



Examen final – Convocatoria de junio de 2006
FUNDAMENTOS DE LA PROGRAMACIÓN

Nombre, apellidos y DNI

PRIMERA PARTE

Consideremos que una imagen está formada por una matriz de números enteros en la que cada elemento contiene el nivel de gris de un píxel. Este valor puede ir de 0 (negro) a 255 (blanco). Dada la clase Imagen ...

```
public class Imagen
{
    private int pixels[] [];
}
```

EJERCICIO 1 (2 puntos)

Implemente un método `reflejoHorizontal()` que refleje horizontalmente la imagen (esto es, lo que hubiese a la izquierda de la imagen original debe quedar a la derecha y lo que hubiese a la derecha debe quedar a la izquierda en la imagen reflejada).

EJERCICIO 2 (4 puntos)

Diseñe e implemente un método `ecualizar()` que ecualice la imagen. La ecualización del histograma de una imagen uniformiza los niveles de gris de una imagen de tal forma que, en imágenes muy concentradas en determinados niveles de gris, se aumenta el contraste. Para ecualizar una imagen:

1. Se calcula el histograma acumulado de la imagen: $c(i) = \sum_{j=0}^i p(j)$
2. Se transforma cada píxel de la imagen de tal forma que, si tenemos un píxel con el nivel de gris i , éste pasará a tener el nivel de gris $255 * c(i) / \text{píxeles}$,

donde $p(j)$ es el número de píxeles de la imagen con el nivel de gris j , $c(i)$ es el número de píxeles con una intensidad menor o igual que i y píxeles es el número de píxeles de la imagen.

EJERCICIO 3 (4 puntos)

Dado el siguiente fragmento de código:

```
static final double N = 2;
static final double PREC = 1e-6;

static double f (double x)
{
    return x*x-N;
}

static double bis (double min, double max)
{
    double med = (min+max)/2;
    if (max-min<PREC) {
        return med;
    } else if (f(min)*f(med)<0) {
        return bis (min,med);
    } else {
        return bis (med,max);
    }
}
```

- a) ¿Qué calcula la llamada a la función recursiva `bis(0,N)`? Si cambiamos el valor de `N`, ¿qué estaríamos calculando? ¿Y si cambiásemos la función `f(x)`?
- b) Implemente un algoritmo iterativo equivalente.



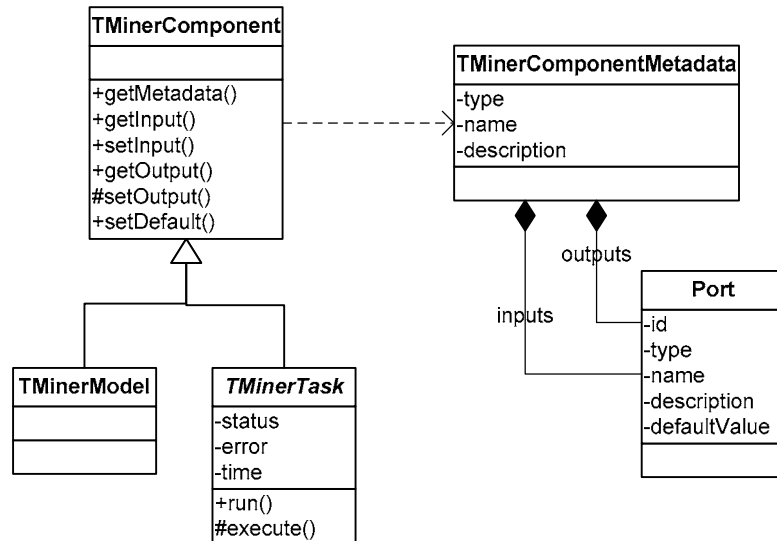
Examen final – Convocatoria de junio de 2006
FUNDAMENTOS DE LA PROGRAMACIÓN

Nombre, apellidos y DNI

SEGUNDA PARTE

EJERCICIO 1 (4 puntos)

Dado el siguiente diagrama UML:



- Declare e implemente adecuadamente en Java las clases que se derivan del diagrama.
- Defina una subclase de `TMinerTask` llamada `Experimento` que redefina el método `execute()` de tal forma que éste sirva para determinar si un número es primo o no (el número se obtiene a través del método `getInput()`, el cual devuelve un `String`, y el resultado del experimento se guarda como una cadena utilizando el método `setOutput()`)
- Diseñe un conjunto de casos de prueba adecuado para comprobar el funcionamiento de la clase implementada en el apartado anterior.
- Implemente un programa en Java que lance el experimento en una hebra independiente.

EJERCICIO 2 (3 puntos)

Diseñe e implemente un programa en Java que lea un fichero de texto y nos dé la frecuencia de las 3 vocales que aparecen más a menudo en el fichero.

EJERCICIO 3 (3 puntos)

Queremos acceder a un componente de tipo `TMinerComponent` de forma remota usando RMI: describa cómo deberíamos declarar la clase `TMinerComponent`, implemente el servidor RMI necesario para poder acceder al componente de forma remota y escriba el código necesario para acceder al componente desde un cliente RMI.