OBJETIVO

Este documento contiene material de apoyo para el boletín T9 del trabajo práctico. En concreto, el documento incluye algunos de los métodos que se añaden en este boletín a los tipos Centro y Departamento, así como las modificaciones a realizar a ProfesorImpl para obtener ProfesorImpl2. 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.

Recuerde que debe añadir cada uno de los métodos cuya implementación se le proporciona en este boletín a la interfaz correspondiente, si aún no figura en ella.

NUEVAS CLASES

Clase `ProfesorImpl2`

```
public class ProfesorImpl2 extends PersonaImpl implements Profesor {
	// Faltan el resto de atributos
	private Map<Asignatura, Double> creditosPorAsignatura;

	public ProfesorImpl2(String dni, String nombre, String apellidos,
			LocalDate fechaNacimiento,String email,Categoria categoría){
		super(dni, nombre, apellidos, fechaNacimiento, email);
		// Falta el resto del constructor: checkers, inicializaciones, ...
		creditosPorAsignatura = new HashMap<Asignatura, Double>();
	}

	public ProfesorImpl2(String dni, String nombre, String apellidos,
			LocalDate fechaNacimiento, String email,
			Categoria categoria, Departamento departamento) {
		super(dni, nombre, apellidos, fechaNacimiento, email);
		// Falta el resto del constructor: checkers, inicializaciones, ...
		creditosPorAsignatura = new HashMap<Asignatura, Double>();
	}

	public List<Asignatura> getAsignaturas() {
		return new ArrayList<Asignatura>(creditosPorAsignatura.keySet());
	}

	public List<Double> getCreditos() {
		return new ArrayList<Double>(creditosPorAsignatura.values());
	}

	public void imparteAsignatura(Asignatura asig, Double dedicacion) {
		// Faltan los checkers
		if (creditosPorAsignatura.containsKey(asig)) {
			actualizaDedicacion(asig, dedicacion);
		} else {
			nuevaAsignatura(asig, dedicacion);
		}
	}
```

Clase `ProfesorImpl2`

```
	private void actualizaDedicacion(Asignatura asig, Double dedicacion) {
		// Falta el checker
		creditosPorAsignatura.put(asig, dedicacion);
	}

	private void nuevaAsignatura(Asignatura asig, Double dedicacion) {
		// Falta el checker
		creditosPorAsignatura.put(asig, dedicacion);
	}

	public void eliminaAsignatura(Asignatura asig) {
		creditosPorAsignatura.remove(asig);
	}

	public Double dedicacionAsignatura(Asignatura asig) {
		Double res = 0.0;

		if (creditosPorAsignatura.containsKey(asig)) {
			res = creditosPorAsignatura.get(asig);
		}

		return res;
	}
	// El resto de la clase no sufre cambios con respecto a ProfesorImpl
}
```

NUEVAS OPERACIONES

Clase `CentroImpl`

```
public class CentroImpl implements Centro {

	public SortedMap<String, Despacho> getDespachosPorProfesor() {
		SortedMap<String, Despacho> res = new TreeMap<String, Despacho>();

		for (Despacho d : getDespachos()) {
			anyadeProfesores(d, res);
		}
		return res;
	}

	private void anyadeProfesores(Despacho d,
					SortedMap<String, Despacho> res) {
		for (Profesor p : d.getProfesores()) {
			res.put(p.toString(), d);
		}
	}

}
```

Clase `DepartamentoImpl`

```
public class DepartamentoImpl implements Departamento {

      public SortedMap<Asignatura, SortedSet<Profesor>>
                                          getProfesoresPorAsignatura() {
            SortedMap<Asignatura, SortedSet<Profesor>> res =
                        new TreeMap<Asignatura, SortedSet<Profesor>>();
            for (Profesor p : getProfesores()) {
                  anyadeAsignaturas(p, res);
            }
            return res;
      }

      private void anyadeAsignaturas(Profesor p,
                        SortedMap<Asignatura, SortedSet<Profesor>> res) {
            for (Asignatura a : p.getAsignaturas()) {
                  if (res.containsKey(a)) {
                        res.get(a).add(p);
                  } else {
                        SortedSet<Profesor> s = new TreeSet<Profesor>();
                        s.add(p);
                        res.put(a, s);
                  }
            }
      }
}
```