

Package `es.upm.dit.fprg24.p2`

Class Inventario

`java.lang.Object`
`es.upm.dit.fprg24.p2.Inventario`

```
public class Inventario
extends Object
```

Constructor Summary

Constructors

Constructor	Description
<code>Inventario(String hospital, int numPlantas, int numAlmacenes, int numArmarios)</code>	Constructor de la clase Inventario.

Method Summary













All Methods

Instance Methods

Concrete Methods



Modifier and Type	Method	Description
void	<code>almacenaDispositivo(DispositivoMedico dispos, int planta, int almacen, int armario)</code>	Almacena un dispositivo médico en un armario específico.
boolean	<code>armarioOcupado(int planta, int almacen, int armario)</code>	Verifica si un armario específico está ocupado (hay un dispositivo dentro).
int[]	<code>buscaDispositivo(String id)</code>	Busca un dispositivo por su identificador y devuelve la ubicación.
double	<code>calcularPrecioPromedio()</code>	Calcula el precio promedio de los dispositivos en el inventario.
<code>DispositivoMedico[]</code>	<code>dispositivosNoOperativos()</code>	Obtiene los dispositivos que no están operativos.

DispositivoMedico	encontrarDispositivoCompa (DispositivoMedico dispos.	Dado un dispositivo, encuentra otro dispositivo en el inventario que sea del mismo tipo, esté operativo, y no esté ocupado.
int[]	encontrarEspacioLibrePara (DispositivoMedico dispos.	Encuentra el primer espacio libre para colocar un dispositivo médico.
boolean	existeArmario (int planta, int almacen, int armario)	Comprueba si existe un armario específico en la estructura de almacenamiento.
boolean	existeDispositivoNoOperat.	Verifica si hay al menos un dispositivo en el inventario que no está operativo,
DispositivoMedico	getDispositivoMasCaro()	Obtiene el dispositivo de mayor valor en el inventario.
DispositivoMedico [][] []	getDispositivos()	Getter del array de dispositivos médicos en el inventario.
String 	getHospital()	Getter del nombre del hospital asociado al inventario.
int	getNumDispositivosEnPlanta (int planta)	Calcula el número de dispositivos almacenados en una planta.
void	liberarArmario (int planta, int almacen, int armario)	Libera un armario al dar de baja un dispositivo del inventario, eliminándolo de su ubicación.
DispositivoMedico	obtenerDispositivo (int planta, int almacen, int armario)	Obtiene el dispositivo médico almacenado en una ubicación específica del inventario.
int[][]	obtenerEspaciosDisponibles	Obtiene todos los espacios disponibles en el inventario, es decir, aquellos que no contienen ningún dispositivo.

String 	obtenerNombreDispositivo (int planta, int almacen, int armario)	Obtiene el nombre del dispositivo médico almacenado en una ubicación específica del inventario.
void	setDispositivos (DispositivoMedico [][] [] dispositivos)	Setter del array tridimensional de dispositivos médicos en el inventario.
void	setHospital (String  hospital)	Setter del nombre del hospital asociado al inventario.
void	transferirDispositivo (int planta, int origenAlmacen, int origenArmario, int destinoAlmacen, int destinoArmario)	Transfiere un dispositivo médico de un armario a otro dentro de la misma planta.
void	transferirDispositivosPor (Tipo tipo, int planta, int origenAlmacen, int destinoAlmacen)	Transfiere todos los dispositivos de un tipo específico desde un almacén de origen a un almacén de destino.
Methods inherited from class java.lang.Object 		
equals  , getClass  , hashCode  , notify  , notifyAll  , toString  , wait  , wait  , wait 		

Constructor Details

Inventario

```
public Inventario(String  hospital,  
                 int numPlantas,  
                 int numAlmacenes,  
                 int numArmarios)  
    throws Exception 
```

Constructor de la clase Inventario.

Parameters:

hospital - Nombre del hospital

numPlantas - Número de plantas en el hospital

numAlmacenes - Número de almacenes por planta

numArmarios - Número de armarios por almacén

responsable - Nombre del responsable del inventario

Throws:

[Exception](#) - Lanza una excepción si el número de plantas, almacenes o armarios no es válido.

Method Details

getHospital

```
public String getHospital()
```

Getter del nombre del hospital asociado al inventario.

Returns:

El nombre del hospital.

setHospital

```
public void setHospital(String hospital)
```

Setter del nombre del hospital asociado al inventario.

Parameters:

hospital - Nombre del hospital.

getDispositivos

```
public DispositivoMedico[][][] getDispositivos()
```

Getter del array de dispositivos médicos en el inventario.

Returns:

Un array tridimensional de dispositivos médicos.

setDispositivos

```
public void setDispositivos(DispositivoMedico[][][] dispositivos)
```

Setter del array tridimensional de dispositivos médicos en el inventario.

Parameters:

dispositivos - Un array tridimensional de dispositivos médicos.

existeArmario

```
public boolean existeArmario(int planta,
                             int almacen,
                             int armario)
```

Comprueba si existe un armario específico en la estructura de almacenamiento.

Parameters:

planta - Número de planta

almacen - Número de almacén

armario - Número de armario

Returns:

True si el armario existe, false en caso contrario.

armarioOcupado

```
public boolean armarioOcupado(int planta,
                              int almacen,
                              int armario)
    throws Exception
```

Verifica si un armario específico está ocupado (hay un dispositivo dentro).

Parameters:

planta - Número de planta

almacen - Número de almacén

armario - Número de armario

Returns:

True si el armario está ocupado, false si está vacío.

Throws:

[Exception](#) - Lanza una excepción si el armario no existe.

obtenerDispositivo

```
public DispositivoMedico obtenerDispositivo(int planta,
                                              int almacen,
                                              int armario)
    throws Exception
```

Obtiene el dispositivo médico almacenado en una ubicación específica del inventario.

Parameters:

planta - Número de planta en la que se encuentra el dispositivo.

almacen - Número de almacén dentro de la planta donde se encuentra el dispositivo.

armario - Número de armario dentro del almacén donde se encuentra el dispositivo.

Returns:

El dispositivo médico almacenado en la ubicación especificada.

Throws:

[Exception](#) - Lanza una excepción si el armario no existe, o está vacío (no contiene ningún dispositivo).

almacenaDispositivo

```
public void almacenDispositivo(DispositivoMedico dispositivo,  
                               int planta,  
                               int almacen,  
                               int armario)  
    throws Exception
```

Almacena un dispositivo médico en un armario específico.

Parameters:

dispositivo - Dispositivo médico a almacenar

planta - Número de planta

almacen - Número de almacén

armario - Número de armario

Throws:

[Exception](#) - Lanza una excepción si el armario ya está ocupado (o no existe) o si el dispositivo es nulo.

liberarArmario

```
public void liberarArmario(int planta,  
                           int almacen,  
                           int armario)  
    throws Exception
```

Libera un armario al dar de baja un dispositivo del inventario, eliminándolo de su ubicación.

Parameters:

planta - Número de planta

almacen - Número de almacén

armario - Número de armario

Throws:

[Exception](#) - Lanza una excepción si no el armario indicado no existe.

obtenerNombreDispositivo

```
public String obtenerNombreDispositivo(int planta,  
                                       int almacen,  
                                       int armario)  
    throws Exception
```

Obtiene el nombre del dispositivo médico almacenado en una ubicación específica del inventario.

Parameters:

planta - Número de planta en la que se encuentra el dispositivo.

almacen - Número de almacén dentro de la planta donde se encuentra el dispositivo.

armario - Número de armario dentro del almacén donde se encuentra el dispositivo.

Returns:

El nombre del dispositivo médico en la ubicación especificada.

Throws:

[Exception](#) - Si el armario está vacío y no contiene ningún dispositivo.

buscaDispositivo

```
public int[] buscaDispositivo(String id)
                           throws Exception
```

Busca un dispositivo por su identificador y devuelve la ubicación.

Parameters:

id - Id del dispositivo a buscar

Returns:

- Devuelve un array de enteros de longitud 3, siendo el primer elemento el número de planta, el segundo el número de almacén y el tercero el número de armario donde se encuentra el dispositivo.

Throws:

[Exception](#) - Lanza una excepción si el dispositivo no se encuentra en el inventario.

encontrarEspacioLibreParaDispositivo

```
public int[] encontrarEspacioLibreParaDispositivo
(DispositivoMedico dispositivo)
                           throws Exception
```

Encuentra el primer espacio libre para colocar un dispositivo médico.

Parameters:

dispositivo - - El dispositivo que se quiere colocar

Returns:

- Devuelve un array de enteros de longitud 3, siendo el primer elemento el número de planta, el segundo el número de almacén y el tercero el número de armario donde se podría colocar el dispositivo.

Throws:

[Exception](#) - - Lanza una excepción si el dispositivo es nulo o si no hay ningún espacio libre

getNumDispositivosEnPlanta

```
public int getNumDispositivosEnPlanta(int planta)
    throws Exception
```

Calcula el número de dispositivos almacenados en una planta.

Parameters:

planta - Número de planta

Returns:

Número total de dispositivos en la planta

Throws:

[Exception](#) - Lanza una excepción si la planta no existe.

getDispositivoMasCaro

```
public DispositivoMedico getDispositivoMasCaro()
    throws Exception
```

Obtiene el dispositivo de mayor valor en el inventario.

Returns:

El dispositivo con el mayor precio en el inventario

Throws:

[Exception](#) - Lanza una excepción si el inventario está vacío.

encontrarDispositivoCompatible

```
public DispositivoMedico encontrarDispositivoCompatible
(DispositivoMedico dispositivo)
    throws Exception
```

Dado un dispositivo, encuentra otro dispositivo en el inventario que sea del mismo tipo, esté operativo, y no esté ocupado.

Parameters:

dispositivo - Dispositivo para el cual se busca otro compatible.

Returns:

Dispositivo compatible, es decir, un dispositivo con el mismo tipo que el dispositivo dado.

Throws:

[Exception](#) - Lanza una excepción si no se encuentra ningún dispositivo compatible.

calcularPrecioPromedio

```
public double calcularPrecioPromedio()
    throws Exception
```


Calcula el precio promedio de los dispositivos en el inventario.

Returns:

El precio promedio de los dispositivos.

Throws:

[Exception](#) - Lanza una excepción si el inventario está vacío.

existeDispositivoNoOperativo

```
public boolean existeDispositivoNoOperativo()
```

Verifica si hay al menos un dispositivo en el inventario que no está operativo,

Returns:

True si existe al menos un dispositivo que no está operativo, False en caso contrario.

dispositivosNoOperativos

```
public DispositivoMedico[] dispositivosNoOperativos()
```

Obtiene los dispositivos que no están operativos.

Returns:

Un array de DispositivoMedico conteniendo los dispositivos que no están operativos.

obtenerEspaciosDisponibles

```
public int[][] obtenerEspaciosDisponibles()
```

Obtiene todos los espacios disponibles en el inventario, es decir, aquellos que no contienen ningún dispositivo.

Returns:

Un array de enteros bidimensional, donde cada subarray representa un espacio disponible. Cada espacio disponible está representado por sus coordenadas: [planta, almacen, armario].

transferirDispositivo

```
public void transferirDispositivo(int planta,
                                  int origenAlmacen,
                                  int origenArmario,
                                  int destinoAlmacen,
                                  int destinoArmario)
    throws Exception
```

Transfiere un dispositivo médico de un armario a otro dentro de la misma planta.

Parameters:

planta - Número de planta

origenAlmacen - Número de almacén de origen
origenArmario - Número de armario de origen
destinoAlmacen - Número de almacén de destino
destinoArmario - Número de armario de destino

Throws:

[Exception](#) - Lanza una excepción si el armario de origen no existe, si no hay un dispositivo en el armario de origen, o si el armario de destino ya está ocupado.

transferirDispositivosPorTipo

```
public void transferirDispositivosPorTipo(Tipo tipo,  
                                           int planta,  
                                           int origenAlmacen,  
                                           int destinoAlmacen)
```

Transfiere todos los dispositivos de un tipo específico desde un almacén de origen a un almacén de destino. Debe imprimir por pantalla un mensaje de error si la planta, origenAlmacen, o destinoAlmacen no existen. Si por algún motivo falla la transferencia de algún dispositivo, deberá imprimir el identificador del dispositivo que ha fallado.

Parameters:

tipo - El tipo de dispositivo que se desea transferir.

planta - La planta donde se encuentran los armarios.

origenAlmacen - El armario de origen desde donde se transferirán los dispositivos.

destinoAlmacen - El armario de destino al que se transferirán los dispositivos.