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)

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.

---

6)

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) 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 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.

---

7)
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 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)

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) 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: “Titulo, Nombre de los autores, precio final, tapa dura”

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.

---

9)

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 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).

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 número de estante más pesado.

---

10)

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) 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. Para ambos libros el precio final surge de adicionar al precio base el IVA (21% del precio base). Además, para los libros electrónicos se adiciona un impuesto de descarga de 2,5$ por MB.
iii-Obtener la representación String del libro, la cual se compone de título, precio final y el nombre de su primer autor.

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.

---

11)
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 y turno libre. Retornar un string con el día y turno asignados. Asuma que hay un turno libre.

-Dado un día D y turno T, liberar ese turno. Asuma que D y T son válidos.

-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.

---

12)
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 un turno T, liberar los turnos de todos los pacientes agendados en ese horario (los 5 días). Retornar un string con los nombres de los pacientes para avisar de la cancelación. Asuma que T es válido.

- Calcular y devolver el día 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.

---

13)
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.

---

14)

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.

---

15)
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.

---

16)

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.
   - Devolver al artista cuyo nombre de obra coincide con uno recibido.

2- Implemente un programa que instancie un catálogo y agreguele dos artistas. A partir del catálogo: obtenga el artista de la obra “el grito” e imprima su nombre artístico.

---

17)

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 el nombre artístico y la categoría.

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.
   - Devolver la cantidad de artistas cuya categoría coincide con una recibida.

2- Implemente un programa que instancie un catálogo y agreguele dos artistas. A partir del catálogo: obtenga y muestre la cantidad de artistas de la categoría “pintura”.

---

18)

1- Una galería de arte quiere reunir información de sus mejores artistas en un catálogo. El catálogo almacena para cada categoría (numeradas de 1..15) al mejor artista. De los artistas se registra nombre artístico y nacionalidad.

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:

- Dado un artista y una categoría, incorporar al artista en dicha categoría del catálogo.
- Obtener la representación string del catálogo (“Categoría 1, Nombre artista, Nacionalidad”) (“Categoría 2, sin artista”) .. (“Categoría 15, Nombre artista, Nacionalidad”).

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

---

19)
1- Una galería de arte quiere reunir información de sus mejores artistas en un catálogo. El catálogo almacena para cada categoría (numeradas de 1..15) al mejor artista. De los artistas se registra nombre artístico y nacionalidad.
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:

- Dado un artista y una categoría, incorporar al artista en dicha categoría del catálogo.
- Devolver la cantidad de artistas cuya nacionalidad coincide con una recibida.

2- Implemente un programa que instancie un catálogo y agreguele dos artistas. A partir del catálogo: obtenga y muestre la cantidad de artistas de nacionalidad “argentino”.

---

20)
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 string.

---

21)

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 válido.

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.

---

22)

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. Dada una persona y un nro. de bono, asignarle a la persona ese nro. de bono en el sorteo. Asuma que el nro. de bono es válido y está disponible.

b. Obtener la persona que compró el nro. de bono B. Asuma que B es un nro. de bono válido. En caso de no haberse asignado aún, retornar null.

c. Ejecutar el sorteo, devolviendo la persona ganadora, teniendo en cuenta que: en los sorteos comunes se generan nros. aleatoriamente hasta obtener un nro. de bono asignado; 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.

---

23)

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 válido.

c. Ejecutar el sorteo, devolviendo un String como se indica a continuación:
Para los sorteos comunes: se elige el nro. de bono ganador aleatoriamente entre los nros. repartidos, debiendo retornar “Premio Neto: ... Ganador: *nombre - dni*”.
Para los sorteos avanzados: se eligen dos nros. de bono aleatoriamente entre los nros. repartidos (quienes se dividirán el premio), debiendo retornar “Premio Neto: … Ganador 1: *nombre - dni* Ganador 2: *nombre - dni*”

*** El Premio Neto se calcula descontando al premio en pesos el 21% de IVA*

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.

---

24)

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. El nro. de bono a otorgar se obtiene generando nros. aleatoriamente hasta obtener uno no asignado. 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 válido. En caso de no haberse asignado aún, retornar null.
c. Ejecutar el sorteo, devolviendo la persona ganadora, teniendo en cuenta que: en los sorteos comunes se generan nros. aleatoriamente hasta obtener un nro. de bono asignado; 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.

---

25)

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. El nro. de bono a otorgar se obtiene generando nros. aleatoriamente hasta obtener uno no asignado. 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 válido. En caso de no haberse asignado aún, retornar null.
c. Ejecutar el sorteo, devolviendo la persona ganadora, teniendo en cuenta que: en los sorteos comunes se generan nros. aleatoriamente hasta obtener un nro. de bono asignado; 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.

---

26)

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.

---

27)

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 la compra sabiendo que: para las compras minoristas es la suma del precio minorista de los productos y se adiciona el 21% de IVA y se descuenta el 10% si es jubilado; para las compras mayoristas es la suma del precio mayorista de los productos y descuenta el 21% de IVA.
iii- Obtener la representación String de la compra siguiendo el ejemplo:
"Nro:… , Productos (código y precios mayorista y minorista):…, Precio a pagar:… "

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.

---

28)

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 la compra sabiendo que: para las compras minoristas es la suma del precio minorista de los productos y se descuenta el 10% al total si es jubilado; para las compras mayoristas es la suma del precio mayorista de los productos y descuenta el IVA (21% del total).
iii- Obtener la representación String de la compra siguiendo el ejemplo:
Compra minorista: "Nro:… , Productos (código y precios mayorista y minorista):…, Precio a pagar:…, Es jubilado:…"
Compra mayorista: "Nro:…, Productos (código y precios mayorista y minorista):…, Precio a pagar:…, CUIT:…"

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.

---

29)

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 y descripción. 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 la compra teniendo en cuenta que es la suma del precio de los productos comprados al que se adiciona el IVA (21% del total).
iii- Obtener la representación String de la compra siguiendo el ejemplo:
Compra minorista: “Nro:… , Productos (código/precio/descripción):…, Precio a pagar:… , Es jubilado: …”
Compra mayorista: “Nro:…, Productos (código/precio/descripción):…, Precio a pagar:…, CUIT:…”

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.

---

30)

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 y descripción. 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 la compra sabiendo que es la suma del precio de los productos comprados. Además, para las compras mayoristas se descuenta el 21% del IVA al total.
iii- Obtener la representación String de la compra siguiendo el ejemplo:
Compra minorista: “Nro:… , Productos (código/precio/descripción):…, Precio a pagar:… , Es jubilado: …”
Compra mayorista: “Nro:…, Productos (código/precio/descripción):…, Precio a pagar:…, CUIT:…”

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.

---

31)
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 y descripción. 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 la compra sabiendo que es la suma del precio de los productos comprados. Además, para las compras minoristas se descuenta un 10% al total si es jubilado, mientras que para las mayoristas se descuenta el 21% del IVA al total.
iii- Obtener la representación String de la compra siguiendo el ejemplo:
“Nro:… , Productos (código/precio/descripción):…, Precio a pagar:…”

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.

---

32)
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.

---

33)

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) 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 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.

---

34)

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 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”.

---

35)

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) 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: “Titulo, Nombre de los autores, precio final, tapa dura”

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.

---

36)

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 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).

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 número de estante más pesado.

---

37)

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) 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. Para ambos libros el precio final surge de adicionar al precio base el IVA (21% del precio base). Además, para los libros electrónicos se adiciona un impuesto de descarga de 2,5$ por MB.
iii- Obtener la representación String del libro, la cual se compone de título, precio final y el nombre de su primer autor.

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.

.