

Taller de Objetos

Paradigmas de Programación
2do Cuatrimestre 2018

Fecha de entrega: 25 de Octubre del 2018

Introducción

En este trabajo modelaremos listas como objetos.

Ejercicios a desarrollar

Ejercicio 1

Definir el objeto `nil` que puede responder a los mensajes:

- `nil`: retorna verdadero
- `cons(o)`, que retorna una lista no vacía cuya cabeza es el objeto `o` y como cola tiene al objeto receptor. Notar que la nueva lista debe poder responder adecuadamente a `nil` y `cons(o)` y a las operaciones `head` and `tail`.

Definir el objeto `lista12` que representa a la lista `[1, 2]`.

Ejercicio 2

Se desea redefinir a la operación `toString` para los objetos de tipo lista, que retorne una string mostrando el contenido de la lista. Por ejemplo, `nil.toString()` evalúa `"[]"`, y `lista12.toString()` evalúa `"[1:[2:[]]]"`, y

Ejercicio 3

Extender las definiciones anteriores de manera tal que las listas provean el método `length` que permite calcular la longitud de lista receptora. Por ejemplo `nil.length()` evalúa `0`, y `lista12.tail().length()` evalúa `1`.

Ejercicio 4

Extender la definición anterior de manera tal que los objetos listas puedan responder al mensaje `copy()` que retorna una nueva lista conteniendo todos los elementos de la lista original. La definición de la operación debe ser tal que

`nil.copy().toString()` evalúa `"[]"`

```
nil.copy() == nil evalúa false
lista12.copy().toString() evalúa "[1:[2:[]]]"
lista12.copy() == lista12 evalúa false
lista12.copy().tail == lista12.tail evalúa false
```

Ejercicio 5

Para listas muy largas se desea evitar recalcularse su longitud reiteradas veces. Suponer que se cuenta con el objeto `listaMuyLarga` que fue construido a partir del objeto `nil` definido en los incisos anteriores. Se desea modificar la definición del método `length` del objeto `listaMuyLarga` (y recursivamente para `listaMuyLarga.tail`) de manera tal que no se deba recalcularse su longitud cada vez que reciba el mensaje `length`. Notar que el comportamiento de `nil` y de los restantes objetos creados anteriormente no deben verse afectados. El nuevo comportamiento de `length` debe ser tal que la primera vez que se invoque a la operación se calculará su longitud y el resultado será utilizado para responder a los siguientes mensajes `length`. Por ejemplo, al evaluar `listaMuyLarga.length()` por primera vez, se evaluará `listaMuyLarga.tail().length()`. Sin embargo, las evaluaciones subsiguientes de `listaMuyLarga.length()` no evaluará `listaMuyLarga.tail().length()`.

Se desea además que al evaluar el siguiente código

```
listaMuyLarga.length()
let a = listaMuyLarga.tail()
a.length()
```

La evaluación `a.length()` no requiera computar nuevamente la longitud de la lista.

Ejercicio 6

Definir la operación `concatenar(l)`, que permite obtener una nueva lista que consiste en la concatenación de la lista original y `l`. El objeto receptor y el parámetro `l` pueden ser listas cortas o largas. La operación `length` de la lista resultante debe comportarse como en el receptor.

Pautas de Entrega

Se debe entregar el código impreso con la implementación de los métodos pedidos. Cada método asociado a los ejercicios debe contar con ejemplos que muestren que exhibe la funcionalidad solicitada. Además, se debe enviar un e-mail conteniendo el código fuente en Javascript a la dirección plp-docentes@dc.uba.ar. Dicho mail debe cumplir con el siguiente formato:

- El título debe ser [PLP;Taller-P0] seguido inmediatamente del nombre del grupo.
- El código Javascript debe acompañar el e-mail y lo debe hacer en forma de archivo adjunto (puede adjuntarse un `.zip` o `.tar.gz`).

No es necesario entregar un informe sobre el trabajo, alcanza con que el código esté **adecuadamente** comentado (son comentarios adecuados los que ayudan a entender lo que no es evidente o explican decisiones tomadas; no son adecuadas las traducciones al castellano del código). Los objetivos a evaluar son:

- Corrección.
- Declaratividad.
- Prolijidad: evitar repetir código innecesariamente y usar adecuadamente los métodos previamente definidos

Importante: se admitirá un único envío, sin excepción alguna. Por favor planifiquen el trabajo para llegar a tiempo con la entrega. No se aceptaran envíos de personas sin grupo.