Navegación por el cuestionario
1 2 3 4 5 6 7 8 9 10
Finalizar revisión

Comenzado el	martes, 12 de marzo de 2019, 20:19
Estado	Finalizado
Finalizado en	miércoles, 13 de marzo de 2019, 20:05
Tiempo empleado	23 horas 45 minutos
Calificación	**8,60** de 10,00 (**86**%)

Pregunta **1**
Correcta
Puntúa 1,00 sobre 1,00
Marcar pregunta

¿Qué clase Java define e implementa el comportamiento común a todas las clases (incluidas las que desarrolle el programador)?

Seleccione una:
- a. `Class`.
- b. No existe tal clase en Java.
- c. `Template`.
- d. `Object`. ✓

La respuesta correcta es: `Object`.

Correcta
Puntos para este envío: 1,00/1,00.

Pregunta **2**
Correcta
Puntúa 1,00 sobre 1,00
Marcar pregunta

¿Qué diferencia existe entre la ligadura dinámica y la ligadura estática?

Seleccione una:
- a. En la ligadura estática la vinculación entre una llamada a un método y el método que finalmente va a ser ejecutado se realiza en tiempo de compilación, mientras que en la ligadura dinámica esa vinculación se lleva a cabo en tiempo de ejecución. ✓
- b. En la ligadura estática la vinculación entre una llamada a un método y el método que finalmente va a ser ejecutado se realiza en tiempo de ejecución, mientras que en la ligadura dinámica esa vinculación se lleva a cabo en tiempo de compilación.
- c. En la ligadura dinámica se puede llamar a métodos que no existen y que en tiempo de ejecución darán lugar a un error.
- d. En realidad no existe ninguna diferencia, son dos formas distintas de llamar al mismo fenómeno.

La respuesta correcta es: En la ligadura estática la vinculación entre una llamada a un método y el método que finalmente va a ser ejecutado se realiza en tiempo de compilación, mientras que en la ligadura dinámica esa vinculación se lleva a cabo en tiempo de ejecución.

Correcta
Puntos para este envío: 1,00/1,00.

Pregunta **3**
Correcta
Puntúa 0,50 sobre 1,00
Marcar pregunta

Tenemos las siguiente clases e interfaces:

Interfaz1
```
public interface Interfaz1
{

   public void metodo();
}
```

Clase A
```
public class A implements Interfaz1
{
   public void metodo1() {
      System.out.println("A 1");
   }

   public void metodo2() {
      System.out.println("A 2");
   }

   public String toString() {
      return "A";
   }

   public void metodo() {
      System.out.println("Interfaz")
;
   }
}
```

Clase C
```
public class C extends A {

   public void metodo1() {
      System.out.println("C 1")
;
   }

   public String toString() {
      return "C";
   }
}
```

Clase D
```
public class D extends C {

   public void metodo2() {
      System.out.println("D 2");
   }

   public void metodo() {
      System.out.println("Clase D")
;
   }
}
```

Creamos la lista elementos y le asignamos 3 valores:

```
ArrayList<Interfaz1> elementos = new ArrayList<Interfaz1>();
elementos.add(new D());
elementos.add(new C());
elementos.add(new A());
```

Elige el resultado de las siguientes expresiones:

Expresión	Resultado	Expresión	Resultado
elementos.get(0).metodo2()	ERROR ✓	elementos.get(0).metodo()	Clase D ✓
elementos.get(1).toString()	C ✓	elementos.get(1).metodo()	Intefaz ✓
elementos.get(2).metodo1()	ERROR ✓	elementos.get(2).toString()	A ✓

Correcta
Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría **0,50/1,00**.

Pregunta **4**
Correcta
Puntúa 1,00 sobre 1,00
Marcar pregunta

¿Qué hay que hacer en Java para crear un objeto polimórfico?

Seleccione una:
- a. Declarar una variable como referencia a un objeto de una clase determinada que tenga clases derivadas y así posteriormente se podrán asignar a esa variable referencias a objetos de subclases de la clase referencia inicial. ✓
- b. Utilizar la palabra reservada `polimorphic`.
- c. En Java no es posible el polimorfismo.
- d. Declarar una variable como referencia a un objeto de una clase determinada y posteriormente asignar a esa variable referencias a objetos de otras clases diferentes.

La respuesta correcta es: Declarar una variable como referencia a un objeto de una clase determinada que tenga clases derivadas y así posteriormente se podrán asignar a esa variable referencias a objetos de subclases de la clase referencia inicial.

Correcta
Puntos para este envío: 1,00/1,00.

Pregunta **5**
Correcta
Puntúa 0,90 sobre 1,00
Marcar pregunta

Tenemos las siguiente clases.

Clase Coche
```
public class Coche {
    protected String marca;
    protected String modelo;

    public Coche(String marca, String modelo
```

Clase CocheElectrico
```
public class CocheElectrico extends Coche {
   private int autonomiaKm;
   public CocheElectrico(String marca, String
modelo, int autonomiaKm) {
      super(marca, modelo);
```

Clase CocheGasolina
```
public class CocheGasolina extends Coche {

   private int kilometros;

   public CocheGasolina(String marca, String
```

```
) {
        this.marca = marca;
        this.modelo = modelo;
    }

    public String toString() {
        return this.getClass().getName() + "
 " + marca + " " + modelo;
    }
}
```

```
        this.autonomiaKm = autonomiaKm;
    }

    public int getAutonomia() {
        return autonomiaKm;
    }

    public int viajar(int km) {
        autonomiaKm -= km;
        return autonomiaKm;
    }

    public void repostar(int autonomiaKm) {
        this.autonomiaKm = autonomiaKm;
    }
}
```

```
modelo, int kilometros) {
        super(marca, modelo);
        this.kilometros = kilometros;
    }

    public int getKilometros() {
        return kilometros;
    }

    public boolean tocaRevision() {
        if (kilometros > 10000) {
            return true;
        }
        return false;
    }
```

Corrige el siguiente programa para que muestre qué coches de gasolina del array necesitan revisión y cuántos son en total.

Si el array tiene los siguientes elementos:

```
[Coche Opel Meriva, CocheElectrico Renault ZOE, CocheGasolina Fiat Punto, Coche Ford Fiest, CocheElectrico Hyundai Ioniq, CocheGasolina Renault Cl
io]
```

El resultado será:

```
CocheGasolina Fiat Punto 15000
1 coches necesitan revision
```

Respuesta: (penalty regime: 10, 20, ... %)

Reiniciar respuesta

```
import java.util.*;
public class CochesMain {
    public static void main(String[] args) {
        Coche[] arrayCoches = {new Coche("Opel", "Meriva"), new CocheElectrico("Renault", "ZOE", 300),
            new CocheGasolina("Fiat", "Punto", 15000), new Coche("Ford", "Fiest"),
            new CocheElectrico("Hyundai", "Ioniq", 200), new CocheGasolina("Renault", "Clio", 1500)};

        System.out.println(Arrays.toString(arrayCoches));
        System.out.println();

        // Coches de gasolina que necesitan revision
        int cuantos = 0;
        for (Coche unCoche : arrayCoches) {
            if (unCoche instanceof CocheGasolina) {
                CocheGasolina CocheGasolinaRevision = (CocheGasolina) unCoche;
                if (CocheGasolinaRevision.tocaRevision()) {
                    System.out.println(CocheGasolinaRevision + " " + CocheGasolinaRevision.getKilometros());
                    cuantos++;
                }
            }
```

	Expected
✓	[Coche Opel Meriva, CocheElectrico Renault ZOE, CocheGasolina Fiat Punto, Coche Ford Fiest, CocheElectrico Hyundai Ioniq, CocheGasolina Renault CocheGasolina Fiat Punto 15000 1 coches necesitan revision

Todas las pruebas superadas. ✓

Question author's solution:

```
import java.util.*;

public class CochesMain {
    public static void main(String[] args) {
        Coche[] arrayCoches = {new Coche("Opel", "Meriva"), new CocheElectrico("Renault", "ZOE", 300),
            new CocheGasolina("Fiat", "Punto", 15000), new Coche("Ford", "Fiest"),
            new CocheElectrico("Hyundai", "Ioniq", 200), new CocheGasolina("Renault", "Clio", 1500)};

        System.out.println(Arrays.toString(arrayCoches));
        System.out.println();

        // Coches de gasolina que necesitan revision
        int cuantos = 0;
        for (Coche unCoche : arrayCoches) {
            if (unCoche instanceof CocheGasolina) {
                CocheGasolina gasolina = (CocheGasolina) unCoche;
                if (gasolina.tocaRevision()) {
                    System.out.println(gasolina + " " + gasolina.getKilometros());
                    cuantos++;
                }
            }
        }
        System.out.println(cuantos + " coches necesitan revision");

    }
}
```

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría **0,90/1,00**.

Pregunta **6**

Correcta

Puntúa 0,90 sobre 1,00

Marcar pregunta

Tenemos las siguiente clases.

Clase A

```
public class A {
    public void metodo1() {
        System.out.println("A 1");
    }

    public void metodo2() {
        System.out.println("A 2");
    }

    public String toString() {
        return "A";
    }
}
```

Clase B

```
public class B extends A {

    public void metodo2() {
        System.out.println("B 2");
        super.metodo2();
    }

    public String toString() {
        return "B";
    }
}
```

Clase C

```
public class C extends B {

    public void metodo2() {
        System.out.println("C 2");
        super.metodo1();
    }
}
```

Clase D

```
public class D extends C {

    public void metodo1() {
        System.out.println("D 1");
    }
}
```

Escribe lo que mostraría en consola el siguiente código:

```
ArrayList<A> elementos = new ArrayList<A>();
elementos.add(new A());
elementos.add(new B());
elementos.add(0, new C());
elementos.add(2, new D());

for (A unElemento : elementos) {
    System.out.println(unElemento);
    unElemento.metodo1();
    unElemento.metodo2();
}
```

Respuesta: (penalty regime: 10, 20, ... %)

```
1 B
```

```
A 1
C 2
A 1
A
A 1
A 2
B
D 1
C 2
A 1
B
A 1
B 2
A 2
```

	Expected	Got	Comment	Mark	
✓	B A 1 C 2 A 1 A A 1 A 2 B D 1 C 2 A 1 B A 1 B 2 A 2	B A 1 C 2 A 1 A A 1 A 2 B D 1 C 2 A 1 B A 1 B 2 A 2	Line 0 right Line 1 right Line 2 right Line 3 right Line 4 right Line 5 right Line 6 right Line 7 right Line 8 right Line 9 right Line 10 right Line 11 right Line 12 right Line 13 right Line 14 right	1	✓

Todas las pruebas superadas. ✓

Question author's solution:

```
B
A 1
C 2
A 1
A
A 1
A 2
B
D 1
C 2
A 1
B
A 1
B 2
A 2
```

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría **0,90/1,00**.

Pregunta **7**

Correcta

Puntúa 1,00 sobre 1,00

Marcar pregunta

Tenemos las siguiente clases.

Clase Animal

```
public class Animal {
   public void metodo1() {
      System.out.println("mamifero 1");
   }

   public void metodo2() {
      System.out.println("mamifero 2");
   }

   public String toString() {
      return "MAMIFERO";
   }
}
```

Clase Mamifero

```
public class Mamifero extends Animal {

   public void metodo1() {
      System.out.println("animal 1");
      System.out.println(super.toString());
   }
}
```

Clase Reptil

```
public class Reptil extends Animal {

   public void metodo2() {
      System.out.println("reptil 2");
      super.metodo2();
   }

   public String toString() {
      return "REPTIL";
   }
}
```

Clase Perro

```
public class Perro extends Mamifero {

   public void metodo1() {
      System.out.println("perro 2");
      super.metodo1();
   }

   public String toString() {
      return "PERRO" + super.toString();
   }
}
```

Escribe lo que mostraría en consola el siguiente código :

```
ArrayList<Animal> animales = new ArrayList<Animal>();
animales.add(new Mamifero());
animales.add(new Animal());
animales.add(1, new Reptil());
animales.add(new Perro());

for (Animal unAnimal : animales) {
   System.out.println(unAnimal);
   unAnimal.metodo2();
   unAnimal.metodo1();
}
```

Respuesta: (penalty regime: 10, 20, ... %)

```
MAMIFERO
mamifero 2
animal 1
MAMIFERO
REPTIL
reptil 2
mamifero 2
mamifero 1
MAMIFERO
mamifero 2
mamifero 1
PERROMAMIFERO
mamifero 2
perro 2
animal 1
MAMIFERO
```

	Expected	Got	Comment	Mark	
✓	MAMIFERO mamifero 2 animal 1 MAMIFERO REPTIL reptil 2 mamifero 2 mamifero 1 MAMIFERO mamifero 2 mamifero 1 PERROMAMIFERO	MAMIFERO mamifero 2 animal 1 MAMIFERO REPTIL reptil 2 mamifero 2 mamifero 1 MAMIFERO mamifero 2 mamifero 1 PERROMAMIFERO	Line 0 right Line 1 right Line 2 right Line 3 right Line 4 right Line 5 right Line 6 right Line 7 right Line 8 right Line 9 right Line 10 right Line 11 right	1	✓

	Expected	Got	Comment	Mark	
	mamifero 2 perro 2 animal 1 MAMIFERO	mamifero 2 perro 2 animal 1 MAMIFERO	Line 12 right Line 13 right Line 14 right Line 15 right		

Todas las pruebas superadas.

Question author's solution:

```
MAMIFERO
mamifero 2
animal 1
MAMIFERO
REPTIL
reptil 2
mamifero 2
mamifero 1
MAMIFERO
mamifero 2
mamifero 1
PERROMAMIFERO
mamifero 2
perro 2
animal 1
MAMIFERO
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta **8**

Correcta

Puntúa 0,90 sobre 1,00

Marcar pregunta

Tenemos las siguiente clases.

Clase Coche

```
public class Coche {
    protected String marca;
    protected String modelo;

    public Coche(String marca, String modelo
) {
        this.marca = marca;
        this.modelo = modelo;
    }

    public String toString() {
        return this.getClass().getName() + "
 " + marca + " " + modelo;
    }
}
```

Clase CocheElectrico

```
public class CocheElectrico extends Coche {
   private int autonomiaKm;
   public CocheElectrico(String marca, String
modelo, int autonomiaKm) {
      super(marca, modelo);
      this.autonomiaKm = autonomiaKm;
   }

   public int getAutonomia() {
      return autonomiaKm;
   }

   public int viajar(int km) {
      autonomiaKm -= km;
      return autonomiaKm;
   }

   public void repostar(int autonomiaKm) {
      this.autonomiaKm = autonomiaKm;
   }
}
```

Clase CocheGasolina

```
public class CocheGasolina extends Coche {

   private int kilometros;

   public CocheGasolina(String marca, String
 modelo, int kilometros) {
      super(marca, modelo);
      this.kilometros = kilometros;
   }

   public int getKilometros() {
      return kilometros;
   }

   public boolean tocaRevision() {
      if (kilometros > 10000) {
         return true;
      }
      return false;
   }
```

Mejora el siguiente programa para que muestre la autonomía de todos los coches eléctricos.

Si el ArrayList tiene los siguientes elementos:

```
[Coche Opel Meriva, CocheElectrico Renault ZOE, CocheGasolina Fiat Punto, Coche Ford Fiest, CocheElectrico Hyundai Ioniq, CocheGasolina Renault Cl
io]
```

El resultado será:

```
1. CocheElectrico Renault ZOE 300
2. CocheElectrico Hyundai Ioniq 200
```

Además, de los datos del coche se muestra la autonomía.

Respuesta: (penalty regime: 10, 20, ... %)

Reiniciar respuesta

```
import java.util.*;

public class CochesTest {
   public static void main(String[] args) {
      ArrayList<Coche> listaCoches = new ArrayList<Coche>();
      listaCoches.add(new Coche("Opel", "Meriva"));
      listaCoches.add(new CocheElectrico("Renault", "ZOE", 300));
      listaCoches.add(new CocheGasolina("Fiat", "Punto", 15000));
      listaCoches.add(new Coche("Ford", "Fiest"));
      listaCoches.add(new CocheElectrico("Hyundai", "Ioniq", 200));
      listaCoches.add(new CocheGasolina("Renault", "Clio", 1500));

      System.out.println(listaCoches);
      System.out.println();

      // Autonomía de todos los coches electricos
        int cont = 0;
      for(Coche unCoche: listaCoches) {
         if(unCoche instanceof CocheElectrico) {
            cont++;
```

	Expected
✓	[Coche Opel Meriva, CocheElectrico Renault ZOE, CocheGasolina Fiat Punto, Coche Ford Fiest, CocheElectrico Hyundai Ioniq, CocheGasolina Renault 1. CocheElectrico Renault ZOE 300 2. CocheElectrico Hyundai Ioniq 200

Todas las pruebas superadas.

Question author's solution:

```
import java.util.*;

public class CochesTest {
   public static void main(String[] args) {
      ArrayList<Coche> listaCoches = new ArrayList<Coche>();
      listaCoches.add(new Coche("Opel", "Meriva"));
      listaCoches.add(new CocheElectrico("Renault", "ZOE", 300));
      listaCoches.add(new CocheGasolina("Fiat", "Punto", 15000));
      listaCoches.add(new Coche("Ford", "Fiest"));
      listaCoches.add(new CocheElectrico("Hyundai", "Ioniq", 200));
      listaCoches.add(new CocheGasolina("Renault", "Clio", 1500));

      System.out.println(listaCoches);
      System.out.println();

      // Autonomia de todos los coches electricos
      int cont = 1;
      for (Coche unCoche : listaCoches) {
         if (unCoche instanceof CocheElectrico) {
            CocheElectrico electrico = (CocheElectrico) unCoche;
            System.out.println(cont + ". " + electrico + " " + electrico.getAutonomia());
            cont++;
         }
      }
   }
```

```
}
```

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría **0,90/1,00**.

Pregunta **9**

Correcta

Puntúa 0,60 sobre 1,00

Marcar pregunta

Tenemos las siguiente clases.

Clase Coche

```
public class Coche {
    protected String marca;
    protected String modelo;

    public Coche(String marca, String modelo
) {
        this.marca = marca;
        this.modelo = modelo;
    }

    public String getMarca() {
      return marca;
    }

    public String toString() {
        return this.getClass().getName() + "
 " + marca + " " + modelo;
    }
}
```

Clase CocheElectrico

```
public class CocheElectrico extends Coche {
   private int autonomiaKm;
   public CocheElectrico(String marca, String
modelo, int autonomiaKm) {
      super(marca, modelo);
      this.autonomiaKm = autonomiaKm;
   }

   public int getAutonomia() {
      return autonomiaKm;
   }

   public int viajar(int km) {
      autonomiaKm -= km;
      return autonomiaKm;
   }

   public void repostar(int autonomiaKm) {
      this.autonomiaKm = autonomiaKm;
   }
}
```

Clase CocheGasolina

```
public class CocheGasolina extends Coche {

   private int kilometros;

   public CocheGasolina(String marca, String
 modelo, int kilometros) {
      super(marca, modelo);
      this.kilometros = kilometros;
   }

   public int getKilometros() {
      return kilometros;
   }

   public boolean tocaRevision() {
      if (kilometros > 10000) {
         return true;
      }
      return false;
   }
```

Mejora el siguiente programa para que pida una marca por teclado y muestre todos los coches de esa marca.

Si el ArrayList tiene los siguientes elementos y se introduce "Renault":

```
[Coche Opel Meriva, CocheElectrico Renault ZOE, CocheGasolina Fiat Punto, Coche Ford Fiest, CocheElectrico Hyundai Ioniq, CocheGasolina Renault Cl
io]
```

El resultado será:

```
CocheElectrico Renault ZOE
CocheGasolina Renault Clio
```

Respuesta: (penalty regime: 10, 20, ... %)

Reiniciar respuesta

```
1  import java.util.*;
2
3  public class CochesTest {
4     public static void main(String[] args) {
5        Scanner teclado = new Scanner(System.in);
6        ArrayList<Coche> listaCoches = new ArrayList<Coche>();
7        listaCoches.add(new Coche("Opel", "Meriva"));
8        listaCoches.add(new CocheElectrico("Renault", "ZOE", 300));
9        listaCoches.add(new CocheGasolina("Fiat", "Punto", 15000));
10       listaCoches.add(new Coche("Ford", "Fiest"));
11       listaCoches.add(new CocheElectrico("Hyundai", "Ioniq", 200));
12       listaCoches.add(new CocheGasolina("Renault", "Clio", 1500));
13
14       System.out.println(listaCoches);
15       System.out.println();
16
17       // Pide una marca y muestra todos los coches de esa marca
18           System.out.print("Introduce marca: ");
19           String marca = teclado.next();
20
```

	Input	Expected
✓	Renault	[Coche Opel Meriva, CocheElectrico Renault ZOE, CocheGasolina Fiat Punto, Coche Ford Fiest, CocheElectrico Hyundai Ioniq, CocheGaso Introduce marca: CocheElectrico Renault ZOE CocheGasolina Renault Clio
✓	Opel	[Coche Opel Meriva, CocheElectrico Renault ZOE, CocheGasolina Fiat Punto, Coche Ford Fiest, CocheElectrico Hyundai Ioniq, CocheGaso Introduce marca: Coche Opel Meriva
✓	Volkswagen	[Coche Opel Meriva, CocheElectrico Renault ZOE, CocheGasolina Fiat Punto, Coche Ford Fiest, CocheElectrico Hyundai Ioniq, CocheGaso Introduce marca:

Todas las pruebas superadas. ✓

Question author's solution:

```
import java.util.*;

public class CochesTest {
   public static void main(String[] args) {
      ArrayList<Coche> listaCoches = new ArrayList<Coche>();
      listaCoches.add(new Coche("Opel", "Meriva"));
      listaCoches.add(new CocheElectrico("Renault", "ZOE", 300));
      listaCoches.add(new CocheGasolina("Fiat", "Punto", 15000));
      listaCoches.add(new Coche("Ford", "Fiest"));
      listaCoches.add(new CocheElectrico("Hyundai", "Ioniq", 200));
      listaCoches.add(new CocheGasolina("Renault", "Clio", 1500));

      System.out.println(listaCoches);
      System.out.println();

      // Pide una marca y muestra todos los coches de esa marca
      Scanner leer = new Scanner(System.in);
      System.out.print("Introduce marca: ");
      String marca = leer.next();
      for (Coche unCoche : listaCoches) {
         String laMarca = unCoche.getMarca();
         if (laMarca.equals(marca)) {
            System.out.println(unCoche);
         }
      }
   }
}
```

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría **0,60/1,00**.

Pregunta **10**

Correcta

Puntúa 0,80 sobre 1,00

Marcar pregunta

Tenemos las siguiente clases.

Clase Coche

```
public class Coche {
    protected String marca;
    protected String modelo;

    public Coche(String marca, String modelo
) {
        this.marca = marca;
        this.modelo = modelo;
    }
```

Clase CocheElectrico

```
public class CocheElectrico extends Coche {
   private int autonomiaKm;
   public CocheElectrico(String marca, String
modelo, int autonomiaKm) {
      super(marca, modelo);
      this.autonomiaKm = autonomiaKm;
   }

   public int getAutonomia() {
```

Clase CocheGasolina

```
public class CocheGasolina extends Coche {

   private int kilometros;

   public CocheGasolina(String marca, String
 modelo, int kilometros) {
      super(marca, modelo);
      this.kilometros = kilometros;
   }
```

```
    public String toString() {
        return this.getClass().getName() + "
 " + marca + " " + modelo;
    }
}
```

```
        return autonomiaKm;
    }

    public int viajar(int km) {
        autonomiaKm -= km;
        return autonomiaKm;
    }

    public void repostar(int autonomiaKm) {
        this.autonomiaKm = autonomiaKm;
    }
}
```

```
    public int getKilometros() {
        return kilometros;
    }

    public boolean tocaRevision() {
        if (kilometros > 10000) {
            return true;
        }
        return false;
    }
```

Mejora el siguiente programa para que muestre los kilometros de todos los coches de gasolina.

Si el ArrayList tiene los siguientes elementos:

```
[Coche Opel Meriva, CocheElectrico Renault ZOE, CocheGasolina Fiat Punto, Coche Ford Fiest, CocheElectrico Hyundai Ioniq, CocheGasolina Renault Cl
io]
```

El resultado será:

```
1. CocheGasolina Fiat Punto 15000
2. CocheGasolina Renault Clio 1500
```

Además, de los datos del coche se muestran los kilometros.

Respuesta: (penalty regime: 10, 20, ... %)

Reiniciar respuesta

```
1  import java.util.*;
2
3  public class CochesTest {
4     public static void main(String[] args) {
5        ArrayList<Coche> listaCoches = new ArrayList<Coche>();
6        listaCoches.add(new Coche("Opel", "Meriva"));
7        listaCoches.add(new CocheElectrico("Renault", "ZOE", 300));
8        listaCoches.add(new CocheGasolina("Fiat", "Punto", 15000));
9        listaCoches.add(new Coche("Ford", "Fiest"));
10       listaCoches.add(new CocheElectrico("Hyundai", "Ioniq", 200));
11       listaCoches.add(new CocheGasolina("Renault", "Clio", 1500));
12
13       System.out.println(listaCoches);
14       System.out.println();
15
16       // Kilometros de todos los coches de gasolina
17       int cont = 0;
18       for(Coche unCoche: listaCoches) {
19          if(unCoche instanceof CocheGasolina) {
20
```

	Expected

✓	`[Coche Opel Meriva, CocheElectrico Renault ZOE, CocheGasolina Fiat Punto, Coche Ford Fiest, CocheElectrico Hyundai Ioniq, CocheGasolina Renault`
	
	`1. CocheGasolina Fiat Punto 15000`
	`2. CocheGasolina Renault Clio 1500`

Todas las pruebas superadas. ✓

Question author's solution:

```
import java.util.*;

public class CochesTest {
   public static void main(String[] args) {
      ArrayList<Coche> listaCoches = new ArrayList<Coche>();
      listaCoches.add(new Coche("Opel", "Meriva"));
      listaCoches.add(new CocheElectrico("Renault", "ZOE", 300));
      listaCoches.add(new CocheGasolina("Fiat", "Punto", 15000));
      listaCoches.add(new Coche("Ford", "Fiest"));
      listaCoches.add(new CocheElectrico("Hyundai", "Ioniq", 200));
      listaCoches.add(new CocheGasolina("Renault", "Clio", 1500));

      System.out.println(listaCoches);
      System.out.println();

      // Kilometros de todos los coches de gasolina
      int num = 1;
      for (Coche unCoche : listaCoches) {
         if (unCoche instanceof CocheGasolina) {
            CocheGasolina gasolina = (CocheGasolina) unCoche;
            System.out.println(num + ". " + gasolina + " " + gasolina.getKilometros());
            num++;
         }
      }
   }
}
```

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría **0,80/1,00**.

Finalizar revisión