1) Un psicólogo necesita un sistema para organizar su agenda semanal. El sistema mantiene para c/u de los 5 días de atención y c/u de los 6 turnos (horarios) del día, la información del paciente que tomó el turno. De los pacientes guarda: nombre, si tiene obra social y costo a abonar por la sesión.

a)Genere las clases necesarias. Implemente constructores para iniciar: el sistema sin pacientes; los pacientes a partir de toda su información.

b)Lea atentamente y luego implemente métodos que permitan:

- Agendar al paciente P en un día D y turno T. Asuma que el turno está libre. D y T son válidos.

- Dado el nombre de un paciente, liberar todos sus turnos.

- Dado un día D y el nombre de un paciente, devolver si el paciente tiene agendado un turno ese día. Asuma que D es válido.

c) Realice un programa que instancie el sistema. Cargue varios pacientes al sistema.

Libere turnos agendados. Para finalizar, imprima el resultado del inciso b.iii

2)

1) Queremos representar estanterías de libros. Una estantería mantiene sus libros organizados en N estantes cada uno con lugar para M libros. Un libro posee título, nombre de su primer autor y peso.

a) Implemente las clases de su modelo, con sus atributos y getters/setters adecuados.

Provea constructores para iniciar: los libros a partir de toda su información; la estantería para N estantes y lugar para M libros por estante (inicialmente no debe tener libros cargados).

b) Implemente los siguientes métodos:

- almacenarLibro: recibe un libro, un nro. de estante y nro. de lugar válidos y guarda al libro en la estantería. Asuma que dicho lugar está disponible.

- sacarLibro: recibe el título de un libro, y saca y devuelve el libro con ese título, quedando su lugar disponible. Tenga en cuenta que el libro puede no existir.

- calcular: calcula y devuelve el libro más pesado de la estantería.

2) Realice un programa que instancie una estantería para 5 estantes y 3 libros por estante. Almacene 7 libros en la estantería. A partir de la estantería: saque un libro e informe su representación String; luego, informe el título del libro más pesado.

3)

1) Queremos representar libros electrónicos e impresos. De cualquier libro se conoce: título, precio base y el nombre de los autores (a lo sumo 8). Adicionalmente: los libros electrónicos tienen el formato (ej: “.pdf”, “.epub”) y su tamaño en MB; mientras que los libros impresos registran si es tapa dura o no.

a) Provea constructores para iniciar los libros a partir de toda su información y sin autores.

b) Agregue a las clases implementadas los métodos necesarios para incorporar la siguiente funcionalidad:

i- Agregar al libro un autor cuyo nombre se recibe. Asuma que hay espacio.

ii- Obtener el precio final del libro teniendo en cuenta que:

- Para los libros electrónicos es el precio base al que se adiciona un impuesto de descarga de 2,5$ por MB.

- Para los libros impresos es el precio base al que se adiciona 500$ si es de tapa dura.

iii-Obtener la representación String del libro, la cual se compone de título, precio final y nombre de sus autores.

2) Realice un programa que instancie un libro electrónico y un libro impreso. Cargue 3 autores a cada uno. Para finalizar, muestre la representación String de los libros.

4)

1- Una galería de arte quiere reunir información de sus artistas en un catálogo. El catálogo almacena a lo sumo 15 artistas. De los artistas se registra nombre artístico y el nombre de su obra exhibida.

a)Genere las clases necesarias. Provea constructores para iniciar: el catálogo (sin artistas); los artistas a partir de toda su información.

b)Implemente los métodos necesarios, en las clases que correspondan, para permitir:

- Agregar un artista al catálogo. Asuma que hay espacio.

- Obtener la representación string del catálogo (“Nombre artista 1, Nombre de su obra”) ..

(“Nombre artista N, Nombre de su obra”).

2- Implemente un programa que instancie un catálogo y agreguele dos artistas. A partir del catálogo: muestre su representación.

5)

Representar dos tipos de sorteos solidarios: común y avanzado. Ambos sorteos

mantienen el valor en pesos del premio y guardan para cada bono (0..99) la información de la persona a la que fue asignado (nombre, dni y colaboración en pesos). Sin embargo, difieren en cierta funcionalidad (se detalla más adelante).

Lea 1 y 2 atentamente y luego implemente.

1- Genere las clases necesarias. Provea constructores para iniciar los sorteos a partir del premio en pesos (los bonos inicialmente no están asignados a nadie).

2- Implemente los métodos necesarios, en las clases que corresponda, para:

a. Agregar una persona al sorteo, devolviendo el nro. de bono que le tocó. Los nros. De bono se reparten en forma consecutiva comenzando de 0. Asuma que hay un nro. De bono disponible.

b. Obtener la persona que compró el nro. de bono B. Asuma que B es un nro. de bono nválido.

c. Ejecutar el sorteo, devolviendo la persona ganadora, teniendo en cuenta que: en los sorteos comunes se elige el nro. de bono ganador aleatoriamente entre los nros. repartidos; en los sorteos avanzados la persona ganadora será la que aportó mayor colaboración en pesos.

3- Realice un programa que instancie un sorteo común y un sorteo avanzado con premio 50.000$. Agregue personas a cada sorteo. Ejecute ambos sorteos e imprima la información obtenida de la ejecución.

6)

1- Queremos representar compras minoristas y mayoristas. De cualquier compra se conoce: nro. y los productos comprados (como máximo N). De cada producto registra: código, precio mayorista y precio minorista. Además: las compras mayoristas tienen el CUIT del comprador; y las minoristas si el comprador es jubilado.

a- Genere las clases necesarias. Provea constructores para iniciar las compras a partir de toda su información, para una cantidad máxima de N productos comprados y sin productos cargados.

b- Implemente métodos en las clases que corresponda para permitir:

i- Agregar a la compra un producto que se recibe. Asuma que hay espacio.

ii- Obtener el precio a pagar por el i-ésimo producto de la compra sabiendo que: para las compras minoristas el precio a pagar por dicho producto es el precio minorista descontando 10% si es jubilado y para las compras mayoristas es el precio mayorista descontando 21% de IVA. Asuma que “i” es válido y se recibe.

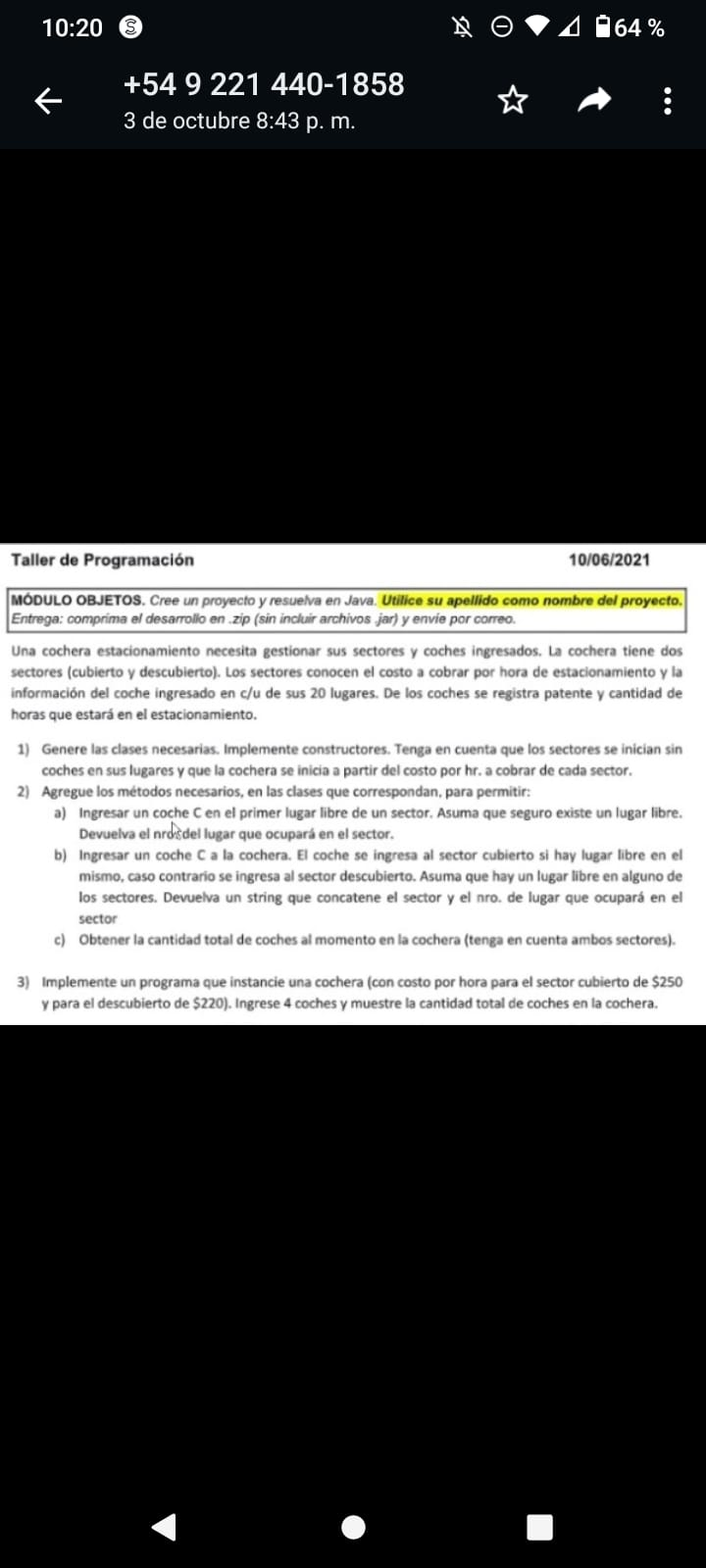
iii- Obtener la representación String de la compra siguiendo el ejemplo:

“Nro:… , Productos comprados (código y precio pagado por el producto):…Total pagado: …”

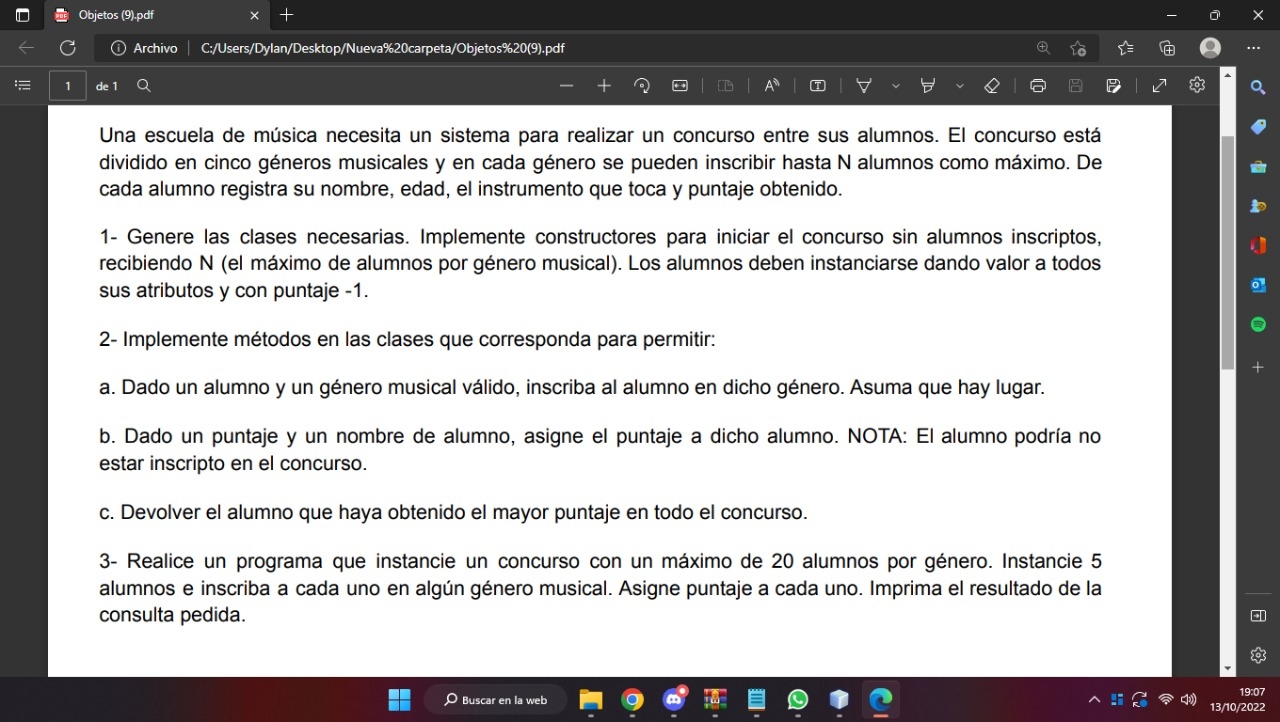
2) Realice un programa que instancie una compra mayorista y una minorista. Cargue 3

productos a cada compra. Para finalizar, muestre la representación String de cada compra.

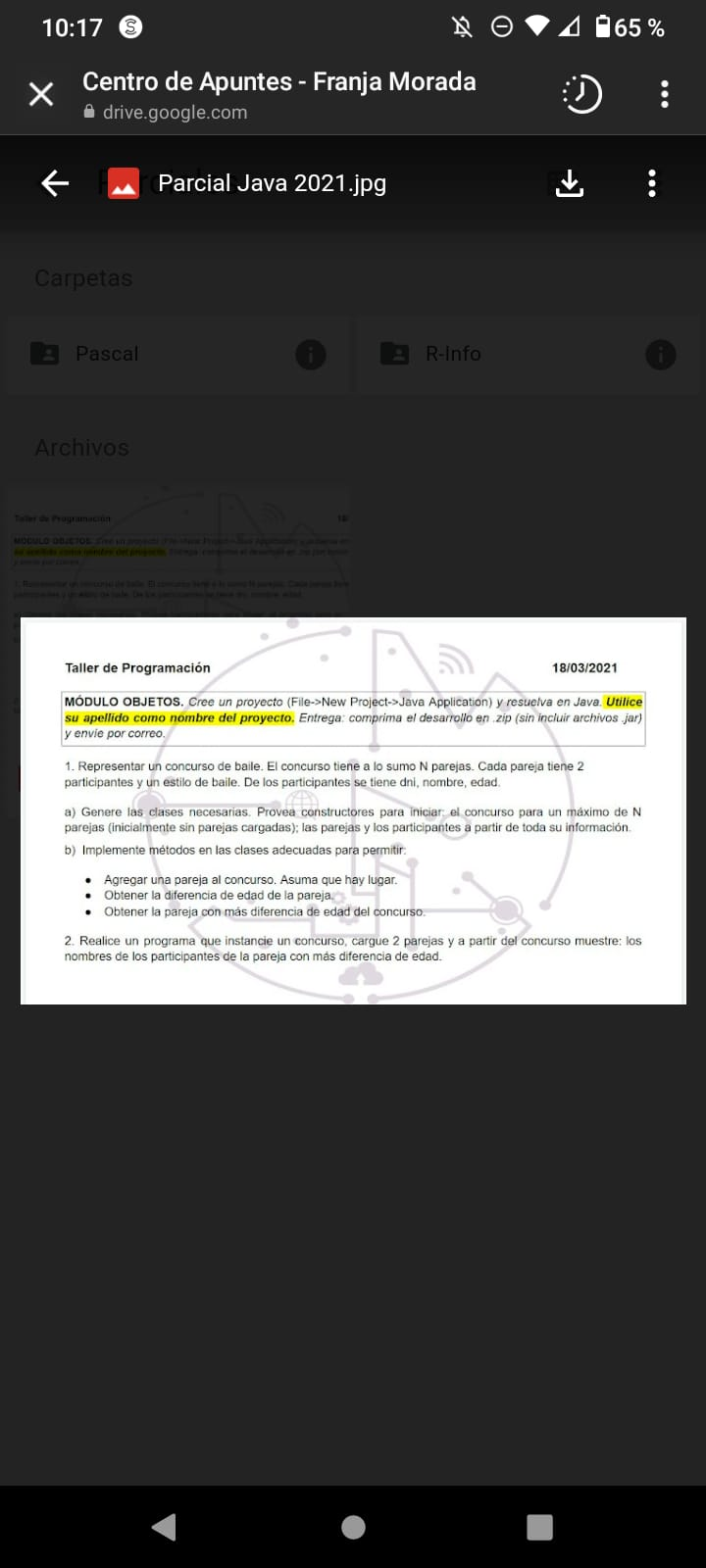
7)



8)



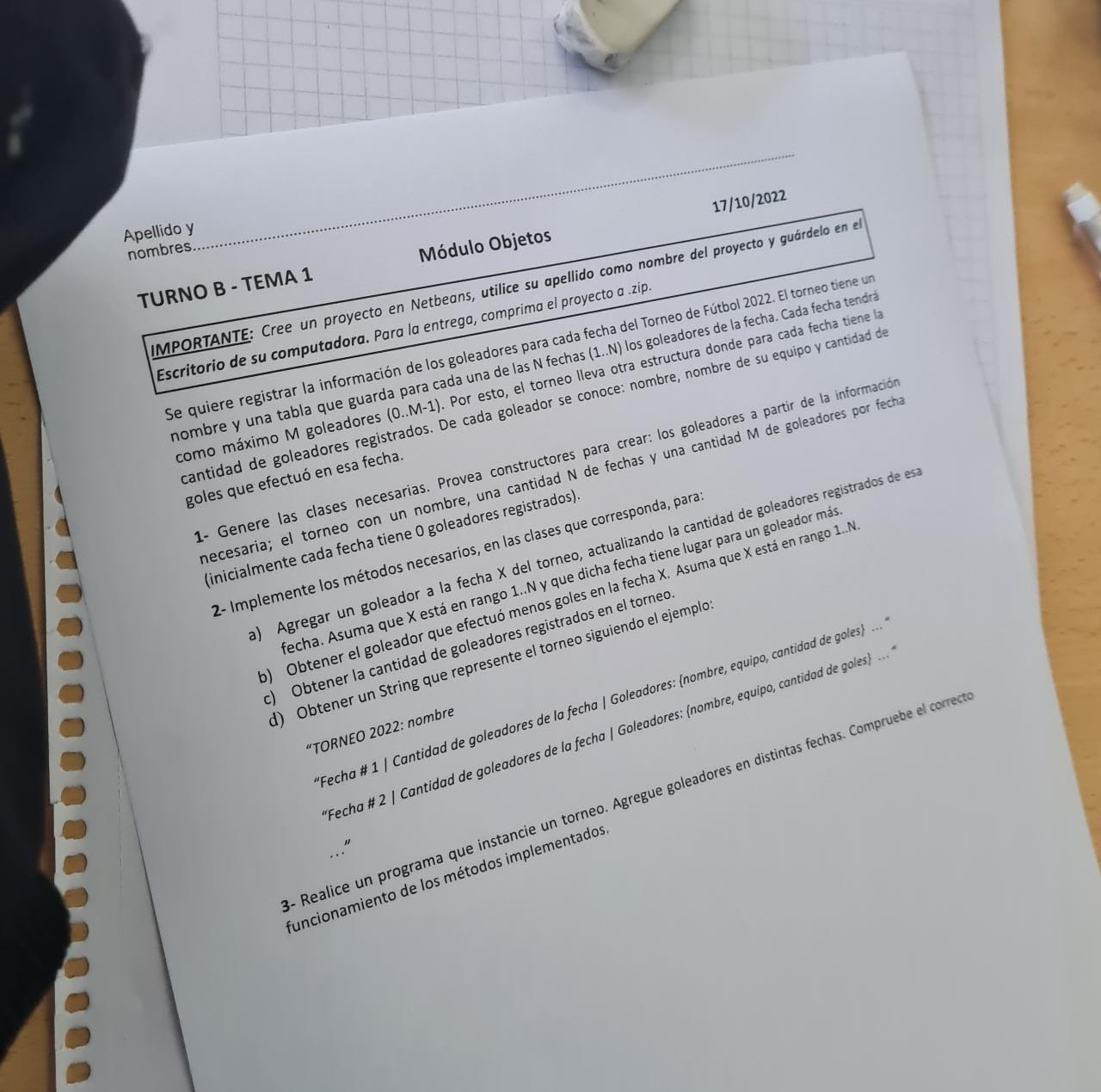
9)



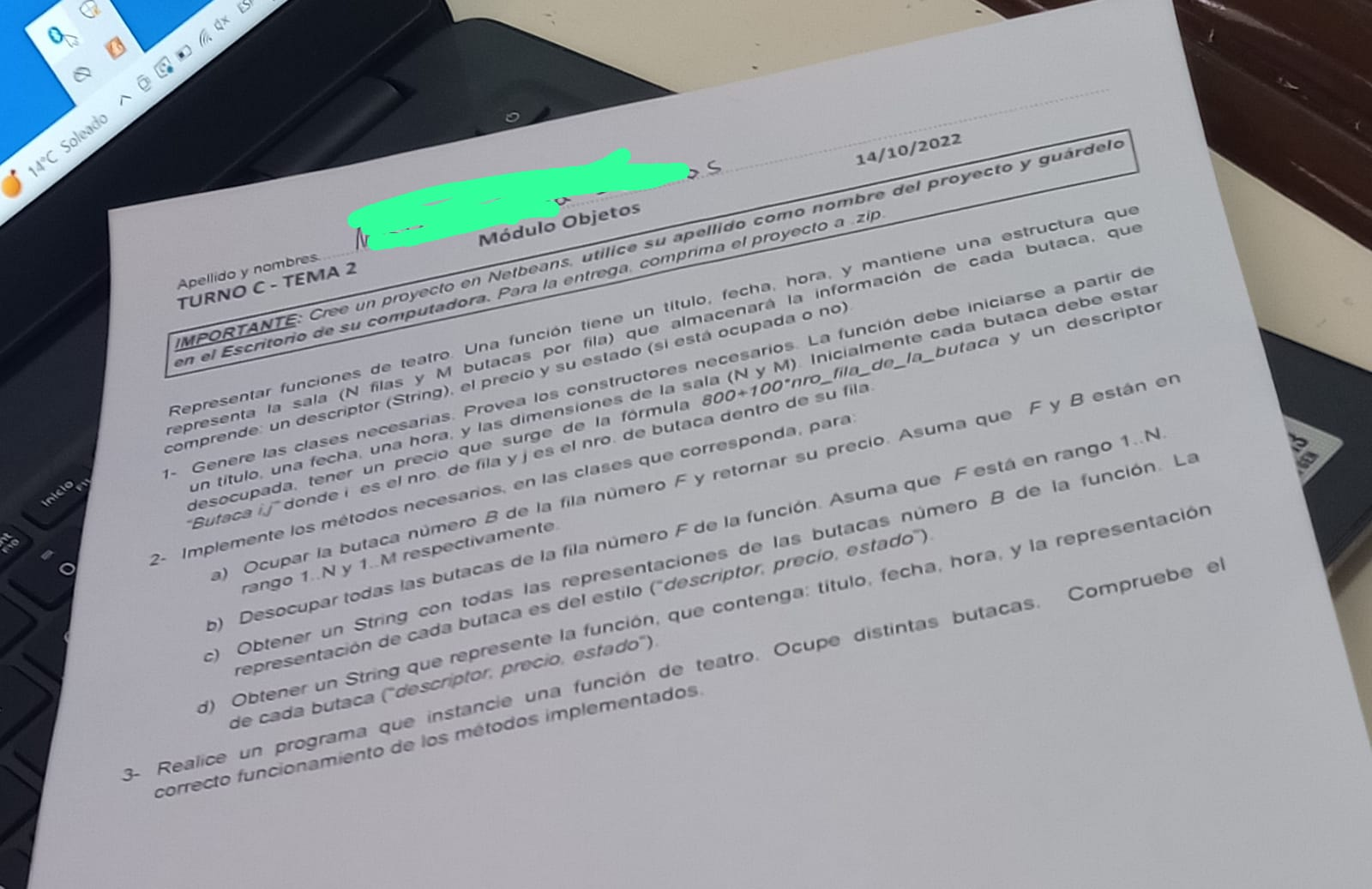
10)



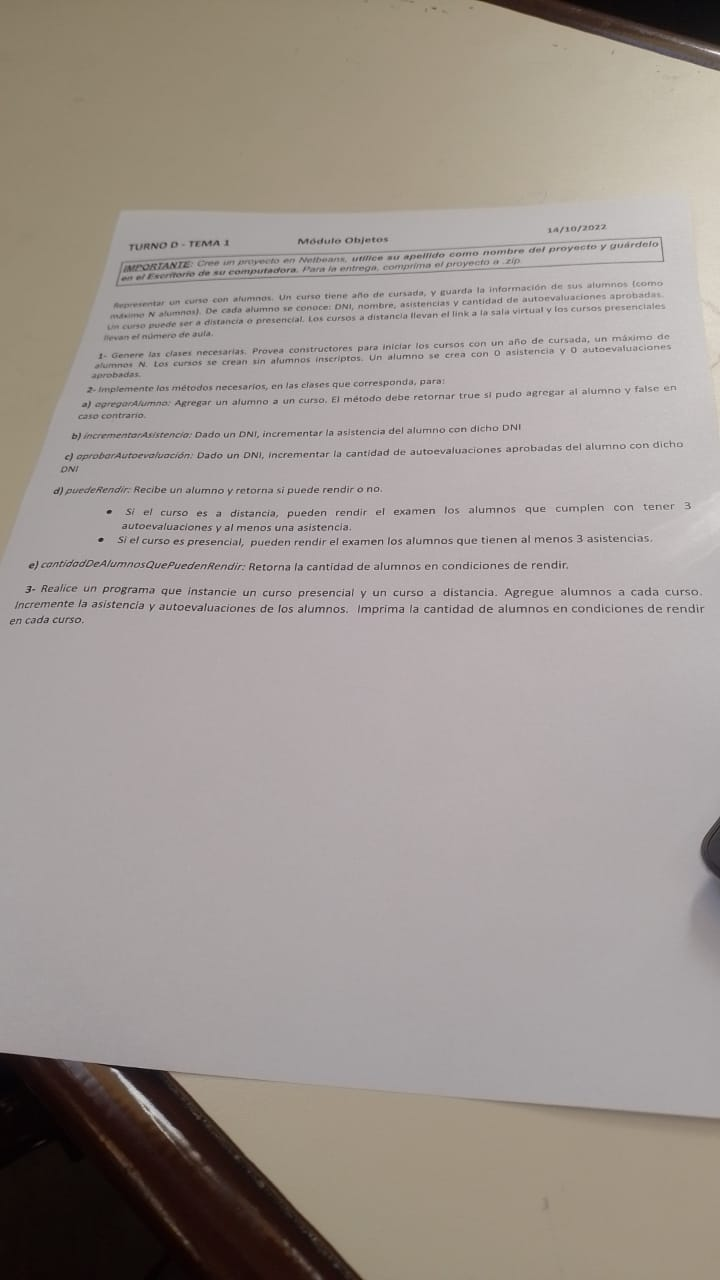
11)



12)



13)



T14

Una editorial universitaria, necesita registrar la información correspondiente a los ejemplares que publicó y se publicarán. Cada **Ejemplar** posee un código identificatorio, cantidad de páginas, un resumen, año de publicación (si aún no fue publicado este valor es cero) y la información del responsable a cargo de la edición. Para el **responsable** se registra su DNI, nombre y apellido.

Existen diferentes tipos de ejemplares:

* **Libro**, que también tiene asociado un título, la cantidad de capítulos del mismo y si es o no una edición de bolsillo.
* **Revista**, que tiene asociado su nombre, un número de volumen y la cantidad de artículos en su interior.

Los ejemplares deben poder imprimirse en un listado que se genera periódicamente, por ello se pide que las mismas definan un mensaje que les permita **retornar un String** con sus datos:

* Para los **libros**, su código identificatorio, su título y el nombre del responsable.
* Para las **revistas**, su nombre y el número de volumen.

Por otro lado, cuando los ejemplares son publicados se agrega el año actual como año de publicación y, en particular para cada tipo de ejemplar:

* Para los **libros**, se concatena al final de su título si el mismo es o no una edición de bolsillo, por ejemplo, si el libro es “Programación en JAVA”, su título se modificará a “Programación en JAVA – De Bolsillo”
* Para las **revistas**, se agrega además el número de volumen (que se solicita mediante el método **getNroVolumen()** de la clase **Generador** provista para el examen).

Se pide:

1. Implementar la clase **Ejemplar** con sus respectivos atributos, constructores y métodos para acceder y modificar atributos.
2. Implemente aquellas **clases**, **métodos** y **objetos** que considere necesarios para representar los diferentes tipos de ejemplares.
3. Implementar la clase **ProgramaEditorial**, su método main y los métodos que considere necesarios para la lectura y almacenamiento de a lo sumo 1000 ejemplares. Luego, haga la impresión de un listado de todos los ejemplares existentes.
4. Escriba todos los métodos que considere necesarios para publicar los ejemplares de la editorial (considere los atributos del estado interno que se verán modificados).

T15

Arba (Agencia de Recaudación de la Provincia de Buenos Aires) procesa de la información correspondiente al impuesto vehicular a pagar por los contribuyentes de la provincia de Buenos Aires. Cada **vehículo** posee su año de fabricación, importe de gestión básico a pagar por el impuesto y la información del propietario. Para cada uno de los **propietarios** se registra su CIT (Clave de Identificación Tributaria), nombre y apellido.

Existen diferentes **tipos** de vehículos:

* **Automotor**, que también tiene asociado una patente, un importe adicional según el año de fabricación y una descripción.
* **Embarcación**, que tiene asociado un código del Registro Especial de Yates (REY), nombre, un tipo de embarcación (lancha, crucero, velero, etc.), eslora (longitud de proa a popa), un tonelaje (peso) y un valor declarado de la embarcación.

Mensualmente se realiza un listado con la información relevante de cada tipo de vehículo, por ello se pide que cada uno defina la visualización de sus datos de la siguiente manera:

* Para los **automotores**, su patente, y año de fabricación.
* Para las **embarcaciones**, su código REY, tipo de embarcación y tonelaje.

La Agencia necesita calcular los costos de los impuestos de la siguiente manera:

* De los **automotores**, el propietario paga mensualmente el importe básico más el adicional calculado según el año de fabricación.
* De las **embarcaciones**, el propietario paga un monto semestral que varía según el valor declarado de la embarcación: valuadas hasta $6.000, pagan el 4% del su valor declarado; entre $6.000 y $180.000, pagan el 2% del valor declarado y, finalmente superiores a $180.000 pagan el 5% de dicho valor declarado.

Se pide:

1. Implementar la clase **Vehículo** con sus respectivos atributos, constructores y métodos para acceder y modificar atributos.

1. Implementar aquellas clases, constructores, métodos y objetos que considere necesarios para representar los diferentes **tipos** de vehículos.

1. Implementar la clase **ProgramaArba**, su método main y aquello que considere necesario para la lectura y almacenamiento de a lo sumo 1000 vehículos.

1. Implementar dentro del programa y de las clases del modelo, los métodos que considere necesarios para la impresión del listado de todos los vehículos existentes y el costo del impuesto a pagar por el mes de diciembre de 2019.

T16

Una escuela de música necesita un sistema para realizar un concurso entre sus alumnos. El

concurso está dividido en cinco géneros musicales y en cada género se pueden inscribir hasta N

alumnos como máximo. De cada alumno registra su nombre, edad, el instrumento que toca y

puntaje obtenido.

1- Genere las clases necesarias. Implemente constructores para iniciar el concurso sin alumnos

inscriptos, recibiendo N (el máximo de alumnos por género musical). Los alumnos deben

instanciarse dando valor a todos sus atributos.

2- Implemente métodos en las clases que corresponda para permitir:

a. Dado un alumno y un género musical válido, inscriba al alumno en dicho género. Asuma que

hay lugar.

b. Dado un género musical válido G, un nombre de alumno N y un puntaje P, asignarle el puntaje

P al alumno N del género G. Considere que el alumno puede no existir.

c. Devolver al alumno que haya obtenido el mayor puntaje en todo el concurso.

d. Devolver un String que contenga la representación del concurso, siguiendo el formato:

“Genero 1 – Total inscriptos: X – representación del alumno 1 - ... - representación del alumno X

...

Genero 5 – Total inscriptos: Y – representación del alumno 1 - ... representación del alumno Y

Alumno de mayor puntaje es: representación del alumno de mayor puntaje”

3- Realice un programa que instancie un concurso con un máximo de 20 alumnos por género.

Instancie 5 alumnos e inscriba a cada uno en algún género musical. Asigne puntajes. Imprima la representación del concurso.

T17

I) Representar dos tipos de urnas electrónicas: electorales (usada en elecciones donde se

presentan varias listas) y de referéndum (usada para saber si la población está a favor/en

contra de la aprobación de una ley).

● Cualquier urna se caracteriza por su número de urna, un contador de votos en blanco y la

información de la zona a la que pertenece (localidad, partido, provincia).

● Una urna electoral es una urna que se caracteriza además por la cantidad de listas

existentes y por mantener un contador de votos para cada lista (son “L” listas, y están

numeradas desde 0).

● Una urna de referéndum es una urna que se caracteriza además por mantener un contador

de votos a favor y un contador de votos en contra.

A) Genere las clases necesarias.

B) Implemente constructores. Las urnas electorales deben iniciarse a partir de un número de

urna “U”, una cantidad de listas a “L”, una zona “Z”. Las urnas de referéndum deben iniciarse a

partir de un número de urna “U” y una zona “Z”. Ambas urnas deben inicialmente poseer los

contadores de votos en 0.

C) Implemente métodos para obtener/modificar (get/set) aquellos atributos que considere

adecuado.

D) Incorpore los métodos listados a continuación:

i. Cualquier urna debe saber responder al mensaje votarEnBlanco que incrementa en 1 el

contador de votos en blanco.

ii. Las urnas electorales deben responder a los mensajes:

● validarNumeroDeLista: recibe un nro. “N” y devuelve un boolean indicando si N es un nro.

de lista válido.

● votarPorLista: recibe un nro. de lista válido “I” y debe incrementar en 1 el contador de

votos de esa lista.

● devolverVotosPorLista: recibe un nro de lista válido “I” y devuelve la cantidad de votos

para esa lista.

iii. Las urnas de referéndum deben responder a los mensajes:

● votarAFavor: incrementa en 1 el contador de votos a favor.

● votarEnContra: incrementa en 1 el contador de votos en contra.

iv. Ambos tipos de urnas deben saber responder al mensaje calcularGanador pero de manera

diferente:

● Para las urnas electorales debe devolver el nro. de la lista ganadora. En caso de empate

entre 2 o más listas, se elegirá una como ganadora – indistintamente.

● Para las urnas de referéndum debe devolver un nro. 0 si hubo empate, -1 si ganó la opción

En Contra, 1 si ganó la opción A Favor.

v. Ambos tipos de urnas deben saber responder al mensaje calcularTotalVotos pero de manera

diferente:

● Para las urnas electorales debe devolver la suma de votos en blanco y votos a favor de cada

lista.

● Para las urnas de referéndum debe devolver la suma de votos en blanco, a favor y en contra.

vi. Cualquier urna debe saber responder al mensaje toString que retorna un String del estilo:

“Urna: nro. Zona: loc-partido-prov Total Votos: cant. total Ganador: nro. de la opción

ganadora”

II) Realice un programa que instancie una urna electoral para la mesa “203” con 5 listas y una

urna de referéndum para la mesa “204”. Luego, lea DNIs de personas que llegan a votar a la

mesa, hasta que se ingresa el DNI 0 (el cual no debe procesarse). Cada persona vota en ambas

urnas de la siguiente manera:

 Para la urna electoral, la persona ingresa un nro. “N”: si “N” corresponde a una lista

válida se debe votar por la lista “N” y en caso contrario se debe votar en blanco.

 Para la urna de referéndum, la persona ingresa un número “M”: si “M” es positivo se

debe votar a favor, si es negativo se debe votar en contra, y si es 0 debe votar en blanco.

Al finalizar la votación, muestre en consola la representación String de cada urna.