

Progettare e implementare una classe `Matrix` per rappresentare matrici  $m \times n$  (con  $m$  righe e  $n$  colonne) di numeri interi. Dotare la classe di:

- metodi `get` e `set` che permettano di leggere/scrivere un elemento della matrice date le sue coordinate. Ad esempio, `m.get(0, 0)` deve restituire l'elemento che si trova nell'angolo in alto a sinistra della matrice  $m$ ;
- metodi `rows` e `columns` che ritornino rispettivamente il numero di righe e di colonne della matrice;
- un metodo `add` tale che `m1.add(m2)` ritorni la matrice ottenuta sommando  $m_1$  ed  $m_2$ ;
- un metodo `mul` tale che `m1.mul(m2)` ritorni la matrice ottenuta moltiplicando  $m_1$  con  $m_2$ ;<sup>1</sup>
- un metodo `pow` tale che `m.pow(n)` ritorni la potenza  $n$ -esima della matrice quadrata  $m$ . Si ricorda che la potenza 0-esima di  $m$  è la matrice identità che ha lo stesso lato di  $m$ .<sup>2</sup>

Specificare le pre-condizioni di ogni costruttore/metodo con una clausola `assert` opportuna.

---

<sup>1</sup> <https://www.andreaminini.org/matematica/algebra-lineare/moltiplicazione-tra-matrici>

<sup>2</sup> <https://www.andreaminini.org/matematica/algebra-lineare/matrice-identita>