

# Java Teszt

## Készítsen egyszerű konzolos alkalmazást az alábbi specifikáció alapján:

- Adva van egy paraméterben állítható méretű két dimenziós aréna.
- Az arénában egyszerre két robot harcolhat.
- A robotok java osztályok melyek egy közös interfészt valósítanak meg.
- A robot osztályokat dinamikusan lehet betölteni.
- A betöltendő osztályokat parancssor paraméterben kapja meg az alkalmazás.
- Egy robot cselekvései körökre vannak osztva – minden robot az adott körben egyszerre hajtja végre a cselekedetét.
- Minden robot le tudja kérdezni a következőket:
  - saját pozíció
  - ellenfél pozíció
  - aréna mérete
  - saját páncél
  - ellenfél páncél
- A robot a következő cselekvésekre képes:
  - lépés adott irányba
  - támadás adott irányba (1 pontot levon az ellenfél páncél pontjaiból)
  - védekezés adott irányba (semlegesíti a pont levonást)
  - várakozás
- Az rendszert úgy kell megírni, hogy:
  - a robotok egy körben csak egy cselekvést tudjanak elvégezni. (pl. Ha már lépett, nem támadhat és nem is léphet újra, ha már támadott nem védekezhetsz, támadhat vagy léphet, stb....-)
  - a robotok ne tudják elhagyni az arénát. Ha egy lépés kívül esne az arénán, akkor az a lépés elveszett és a robot nem mozdul semerre.
  - a cselekvések párhuzamosak, ha egyszerre támadnak – egyszerre sérülnek, ha egyszerre lépnek ugyanoda – egyikük lépése sem érvényesül.
- Az egy kör - egy cselekvés szabályt úgy kell implementálni, hogy a leszármazott osztályokban csalásra ne legyen lehetőség. (Reflection-t nem kell figyelembe venni)
- A robotok paraméterben megadott páncél ponttal rendelkeznek, ha ez elfogy, a küzdelemnek vége és a talpon maradt robot nyer.
- A meccs paraméterben megadott körig tart, ha mindkét robot talpon van még, akkor pontozással a kevésbé sérült robot nyer.
- Minden más esetben a végeredmény döntetlen.

Példák a konzol képernyőre:

### Példa 1:

```
4. kör
#####
#.....#
#..A....#
#.....#
#.....#
#.....#
#.....B..#
#.....#
#.....#
#####

"A" robot: Wolf.class
Páncél: 8/10

"B" robot: Fox.class
Páncél: 10/10
```

### Példa 2:

```
#####  
#_|_|_|_|_|_|_|_|_|_|_|_#  
#_|_|A|_|_|_|_|_|_|_|_|_|_|_# "A" robot: Wolf.class  
#_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_# Páncél: 8/10  
#_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_#  
#_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_#  
#_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_# "B" robot: Fox.class  
#_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_# Páncél: 10/10  
#_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_#  
#_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_#  
#####
```

## Alternatív konzol kimenet

## Lanterna - Java library for creating text-based GUIs

Ezzel a könyvtárral lehetőség nyílik a konzol teljes kihasználására, animált képernyőkre, billentyű leütések kezelésére, stb. Használata teljesen opcionális.

- <https://github.com/mabe02/lanterna>
- <https://github.com/mabe02/lanterna/tree/master/src/test/java/com/googlecode/lanterna/tutorial>