[ED185] TAD Números Arbitrariamente Grandes (BigNumber)

Neste problema deverá apenas submeter uma classe **BigNumber** (e não um programa completo).

O problema

A sua tarefa é criar uma classe **BigNumber**, que implementa um número arbitrariamente grande (com tantos dígitos quanto o necessário). A classe deverá suportar quaisquer números inteiros positivos, e deverá implementar os seguintes métodos:

- **Construtores:**
  - **BigNumber(String n)** - Inicializa o número a partir da String *n* (supondo que a String só contém dígitos e não tem zeros à esquerda)
- **Métodos padrão:**
  - **public boolean equals(BigNumber n)** - devolve *true* caso o número seja igual a n, ou *false* caso contrário
  - **public String toString()** - devolve uma String representando o número (os dígitos em si)
- **Operações:**
  - **public BigNumber add(BigNumber n)** - devolve um novo BigNumber igual à soma de *n* com o próprio objecto (*this* + *n*).
  - **public BigNumber multiply(BigNumber n)** - devolve um novo BigNumber igual à multiplicação de *n* com o próprio objecto (*this* * *n*).

Um exemplo de utilização seria:

```
class TestBigNumber {
   public static void main(String[] args) {
      BigNumber n1 = new BigNumber("1234567890");
      System.out.println(n1); // Escreve "1234567890"

      BigNumber n2 = new BigNumber("42");
      BigNumber n3 = new BigNumber("1234567890");
      System.out.println(n1.equals(n2)); // Escreve "false"
      System.out.println(n1.equals(n3)); // Escreve "true"

      BigNumber n4 = new BigNumber("46711237126582920746212");
      BigNumber n5 = new BigNumber("8765432110");
      BigNumber n6 = n1.add(n3);
      System.out.println(n6); // Escreve "2469135780"
      BigNumber n7 = n1.add(n4);
      System.out.println(n7); // Escreve "4671123712658415531402"
      BigNumber n8 = n1.add(n5);
      System.out.println(n8); // Escreve "10000000000"

      BigNumber n9 = n1.multiply(n3);
      System.out.println(n9); // Escreve "1524157875019052100"
      BigNumber n10 = n1.multiply(n4);
      System.out.println(n10); // Escreve "57668193458655139375688174332680"
   }
}
```

Input e Output

Deverá apenas submeter a classe **BigNumber**. O Mooshak irá chamar criar várias instâncias da sua classe usando o construtores definido e irá fazer uma série de testes aos métodos por si implementados (como mostrado no exemplo de utilização).

É garantido que o números guardados serão sempre positivos e que nunca terão mais do que 1000 dígitos.

É garantido que os métodos são chamados de forma correcta (os argumentos fazem sentido e não geram excepções).
