

Pràctica Programació: BENDER

Per un dispositiu mòbil, la capacitat per navegar a través del seu entorn és molt important. Ens referim a que ha de poder localitzar la seva posició dins un mapa i ha d'ésser capaç de planificar un camí cap a l'objectiu.

Suposarem que el robot disposa d'un localitzador que en tot moment ens indica les coordenades de la seva posició. Aiximateix, el sistema té unes regles ben definides:

1. El robot comença a la posició del mapa assenyalada amb el caràcter «X».
2. El robot sempre va inicialment cap al sud del mapa.
3. Les parets estan representades pel caràcter «#».
4. Quan el robot troba un obstacle, aleshores intenta canviar la seva direcció, en aquest ordre: S (South), E (East), N (North), W (West).
5. El robot finalitza el seu camí en arribar al destí, assenyalat amb el caràcter «\$».
6. En el mapa hi ha també dispositius «inversors», assenyalats amb el caràcter «I». Quan el robot troba un inversor, canvia la prioritat de les seves direccions: N (North), W (West), S (South), E (East).
7. Opcionalment, us podeu trobar dispositius teleportadors, assenyalats amb el caràcter «T».

Quan el robot entra en un teleportador, surt per l'altre (més pròxim), conservant la direcció i el sentit del seu moviment. El robot es mou només una casella cada cop: cap a dalt (N), cap abaix (S), cap a l'esquerre (W) o cap a la dreta (E). Exemples:

```
#####  
#  X  #  
#      #  
#      #  
#      #  
#      #  
#  $  #  
#      #  
#####
```

Camí del robot: SSSS

```
#####
#  X  #
#      #
#      #
#      #
#      #
#      #
#   $  #
#####
```

Camí del robot: SSSSSEE

La vostra tasca

En honor a la gran sèrie «Futurama», anomenarem el nostre robot «Bender».

Heu d'implementar una classe en Java anomenada «Bender», que implementa el sistema de navegació esmentat abans.

Aquí teniu l'esquelet:

```
class Bender {
    // Constructor: ens passen el mapa en forma d'String
    public Bender(String mapa) { }
    // Navegar fins a l'objectiu («$»).
    // El valor retornat pel mètode consisteix en una cadena de
    // caràcters on cada lletra pot tenir
    // els valors «S», «N», «W» o «E»,
    // segons la posició del robot a cada moment.
    public String run() { return ""; }
}
```

Penseu que el mapa s'enviarà com una cadena de caràcters. Cada línia representa un troç del mapa. Us convindrà tenir un array bidimensional intern per representar les parets i els distints objectes.