

Instrucciones

- Lee cuidadosamente el examen hasta el final. Algunas decisiones de diseño tomadas para implementar las primeras partes pueden afectar a la complejidad en la implementación de las últimas.
- En el examen deberás partir de la *práctica 5* realizada durante el curso. **En breve se indicará el procedimiento de acceso a dicho código.**
- El código entregado *debe compilar*.
- En la corrección del examen se valorará el funcionamiento, la claridad del código, el uso conveniente de los medios proporcionados por la Programación Orientada a Objetos (herencia, polimorfismo...) y comentarios. Para la evaluación se tendrá en cuenta tanto lo pedido en el examen como el código del que se parte (implementación de la práctica 5). Romper la encapsulación de las clases (acceso a atributos privados y protegidos desde clases externas, utilización de atributos públicos, etc.) implica suspender el examen.
- Incluye, además del código, una descripción de los pasos y modificaciones que has realizado para implementar lo que se pide en el examen. Este informe de cambios puedes hacerlo en papel o adjuntarlo como un fichero de texto con el resto de los archivos de código fuente. Adjunta este fichero de texto `descripcion.txt` con el resto de los archivos de código fuente. Deja el fichero en “el raíz” del .zip entregado.
- La puntuación de cada apartado está calculada *sobre 10 puntos*, aunque supondrá el 40% de la nota final.

Consejos

Es preferible realizar el examen en el orden siguiente:

1. Implementar cada uno de los apartados. Tras la implementación de cada parte, comprobar que el código compila y funciona y guardar una copia del mismo. De este modo siempre tendremos algo que entregar que sabemos que compila y que funciona (aunque no esté completo).
2. Revisar y corregir los posibles errores o deficiencias que se arrastren de la práctica base, pues ese código también se evalúa.

Instrucciones de entrega

- Para entregar la solución al examen, crea un fichero zip. En él debes incluir todo el proyecto una vez limpiado de archivos intermedios, y con el fichero `alumnos.txt` con tu nombre completo.
- Nombra al fichero `NN_Apellido1Apellido2.zip`, donde NN indica el número de grupo (con dos dígitos).
- La estructura del fichero será la misma que la utilizada durante el curso.
- Si optas por escribir la descripción de los pasos y modificaciones realizadas durante el examen en un fichero de texto, inclúyelo en el raíz del archivo comprimido.
- Para entregar el examen, se utilizará el mecanismo de entregas disponible en el laboratorio. En particular, cerca del final del examen se habilitará la unidad U: en la que deberás dejar la solución al examen. Si deseas entregarlo antes, indícaselo al profesor para que habilite la unidad.
- Antes de abandonar el laboratorio debes pasar por el puesto del profesor para asegurarte de que lo que se ve en el puesto del profesor es lo que has entregado y firmar en la hoja de entregas.

Enunciado

Vamos a ampliar la funcionalidad de la **Práctica 5** incluyendo un banco de registros a nuestra CPU. El banco de registros se compone de 4 registros (nombrados R0, R1, R2 y R3), cada uno de ellos con capacidad para un entero (como las celdas de memoria o la pila).

Para poder almacenar información en los registros vamos a añadir a nuestra CPU la instrucción **MOVE Rx**, que guarda el valor de la cima de la pila en el registro **Rx** del banco de registros.

Así mismo, vamos a crear una nueva instrucción que hace uso del banco de registros y que nos facilita la creación de bucles: la instrucción **LOOPDEC Rx n**. Esta instrucción primero decrementa el valor del registro **Rx** y después comprueba su valor: si es mayor que cero, salta (cambia el PC) para ejecutar la instrucción **n**; si es cero entonces no hace nada más y la ejecución continuará con la instrucción en la posición **PC+1**.

Por último modificaremos al interfaz de Swing para que se pueda mostrar el estado del banco de registros.

[2 puntos] Parte A

Crea una clase **BancoRegistros** con todos los métodos necesarios para poder interactuar con ella. Inicialmente todos los registros tienen valor 0. Integra el banco de registros con la CPU.

Se valorará positivamente la creación de un banco de registros que sea extensible en cuanto al número de registros que almacena.

[1,5 puntos] Parte B

Implementa la instrucción **MOVE Rx** y añádela al repertorio de instrucciones de la CPU. El parseo de la instrucción falla si el nombre del registro no es válido. Al ejecutar la instrucción se guardará el valor de la cima de la pila en el registro **Rx**, produciendo un error en caso de que la pila esté vacía.

[1,5 puntos] Parte C

Haz que en el modo Interactivo podamos ver el estado del banco de registros junto con el estado de la CPU. A modo de ejemplo, este sería el estado de la CPU tras haber realizado algunas instrucciones que hacen uso de los registros.

El estado de la máquina tras ejecutar la instrucción es:

Memoria: [0]:0 [1]:1

Pila de operandos: <vacía>

Registros: [R0]: 10 [R1]: 0 [R2]: 0 [R3]: 5

[2 punto] Parte D

Implementa la instrucción **LOOPDEC Rx n** que hace uso del registro **Rx** para decidir si salta o no a la instrucción **n** tal y como se explicó anteriormente. El parseo de la instrucción falla si el nombre del registro no es válido. La ejecución de la instrucción ha de fallar si el registro tiene un valor negativo después de decrementar el valor del mismo (evitando de esta forma la ejecución de bucles infinitos). En este caso, deberá mostrar el siguiente mensaje de error si lo estamos ejecutando en modo Batch o Interactivo:

Error ejecutando LOOPDEC R0 0: Valor negativo en registro R0

[3 puntos] Parte E

Haz las modificaciones necesarias para hacer que el banco de registros se muestre en la interfaz de Swing. Como se puede ver en la Figura, el banco de registros aparece en un panel en la parte superior de la interfaz.

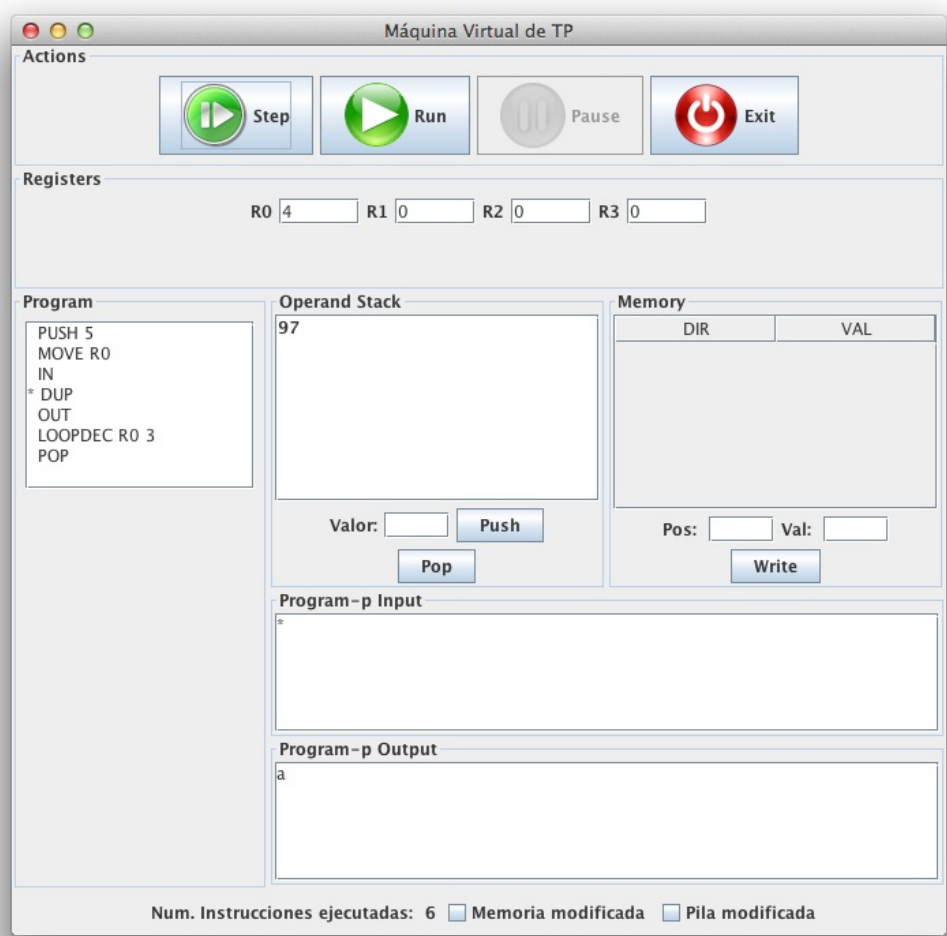


Figure 1: El aspecto de la interfaz