

Problema: La calculadora de números grandes

Algunos lenguajes de programación tienen limitaciones para representar números grandes, esto es que tengan más dígitos que 2^{32} o 2^{64} . Una solución a este problema es utilizar estructuras de datos para representar números sin límite en la cantidad de dígitos, las bien conocidas librerías de números de alta precisión. En este ejercicio los estudiantes realizarán la especificación e implementación de una calculadora para este tipo de números.

Requerimientos del programa

R1: permitir el uso de enteros de muchos dígitos representados usando una estructura de datos que llamarán BigInt.

R2: sumar números representados con BigInt.

R3: restar números representados con BigInt.

R4: multiplicar números representados con BigInt.

R5: dividir (división entera) números representados con BigInt.

R6: retornar el módulo números representados con BigInt

R7: limpiar la entrada de la calculadora para ingresar una nueva operación

R8: Permitir el ingreso de varias operaciones utilizando paréntesis para evitar ambigüedades y realizar la operación solo cuando ocurra un evento (botón)

R89 regresar sobre las entradas para editar la operación antes de calcular el valor de una operación

R10: guardar el último resultado

R10: informar al usuario que ha escrito mal una operación y mostrar una sugerencia de cómo corregirla

Condiciones de la entrega:

1. Cada grupo debe diseñar e implementar una interfaz gráfica de usuario para la calculadora.
2. De ninguna manera es permitido transformar un BigInt a entero para operar con números, todas las operaciones deben implementarse con BigInts. Tampoco se puede utilizar BigInteger.
3. El manejo de la calculadora debe realizarse utilizando estructuras de datos como pilas, colas o listas.
4. Todas las estructuras de datos utilizadas (excepto arreglos) deben ser construidas por los estudiantes. Cada estructura debe tener los documentos de diseño respectivos.
5. Debe crear clases y los métodos de prueba utilizando JUnit.
6. El diseño se debe hacer MINIMIZANDO el tiempo de ejecución de todas las operaciones y el espacio que ocupan las estructuras de datos, garantizando que la aplicación tanto con números grandes, y pequeños.

7. La entrega debe contener un documento en donde se justifiquen las decisiones de diseño tomadas, se debe explicar claramente por que se escogieron las estructuras de datos utilizadas, tomando como base para la justificación la complejidad temporal y de memoria.

Los 3 mejores ejercicios tienen una bonificación en la nota del parcial.

Criterios para escoger los mejores ejercicios:

- Proyecto que permite evidenciar la funcionalidad de la calculadora con la mayor cantidad de dígitos. Si hay empate entonces se tomará en cuenta la mayor cantidad de operaciones.
- mejor diseño de la interfaz.