

Centro de estudios

Carrera: Analista Programador Web

Materia: Lenguajes I

Profesor: Gauto, David Alejandro

HabitacionHotel

Trabajo Final

1. Una agencia de viajes mantiene información referida a los hoteles y cabañas que ofrece a sus clientes. La clase Alojamiento se introduce para factorizar atributos y comportamiento. En la aplicación todos los alojamientos son instancias de una de las clases derivadas: Habitación, Hotel o Cabaña (ambas clases heredan de alojamiento). La clase Agencia se implementa como una tabla. La clase cliente tiene acceso a los alojamientos por posición.

Algunas posiciones pueden ser nulas.

- a) Implemente el diagrama de clases.
- b) Implemente una clase tester que verifique los servicios provistos por la clase Agencia.

Alojamiento*	Agencia
< <atributos de="" instancia="">></atributos>	< <atributos de="" instancia="">></atributos>
codigo: entero	alojamientosAgencia: Alojamiento[]
domicilio: String	
TV: boolean	
< <constructores>></constructores>	< <constructor>></constructor>
Alojamiento(c: entero, d: String, t: boolean)	Agencia(n: entero)
< <comandos>></comandos>	< <comandos>></comandos>
establecerDomicilio(d: String)	insertarAlojamiento(a: Alojamiento, p: entero)
establecerTV (t: boolean)	eliminarAlojamiento(pos: entero) eliminarAlojamiento(a:
< <consultas>></consultas>	Alojamiento)
obtenerCodigo(): entero	< <consultas>> cantAlojamientos():</consultas>
obtenerDomicilio(): String	entero
tieneTV(): boolean toString	recuperarAlojamiento(pos:entero): Alojamiento
(): String	recuperarPosicion(a: Alojamiento): entero
igualCodigo(a: Alojamiento): boolean estrellas():	estaAlojamiento(c: entero): Alojamiento estaLlena():
entero	boolean
	hayAlojamientos(): boolean masEstrellas(cantEst: entero):
	Agencia

Cabaña

<<atributos de instancia>> precioXPersona: real piscina: boolean <<constructores>> HabitacionHotel(c:entero,d:String,t,p:boolean, <<comandos>> establecerPrecioXPersona(r: real) establecerPiscina(p: boolean) <<consultas>> obtenerPrecioXPersona(): real tienePiscina(): boolean estrellas(): entero

<<atributos de instancia>> precioXDia: real cantHabitaciones: entero cantBanios: entero

<<constructores>>

Cabaña(c:entero,d:String,t:boolean,p:real, ch,cb: entero)

<<comandos>>

establecerPrecioXDia(r: real)

<<consultas>>

obtenerPrecioXDia(): real

- estrellas(): entero en Alojamiento retorna 2 si el alojamiento tiene TV, 1 en caso contrario.
- estrellas(): entero en HabitacionHotel retorna la cantidad de estrellas de un Alojamiento más 1 si tiene piscina.
- recuperarPosicion(a: Alojamiento): entero busca por identidad.
- esta Alojamiento (c: entero): Alojamiento busca por código.

En un balneario se pueden alquilar sombrillas con capacidad para 4 personas o sombrillas

especiales, más amplias, para 6 personas y

servicio de camarero. En las especiales,

también se pueden agregar reposeras extras abonando un costo adicional por unidad. El

estacionamiento es opcional en ambos casos.

Cada nuevo alquiler se inserta al final de la

- masEstrellas(cantEst: entero): Agencia en Agencia retorna una nueva agencia sólo con los alojamientos que tengan más de cantEst estrellas.
- hayAlojamientos(): boolean en Agencia retorna true si y sólo si en la tabla hay algún alojamiento.

2.

ColeccionAlquiler

<<atributos de instancia>>

ca: Sombrilla[] cant: entero

<<Constructor>>

ColeccionAlquiler(n: entero)

<<Comandos>> insertar(s: Sombrilla) eliminar(c: entero)

<<Consultas>>

totalAlquiler(n: entero): real

costo(): real

dias * costo base .Si contrató estacionamiento: dias * (costo base + extras)

Sombrilla

<<atributos de clase>>

costoBase = 200

extras = 100

<<atributos de instancia>>

numero: entero dias: entero

estacionamiento: boolean

<<constructor>>

Sombrilla(n, d: entero, e: boolean)

<<comandos>>

<<consultas>> costo(): real obtenerNum(): int obtenerDias(): int

SombrillaEspecial

<atributos de instancia>>

Reposera: entero

<<constructor>> SombrillaEspecial(n,

d, r: entero, e: boolean)

<<comandos>>

<<consultas>> costo(): real

colección incrementando la cantidad. La colección NO está ordenada. Si se almacenan n alquileres, se ocupan las primeras n componentes del arreglo.

insertar(s: Sombrilla) agrega la sombrilla pasada por parámetro a la colección. Requiere haber controlado que existe lugar disponible. eliminar(c: entero) elimina de la colección el objeto correspondiente al número de sombrilla pasado por parámetro. La colección debe quedar compactada nuevamente.

totalAlquiler(n: entero): real devuelve el monto total que se obtendrá por las sombrillas que superen los n días de alquiler.

Implementar las tres clases indicadas en el diagrama.

costo(): real

140% del costo de la sombrilla más (1/3*extras) por reposera por día