

Simulación a Eventos Discretos

Tema 3: Programación

Programación de un modelo SED

- Veremos las nociones básicas para la programación de un modelo computacional de SED en un lenguaje imperativo en base al ejemplo del hospital simple programado en Java.
- Ver libro de Davies y O'Keefe, capítulos 1, 2 y 3. Notar que en el libro se programa en Pascal, utilizando la biblioteca Pascal SIM. En este curso programamos en modelo en Java sin utilizar bibliotecas auxiliares (salvo la de manejo de colas y listas).
- El código completo está disponible en el sitio EVA del curso. Se utilizará como referencia para esta clase y como base para algunos ejercicios de laboratorio.
- En este curso no vamos a construir otros modelos en Java.

Hospital simple en Java

Compuesto por tres archivos .java:

- `Entity.java`: Contiene la definición e implementación del tipo entidad.
- `Simulator.java`: Contiene el código del motor de la simulación (calendario, ejecutivo) y los componentes del modelo del hospital simple, incluyendo eventos, colas y recursos.
- `HospitalSimple.java`: Contiene el programa que ejecuta la simulación.

Entity.java (**ver archivo**)

- Clase que representa una entidad, en este caso a un paciente del hospital.
- Tiene atributos para identificar de forma única a la entidad, el número del evento B agendado para ocurrir con esa entidad y el tiempo en que ocurrirá ese evento (`id`, `nextBEvent`, `time`, respectivamente).
- Operación `Entity` para construir una entidad, denominada *constructor*.
- Operaciones para obtener y setear atributos de la entidad (`get`, `set`, respectivamente).

Simulator.java (ver archivo)

- Contiene el motor de la simulación y el modelo del hospital simple.
- Tiene atributos y operaciones del simulador y del modelo.
- Principales operaciones del simulador (`cause`, `init`, `run`) en su mayoría no se deben modificar si queremos construir otro modelo. Solamente hay que cambiar las inicializaciones dependientes del modelo en la operación `init` y agregar los eventos B y C específicos del modelo en la operación `run`.
- Principales operaciones del modelo. Observar que modifican el estado del sistema (cola de espera, disponibilidad de camas). La operación `cause` agenda eventos B y la entidad `current` es la que se acaba de quitar del calendario y por lo tanto es la que está ejecutando el evento en cuestión.

HospitalSimple.java (ver archivo)

- Clase que representa el programa que ejecuta la simulación.
- Crea una instancia del simulador con sus respectivos parámetros (`new Simulator`).
- Invoca la operación de inicialización (`init`).
- Ejecuta la simulación (`run`).

Compilación y ejecución

- Colocar todos los archivos `.java` en una misma carpeta.
- Compilar todos los archivos fuente haciendo `javac *.java`.
- Ejecutar el programa haciendo `java HospitalSimple`.
- Observar la salida en pantalla de la simulación (mensajes de los eventos, contenido del calendario).
- Modificar los parámetros del modelo (tiempo entre arribos, tiempo de estadía, cantidad de camas) en el archivo `HospitalSimple.java`. Compilar nuevamente, ejecutar y observar los nuevos resultados.