es posible?

24. Crear un método setNombre y getNombre para poder asignar y recuperar el nombre del caballo.
25. Desde el main, invoca al metodo setNombre para darle al objeto **caballo1** el nombre de “**Bolita**”
26. Crear un método setColor y getColor para poder asignar y recuperar el color del caballo.
27. Desde el main, invoca al metodo setColor para darle al objeto **caballo1** el color “**Marrón**”
28. Crear un método setEdad y getEdad para poder asignar y recuperar la edad del caballo.
29. Desde el main, invoca al metodo setEdad para darle al objeto **caballo1** el valor 2.
30. Crear un método setCarrerasGanadas y getCarrerasGanadas para poder asignar y recuperar el numero de carreras del caballo.
31. Desde el main, invoca al metodo setCarrerasGanadas para darle al objeto **caballo1** el valor 10.
32. Escribir un método público en la clase Caballo: toString() que devuelva en una cadena todos los datos del caballo de esta manera:

```
Caballo [nombre=" + nombre + ", color=" + color + ", edad=" +  
edad + ", carreras Ganadas=" + this.carrerasGanadas + "];
```

33. Desde el man, invocar el método toString del objeto caballo1
34. Crear un constructor para la clase Caballo que reciba como parámetros los 4 atributos del caballo: nombre, color, edad y carrerasGanadas. ¿Qué ocurre en el main con caballo1?
35. Desde el main, crear un objeto nuevo **caballo2** con el constructor con parámetros creado en el apartado anterior con los valores: “Fideo”, “blanco”, 4 y 20.
36. Desde el main, obtener el numero de carreras ganadas del objeto **caballo2**.
37. Desde el main, invocar el método toString para ver los datos del caballo2.?
38. Crear un constructor sin parámetros en la clase Caballo dandole al nombre y color cadenas vacias y a la edad y al numero de carreras el valor 0.
39. Crear un objeto **caballo3** con el constructor sin parametros
40. Invocar desde el main el metodo **toString()** con el **caballo3** y mostrar el resultado por pantalla.
41. Asigarle al objeto **caballo3** los datos nombre, edad, carreras ganadas y color que elijais.
42. Invocar al metodo **toString()** con el caballo3 para ver los datos.