

Trabajo Práctico 3

Programación Orientada a Objetos

Paradigmas de Lenguajes de Programación – 1er cuat. 2018

Fecha de entrega: 26 de Junio de 2018

Objetivo

En este Trabajo Práctico vamos a continuar con las expresiones regulares que presentamos en el TP anterior. La idea es explorar las características que nos brinda un lenguaje de objetos como lo es *Smalltalk* para modelar este dominio.

Recordatorio

La sintaxis de las expresiones regulares está definida según la siguiente gramática:

- (1) $E ::=$
- (2) $E ::= \text{symbol}$
- (3) $E ::= E E$
- (4) $E ::= E \mid E$
- (5) $E ::= E *$

La regla (1) permite aceptar la cadena vacía ($""$). La regla (2) permite aceptar un símbolo, el cual puede ser cualquier carácter (letras, dígitos, etc.). La regla (3) es la concatenación de dos expresiones regulares. La regla (4) permite aceptar alguna de las dos expresiones regulares pero no necesariamente ambas. La regla (5) permite aceptar cero o más veces una expresión regular.

1. Implementación

Ejercicio 1. Implementar las clases y métodos (de clase y/o instancia) necesarios para modelar las expresiones regulares y pasar los tests de la clase `TestEj1`. Los mensajes binarios `|` y `,` representan la unión y concatenación de las expresiones regulares respectivamente. Si bien los tests sólo utilizan los símbolos `a` y `b`, la implementación deberá poder soportar cualquier otra letra como símbolo posible.

Ejercicio 2. Implementar los siguientes métodos para que pasen los tests de la clase `TestEj2`:

- `hasStar` que devuelve `true` si la expresión regular receptora del mensaje contiene una estrella, o `false` en caso contrario.
- `printString` que devuelve la representación en texto de la expresión regular receptora del mensaje.

Ejercicio 3. Implementar el método `length:` que recibe un bloque como parámetro el cual indica el criterio utilizado para calcular la longitud de la expresión regular receptora del mensaje (por ejemplo, si queremos calcular la longitud máxima o mínima). La implementación debe poder pasar los tests de la clase `TestEj3`.

Ejercicio 4. Implementar el método `match:` que recibe una cadena como parámetro e indica si la expresión regular receptora del mensaje la acepta o no. La implementación debe poder pasar los tests de la clase `TestEj4`.

Pautas de entrega

El entregable debe contener:

- un archivo `.st` con todas las clases implementadas
- versión impresa del código, comentado adecuadamente (puede ser el propio `.st` sin los tests)
- **No** hace falta entregar un informe sobre el trabajo.

Se espera que el diseño presentado tenga en cuenta los siguientes factores:

- definición adecuada de clases y subclases (si corresponde), con responsabilidades bien distribuidas
- uso de polimorfismo y/o delegación para evitar exceso de condicionales (no usar más de un `ifTrue:`, `ifFalse:` o `ifTrue:ifFalse:`; sí se pueden usar libremente mensajes como `ifNil:`, `ifEmpty:`, `ifNotEmpty:`, `detect:ifNone:`, `detect:ifFound:`, `whileTrue:`, etc.)
- intento de evitar código repetido utilizando las abstracciones que correspondan.

Consulten todo lo que sea necesario.

Consejos y sugerencias generales

- Lean al menos el primer capítulo de *Pharo by example*, en donde se hace una presentación del entorno de desarrollo.
- Explorar la imagen de Pharo suele ser la mejor forma de encontrar lo que uno quiere hacer. En particular tengan en cuenta el buscador (**shift+enter**) para ubicar tanto métodos como clases.
- No se pueden modificar los test entregados, si los hubiere, aunque los instamos a definir todos los tests propios que crean convenientes.

Importación y exportación de paquetes

En Pharo se puede importar un paquete arrastrando el archivo del paquete hacia el intérprete y seleccionando la opción “**FileIn entire file**”. Otra forma de hacerlo es desde el “**File Browser**” (botón derecho en el intérprete > **Tools** > **File Browser**, buscar el directorio, botón derecho en el nombre del archivo y elegir “**FileIn entire file**”).

Para exportar un paquete, abrir el “**System Browser**”, seleccionar el paquete deseado en el primer panel, hacer click con el botón derecho y elegir la opción “**FileOut**”. El paquete exportado se guardará en el directorio **Contents/Resources** de la instalación de Pharo (o en donde está la imagen actualmente en uso).