deep se
dependable evolvable pervasive software engineering group

Exercise Session – Matrix Computation

Federica Filippini

Politecnico di Milano

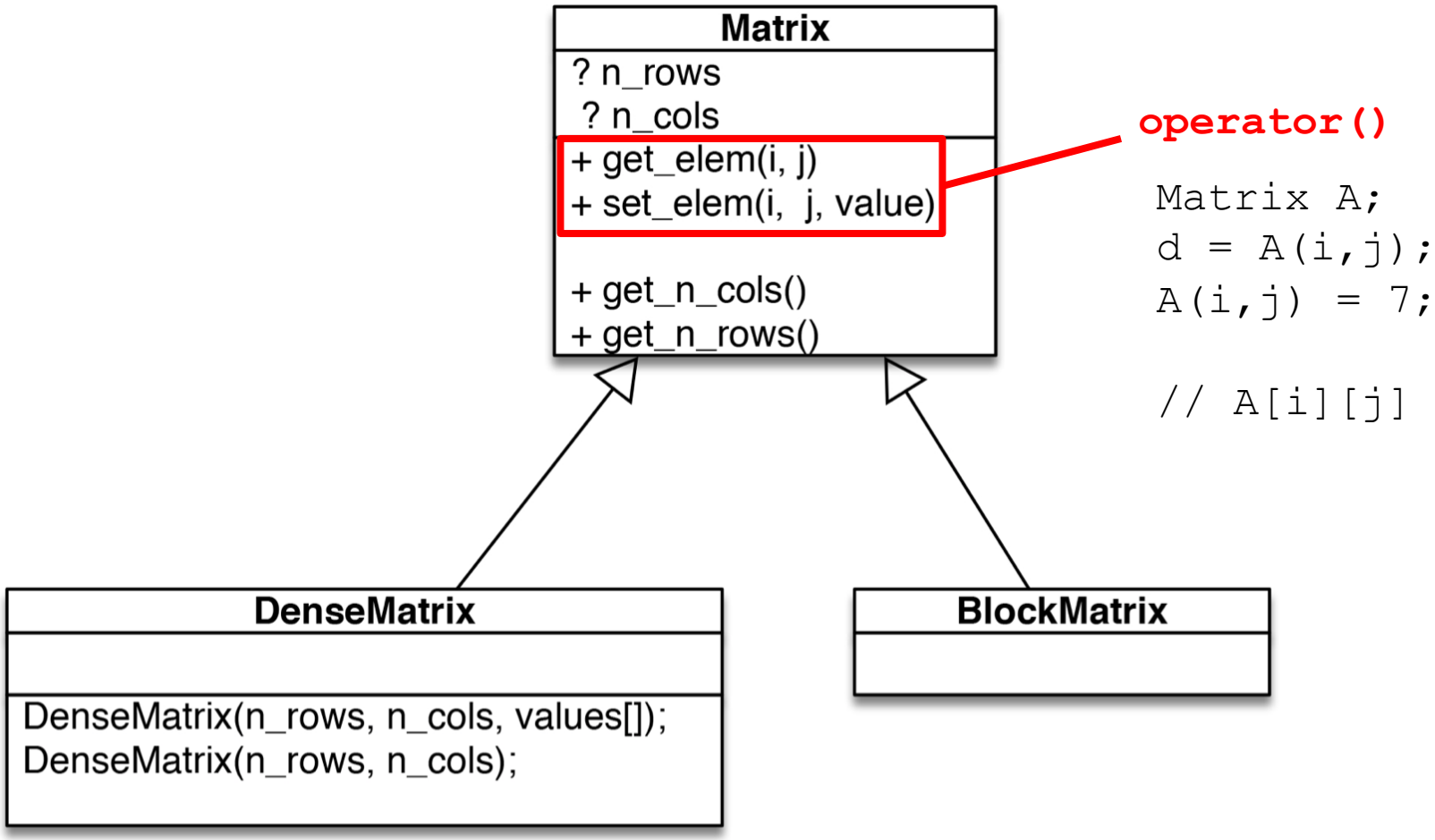
federica.filippini@polimi.it

POLITECNICO DI MILANO

Goal

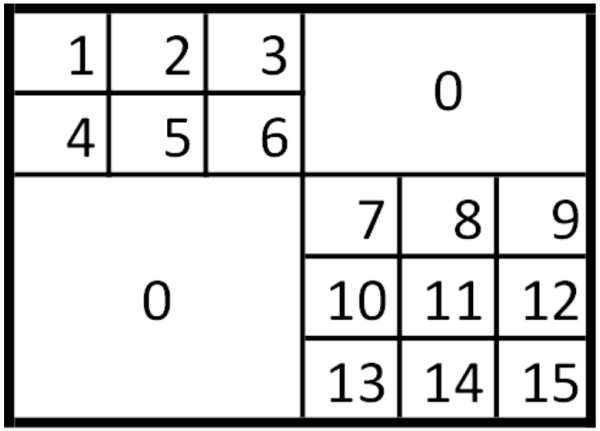
- Implement a program for **matrix computation**, that manages both **dense and block matrices** of `double` precision numbers.
- **Dense matrices** are allocated when created.
- **Block matrices** are ideal to store **sparse** data and their size and memory storage are **changed dynamically** when blocks are added.

Class hierarchy



`Block` class

- It includes a `DenseMatrix`
- It is characterized by the indexes of its top left and bottom right elements



- The values of a new block are provided per row in a single vector `vals` (e.g., {1,2,3,4,5,6}).

Required methods

- **`add_block()`**, which receives a new block as parameter. The method should also update the matrix size accordingly.
- get-like implementation of **`operator`**`()`, which
  - receives as parameters the row and column indexes of the element to be read and returns its value,
  - prints an error message if the indexes are out of range,
  - returns 0 if the indexes are within the range, but for an element not explicitly initialized in a block.
- set-like implementation of **`operator`**`()`, which
  - receives as parameters the row and column indexes of an element and returns a reference to the element in the matrix,
  - adds a new block with a single element if the element was not previously allocated in a block.