

Package es.upm.dit.fprg24.p3

Class MonitorDeDispositivos

java.lang.Object[↗]

es.upm.dit.fprg24.p3.MonitorDeDispositivos

```
public class MonitorDeDispositivos  
extends Object↗
```

Clase MonitorDeDispositivos para gestionar el registro y el uso de dispositivos médicos en diferentes áreas del hospital.

Field Summary

Fields

Modifier and Type	Field	Description
private Map [↗] <DispositivoMedico, String [↗] >	registroDispositivos	Diccionario para almacenar cada dispositivo que está actualmente en funcionamiento junto con el nombre del área o unidad en el que se encuentra.

Constructor Summary

Constructors

Constructor	Description
MonitorDeDispositivos()	Constructor que inicializa el registro de dispositivos como un HashMap vacío.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
String [↗]	borrarDispositivo (DispositivoMedico dispos.	Elimina un dispositivo del registro.

double	calcularConsumoTotalEnArea (String [↗] area)	Calcula el consumo energético total de todos los dispositivos en un área específica, por cada hora de uso.
int	contarDispositivosEnUso()	Cuenta el número total de dispositivos en uso (es decir, el número total de dispositivos en el registro).
int	contarDispositivosPorTipo (Tipo tipo)	Cuenta cuántos dispositivos de un tipo específico están en uso.
boolean	dispositivoEnRegistro (DispositivoMedico dispos.	Comprueba si un dispositivo se encuentra en el registro.
Map [↗] <DispositivoMedico, String [↗] >	getRegistroDispositivos()	Obtiene el registro completo de dispositivos y sus áreas asignadas.
void	imprimirDispositivosPorArea	Imprime los dispositivos agrupados por área en un formato organizado.
void	incluirDispositivo (DispositivoMedico dispos. String [↗] area)	Añade un dispositivo al registro y lo asigna a un área.
List [↗] <DispositivoMedico>	listarDispositivosPorArea (String [↗] area)	Lista los dispositivos de un área específica.
Map [↗] <Boolean [↗] ,List [↗] <DispositivoMedico>>	listarDispositivosPorEsta	Lista los dispositivos agrupados según estén en garantía o no en el año actual.
String [↗]	obtenerAreaDispositivo (DispositivoMedico dispos.	Obtiene el área donde se encuentra un dispositivo específico.
Set [↗] <DispositivoMedico>	obtenerConjuntoDispositivos	Obtiene el conjunto de dispositivos actualmente en uso.
DispositivoMedico	obtenerDispositivoMasCaro (String [↗] area)	Devuelve el dispositivo más caro asignado a un área específica.

InstrumentoClinico	obtenerDispositivoMasUtil	Devuelve el instrumento clínico más utilizado en términos de horas de uso.
List[↗] <DispositivoMedico>	obtenerDispositivosEnMant	Obtiene una lista de todos los dispositivos que necesitan mantenimiento.
void	setRegistroDispositivos (Map [↗] <DispositivoMedico, String [↗] > registroDispositivos)	Asigna un nuevo registro de dispositivos al monitor.
void	trasladarDispositivosDeAr (String [↗] areaOrigen, String [↗] areaDestino)	Reorganiza los dispositivos asignados a un área específica, trasladándolos a otra área.

Methods inherited from class java.lang.Object[↗]

clone[↗], equals[↗], finalize[↗], getClass[↗], hashCode[↗], notify[↗], notifyAll[↗], toString[↗], wait[↗], wait[↗], wait[↗]

Field Details

registroDispositivos

```
private Map↗<DispositivoMedico,String↗> registroDispositivos
```

Diccionario para almacenar cada dispositivo que está actualmente en funcionamiento junto con el nombre del área o unidad en el que se encuentra.

Constructor Details

MonitorDeDispositivos

```
public MonitorDeDispositivos()
```

Constructor que inicializa el registro de dispositivos como un HashMap vacío.

Method Details

getRegistroDispositivos

```
public Map↗<DispositivoMedico,String↗> getRegistroDispositivos()
```

Obtiene el registro completo de dispositivos y sus áreas asignadas.

Returns:

Diccionario con dispositivos y sus respectivas áreas.

setRegistroDispositivos

```
public void setRegistroDispositivos  
(Map<DispositivoMedico,String> registroDispositivos)
```

Asigna un nuevo registro de dispositivos al monitor.

Parameters:

registroDispositivos - Diccionario con dispositivos y sus áreas asignadas.

dispositivoEnRegistro

```
public boolean dispositivoEnRegistro(DispositivoMedico dispositivo)
```

Comprueba si un dispositivo se encuentra en el registro.

Parameters:

dispositivo - El dispositivo a buscar.

Returns:

True si el dispositivo está en el registro, False en caso contrario.

incluirDispositivo

```
public void incluirDispositivo(DispositivoMedico dispositivo,  
                               String area)  
    throws Exception
```

Añade un dispositivo al registro y lo asigna a un área. Además, si se incluye correctamente, se deberá marcar el dispositivo como ocupado.

Parameters:

dispositivo - El dispositivo a incluir.

area - El área en la que se usará el dispositivo.

Throws:

Exception - Si el dispositivo ya está en el registro.

obtenerConjuntoDispositivosEnUso

```
public Set<DispositivoMedico> obtenerConjuntoDispositivosEnUso()
```

Obtiene el conjunto de dispositivos actualmente en uso.

Returns:

Conjunto de dispositivos registrados.

contarDispositivosEnUso

```
public int contarDispositivosEnUso()
```

Cuenta el número total de dispositivos en uso (es decir, el número total de dispositivos en el registro).

Returns:

Número de dispositivos registrados.

borrarDispositivo

```
public String↗ borrarDispositivo(DispositivoMedico dispositivo)  
                               throws Exception↗
```

Elimina un dispositivo del registro. Además, si se elimina correctamente, se deberá marcar el dispositivo como no ocupado.

Parameters:

dispositivo - El dispositivo a eliminar.

Returns:

El área a la que estaba asignado el dispositivo.

Throws:

Exception[↗] - Si el dispositivo no está en el registro.

listarDispositivosPorEstado

```
public Map↗<Boolean↗, List↗  
<DispositivoMedico>> listarDispositivosPorEstado()
```

Lista los dispositivos agrupados según estén en garantía o no en el año actual. El método debería funcionar independientemente de en qué año se ejecute.

Returns:

Diccionario con listas de dispositivos agrupados según estén en garantía o no.

contarDispositivosPorTipo

```
public int contarDispositivosPorTipo(Tipo tipo)
```

Cuenta cuántos dispositivos de un tipo específico están en uso.

Parameters:

tipo - Tipo de dispositivo a buscar.

Returns:

Número de dispositivos del tipo especificado.

obtenerAreaDispositivo

```
public String obtenerAreaDispositivo(DispositivoMedico dispositivo)
    throws Exception
```

Obtiene el área donde se encuentra un dispositivo específico.

Parameters:

dispositivo - El dispositivo a buscar.

Returns:

El nombre del área donde se encuentra el dispositivo.

Throws:

Exception - Si el dispositivo no está registrado.

listarDispositivosPorArea

```
public List<DispositivoMedico> listarDispositivosPorArea(String area)
```

Lista los dispositivos de un área específica.

Parameters:

area - Área donde buscar dispositivos.

Returns:

Lista de dispositivos asignados al área especificada.

obtenerDispositivoMasUtilizado

```
public InstrumentoClinico obtenerDispositivoMasUtilizado()
```

Devuelve el instrumento clínico más utilizado en términos de horas de uso.

Returns:

El dispositivo con más horas registradas.

obtenerDispositivosEnMantenimiento

```
public List<DispositivoMedico> obtenerDispositivosEnMantenimiento()
```

Obtiene una lista de todos los dispositivos que necesitan mantenimiento.

Returns:

Lista de dispositivos que requieren mantenimiento.

obtenerDispositivoMasCaroEnArea

```
public DispositivoMedico obtenerDispositivoMasCaroEnArea(String area)
    throws Exception
```

Devuelve el dispositivo más caro asignado a un área específica.

Parameters:

area - El área en la que buscar el dispositivo más caro.

Returns:

El dispositivo con el precio más alto en el área especificada.

Throws:

`Exception` - Si no hay dispositivos asignados al área.

calcularConsumoTotalEnArea

```
public double calcularConsumoTotalEnArea(String area)
```

Calcula el consumo energético total de todos los dispositivos en un área específica, por cada hora de uso.

Parameters:

area - El área en la que calcular el consumo total.

Returns:

El consumo total en kWh.

imprimirDispositivosPorArea

```
public void imprimirDispositivosPorArea()
```

Imprime los dispositivos agrupados por área en un formato organizado.

trasladarDispositivosDeArea

```
public void trasladarDispositivosDeArea(String areaOrigen,
    String areaDestino)
    throws Exception
```

Reorganiza los dispositivos asignados a un área específica, trasladándolos a otra área.

Parameters:

areaOrigen - El área de origen de los dispositivos.

areaDestino - El área de destino de los dispositivos.

Throws:

[Exception](#) - Si no hay dispositivos en el área de origen.