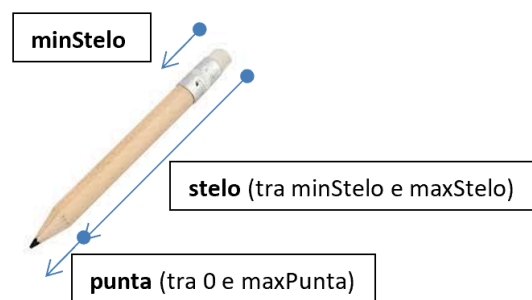


La classe Matita. Vi chiediamo di scrivere una classe pubblica `Matita` per rappresentare virtualmente matite. Una matita è definita come uno stelo (una lunghezza intera in millimetri, da un minimo **`minStelo`** a un massimo **`maxStelo`**) seguita da una punta (un intero da 0 a un massimo **`maxPunta`**).

Fissate nella definizione della classe dei valori per massimi e minimi, per esempio: **`minStelo=10`**, **`maxStelo=200`**, **`maxPunta=5`**. Il prossimo disegno riassume il significato appena visto dei diversi attributi.



(i) **`minStelo`**, **`maxStelo`**, **`maxPunta`** sono attributi interi **pubblici**, **statici** e **final** della classe `Matita` (non legati a un oggetto ma alla classe). Invece `stelo` e `punta` sono attributi interi **dinamici**.

(ii) Il costruttore di **`Matita`** consente di costruire una matita con punta di lunghezza massima dato lo `stelo`. Un assert impedisce lunghezze non accettabili dello `stelo`.

(iii) La classe ha i metodi **`get`** per `stelo` e `punta` e nessun metodo **`set`**: non consento di cambiare la lunghezza a una matita.

(iv) Un metodo "disegna" restituisce "true" (successo) se la matita ha almeno 1mm di punta, e "false" (fallimento) altrimenti. Nel primo caso usa la matita fino a ridurre la punta di un 1mm.

(v) Un metodo "tempera" restituisce "true" (successo) se la matita è più lunga del minimo e "false" (fallimento) altrimenti. Nel primo caso riduce lo `stelo` di 1mm e allunga la punta di 1mm, a meno che la lunghezza della punta sia già il massimo. In questo caso la matita si accorcia ma la punta resta invariata.

Scrivete **`Matita.maxStelo`** per richiamare il massimo dello `stelo` (attributo statico). Scrivete **`Math.min`** per richiamare il metodo statico `min(x,y)` della classe `Math`, che calcola il

minimo. Includete una classe TestMatita per sperimentare la classe Matita: eseguirla e controllate che i risultati siano sensati.