Práctica 5: Implementación del MVC

Fecha de entrega: 25 de mayo de 2014. 16:00

Objetivo: Uso del patrón arquitectónico Modelo-Vista-Controlador (MVC).

1. Introducción

La práctica anterior sufría de un alto acoplamiento entre el la implementación de la Máquina Virtual en sí y la manera de presentarlo al usuario. Además, no permitía la coexistencia de ambas interfaces (swing y consola) o de otras interfaces adicionales. En esta nueva versión vamos a cambiar principalmente la arquitectura de nuestra práctica, implementando un patrón Modelo Vista Controlador (MVC).

Para ver los beneficios de esta nueva arquitectura nuestra aplicación va a tener una funcionalidad adicional: La vista de Swing añade en la parte inferior una barra de estado que mostrará el estado de la máquina virtual a medida que se ejecutan las instrucciones, como se puede ver en la Figura 1.

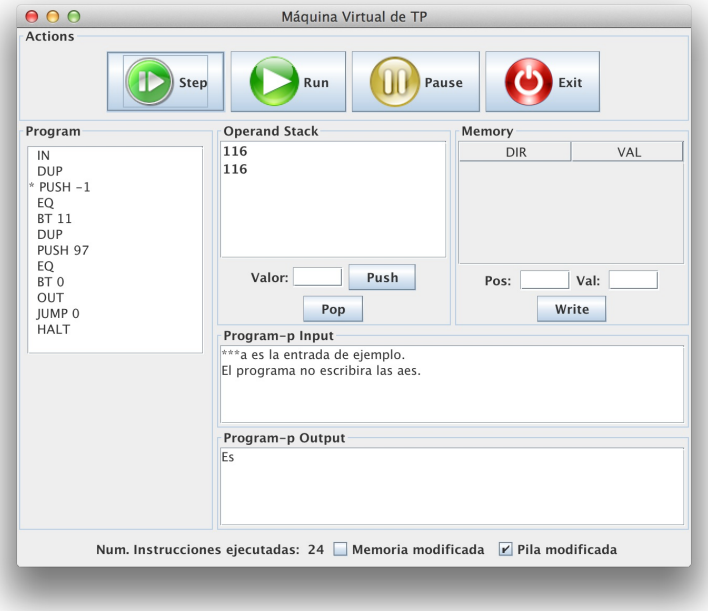
Antes de continuar os recomendamos que penséis la repercusión que tendría implementar este cambio usando la estructura actual.

La implantación del MVC más las directrices de diseño que se han explicado en clase hacen que tengamos que añadir algunos cambios adicionales a nuestra máquina virtual, que se detallarán en las próximas secciones.

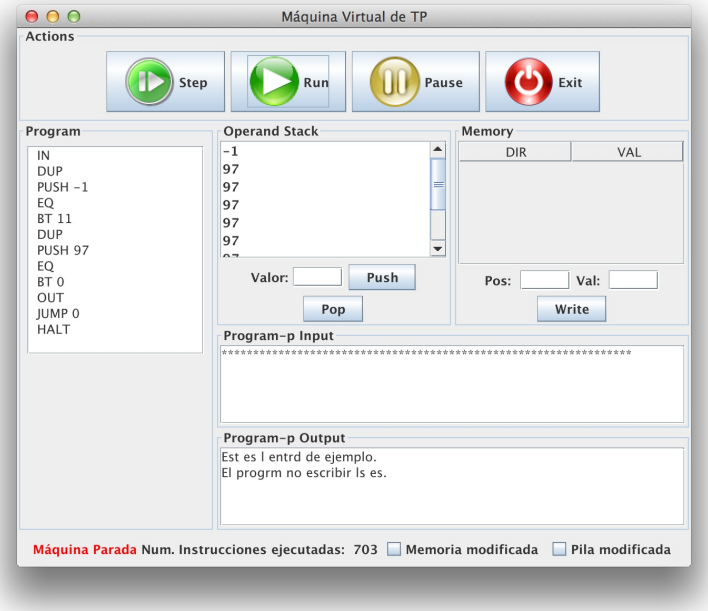
2. Modelo-Vista-Controlador

Para mejorar nuestro diseño vamos a implementar el patrón MVC. En este patrón de arquitectura se distinguen tres partes

- **Modelo**. Está compuesto por la clase que representa la máquina virtual, que será el punto de entrada para ejecutar acciones en el modelo, y el resto de clases usadas por él (las que representan la pila, la memoria, las instrucciones ...). En el modelo tendremos



(a) Tras ejecutar algunas instrucciones



(b) Tras terminar la ejecución

Figura 1: Barra de estado en varios momentos durante la ejecución de un programa

que eliminar todas la referencias a los componentes de Swing, así como todos los *System.out* que presentaban la información por consola. En su lugar, tendremos que avisar a los observadores sobre los cambios producidos durante la ejecución. El modelo puede proporcionar al menos tres observadores distintos que aglutinan los tres tipos de eventos que pueden ocurrir:

  - **Observadores de la Máquina Virtual**. Tiene los métodos que se han de implementar si queremos ser avisados de los eventos globales de la ejecución de la máquina virtual: cuando se inicializa la máquina virtual con un programa, cuando se comienza y se termina la ejecución de una instrucción correctamente, cuando se producen errores de ejecución o cuando la máquina queda parada, entre otros.
  - **Observadores de la Pila**. Eventos relacionados con el cambio de estado de la pila de la máquina virtual.
  - **Observadores de la Memoria**. Eventos relacionados con el cambio de estado de la memoria de la máquina virtual.

- **Vista**. Para cada interfaz habrá una (o varias, en el caso de swing) clases responsables de presentar el estado de la máquina virtual al usuario. Dependiendo de la clase concreta, ésta deberá ser observadora de más o menos tipos de eventos dentro de los proporcionados por el modelo.

- **Controlador**. Podemos llegar a tener tantos controladores como modos de ejecución de la máquina virtual (modo batch, interactivo y swing). Opcionalmente, podemos tener una clase abstracta `Controller` que define la funcionalidad básica común a los dos controladores. Los controladores relacionados con la ejecución de consola serán responsables de, si es necesario, leer los comandos de entrada de usuario, parsearlos y ejecutarlos sobre la máquina virtual. En el caso del controlador de Swing, éste se encarga de mandar las peticiones de ejecución de comandos que se realizan usando las vistas de Swing.

3. Parte Opcional: Pausar la ejecución de la máquina virtual

Al pulsar el botón "run" estamos bloqueando la hebra de Swing por lo que la ejecución de un programa largo "congela" la ventana de Swing: la ventana no muestra los estados intermedios por los que pasa la máquina virtual y los botones no reaccionan a nuestras acciones mientras que el programa largo se está ejecutando.

Se propone como parte opcional la implementación de un botón "run" mejorado que no bloquee la hebra de swing sino que vaya ejecutando una instrucción cada pocos milisegundos y pueda pararse con el botón "pause" que aparece en la Figura 1. Para evitar problemas de ejecución, el inicio del "run" inactiva todos los botones de la ventana salvo el botón de "Pause" y "Exit", mientras que la finalización de la ejecución restablece el estado de los botones a la situación inicial.

4. Entrega de la práctica

La práctica debe entregarse utilizando el mecanismo de entregas del campus virtual, no más tarde de la fecha indicada en la cabecera de la práctica.

Es indispensable que el código fuente supere los tests de unidad proporcionados y que el fichero enviado pase el programa validador publicado.

Sólo uno de los dos miembros del grupo debe hacerlo, subiendo al campus un fichero llamado `GrupoNN.zip`, donde `NN` representa el número de grupo con dos dígitos.

El fichero debe tener al menos el siguiente contenido[1]:

- Directorio `src` con el código de todas las clases de la práctica.
- Fichero `alumnos.txt` donde se indicará el nombre y apellidos de los componentes del grupo.

[1] Puedes incluir también opcionalmente los ficheros de información del proyecto de Eclipse