OBJETIVO

Este documento contiene material de apoyo para el boletín T6 del trabajo práctico. En concreto, el documento incluye las interfaces y clases que se trabajarán en el aula. El profesor las explicará en la sesión de laboratorio y responderá a sus dudas al respecto. El alumno, por su parte, debe añadir el código a su proyecto de curso, y realizar por su cuenta el resto de ejercicios del boletín.

INTERFACES

Interfaz `Departamento`

```
package fp.grados.tipos;

import java.util.Set;

public interface Departamento extends Comparable<Departamento> {
	String getNombre();

	Set<Asignatura> getAsignaturas();

	void nuevaAsignatura(Asignatura a);

	void eliminaAsignatura(Asignatura a);

	// TODO: Para hacer en casa
	// Set<Profesor> getProfesores();
	// void nuevoProfesor(Profesor p);
	// void eliminaProfesor(Profesor p);
}
```

Interfaz `Centro`

```
package fp.grados.tipos;

import java.util.Set;

public interface Centro extends Comparable<Centro> {
	String getNombre();

	String getDireccion();

	Integer getNumeroPlantas();

	Integer getNumeroSotanos();

	Set<Espacio> getEspacios();

	void nuevoEspacio(Espacio e);

	void eliminaEspacio(Espacio e);
}
```

CLASES

Clase `DepartamentoImpl`

```
package fp.grados.tipos;

import java.util.HashSet;
import java.util.Set;

public class DepartamentoImpl implements Departamento {
	private String nombre;
	private Set<Asignatura> asignaturas;

	public DepartamentoImpl(String nombre) {
		this.nombre = nombre;
		this.asignaturas = new HashSet<Asignatura>();
	}

	public String getNombre() {
		return nombre;
	}

	public Set<Asignatura> getAsignaturas() {
		return new HashSet<Asignatura>(asignaturas);
	}

	public void nuevaAsignatura(Asignatura a) {
		asignaturas.add(a);
		// Ponemos a "este objeto", es decir, a this, como el departamento de la
		// asignatura a.
		a.setDepartamento(this);
	}

	public void eliminaAsignatura(Asignatura a) {
		if(asignaturas.contains(a)){
			asignaturas.remove(a);
			// Al poner el departamento a "null" indicamos
			// que esa asignatura se ha quedado
			// "huérfana"
			a.setDepartamento(null);
		}
	}

	public int compareTo(Departamento d) {
		return getNombre().compareTo(d.getNombre());
	}

	public String toString() {
		return getNombre();
	}

	public boolean equals(Object o) {
		boolean res = false;

		if (o instanceof Departamento) {
			Departamento d = (Departamento) o;
			res = getNombre().equals(d.getNombre());
		}

		return res;
	}

	public int hashCode() {
		return getNombre().hashCode();
	}
}
```

Clase `AsignaturaImpl`

```
public class AsignaturaImpl implements Asignatura {
	private String nombre;
	private String codigo;
	private Double creditos;
	private TipoAsignatura tipo;
	private Integer curso;
	// Nuevo en T6
	private Departamento departamento;

	public AsignaturaImpl(String nombre, String codigo, Double creditos,
			TipoAsignatura tipo, Integer curso) {
		checkCodigo(codigo);
		checkCreditos(creditos);
		checkCurso(curso);

		this.nombre = nombre;
		this.codigo = codigo;
		this.creditos = creditos;
		this.tipo = tipo;
		this.curso = curso;
		// Nuevo en T6
		this.departamento = null;
	}

	//El resto de métodos se quedan como estaban, y se añaden los siguientes

	public AsignaturaImpl(String nombre, String codigo, Double creditos,
			TipoAsignatura tipo, Integer curso, Departamento departamento) {
		checkCodigo(codigo);
		checkCreditos(creditos);
		checkCurso(curso);

		this.nombre = nombre;
		this.codigo = codigo;
		this.creditos = creditos;
		this.tipo = tipo;
		this.curso = curso;
		setDepartamento(departamento);
	}

	public Departamento getDepartamento() {
		return departamento;
	}

	public void setDepartamento(Departamento nuevoDepartamento) {
		// Si el nuevo departamento es el mismo que ya había no haríamos nada.
		if (nuevoDepartamento != this.departamento) {
			Departamento antiguoDepartamento = this.departamento;
			this.departamento = nuevoDepartamento;
			if (antiguoDepartamento != null) {
				// Si antes la asignatura pertenecía a otro departamento,
				// tenemos que actualizar ese departamento eliminando a
				// "esta asignatura"(es decir, a this) de las asignaturas
				// que conforman el departamento.
				antiguoDepartamento.eliminaAsignatura(this);
			}
			if (nuevoDepartamento != null) {
				// Si la asignatura pertenece a un nuevo departamento,
				// tenemos que actualizar este departamento para incluir a
				// "esta asignatura" (es decir, a this) en las asignaturas
				// que conforman el departamento.
				nuevoDepartamento.nuevaAsignatura(this);
			}
		}
	}
}
```

TESTS

La clase `TestDepartamento` se proporciona en un fichero independiente.