1. Una empresa argentina de Luthieres utiliza una aplicación informática codificada en Java que utiliza una entidad llamada **Mandocleta** que permite gestionar los datos de cada una de sus creaciones de este tipo de instrumentos.

   Añadir a la clase un **constructor por defecto** y un **constructor parametrizado** de forma que los objetos se instancien en un **estado consistente**.

   Entre otros recursos la entidad dispone del siguiente **atributo**:

   - ✔ `double velocidad;`

     Velocidad en m/s a la que se desplaza el instrumento por el escenario. Se puede ir hacia atrás.

2. Una clínica veterinaria canina de **Reikiavik** utiliza una aplicación informática codificada en Java que utiliza una entidad llamada **Perro** que permite gestionar los datos sanitarios de cuarentena de los perros que entran al país.

   Añadir a la clase un **constructor por defecto** y un **constructor parametrizado** de forma que los objetos se instancien en un **estado consistente**.

   Entre otros recursos la entidad dispone del siguiente **atributo**:

   - ✔ `boolean cuarentenaOK;`

     Indica si el animal en cuestión ha pasado (true) o no (false) la cuarentena de entrada a Islandia.

3. Una inteligencia artificial llamada **HAL-9000** dedicada a las simulación del crecimiento del perejil en condiciones de microgravedad utiliza una aplicación informática codificada en Java que utiliza una entidad llamada **Maceta** que permite dar ánimos a cada una de las simulaciones de dichos vegetales para que crezcan más.

   Añadir a la clase un **constructor por defecto** y un **constructor parametrizado** de forma que los objetos se instancien en un **estado consistente**.

   Entre otros recursos la entidad dispone del siguiente **atributo**:

   - ✔ `String nombre;`

     Nombre utilizado por HAL-9000 para dar ánimos a la planta. Por defecto se llamará "Anita".

4. La asociación cultural **Tim & Fire**, conocida por su promoción del Folclore Tradicional de la isla de **Lanzarote**, distribuye llaveros con el logo del **Plátano de Canarias** entre los fabricantes de **queso de vaca** del **Alto Bidasoa** en **Guipúzcoa**. Para mantenerse en contacto con sus simpatizantes, la asociación maneja una aplicación informática codificada en Java que utiliza una entidad llamada **Contacto** que permite saludar a cada uno de los contactados dependiendo si son mayores de edad o no.

   Añadir a la clase un **constructor por defecto** y un **constructor parametrizado** de forma que los objetos se instancien en un **estado consistente**.

   Entre otros recursos la entidad dispone de los siguientes **atributos**:

   - ✔ `String alias;`

     Nombre informal del contacto. Por defecto será "Aupatú"

   - ✔ `int edad;`

     Edad del contacto expresada en años. Nunca inferior a 18 años.

5. El grupo de Rap llamado **Flander's Tutiplén** afincado en la ciudad de **Torredembarra** en **Tarragona**, ha organizado un **concurso de cocido madrileño** entre los vecinos del barrio donde está su local de ensayos. Para ello han encargado una aplicación informática codificada en Java que utiliza una entidad llamada **MasterChef** que permite gestionar el presupuesto de cada uno de los equipos de cocineros. Un equipo tendrá un **mínimo** de 2 cocineros y un **máximo** de 4, y su presupuesto no podrá superar los 30€.

   Añadir a la clase un **constructor por defecto** y un **constructor parametrizado** de forma que los objetos se instancien en un **estado consistente**.

   Entre otros recursos la entidad dispone de los siguientes **atributos**:

   - ✔ `int miembros;`

     Número de personas del equipo. Por defecto: 3

   - ✔ `double presupuesto;`

     Presupuesto del equipo en €. Por defecto: 21

6. La gerencia del gimnasio **Nenazas Gym** dispone de una aplicación informática codificada en Java que utiliza una entidad llamada **Cafetera** que permite manejar los pedidos de la máquina de "vending" de la recepción.

   Añadir a la clase un **constructor por defecto** y un **constructor parametrizado** de forma que los objetos se instancien en un **estado consistente**.

   Entre otros recursos la entidad dispone de las siguientes **constantes**:

   - ✔ `CAFE       = 0;`
   - ✔ `TE         = 1;`
   - ✔ `BATIDO     = 2;`
   - ✔ `CON_AZUCAR = 4;`
   - ✔ `SIN_AZUCAR = 8;`

   Entre otros recursos la entidad dispone de los siguientes **atributos**:

   - ✔ `int servicio;`

     Codifica la petición actual. Por defecto: 8 (Café sin azúcar)

7. La clase **Posicion** encapsula los datos de posición asociados a un píxel de la **pantalla gráfica táctil de un teléfono móvil** de 800 pixeles de ancho por 600 pixeles de alto. A los efectos de este ejercicio esta clase se estructura a partir de dos números enteros referidos a un cuadrante cartesiano positivo que tiene su origen de coordenadas en la esquina superior izquierda de la pantalla. Uno de ellos se refiere al eje de abscisas y el otro para el eje de ordenadas. El **origen de coordenadas** se sitúa en la **esquina superior izquierda** lo que determina que la primera posición visible corresponda a las coordenadas [0, 0]. Por tanto la última coordenada visible corresponde a las coordenadas [799, 599] que se sitúa en la esquina inferior derecha.

   Añadir a la clase un **constructor por defecto** y un **constructor parametrizado** de forma que los objetos se instancien en un **estado consistente**.

Campos

- Tamaño de pantalla
```
public final static int W_SCR = 800; // Ancho
public final static int H_SCR = 600; // Alto
```
- Posición por defecto
```
public final static int X_INI = W / 2;
public final static int Y_INI = H / 2;
```
- Coordenadas
```
private int x;
private int y;
```

8. La clase **Nif** encapsula el NIF de una persona. A los efectos de este ejercicio el formato del NIF consiste básicamente en un número entero de ocho dígitos numéricos más un dígito alfabético de control. El dígito de control del NIF, que sirve como código de garantía de seguridad, se obtiene dividiendo el número de DNI entre 23 y al resto de dicha división que deberá estar comprendido entre 0 y 22 se le asigna la letra según la equivalencia:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

   Todo objeto instanciado tendrá siempre un NIF correcto.

   Añadir a la clase un **constructor por defecto** y un **constructor parametrizado** de forma que los objetos se instancien en un **estado consistente**.

   Campos

   - Lista de letras
```
public final static String LETRAS = "TRWAGMYFPDXBNJZSQVHLCKE";
```
   - Valores limite para el DNI
```
public final static int DNI_MIN = 10000000; (POR DEFECTO)
public final static int DNI_MAX = 99999999;
```
   - Número de DNI
```
private int dni;
```

9. La clase **Hora** encapsula una hora. A los efectos de este ejercicio el formato de hora consiste básicamente en **tres números enteros**. Uno de ellos para las **horas** (entre 0 y 23), otro para los **minutos** (entre 0 y 59) y el tercero para los **segundos** (entre 0 y 59).

   Añadir a la clase un **constructor por defecto** y un **constructor parametrizado** de forma que los objetos se instancien en un **estado consistente**.

   Atributos

   - Horas, minutos y segundos.
```
private int h;
private int m;
private int s;
```

10. La clase **Fecha** encapsula una fecha. A los efectos de este ejercicio el formato de fecha consiste básicamente en tres números enteros. Uno de ellos para el año (positivo o negativo), otro para el mes (entre 1 para enero hasta 12 para diciembre) y el tercero para el día (entre 1 y 28, 29, 30 o 31, dependiendo del mes y el año). Para determinar si un año es bisiesto se atendrá a la siguiente regla: "Si un año es divisible por 400 es bisiesto, en caso contrario si es divisible por 100 entonces no es bisiesto, pero si esto último no sucede y es divisible por 4 entonces es bisiesto, y si esto último tampoco acontece entonces definitivamente no es bisiesto".

    Añadir a la clase un **constructor por defecto** y un **constructor parametrizado** de forma que los objetos se instancien en un **estado consistente**.

Atributos

- Nombres de los días de la semana
```
public final static String[] DIAS = {"lunes", "martes",
       "miércoles", "jueves", "viernes", "sábado", "domingo"};
```
- Nombres de los meses del año
```
public final static String[] MESES = {"enero", "febrero",
       "marzo", "abril", "mayo", "junio", "julio", "agosto",
       "septiembre", "octubre", "noviembre", "diciembre"};
```
- Nombres de las estaciones del año
```
public final static String[] ESTACIONES = {"primavera",
       "verano", "otoño", "invierno"};
```
- Día, mes y año.
```
private int dia;
private int mes;
private int any;
```

11. La clase **Color** encapsula los datos de un color codificado en HTML. A los efectos de este ejercicio el formato consiste en tres números enteros referidos a los componentes de color rojo, verde y azul respectivamente. Sus valores individuales van desde 0 para ausencia de color hasta 255 para presencia de color. Si un canal se pasa de rango se pondrá el límite más próximo. Todo color instanciado tendrá siempre un color válido.

    Añadir a la clase un **constructor por defecto** y un **constructor parametrizado** de forma que los objetos se instancien en un **estado consistente**.

    Atributos

    - Niveles de color
```
public static final int NIV_MIN = 0; (Por defecto)
public static final int NIV_MAX = 255;
```
    - Rojo, verde y azul.
```
private int r;
private int v;
private int a;
```