



OBJETIVO

Este documento contiene material de apoyo para el boletín T4 del trabajo práctico. En concreto, el documento incluye modificaciones de las clases y los tests que se trabajarán en el aula. **Sólo se muestran los métodos nuevos con respecto a las clases implementadas en el boletín anterior.** El profesor los 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.

CLASES

Clase AsignaturaImpl

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

    if (o instanceof Asignatura) {
        Asignatura a = (Asignatura) o;
        res = getCodigo().equals(a.getCodigo());
    }

    return res;
}

public int hashCode() {
    return getCodigo().hashCode();
}

public int compareTo(Asignatura a) {
    return getCodigo().compareTo(a.getCodigo());
}
```

Clase BecaImpl

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

    if (o instanceof Beca) {
        Beca b = (Beca) o;
        res = getCodigo().equals(b.getCodigo())
            && getTipo().equals(b.getTipo());
    }

    return res;
}

public int hashCode() {
    return getCodigo().hashCode() + getTipo().hashCode() * 31;
}

public int compareTo(Beca b) {
    int res = getCodigo().compareTo(b.getCodigo());

    if (res == 0) {
        res = getTipo().compareTo(b.getTipo());
    }

    return res;
}
```



Clase PersonaImpl

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

    if (o instanceof Persona) {
        Persona a = (Persona) o;
        res = getDNI().equals(a.getDNI())
            && getNombre().equals(a.getNombre())
            && getApellidos().equals(a.getApellidos());
    }

    return res;
}

public int hashCode() {
    return getDNI().hashCode() + getNombre().hashCode() * 31 +
        getApellidos().hashCode() * 31 * 31;
}

public int compareTo(Persona p) {
    int res = getApellidos().compareTo(p.getApellidos());

    if (res == 0) {
        res = getNombre().compareTo(p.getNombre());

        if (res == 0) {
            res = getDNI().compareTo(p.getDNI());
        }
    }

    return res;
}
```

TESTS

Clase TestAsignatura

```
public static void main(String[] args) {
    //...
    // Añadir después llamadas a casos de prueba anteriores
    testIgualdad();
    testOrden();
}

private static void testIgualdad() {
    System.out.println("\n=====Probando igualdad");
    // Creamos un objeto
    Asignatura a1 = new AsignaturaImpl("Fundamentos de Programación", "2050001", 12.0,
        TipoAsignatura.ANUAL, 1);
    // Creamos un objeto igual al anterior
    Asignatura a2 = new AsignaturaImpl("Fundamentos de Programación", "2050001", 12.0,
        TipoAsignatura.ANUAL, 1);
    // Creamos un objeto distinto
    Asignatura a3 = new AsignaturaImpl("Estructura de Computadores", "2050009", 6.0,
        TipoAsignatura.SEGUNDO_CUATRIMESTRE, 1);

    System.out.println("Código hash del objeto a1 (" + a1 + "): " + a1.hashCode());
    System.out.println("Código hash del objeto a2 (" + a2 + "): " + a2.hashCode());
    System.out.println("Código hash del objeto a3 (" + a3 + "): " + a3.hashCode());

    System.out.println("¿Es a1 igual a a2? (debe ser true): " + a1.equals(a2));
    System.out.println("¿Es a1 distinto de a3? (debe ser true): " + !a1.equals(a3));
}

private static void testOrden() {
    System.out.println("\n=====Probando orden natural");
    // Creamos cuatro objetos tales que menor < igual1 == igual2 < mayor
    Asignatura menor = new AsignaturaImpl("Fundamentos de Programación", "2050001",
        12.0, TipoAsignatura.ANUAL, 1);
    Asignatura igual1 = new AsignaturaImpl("Estructura de Computadores", "2050009",
        6.0, TipoAsignatura.SEGUNDO_CUATRIMESTRE, 1);
    Asignatura igual2 = new AsignaturaImpl("Estructura de Computadores", "2050009",
        6.0, TipoAsignatura.SEGUNDO_CUATRIMESTRE, 1);
    Asignatura mayor = new AsignaturaImpl("Ingeniería de Requisitos", "2050020",
        6.0, TipoAsignatura.PRIMER_CUATRIMESTRE, 3);

    System.out.print("(debe ser ANTES) ");
    compara(menor, igual1);
    System.out.print("(debe ser MISMA POSICIÓN) ");
    compara(igual1, igual2);
    System.out.print("(debe ser ANTES) ");
    compara(igual2, mayor);
}

// Métodos auxiliares
private static void compara(Asignatura a1, Asignatura a2) {
    System.out.print("El objeto <" + a1 + ">");
    if (a1.compareTo(a2) < 0) {
        System.out.print(" va ANTES que el objeto ");
    }
    else if (a1.compareTo(a2) > 0) {
        System.out.print(" va DESPUÉS que el objeto ");
    }
    else {
        System.out.print(" va en la MISMA POSICIÓN que el objeto ");
    }
    System.out.println("<" + a2 + ">");
}
```

Clase TestBeca

```
public static void main(String[] args) {
    ///...
    // Añadir después llamadas a casos de prueba anteriores
    testIgualdad();
    testOrden();
}

private static void testIgualdad() {
    System.out.println("\n=====Probando igualdad");
    // Creamos dos objetos iguales (mismo código, mismo tipo de beca,
    // el resto distinto)
    Beca b1 = new BecaImpl("ABC1234",2000.0,12,TipoBeca.EMPRESA);
    Beca b2 = new BecaImpl("ABC1234",3000.0,6,TipoBeca.EMPRESA);
    // Creamos objetos distintos (distinto código, distinto tipo de beca)
    Beca b3 = new BecaImpl("ZWQ9877",2000.0,12,TipoBeca.EMPRESA);
    Beca b4 = new BecaImpl("ABC1234",3000.0,6,TipoBeca.MOVILIDAD);

    System.out.println("Código hash del objeto b1 (" +b1+" ): "+b1.hashCode());
    System.out.println("Código hash del objeto b2 (" +b2+" ): "+b2.hashCode());
    System.out.println("Código hash del objeto b3 (" +b3+" ): "+b3.hashCode());
    System.out.println("Código hash del objeto b4 (" +b4+" ): "+b4.hashCode());

    System.out.println("¿Es b1 igual a b2? (debe ser true): "+b1.equals(b2));
    System.out.println("¿Es b1 distinto de b3? (debe ser true): "+!b1.equals(b3));
    System.out.println("¿Es b1 distinto de b4? (debe ser true): "+!b1.equals(b4));
}

private static void testOrden() {
    System.out.println("\n=====Probando orden natural");
    // Creamos cuatro objetos tales que menor < igual1 == igual2 < mayor
    Beca menor = new BecaImpl("ABC1234",10000.0,6,TipoBeca.ORDINARIA);
    Beca igual1 = new BecaImpl("ABC1235",10000.0,6,TipoBeca.ORDINARIA);
    Beca igual2 = new BecaImpl("ABC1235",10000.0,6,TipoBeca.ORDINARIA);
    Beca mayor = new BecaImpl("ABC1235",10000.0,6,TipoBeca.EMPRESA);

    System.out.print("(debe ser ANTES) ");
    compara(menor,igual1);
    System.out.print("(debe ser MISMA POSICIÓN) ");
    compara(igual1,igual2);
    System.out.print("(debe ser ANTES) ");
    compara(igual2,mayor);
}

// Métodos auxiliares
private static void compara(Beca b1, Beca b2) {
    System.out.print("El objeto <" +b1+">");
    if (b1.compareTo(b2)<0) {
        System.out.print(" va ANTES que el objeto ");
    }
    else if (b1.compareTo(b2)>0) {
        System.out.print(" va DESPUÉS que el objeto ");
    }
    else {
        System.out.print(" va en la MISMA POSICIÓN que el objeto ");
    }
    System.out.println("<" +b2+">");
}
```

Clase TestPersona

```
public static void main(String[] args) {
    //...
    // Añadir después llamadas a casos de prueba anteriores
    testIgualdad();
    testOrden();
}

private static void testIgualdad() {
    System.out.println("\n=====Probando igualdad con dos objetos iguales");
    // Creamos dos objetos iguales (mismo dni, nombre y apellidos)
    Persona p1 = new PersonaImpl("12345678Z", "Juan", "Nadie Nadie",
        LocalDate.of(1950, 3, 15), "juan.nadie@gmail.com");
    Persona p2 = new PersonaImpl("12345678Z", "Juan", "Nadie Nadie",
        LocalDate.of(1980, 6, 25), "juan.nadie2@gmail.com");

    // Creamos objetos distintos (distinto dni, distinto nombre, distintos
    // apellidos)
    Persona p3 = new PersonaImpl("12345677J", "Juan", "Nadie Nadie",
        LocalDate.of(1952, 1, 10), "nadie@gmail.com");
    Persona p4 = new PersonaImpl("12345678Z", "Antonio", "Nadie Nadie",
        LocalDate.of(1950, 3, 15), "juan.nadie@gmail.com");
    Persona p5 = new PersonaImpl("12345678Z", "Juan", "Otros Apellidos",
        LocalDate.of(1950, 3, 15), "juan.nadie@gmail.com");

    System.out.println("Código hash del objeto p1 (" + p1 + "): " + p1.hashCode());
    System.out.println("Código hash del objeto p2 (" + p2 + "): " + p2.hashCode());
    System.out.println("Código hash del objeto p3 (" + p3 + "): " + p3.hashCode());
    System.out.println("Código hash del objeto p4 (" + p4 + "): " + p4.hashCode());
    System.out.println("Código hash del objeto p5 (" + p5 + "): " + p5.hashCode());

    System.out.println("¿Es p1 igual a p2? (debe ser true): " + p1.equals(p2));
    System.out.println("¿Es p1 distinto de p3? (debe ser true): " + !p1.equals(p3));
    System.out.println("¿Es p1 distinto de p4? (debe ser true): " + !p1.equals(p4));
    System.out.println("¿Es p1 distinto de p5? (debe ser true): " + !p1.equals(p5));
}

private static void testOrden() {
    System.out.println("\n=====Probando orden natural");
    // Creamos cuatro objetos tales que menor < igual1 == igual2 < mayor
    Persona menor = new PersonaImpl("12345678Z", "Juan", "Nadie Nadie",
        LocalDate.of(1950, 3, 15), "juan.nadie@gmail.com");
    Persona igual1 = new PersonaImpl("12345677J", "Juana", "Nadie Nadie",
        LocalDate.of(1980, 7, 13));
    Persona igual2 = new PersonaImpl("12345677J", "Juana", "Nadie Nadie",
        LocalDate.of(1980, 7, 13));
    Persona mayor = new PersonaImpl("12345679S", "Juana", "Nadie Nadie",
        LocalDate.of(1980, 7, 13));

    System.out.print("(debe ser ANTES) ");
    compara(menor, igual1);
    System.out.print("(debe ser MISMA POSICIÓN) ");
    compara(igual1, igual2);
    System.out.print("(debe ser ANTES) ");
    compara(igual2, mayor);
}

// Métodos auxiliares
private static void compara(Persona p1, Persona p2) {
    System.out.print("El objeto <" + p1 + ">");
    if (p1.compareTo(p2) < 0) {
        System.out.print(" va ANTES que el objeto ");
    } else if (p1.compareTo(p2) > 0) {
        System.out.print(" va DESPUÉS que el objeto ");
    } else {
        System.out.print(" va en la MISMA POSICIÓN que el objeto ");
    }
    System.out.println("<" + p2 + ">");
}
```