PROGRAMACIÓN - EXAMEN FINAL 3º EVALUACIÓN

C.F.G.S 1º D.A.M. 14-Mayo-2021 I.E.S. Fernando Aguilar Quignón

Instrucciones para la realización del examen:

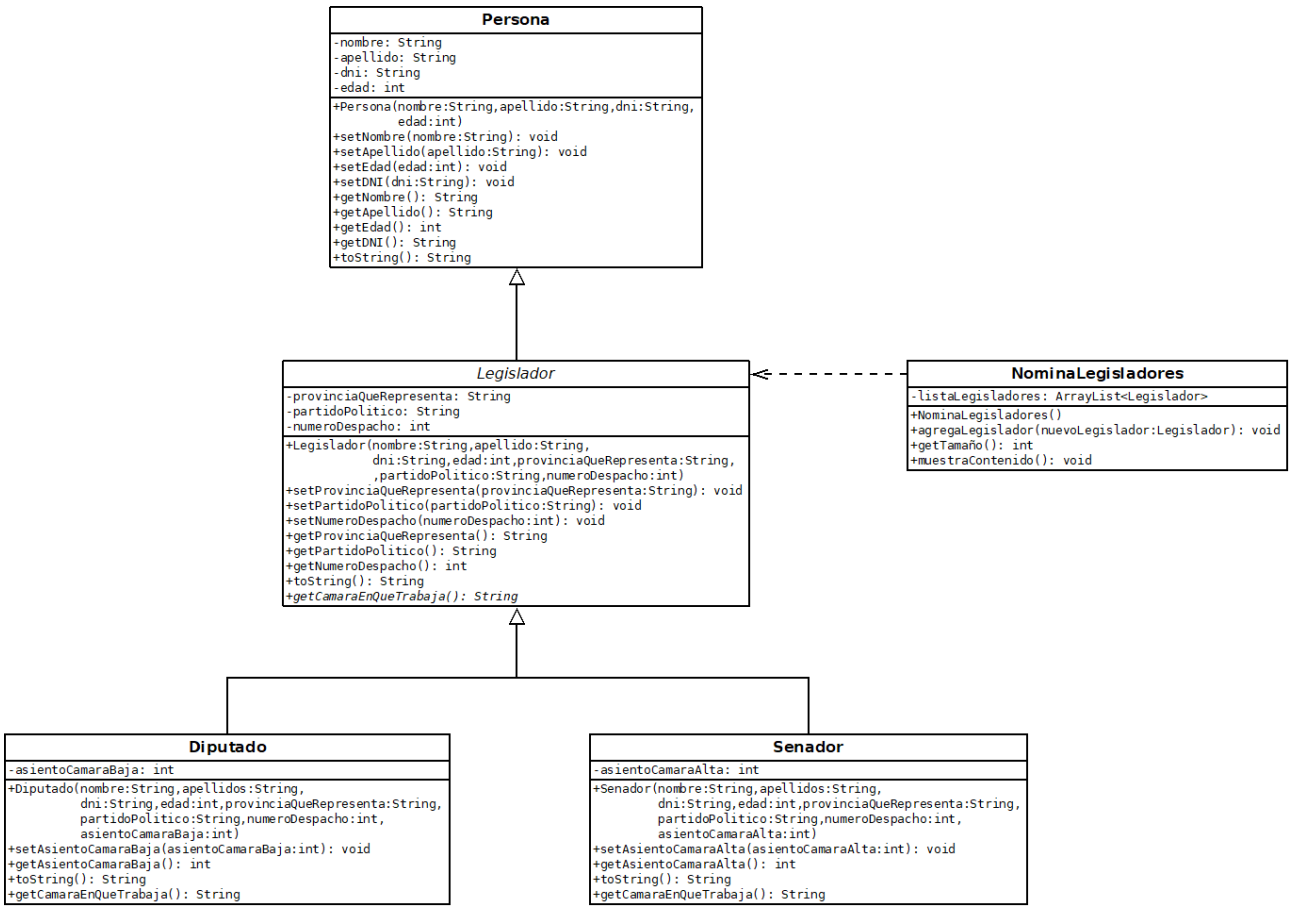
- Al entregar tu examen, debes ser capaz de explicar cada parte de tu código. No está permitida la copia de compañeros u otro medio. No puedes comunicarte con tus compañeros de ninguna manera mientras el examen está en progreso. En caso de detectar copias, el examen se puntuará con un 0.

- Tu pantalla está siendo monitorizada y solo puedes tener abierto este examen, y NetBeans. No está permitido abrir ningún otro documento, ni ningún navegador web.

- Al finalizar el examen debes entregar en classroom un único proyecto con cada uno de los ejercicios organizados por paquetes.

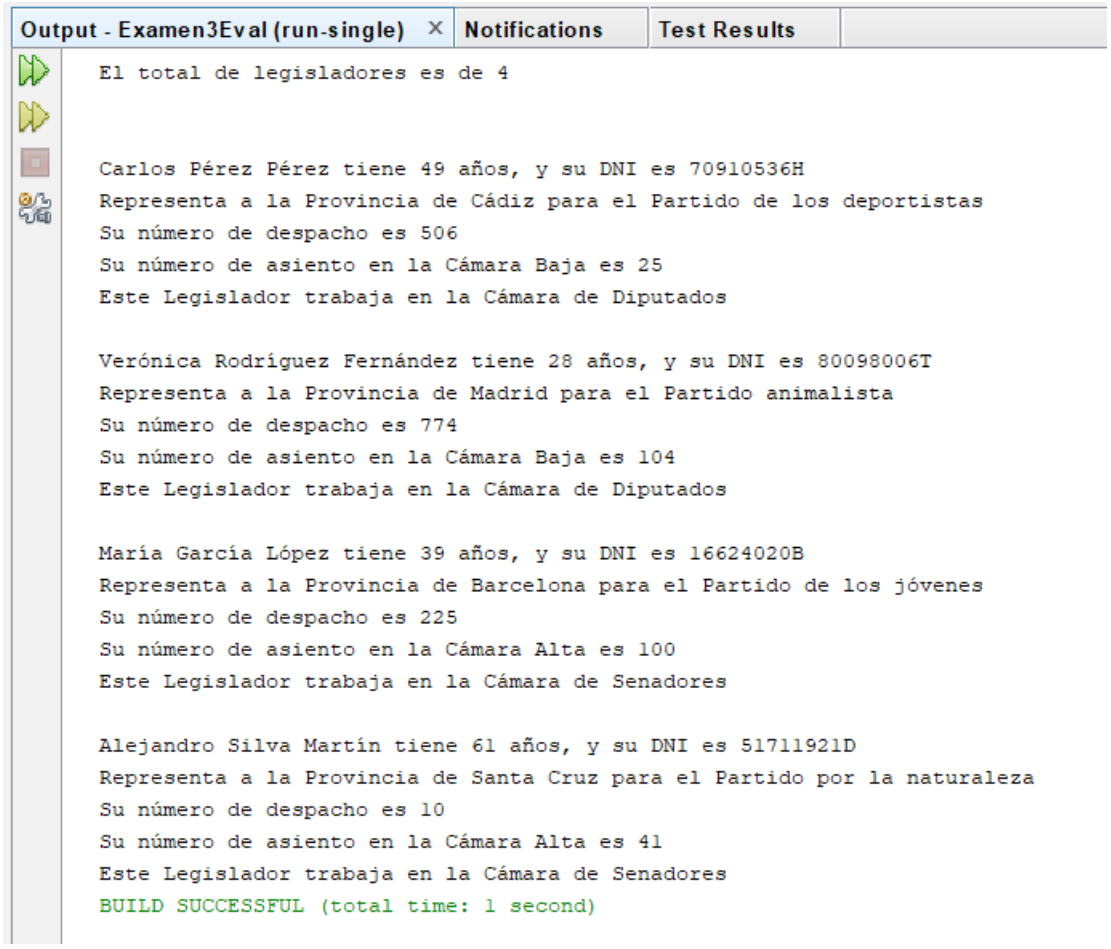
Ejercicio 1. (3,5 puntos)

Escribe el código correspondiente al siguiente diagrama UML:



Puntuación de cada apartado del ejercicio:

- Método toString() de la clase Persona (0,25 puntos).
- Crear la clase Legislador (0,5 puntos).
- Crear la clase Diputado (0,5 puntos).
- Crear la clase Senador (0,5 puntos).
- Crear la clase NominaLegisladores. En esta clase, el método muestraContenido () indicará para cada legislador si trabaja en la cámara de diputados o en la cámara de senadores. Para que este método se califique positivamente es obligatorio reutilizar el máximo código posible. (0,75 puntos).
- Crear una nueva clase main que muestre el correcto funcionamiento del programa haciendo lo siguiente: (1 punto)
    * Crea dos diputados y dos senadores con todos sus parámetros.
    * Reutiliza las clases y métodos del programa oportunos para mostrar por consola una salida igual a la siguiente:



Otras indicaciones para la realización del ejercicio:

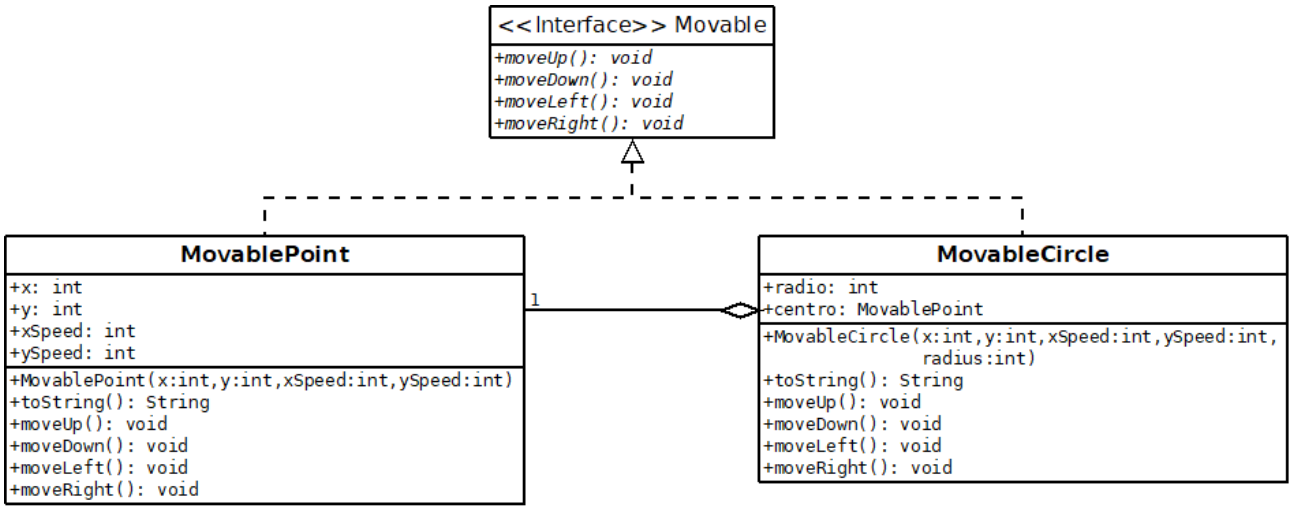
- Es estrictamente obligatorio seguir la estructura proporcionada en el diagrama UML.

- En este ejercicio se valorará sobre todo el uso del polimorfismo, y la reutilización de la mayor cantidad de código posible.

Ejercicio 2. (2,5 puntos)

Supongamos que tenemos un conjunto de objetos con algunos comportamientos comunes: podrían moverse hacia arriba, hacia abajo, hacia la izquierda o hacia la derecha. Los comportamientos exactos (tales como cómo moverse y qué tan lejos moverse), dependen de los propios objetos.

Escribe el código correspondiente al siguiente diagrama UML:



Puntuación de cada apartado del ejercicio:

- Crear la clase MovablePoint (0,25 puntos)

- Crear la clase MovableCircle (0,5 puntos)

- Escribe una nueva clase llamada MovableRectangle, que se compone de dos MovablePoints (que representan las esquinas superior izquierda y inferior derecha), y que implementa la Interfaz Movable. Asegúrate de que los dos puntos tengan la misma velocidad (0,25 puntos).

- Crea las siguientes excepciones, que deben ser comprobadas dentro del constructor de la clase MovablePoint y propagadas:

* PuntoFueraDeGraficaException: Saltará cuando “x” o “y” sean mayor de 50 y se mostrará por pantalla el mensaje: "Error: el punto es demasiado grande, fuera de gráfica".

* ExcesoDeVelocidadException: Saltará cuando “xSpeed” o “ySpeed” sean mayor de 200 y se mostrará por pantalla el mensaje: "Error: Velocidad en datos de entrada demasiado alta". (0,5 puntos)

- Crear una nueva clase main de nombre “TestMovable” que muestre el correcto funcionamiento del programa haciendo lo siguiente: (1 punto)

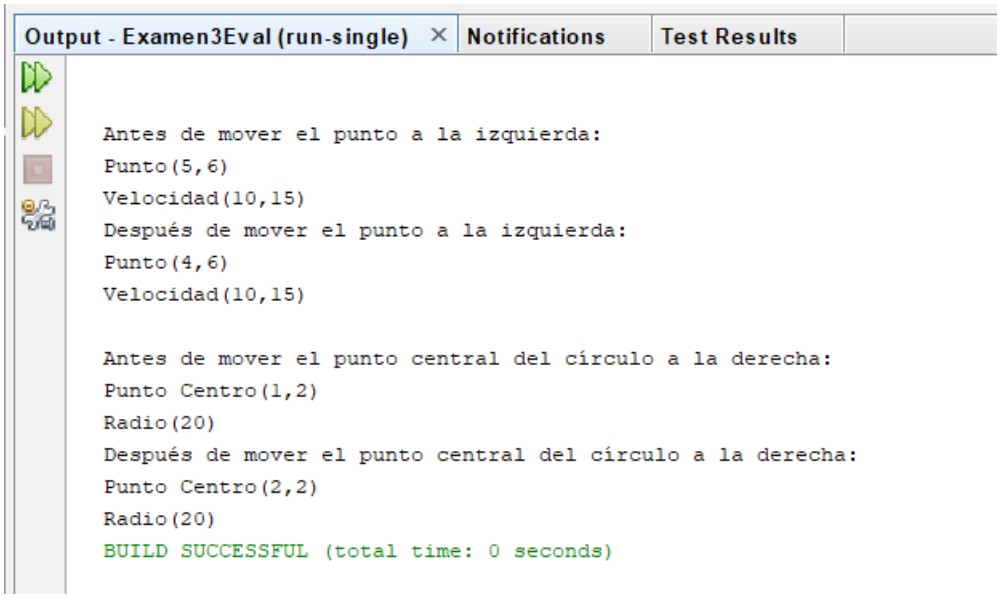
* Captura y gestiona las excepciones.

* Crea un objeto de tipo MovablePoint.
* Muestra el punto por pantalla antes de desplazarlo.
* Mueve el punto a la izquierda.
* Muestra el punto por pantalla después de desplazarlo.

* Crea un objeto de tipo MovableCircle.
* Muestra el circulo por pantalla antes de desplazarlo.
* Mueve el circulo a la derecha.
* Muestra el circulo por pantalla después de desplazarlo.

* Código que haga saltar la excepción PuntoFueraDeGraficaException (coméntalo)
* Código que haga saltar la excepción ExcesoDeVelocidadException (coméntalo)
* Usa el método getMessage() al gestionar las excepciones para mostrar el mensaje de error.

* Reutiliza las clases y métodos del programa oportunos para mostrar por consola una salida igual a la siguiente:



Otras indicaciones para la realización del ejercicio:

- Es estrictamente obligatorio seguir la estructura proporcionada en el diagrama UML.

- En este ejercicio se valorará sobre todo el uso del polimorfismo, y la reutilización de la mayor cantidad de código posible.

Para la realización de los siguientes ejercicios, utiliza como base de datos el fichero "examen.db" proporcionado. La estructura de la base de datos es la siguiente:

```
CREATE TABLE "EQUIPO" (
        "nomeq" varchar(20),
        "descripcion" varchar(100),
        PRIMARY KEY("nomeq")
);
```

```
CREATE TABLE "CICLISTA" (
        "dorsal" INTEGER,
        "nombre" varchar(30),
        "edad" INTEGER,
        "nomeq" varchar(20),
        PRIMARY KEY("dorsal"),
        FOREIGN KEY("nomeq") REFERENCES "EQUIPO"("nomeq")
);
```

Ejercicio 3. (1,5 puntos)

Realiza una conexión con SQLite al fichero "examen.db" proporcionado. Una vez conectado con la BBDD, nuestro programa debe hacer la siguiente consulta SQL:

- Listar todas las columnas de la tabla "equipos".

Para la realización de este ejercicio debes usar **obligatoriamente** una sentencia de tipo "Statement" sin parámetros.

Ejercicio 4. (2,5 puntos)

Realiza una conexión con SQLite al fichero "examen.db" proporcionado. Una vez conectado con la BBDD, nuestro programa debe hacer las siguientes consultas SQL:

- Inserta en la tabla EQUIPO el equipo "ONCE" con la descripción "Jose Luis Pérez".
- Inserta en la tabla CICLISTA un registro con los siguientes datos (99, "Juan Antonio", 25, "ONCE").
- Lista todas las columnas de la tabla CICLISTA.

Para la realización de este ejercicio debes usar **obligatoriamente** sentencias de tipo "PreparedStatement". En el caso de las instrucciones INSERT serán con parámetros, y en el caso de la instrucción SELECT será sin parámetros.

Utiliza los métodos que proporciona "PreparedStatement" para mostrar por pantalla con un mensaje cuántas filas se han visto afectadas en la tabla equipo y cuantas filas se han visto afectadas en la tabla ciclista.