Prueba 2 de Evaluación Continua

Análisis y Diseño de Software (2010/2011)

Contesta en esta misma hoja

Apellidos: **Nombre:**

Apartado 1 (1.5 puntos)

Señala los errores de compilación (si los hubiera), indica por qué son errores, y cómo los corregirías:

```
class ClasePadre {
	private int id;
	public ClasePadre(int id) {
		this.id = id;
	}

	public abstract void imprime();
}

class ClaseHija extends ClasePadre {
	String id;

	public ClaseHija() {
		id = "ID";
	}
	public void imprime() {
		System.out.println("IDs = "+this.id+", "+super.id);
	}
}

public class Apartado1 {
	public static void main(String[] args) {
		ClasePadre p = new ClaseHija();
		p.imprime();
	}
}
```

Errores y Corrección:

Segunda prueba evaluación continua ADS. Curso 2010/2011

Prueba 2 de Evaluación Continua

Análisis y Diseño de Software (2010/2011)

Contesta en esta misma hoja

Apellidos:
Nombre:

Apartado 2 (2.5 puntos)

Señala los errores de compilación del método *main* (si los hubiera). Después de eliminar las líneas que provocan los errores, indica los errores de ejecución. Para cada error – de compilación o ejecución – escribe una línea que justifique por qué se produce. Después de eliminar las líneas que los provocan, indica la salida del programa:

```
class Clase1 {
	double atrib = 4.5;
	protected String name = "Clase1";
	public double incrementar(int param) {
		System.out.println("Incrementando en Clase 1");
		return atrib+param;
	}
}
class Clase2 extends Clase1 {
	private static int num = 0;
	public Clase2(double d) {
		Clase2.num++;
		this.atrib = d;
	}
	@Override public double incrementar(int p) {
		System.out.println("Incrementando en Clase 2");
		return super.atrib*p;
	}
	public void imprime() { System.out.println("Info de Clase2 = "+atrib+", "+num);}
}
public class Apartado2 {
  public static void main(String[] args) {
	 Clase1 c11 = new Clase1();                  // línea 1
	 Clase1 c12 = new Clase2(2.5);               // línea 2
	 Clase2 c21a = new Clase1();                 // línea 3
	 Clase2 c21b = (Clase2) new Clase1();        // línea 4
	 Clase2 c22a = new Clase2();                 // línea 5
	 c12.incrementar(3);                         // línea 6
	 Clase2 c22b = new Clase2(4);                // línea 7
	 c22b.incrementar(2);                        // línea 8
	 c12.atrib = 8;                              // línea 9
	 c11.name = "Objeto c11";                    // línea 10
	 c12.imprime();                              // línea 11
	 c22b.imprime();                             // línea 12
  }
}
```

Errores de compilación en líneas:

Errores de ejecución en líneas:

Salida:

Segunda prueba evaluación continua ADS. Curso 2010/2011

Prueba 2 de Evaluación Continua

Análisis y Diseño de Software (2010/2011)

Contesta al dorso. Utiliza hojas adicionales en caso necesario

Apellidos:
Nombre:

Apartado 3 (6 puntos)

Se quiere construir una aplicación que emule un sistema sencillo de ficheros con control de acceso. El sistema contiene ficheros de datos y ejecutables, los primeros se pueden abrir en modo lectura o escritura, mientras que los últimos sólo en modo ejecución. Los usuarios tienen un privilegio (un número del 0 al 10), y los ficheros tienen un nivel de acceso. Así, sólo los usuarios con privilegio mayor o igual al nivel de acceso de un fichero pueden abrirlo. Un usuario no puede abrir un fichero dos veces (incluso en modos distintos) sin haberlos liberado antes. Ningún fichero se puede abrir por dos usuarios a la vez, pero existe una excepción: si el fichero es ejecutable, y el usuario tiene nivel de acceso 10, entonces puede acceder a él sin importar si el fichero ya está ocupado o no.
Escribe las clases necesarias para que la salida resultante sea la que se muestra más abajo.

```
public class Ficheros {
  public static void main(String[] argc) {
    Usuario user1 = new Usuario("Juan", 5);           // Nombre y privilego
    Usuario user2 = new Usuario("Eduardo", 6);        // Nombre y privilegio
    Usuario admin = new Usuario("Administrador", 10); // Nombre y privilegio

    FicheroDatos fd = new FicheroDatos("Examen", 6, 6); //Nombre, privi. lectura y escritura
    FicheroEjecutable fe = new FicheroEjecutable("Java", 4); // Nombre, privi. ejecución

    user1.accedeFichero(fd, ModoAcceso.LECTURA); // No suficiente permiso (necesita un 6)
    user1.accedeFichero(fe);                     // OK

    user2.accedeFichero(fd, ModoAcceso.LECTURA);  // OK
    user2.accedeFichero(fe, ModoAcceso.ESCRITURA);// No accesible para escritura

    admin.accedeFichero(fd, ModoAcceso.ESCRITURA);// NO: ya se está usando
    admin.accedeFichero(fe, ModoAcceso.EJECUCION);// OK, ya que privilegio de admin = 10

    System.out.println("Ficheros abiertos por "+user1+" = "+user1.getFicheros());
    System.out.println("Ficheros abiertos por "+user2+" = "+user2.getFicheros());
    System.out.println("Ficheros abiertos por "+admin+" = "+admin.getFicheros());

    user1.liberaFichero(fe);
    user2.liberaFichero(fd);
    admin.liberaFichero(fe);
  }
}

Salida:
Ficheros abiertos por Juan = [Java]
Ficheros abiertos por Eduardo = [Examen]
Ficheros abiertos por Administrador = [Java]
```

Nota: Aunque en el programa no se ha usado, *accedeFichero* debe devolver **true** o **false** dependiendo de si el usuario ha podido acceder al fichero o no.

Segunda prueba evaluación continua ADS. Curso 2010/2011