

Apellidos

Nombre

Instrucciones para la realización del examen:

- El alumno debe entregar una carpeta con las soluciones al examen. El nombre de la carpeta debe estar formado por el número de lista seguido de las iniciales. Por ejemplo, Facundo Romuedo Piladro que es el número 8 de la lista entregaría una carpeta con nombre **Ex08frp**.
- Los ficheros o carpetas correspondientes a las soluciones se deben nombrar igual que la carpeta junto con el número del ejercicio, por ejemplo **Ex08frp1.java**, **Ex08frp2.java**, etc. en caso de ficheros o **Ex08frp1**, **Ex08frp2**, etc. en caso de carpetas.
- En cada programa se deben incluir comentarios con el nombre completo del alumno, el curso, la fecha y el modelo de examen.

1. Crea la clase **Rectangulo** con los atributos **alto** (de tipo entero), **ancho** (de tipo entero) y **estaRelleno** (de tipo booleano). Crea el **constructor** y el método **toString**. Prueba la clase mediante un programa que cree y pinte al menos dos rectángulos, uno relleno y otro hueco. No es necesario utilizar menús ni pedir los datos por teclado, se pueden crear directamente los objetos de prueba en varias líneas.
2. Crea la función de manejo de arrays que tenga la siguiente cabecera y que haga lo que se especifica en los comentarios (puedes incluirla en tu propia biblioteca de rutinas):

```
public static int[] extraePares(int x[]) // Devuelve un array con todos los números pares que se
// encuentren en otro array que se pasa como parámetro.
// Lógicamente el tamaño del array que se devuelve será
// menor o igual al que se pasa como parámetro.
```

Utiliza esta función en un programa para comprobar que funcionan bien. Si en el array no existe ningún número par, se devuelve un array con el número -1 como único elemento.

3. Escribe un programa que genere una secuencia de 6 fichas de dominó perfectamente colocadas. Cada ficha tiene dos mitades y cada mitad contiene un número de puntitos del 0 (blanca) al 6. Para que dos fichas consecutivas estén bien colocadas, deben coincidir las dos mitades que se ponen juntas. Hay que tener en cuenta que la ficha [2 | 4] es la misma que la ficha [4 | 2]. Cada ficha que va saliendo debe ser un objeto de la clase Ficha.

Ejemplo:

```
[ 1 | 4 ] [ 4 | 5 ] [ 5 | 3 ] [ 3 | 3 ] [ 3 | ] [ | 6 ]
```

De forma ideal, en la secuencia mostrada, no se pueden repetir las fichas. Si se controla que no se repitan las fichas, el ejercicio se puntuará con 2'5 pts y si no se controla la repetición pero todo lo demás está bien, se puntuará con 2 pts.

4. El ayuntamiento nos va a contratar para implementar un nuevo programa que llevará el control de los aparcamientos municipales. Crea la clase **Tique** de tal forma que cuando se crea un nuevo tique de aparcamiento se guarde el momento en el que es creado. Cada tique llevará un número consecutivo, empezando por el 1. El precio es de 2 céntimos por minuto (0'02 €).

El siguiente programa genera la salida que se muestra:

```
public class Ejercicio04 {
    public static void main(String[] args) {
        Tique t1 = new Tique(10, 5);
        Tique t2 = new Tique(14, 30);
        Tique t3 = new Tique(15, 00);
        t1.paga(11, 00);
        t3.paga(17, 45);
        t2.paga(18, 20);
    }
}
```

Tique nº 1. Debe pagar 1,10 euros. Gracias.

Tique nº 3. Debe pagar 3,30 euros. Gracias.

Tique nº 2. Debe pagar 4,60 euros. Gracias.