Taller de Prototipado

Verano 2019

Fecha límite entrega: 27 de Febrero del 2019 hasta las 16:00

1. Introducción

Este taller consiste en implementar polinomios en Javascript. Para esto, definiremos primero a los monomios y a las sumatorias de monomios en una variable. Un *monomio* es una expresión que contiene un coeficiente, una parte literal, y un grado, por ejemplo $3x^8$, donde 3 es el coeficiente, x es la parte literal y 8 es el grado. El grado de un polinomio es siempre un entero no negativo. Además, asumiremos en este trabajo que la parte literal es siempre x. Por lo tanto representaremos a un monomio por su coeficiente, que será un número real, y un grado. Una *sumatoria* de monomios se define recursivamente como:

- Un *monomio*
- Una suma (+) de dos *sumatorias*.

Por ejemplo $(7x^2 + 8x) + (-3x + 0)$ es una sumatoria (notar que se han omitido los exponentes 1 y los términos x^0.

A continuación utilizaremos como ejemplos a los siguientes polinomios.

$$\mathsf{s}_1 = (7x^2 + 8x) + (-3x + 4)$$
$$\mathsf{s}_2 = x^4 + (-x^4 + x^3)$$

2. Implementación

Ejercicio 1

Proveer los objetos y/o constructores necesarios que permitan generar *sumatorias*.

Ejercicio 2

Definir la operación evaluar sumatorias que, dado un entero, retorne el resultado de evaluar la sumatoria con el valor dado. Por ejemplo, s_1.evaluar(1) retorna 16.

Ejercicio 3

Definir la operación gradoMayor que retorne el grado del/de los monomios de mayor grado en la suma. Por ejemplo, s_1.gradoMayor() retorna 2 y s_2.gradoMayor() retorna 4.

Útil: `Math.Max`.

Ejercicio 4

Definir la operación coefDeGrado que dado un entero g retorne la suma de los coeficientes de todos los monomios de grado g. Por ejemplo, s_1.coefDeGrado(1) retorna 5, s_1.coefDeGrado(5) retorna 0, y s_2.coefDeGrado(4) retorna 0.

Ejercicio 5

Definir grado() que retorna el máximo n tal que coefDeGrado(n) es distinto de 0. Por ejemplo, s_1.grado() retorna 2, y y s_2.grado() retorna 3.

Ejercicio 6

Definir toString que retorna una string que representa la sumatoria en forma polinómica, es decir, la suma de monomios de distinto grado ordenados decrecientemente por su grado. Por ejemplo, s_1.toString() retorna "$7x^2 + 5x + 4$" y s_2.toString() retorna "x^3".

Ejercicio 7

Definir `aPolinomio` que retorna una sumatoria que representa al receptor en su forma polinómica. Por ejemplo, s_1.aPolinomio() retorna el objeto sumatoria que representa a $7x^2 + (5x + 4)$.

Ejercicio 8

Modificar su solución de forma tal que la invocación a las operaciones grado, toString y aPolinomio sobre formas polinómicas se resuelvan de manera más eficiente que sobre una sumatoria.

3. Pautas de evaluación y entrega

Los objetivos a evaluar con este taller son:

- Corrección.
- Declaratividad.
- Prolijidad. Se deberá evitar repetir código innecesariamente y usar adecuadamente las funciones previamente definidas (tener en cuenta tanto las funciones definidas en el enunciado como las definidas por ustedes mismos).
- Uso adecuado de prototipos y funciones constructoras.

Recomendamos encarecidamente que aprovechen las clases destinadas al taller para intentar terminar los ejercicios. Aquellos grupos que no cuenten con las correcciones al finalizar la clase del 25 de febrero, deberán entregar el código antes de la fecha límite de entrega utilizando el Campus Virtual y presentando su versión impresa (**obligatoria)**.

Para hacerlo, *uno* de los dos integrantes del grupo deberá ingresar a la tarea "Taller de programación orientada a objetos" en la sección "Talleres" de la página del Campus. Allí deberá subir un archivo `.zip` o `.tar.gz` que contenga el código javacript correspondiente. En el comentario que acompaña a la entrega, deberá especificar nombre y apellido de su compañero de grupo.