

Control Intermedio de Ingeniería del Software II

13 Abril 2005

Ejercicio 1 (5 puntos)

Dada la siguiente especificación de una función “C”:

```
typedef struct {
    int fila;
    char columna;
} Posición;

int MovimientoValido ( Posición pinicial, Posición pfinal);
/*
    Esta función comprueba si el movimiento hecho por el jugador se sale del tablero
    (8X8). Los parámetros de entrada son:

    pinicial: posición inicial de la ficha.
    Pfinal: posición final de la ficha.
    La fila es un número entero en el intervalo [1..8]. La columna es una letra en el
    intervalo [A..H]

    La función devuelve 1 en caso de que la posición final sea válida, cualquier otro
    número si no es válida. */
```

a) (3 puntos) Crear una tabla de clases de equivalencia en la que se indiquen:

- Dato de entrada o atributo que se analiza
- Clases válidas para ese dato de entrada
- Clases no válidas para ese dato de entrada.

Dato	Clase válida	Clase no válida
Pinicial	Fila en [1..8] y Columna en [A..H]	Cualquier posición en la que la fila \notin [1..8], la columna \notin [A..H] o ambas cosas
Pfinal	Fila en [1..8] y Columna en [A..H]	Cualquier posición en la que la fila \notin [1..8], la columna \notin [A..H] o ambas cosas
MovimientoValido	1	Cualquier otro número entero

b) (2 puntos) Definir las combinaciones de las clases de equivalencia anteriores necesarias para cubrir todas las clases de equivalencia.

Salida	Entrada	
	pinicial	pfinal
1	Fila en [1..8] y Columna en [A..H]	Fila en [1..8] y Columna en [A..H]
≠1	Fila \notin [1..8], la columna \notin [A..H] o ambas cosas	Fila en [1..8] y Columna en [A..H]
≠1	Fila en [1..8] y Columna en [A..H]	Fila \notin [1..8], la columna \notin [A..H] o ambas cosas

Ejercicio 2 (2 puntos)

Como jefe de un proyecto software has instalado una aplicación para gestionar los pacientes de una clínica de Madrid. Identifica las diferencias que existirán en el contrato de mantenimiento para los siguientes casos:

- a) Se sigue una estrategia planificada que incluye actualización de versiones donde únicamente se contemplan adaptaciones y mejoras o correcciones identificadas por el equipo de mantenimiento de tu empresa.
- b) Se sigue una estrategia combinada con mantenimiento correctivo no planificado y mantenimiento perfectivo, adaptativo y correctivo planificado según la urgencia de las peticiones.

En el b) hay que:

- definir formularios de petición de cambio para los usuarios,
- establecer todos los procedimientos necesarios para la comunicación entre usuario/cliente-empresa desarrolladora (definir cómo y por qué medio se van a hacer llegar las peticiones, plazo de contestación, interlocutores válidos, etc.),
- el alcance del presupuesto (qué tipo de cambios entran en el contrato o cuál es el máximo esfuerzo) ya que interviene cliente y realiza él las peticiones,
- precio extra si las peticiones sobrepasan el alcance determinado,
- establecer un protocolo o una definición para saber cuando una petición se considera urgente.

Ejercicio 3 (3 puntos)

Contesta breve y justificadamente a las siguientes cuestiones:

- a) Un código mal estructurado y sin comentarios, pero que funciona bien, está comenzando a ser muy costoso de mantener. ¿Qué aplicarías para resolver esta situación, reestructuración o ingeniería inversa?

Las dos son posibles:

- Si tengo diseño y sólo voy a trabajar con el código → reestructuración.
- Si no tengo el diseño y lo necesito → Ingeniería Inversa y a partir del diseño, obtengo el código bien estructurado por ingeniería directa.

- b) Al aplicar ingeniería inversa en un sistema software, ¿qué relación existe habitualmente entre la completitud de la información y el nivel de abstracción de la misma?

Normalmente la completitud de la información extraída decrece a medida que el nivel de abstracción es más alto.

- c) ¿En qué puntos del ciclo de vida de un proyecto software se pueden llevar a cabo actividades para mejorar la mantenibilidad del software?

A lo largo de todo el ciclo de vida.

- d) ¿En qué momento del ciclo de vida se debe realizar el plan de gestión de configuraciones?

En la fase de planificación porque es al comienzo del proyecto donde hay que identificar los elementos de configuración y definir las líneas base del proyecto y porque es una actividad que comienza desde el inicio del desarrollo del proyecto.

- e) En un proyecto software, ¿cuándo empieza y cuando acaba la actividad de control de cambios?

Comienza al inicio del proyecto y finaliza cuando se retira el producto, es decir, dura todo el ciclo de vida del proyecto. De una manera más o menos formal pero siempre hay que controlar los cambios que se realizan en el software que, por su propia naturaleza, evoluciona continuamente.