

# Práctica 3.<sup>a</sup>

## Programación de monitores en Ada

Escuela de Ingeniería y Arquitectura  
Depto. de Informática e Ingeniería de Sistemas

### 1. Objetivos

1. Aprender a programar objetos protegidos de Ada, basados en el concepto de monitor.
2. Resolver problemas de sincronización de procesos utilizando objetos protegidos.
3. Profundizar en el modelo de concurrencia de Ada.

### 2. El problema del club de golf

En esta sección se presenta en detalle el problema objeto de esta práctica y se establecen los objetivos concretos del ejercicio a realizar. Como guía, se detallan determinadas pautas de implementación que pueden ser de utilidad.

#### 2.1. Descripción del problema

Considérese un club de golf tan exquisito que únicamente permite jugar con los palos y pelotas que él mismo alquila. La cantidad de material disponible en el club es reducida: concretamente, 20 pelotas y 20 palos de golf. Como paso previo a jugar, cada socio del club debe solicitar el alquiler del material necesario. Hay dos clases de socios: novatos y con experiencia. Los jugadores con experiencia siempre alquilan una única pelota y entre 2 y 5 palos para jugar. Un jugador novato suele alquilar un número mayor de pelotas (nunca más de 5 pelotas) y dos únicos palos.

## 2.2. Tareas a realizar

### Tarea 1

En base a la descripción previa, desarrollad un programa Ada que simule la actividad que un grupo de jugadores lleva a cabo en las instalaciones del mencionado club. Concretamente, habrá 7 jugadores con experiencia y otros 7 novatos. Cada jugador se identificará a través de un número entero comprendido entre 1 y 14, seguido de un signo ‘+’ si se trata de un jugador con experiencia y de un signo ‘-’ si se trata de un jugador novato.

Cada jugador muestra, repetidamente, el siguiente comportamiento, independientemente de su nivel de experiencia: decide jugar y reserva el material necesario, juega durante un tiempo, devuelve el material que ha usado, y, finalmente, descansa durante un rato. En ningún caso podrá comenzar a jugar si no dispone del material necesario. Del mismo modo, no iniciará su descanso hasta haber devuelto el material usado para jugar.

Para entender cuál es el funcionamiento del sistema implementado, cada jugador deberá escribir en la pantalla mensajes indicando su comportamiento:

- Antes de realizar una reserva deberá escribir un mensaje donde muestre su identidad y el número de recursos solicitados.
- Una vez haya conseguido reservar el material que necesita, escribirá un mensaje indicando su identidad, los recursos alquilados y su intención de comenzar a jugar.
- Antes de realizar una devolución deberá escribir un mensaje donde muestre su identidad y los recursos devueltos.
- Una vez completada la devolución, escribirá un mensaje indicando su identidad, los recursos devueltos y su intención de descansar.

Para dar homogeneidad al formato de los mensajes, la representación de la identidad de un jugador y el material que reserva/usa/va a devolver/ha devuelto se ajustará al siguiente formato: “*identidad*[*número de pelotas, números de palos*] *acción que realiza*”. Por ejemplo, “2+[1,3] reserva” representa que el jugador con identificador 2 (con experiencia) reserva 1 pelota y 3 palos.

### Tarea 2

Una vez desarrollado el programa anterior, en un fichero denominado «*tercera.txt*» analizad el comportamiento del sistema desde el punto de vista de la equidad, la existencia o no de esperas innecesarias, la posibilidad de bloqueos, etc. incluid en el fichero una traza de ejecución del programa. En caso de detectar problemas como los anteriores,

plantead posibles soluciones a los mismos.

### Tarea 3 (Optativa)

Desarrollad una segunda versión del programa Ada donde los jugadores se organizan de dos en dos (asumiremos que todos son novatos) para compartir tanto las pelotas como los palos de golf. El funcionamiento debe ser como sigue: cuando un jugador llega al club, espera a que haya otro; acuerdan entonces cuántas pelotas van a pedir y, mientras uno de ellos las pide, el otro pide los dos palos necesarios. Cuando ambos han obtenido lo que esperaban, van juntos al terreno de golf (es decir, el primero que acaba debe esperar al otro).

### 2.3. Pautas de implementación

Para implementar el simulador anterior, se recomienda codificar un programa denominado `simulador`, un tipo de tarea denominada `jugador` y un objeto protegido denominado `club`.

El objeto protegido `club` gestiona la concurrencia en el uso del material disponible en el club. Como mínimo, deberá encapsular dos variables privadas que representen el número de pelotas y palos de golf disponibles y procedimientos, funciones o *entries* para su reserva y devolución. El objeto protegido se podrá configurar fácilmente en cuanto a la cantidad de recursos inicialmente disponibles.

Las tareas del tipo `jugador` implementan los procesos del sistema. No es necesario definir un tipo de tarea diferente para cada tipo de jugador. Las tareas deben tener comenzar su ejecución citándose con el programa principal para que este les indique cuál es su identificador y el tipo de jugador al que pertenece. El comportamiento que exhibe un jugador (reservar material, jugar, devolver material y descansar) deberá ser repetido por cada tarea un número de veces concreto que también será indicado por el programa principal a través de la cita de inicialización. El tiempo que dedica a jugar y descansar será aleatorio (no superior a un segundo para cada actividad) y podrá ser diferente en cada ocasión.

Finalmente, el procedimiento `simulador` es el programa principal, encargado de configurar y ejecutar el sistema. Declarará los elementos que lo constituyen: jugadores, club y material disponible inicialmente. En una primera versión, configurad los 14 jugadores (7 con experiencia y 7 novatos) para que repitan su comportamiento 5 veces. Se recomienda experimentar cómo se comportaría el sistema con otras configuraciones alternativas, por ejemplo, modificando el número de cada tipo de jugadores, la cantidad inicial de material, el número de iteraciones de comportamiento de cada jugador, etc.

### 3. Entrega de material

- Como resultado de esta práctica hay que enviar por correo electrónico al profesor de prácticas ([latre@unizar.es](mailto:latre@unizar.es)) un fichero comprimido en formato ZIP denominado «pract3.zip» que contenga el código fuente solicitados y el informe «tercera.txt».
- Fecha límite de entrega: 9 de mayo de 2012.