

IMPORTANTE: Cree un proyecto y resuelva en Java. Utilice su apellido como nombre del proyecto. Entrega: comprima el desarrollo en .Zip (no debe incluir ningún archivo .jar) y envíe por correo .

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) 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 el primer día que tenga libre el turno T. Retornar el día

asignado. Asuma que hay un día con dicho turno libre y que T es válido.

- Dado un día D y el nombre de un paciente, liberar el turno ocupado por el paciente en ese día.

- Calcular y devolver la cantidad de turnos agendados del paciente nombre N.

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.

3) 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 un día D, liberar los turnos de todos los pacientes agendados en ese día (los 6 turnos). Retornar un string con los nombres de los pacientes para avisar de la cancelación. Asuma que D es válido.

-Calcular y devolver el costo total que recaudará el psicólogo en la atención semanal.

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

4) 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 el primer turno libre del día D. Retornar el turno (horario) asignado. Asuma que hay un turno libre ese día y que D es válido.

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

-Calcular y devolver el turno (es decir, horario) con más pacientes agendados.

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.

5) 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.

6) 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) Implemente las clases del modelo con sus atributos y getters/setters adecuados. 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.

ji- 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.

ii-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.

7) 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 y lo almacena en el primer estante que tenga lugar. Asuma que hay espacio para almacenar el libro.

- sacarLibro: saca y devuelve el libro que se encuentra en el estante X, lugar Y (X e Y se reciben y son válidos). Dicho lugar debe quedar disponible.

- calcular: calcula y devuelve la cantidad de libros de un autor cuyo nombre se recibe.

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 la cantidad de libros de "Borges".

8) 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) Implemente las clases del modelo con sus atributos y getters/setters adecuados. 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. El precio final es el precio base al que se adiciona el IVA (21% del precio base).

iii- Obtener la representación String del libro siguiendo el formato de ejemplo:

Libro electrónico: "Título, Nombre de los autores, precio final, formato, tamaño en MB"

Libro impreso: "Título, Nombre de los autores, precio final, tapa dura"

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.

9) 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 y un nro. de estante válido, y lo almacena en el primer

lugar libre de dicho estante. Asuma que hay espacio para almacenar el libro.

- sacarLibro: saca y devuelve el libro que se encuentra en el estante X, lugar Y (X e Y se reciben y son válidos). Dicho lugar debe quedar disponible.

- Calcular: calcula y devuelve el número del estante más pesado (teniendo en cuenta el peso de sus libros).