T2: Herencia y polimorfismo.

1. Diseñar y codificar una aplicación informática para una compañía de gestión aeroportuaria satisfaciendo los siguientes requisitos:
   - Para cada aeropuerto es necesario saber:
     a) Todas las compañías de vuelos que operan en él.
     b) Nombre del aeropuerto, la ciudad donde se ubica y el país al que pertenece.
   - Cada compañía se caracteriza por el nombre y la lista de vuelos que ofrece.
   - Los vuelos están definidos por su identificador, la ciudad de origen, la ciudad de destino, el precio del viaje, la lista de pasajeros, el número máximo de pasajeros permitidos en el vuelo y el número real de pasajeros que ha reservado asiento en el avión.
   - Los aeropuertos pueden ser privados o públicos.
     a) Los aeropuertos privados tienen una serie de empresas que les patrocinan y es necesario saber el nombre de cada una de las empresas.
     b) Para los aeropuertos públicos se requiere saber la cantidad de dinero correspondiente a la subvención gubernamental.
   - Se necesita gestionar también la información de los pasajeros.
     a) Para cada pasajero se necesita saber nombre, número de pasaporte y nacionalidad.

La aplicación tendrá un menú con las siguientes opciones:

a) Consultar los aeropuertos gestionados, indicando separadamente los aeropuertos públicos y los privados. Para cada uno de ellos deberá mostrar su nombre, la ciudad de ubicación y el país al que pertenece.
b) Visualizar las empresas que patrocinan un determinado aeropuerto en caso que sea privado, o la cuantía de la subvención en caso que se trate de un aeropuerto público.
c) Para un determinado aeropuerto, se debe poder mostrar la lista de compañías que vuelan desde ese aeropuerto.
d) Para una determinada compañía que opera en un aeropuerto concreto, listar todos los posibles vuelos que dicha compañía ofrece, mostrando su identificador, la ciudad de origen y destino y el precio de vuelo.
e) Mostrar todos los posibles vuelos (identificador) que parten de una ciudad origen a otra ciudad destino (indicadas por el usuario) y mostrar su precio.
f) Almacenar en un fichero secuencial de texto, el nombre y ciudad de todos los aeropuertos junto con el nombre de todas las compañías que operan en cada aeropuerto. Mostrar por pantalla el contenido del fichero comprobando que los datos son correctos.

2. En una mediateca tenemos disponibles tres tipos diferentes de documentos: Libros, DVDs (video) y CDs (audio). Todos los documentos se caracterizan por una fecha de edición, un título, un valor nominal (en euros), y un atributo que indica si el documento está prestado o no. A su vez, los libros deben incluir el autor, los DVDs deben incluir el director de la película, y los CDs el artista correspondiente. Si una vez prestado un documento no se devuelve en la fecha debida existe una penalización en todos ellos. En los libros la penalización se obtiene como un 5% de su valor nominal. En los DVDs es un 6% del valor nominal más un valor fijo de 2 euros. En los CDs se aplica un 10% del valor nominal más un valor fijo de 2 euros. A su vez, los socios de la mediateca se caracterizan por su nombre, DNI y número de socio. Los socios de la mediateca pueden sacar en préstamo un máximo de 2 documentos. En caso de retraso en la devolución de los documentos prestados se aplica al socio una penalización obtenida como la suma de las penalizaciones de cada documento prestado.

    a) Diseñar el diagrama de clases en notación UML indicando las relaciones entre las mismas.
    b) Implementar en Java las clases y / o interfaces resultantes.
    c) Implementar un programa principal que defina tres documentos de tipos distintos. Los documentos pueden inicializarse directamente en el programa principal. A continuación cree un objeto de clase socio que saque en préstamo los tres documentos anteriores. Finalmente, imprimir los datos del socio así como la penalización correspondiente por un retraso en la devolución de los tres documentos. Utilizar una matriz para almacenar los objetos documento haciendo uso del polimorfismo.

3. Una empresa de envío de paquetes tiene varias sucursales en todo el país. Cada sucursal está definida por su número de sucursal, dirección y ciudad. Además, conviene distinguir si la sucursal es Principal o Secundaria. En caso de ser sucursal Principal se debe saber el número de sucursales secundarias que dependen de ella y en el caso de la Secundaria es necesario saber el nombre de la sucursal de la que depende. Para calcular el precio que cuesta enviar cada paquete, las sucursales tienen en cuenta el peso del paquete y la prioridad, sabiendo que se cobra 1 euro por kilo, 10 euros más si la prioridad es alta y 20 si es Express. Cada paquete enviado tendrá un número de referencia. Se pide:

a) Diseñar el diagrama de clases en notación UML indicando las relaciones entre las mismas.
b) Implementar en Java las clases y / o interfaces resultantes.
c) Construir un programa principal donde se cree un objeto sucursal Secundaria que dependa de una sucursal Principal llamada “Principal1”. El programa debe imprimir el precio y el peso de 3 paquetes, uno es de 20 kg y sin prioridad, otro de 30 kg y prioridad alta, y uno de 3 kg con prioridad Express. El objeto Sucursal Secundaria deberá crearse mediante una referencia polimórfica.

4. Se pretende automatizar el sistema de una empresa dedicada al alquiler de viviendas. Los tipos de viviendas que la empresa ofrece son pisos y chalets. De todas las viviendas se registra u identificador (un valor entero), metros cuadrados y precio mensual de alquiler. De los pisos se debe saber si están amueblados y el año de construcción. Por otra parte, de los chalets se debe saber si tienen piscina. Para cada tipo de vivienda también se desea conocer el nombre y DNI de los tres últimos inquilinos. Se pide:
   a) Diseñar el diagrama de clases en notación UML indicando las relaciones entre las mismas.
   b) Implementar en Java las clases y / o interfaces resultantes.
   c) Construir un programa principal donde se guarde un objeto piso y uno chalet en una única matriz. Posteriormente, se debe consultar usando esa matriz el nombre y DNI de cada uno de los tres últimos inquilinos que ha tenido esa vivienda.

5. Construir un programa para una biblioteca que contiene libros y revistas. Ambas publicaciones tienen un código, un año de publicación y un atributo prestado. Cuando se crea la publicación, el atributo prestado está a falso porque la publicación no está prestada. Las revistas tienen un número y los libros un nombre del autor. Ambos tipos de publicaciones deben tener métodos de consulta de todos sus atributos Para prevenir posibles cambios en el programa se tiene que implementar una interfaz que contenga los métodos prestar(), devolver(), y prestado(). La clase publicación implementará dicha interfaz.

El programa deberá obtener el número de publicaciones prestadas. Para ello, la clase principal contendrá dos métodos:

- Un método que cuente las publicaciones prestadas. Dicho método recibirá como parámetro una lista de las publicaciones existentes en la biblioteca y devolverá cuántas hay prestadas.
- El método main, que creará una matriz de referencias polimórficas de tipo publicación, invocará al método con el que se obtienen el número de publicaciones prestadas e imprimirá el número de publicaciones prestadas. También se deberá imprimir los datos de todas las publicaciones. Como ejemplo, crear una matriz de 2 libros y 2 revistas, y considerar que se prestan un libro y una revista.

Se debe:

a) Diseñar el diagrama de clases en notación UML indicando las relaciones entre las mismas.
b) Implementar en Java las clases y / o interfaces resultantes.
c) Construir un programa principal en el que se realicen las operaciones descritas anteriormente (cálculo de publicaciones prestadas usando un método específico para tal fin e impresión en el método main del número de publicaciones prestadas).

6. Construir un programa en el que se defina una clase Madre, una clase Hija que herede de la clase Madre y una clase Nieta que herede de la clase Hija. Todas las clases deben tener un método que muestre por pantalla su descripción. Con la anterior estructura de clase, implemente dos métodos main en los que creará un objeto de cada tipo de clase con el que se invocará al método que muestra su descripción por pantalla:
   a) En el primer método no se hará uso del polimorfismo para invocar el método de descripción.
   b) En el segundo se hará uso del polimorfismo para invocar el método de descripción.

7. Implementar un programa que defina tres diferentes clases de deporte (Fútbol, Baloncesto, Rugby), que hereden de otra clase inicial genérica “Deporte” y redefinan el mismo método que describe el tipo de puntuación empleado en dicho de porte. Crear un array de 6 objetos que pertenezcan aleatoriamente a alguno de estos deportes y ejecuten el mismo método de su clase.

8. En un banco se tienen una serie de clientes caracterizados por su NIF, número de cuenta y saldo. Dentro de estos clientes existe un cliente especial llamado preferente. Dicho cliente se caracteriza porque posee varios fondos de inversión. Cada fondo se caracteriza por su número y el saldo mensual. Se pide:
   a) Diseñar el diagrama de clases en notación UML indicando las relaciones entre las mismas.
   b) Implementar en Java las clases y / o interfaces resultantes.
   c) Construir un programa principal donde se creen un cliente y un cliente preferente introduciendo directamente los valores de sus atributos en variables. No es necesario leer estos datos por teclado. Para cada uno de ellos se debe poder consultar todos los atributos que contiene. Además, para el caso del cliente preferente se debe poder consultar también los datos de cada uno de sus fondos (número y el saldo mensual). El número de fondos que posee el cliente preferente y las características de cada fondo deben introducirse por teclado.

9. Una tienda de informática vende dos tipos de artículos (software y hardware) ambos caracterizados por su código (de tipo char) y su descripción. Además, en los productos software hay que indicar el tipo de producto (con una cadena), y en los hardware se debe indicar si es un periférico o no. Para saber el precio de un artículo utilice una interfaz que indique que para un artículo cuyo código es A el precio es 100,0 euros; si el código es B, el precio es 50,3; para un artículo con código C, el precio es 150,50 euros. En el caso de ser un producto hardware, los artículos sólo serán A o B, y si es un periférico, el precio se incrementará en un 10 %. Los productos software serán exclusivamente B o C, incrementándose el precio del artículo en un 5% si el software es de tipo ProgramaJuegos.
   a) Diseñar el diagrama de clases en notación UML, indicando las relaciones entre las mismas.
   b) Implementar en Java las clases y / o interfaces resultantes.
   c) Construir un programa principal donde se cree un objeto software y otro hardware y se muestre el precio y demás características de cada objeto.

10. Una empresa de transporte de viajeros tiene una flota de autobuses. Cada autobús se caracteriza por un número de identificación, un conductor y un precio base del viaje. A su vez, el conductor se caracteriza por un nombre y un salario. Puede haber dos tipos de autobuses: urbanos e interurbanos. La diferencia entre ellos es la forma en la que se calcula el precio del viaje. En todos los casos debe incrementarse el precio base en una cantidad que se calcula de diferente forma en cada tipo de autobús. En el caso de los autobuses urbanos, el precio está en función de la ruta. Si la ruta es A, el precio del billete es el precio base + 10 % de dicho precio. El precio de cualquier otra ruta es el precio base + 20 %. En el ccaso de los autobuses interurbanos, el precio se calcula en función del número de kilómetros, multiplicando éste por el precio base. Se considera que la ruta y el número de kilómetros son variables que caracterizan al tipo de autobús correspondiente. Se pide:
    a) Diseñar el diagrama de clases en notación UML indicando las relaciones entre las mismas.
    b) Implementar en Java las clases y / o interfaces resultantes.
    c) Construir un programa principal que muestre:
    - El precio de un viaje en el autobús urbano número 1 que sigue la ruta B, sabiendo que el precio base es de 0,4 euros.

- El precio del viaje en el autobús interurbano número 2, que recorre 1000 km, cuyo precio base es de 0,05 euros.

En ambos casos se debe imprimir el precio del billete de cada autobús, su número de identificación y el nombre del conductor que es el mismo en ambos autobuses. Utilice una matriz para almacenar los dos objetos autobús haciendo uso del polimorfismo.

Nota: no considere el precio total del billete como un atributo de autobús.

11. Se pretende controlar las personas que están involucradas en un campeonato de tenis: jueces, recogepelotas y azafatas, todos ellos caracterizados por su nombre y edad. Además, del juez es conveniente conocer sus años de experiencia; de los recogepelotas si es la primera vez que ejercen; y de las azafatas, el nombre de la empresa de la cual dependen. Se debe poder calcular el salario de cada uno, el cual se calculará multiplicando el número de horas trabajadas por el precio de la hora. Se sabe que la hora del juez cuesta 25 euros, la del recogepelotas 10 euros y la de la azafata 20 euros. Se pide:
    a) Diseñar el diagrama de clases en notación UML indicando las relaciones entre las mismas. Debe incluir métodos para la consulta de todos los atributos de ejemplar, y un método calcularSalario() que recibirá como parámetro el número de horas calculadas. El método calcularSalario() junto con el precio / hora de todas las personas involucradas deberá definirse en una interfaz.
    b) Implementar en Java las clases y / o interfaces resultantes.
    c) Construir un programa principal donde en una matriz se introduzca un objeto de cada tipo de persona y se muestre a continuación el nombre de cada persona y su salario, considerando que todos han trabajado 10 horas.

12. Una federación de baloncesto está integrada por un conjunto de clubs. Cada club está formado por 10 jugadores, un entrenador y un presidente. Cada club tiene un presupuesto con el que debe afrontar el pago de los sueldos de las personas que lo componen.
    Todas las personas cobran un sueldo base determinado por el club (entrenador: 5000 euros / mes, presidente: 10000 euros / mes, jugador: 2000 euros / mes). Los jugadores se imponen un tanto por ciento del sueldo base que quieren que Hacienda les retenga, por lo que su sueldo base cada mes disminuye en esa proporción. Además, el sueldo del presidente está en función de los años que lleva en el cargo, cobrando un 1 % más del sueldo base por año en el cargo. El sueldo de cada entrenador es igual a su sueldo base más una prima determinada por el entrenador al ser contratado. El sueldo base de cada persona se tomará de una interfaz que implementará el Club.
    Para cada jugador es necesario conocer su nombre, DNI, posición en la que juega ("pivot", "base", "alero", etc.) y la retención que ha elegido. La federación debe conocer del entrenador su nombre, DNI, número de carnet de entrenador (tipo entero) y la prima. Por último, del presidente debe conocerse su nombre, DNI y años de experiencia en el cargo. Para cada persona (independientemente de su cargo) se podrá calcular el sueldo que percibe, para ello se pasará como parámetro su sueldo base. Además el sueldo que realmente se percibe será un atributo de cada tipo de persona. El club debe poder determinar si lo que debe pagar en salarios excede su presupuesto. Se pide:
    a) Diseñar el diagrama de clases en notación UML indicando las relaciones entre las mismas.
    b) Implementar en Java las clases y / o interfaces resultantes.
    c) Construir un programa principal donde, una vez creado un objeto Club, se determine e imprima el superávit o déficit del club una vez pagados los salarios. Además, se debe imprimir un mensaje si el club queda en números rojos (déficit). Por simplicidad considerar sólo 2 jugadores.

13. Se pretende gestionar el pago de las matrículas de los alumnos de una universidad. Para ello necesitamos conocer el DNI del alumno, el número de asignaturas de primera convocatoria de las que se matricula y el número de asignaturas en segunda o superior convocatoria. Por otro lado, en una clase Matriculación se calcula el dinero que debe pagar cada alumno por su matricula, sabiendo que todas las asignaturas tienen 7 créditos. En las asignaturas de primera convocatoria el crédito vale 10 euros y en las de segunda o superior 20 euros. En la clase Matriculación se guarda el número de matrículas que se van realizando y la cantidad total de dinero acumulado. Se debe usar una interfaz para definir el número de créditos de las asignaturas y el precio del crédito.
    a) Diseñar el diagrama de clases en notación UML indicando las relaciones entre las mismas.
    b) Implementar en Java las clases y / o interfaces resultantes.
    c) Construir un programa principal donde se calcule lo que tienen que pagar tres alumnos. El primero tendrá cuatro asignaturas de primera convocatoria y dos de segunda; el segundo cinco asignaturas de primera y una de tercera; y el tercer alumno seis de primera y ninguna de segunda.

14. Se quiere informatizar un hospital y para ello se requiere registrar la información de médicos y enfermeros. De cada uno de ellos se conoce su nombre, DNI y días de vacaciones (25 días). De los médicos, además, se conoce su especialidad y el número de guardias que han realizado en el último año. En el caso de los enfermeros, hay que saber su antigüedad. Aparte de estos erquisitos, hay que tener en cuenta que:
    - Un médico debe poder conocer el nombre, DNI y teléfono de sus pacientes, pudiendo consultar todos los datos y cambiar el teléfono.
    - Tanto de los médicos como del os enfermeros se debe poder calcular los días que les corresponden de vacaciones. En el caso de un médico, además de los días base, le corresponde un día más por cada guardia realizada el año anterior. En el caso de un enfermero, le corresponde un día más por cada 2 años de antigüedad.
    - Es necesario construir un programa principal donde un médico tenga, en este caso concreto, 2 pacientes y se muestren los nombres de los 2 pacientes. El médico habrá realizado 10 guardias el año anterior y se consultará cuantos días de vacaciones le corresponden este año. De un enfermero, que tiene 6 años de antigüedad, se consultará el número de días de vacaciones que le corresponden.

    Se pide:
    a) Dibujar el diagrama UML de cada clase incluyendo las relaciones entre ellas.
    b) Implementar cada una de las clases en Java
    c) Construir un programa principal siguiendo la especificación del tercer punto del enunciado. Debe utilizarse polimorfismo.

15. Una empresa de transporte de mercancías tiene una flota de camiones. Cada camión se caracteriza por un número de identificación, un conductor y una carga mínima que puede transportar. A su vez, el conductor se caracteriza por un nombre y un número de horas que trabaja a la semana. Puede haber dos tipos de camiones: con remolque o sin remolque. La diferencia entre ellos es la forma en la que se calcula la carga máxima que pueden transportar. En todos los cases debe incrementarse la carga mínima en una cantidad que se calcula de diferente forma en cada tipo de camión. En el caso de los camiones con remolque, la carga máxima está en función del tamaño del remolque. Si el tamaño del remolque es menor de 500 metros cúbicos, la carga máxima es la carga mínima más un 20 % de dicha carga. La carga a transportar de cualquier otro camión con remolque es la carga mínima + 40 % de dicha carga. En el caso de los camiones sin remolque, la carga máxima se calcula en función de su peso, multiplicando éste por la carga mínima. Se considera que el tamaño del remolque y el peso son variables que caracterizan al tipo de camión correspondiente. Se pide:
    a) Diseñar el diagrama de clases en notación UML indicando las relaciones entre las mismas.
    b) Implementar en Java las clases y / o interfaces resultantes.
    c) Construir un programa principal que calcule e imprima:
    - La carga máxima que puede transportar un camión con remolque si este tiene 450 metros cúbicos
    - La carga máxima que puede transportar un camión sin remolque si este pesa 400 kg.
    - También se debe imprimir la identificación y el nombre del conductor de cada camión. Utilice una matriz para almacenar los dos objetos camión, haciendo uso del polimorfismo.