Tic-tac-toe

XOX

OXO

OXO

A Tic-tac-toe a háromszor hármas táblán játszott amőba játék. Két játékos felváltva rak, az egyik O-kat, a másik X-eket. Az nyer, akinek előbb sikerül három azonos jelet elhelyeznie egy sorban, egy oszlopban vagy átlóban.

Ha nem ismered, és ennyiből nem világos, írd be a Google-be, hogy Tic-tac-toe és játssz párat a megjelenő játékkal!

A feladatod, hogy implementálj egy tic-tac-toe játék-táblát és két egyszerű stratégiájú tic-tac-toe játékost.

A programot elkezdtük, megírtuk a szükséges interface-eket és néhány osztályt is. A te dolgod csak az, hogy befejezd. Figyelj rá, hogy miközben a feladatot megoldod, a projektben található interface-eket ne módosítsd! Megoldásod során nem kell felhasználói felületet készítened, és main metódusra se lesz szükséged, csak a unit testek futtatásával fogjuk ellenőrizni a munkád. A projekt tartalmaz egy unit tesztet (TicTacToeTest.java), ezzel tudod ellenőrizni a megoldásodat. A unit teszt futtatásához az adott fájlon csak egy jobbklikk > “Test file” opciót kell választanod. (Természetesen, ha szükséged van rá, írhatsz main metódust, de ez nem része a feladatnak, pont sem jár érte.)

* Fork-old ([howto](https://confluence.atlassian.com/bitbucket/forking-a-repository-221449527.html)) saját felhasználódhoz a <https://bitbucket.org/aurettoworks/progmatic-tictactoeexam/src/master/> repository-ban található projektet, majd nyisd meg a NetBeans-ben!  
  A projekt megnyitása után a projekt mappáján kattints jobb gombbal, majd válaszd ki a “Clean and build” opciót. (Ugyanis ez egy maven-es projekt.)  
  (1p)
* Tanulmányozd a projektet! Vedd észre, hogy a benne szereplő osztályok különböző package-ekben találhatók! Nézd meg az összes osztályt, interface-t, olvasd el a kommenteket! Fordíts ez után külön figyelmet a Board és a Player interfacekre, továbbá az AbstractPlayer osztályra!
* Valósíts meg egy Board implementációt!
* Valósítsd meg a getCell metódust. A   
  PlayerType getCell(int rowIdx, int colIdx) throws CellException   
  metódussal le tudjuk kérdezni a tábla rowIdx-edik sorának colIdx-edik elemét. Figyelj arra is hogy a metódus exception-t dobjon, ha a kérdezett sor vagy oszlop nincs a táblán.   
  (2p)
* Hasonlóan valósítsd meg a put metódust (ez rak egy kört vagy X-et a táblára). Figyelj rá, hogy ez (a fenti ellenőrzés mellett) akkor is kivételt dobjon, ha az adott cellában már van O vagy X!   
  (2p)
* Valósítsd meg a   
  boolean hasWon(PlayerType p)  
  metódust! Ez akkor adjon vissza igazat, ha a p játékos nyert (vagyis van a táblán három egymás melletti jele vízszintes függőleges vagy átlós irányban)  
  (3p)
* Valósítsd meg a   
  List<Cell> emptyCells();  
  metódust! A Board interface kommentjei segítenek abban, hogy mi a metódus feladata!  
  (2p)
* Nézd meg a TicTacToeTest osztályt! Ez egy JUnit teszt és majdnem kész van. Csak három metódust kell benne helyesen megírnod, ezek:  
  private Board getBoardImpl()  
  private Player simplePlayerImpl(PlayerType p)  
  private Player victoryAwarePlayerImpl(PlayerType p)  
  Írd meg most a getBoardImpl metódust az előbbi feladatban megvalósított Board implementációd segítségével. A unit teszt segítségével ellenőrizd, hogy a Board implementációd helyesen működik. Ne várd most, hogy a TicTacToeTest minden tesztesete hibátlan legyen, ahhoz előbb még játékosokat is kell készítened!  
  (A getBoardImpl() metódusodnak csak példányosítania kell egyet a 3-as feladatban elkészített osztályodból!)  
  (1p)
* Készíts el egy egyszerű játékost, aki mindig az első olyan cellára tesz, ami még üres. (Ha nincs már üres cella a táblán, akkor ez a metódus null-t adjon vissza!) Ehhez készíts egy új osztályt, és származtasd ezt az AbstractPlayer osztályból! Ellenőrizd megoldásod a TicTacToeTest   
  private Player simplePlayerImpl(PlayerType p)   
  metódusának megírásával és a teszt futtatásával!  
  (2p)
* Készíts egy olyan játékost, aki annyival okosabb az előző egyszerű játékosnál, hogy észreveszi, ha a következő lépésben tud nyerni, és akkor oda rak, ahová kell, ahhoz, hogy nyerjen. Ehhez készíts egy új osztályt, és származtasd ezt az AbstractPlayer osztályból!  
  Ellenőrizd megoldásod a TicTacToeTest   
  private Player victoryAwarePlayerImpl(PlayerType p)   
  metódusának megírásával és a teszt futtatásával!  
  (4p)

Ha kész vagy, commitold a megoldásodat, és push-old! (Ellenőrizd a webes felületen, hogy a SAJÁT repository-dban látod-e a kommitot!) Ha készen adj hozzáférést a repodhoz [nyilasypeter@gmail.com](mailto:nyilasypeter@gmail.com)-nak és aurettoworks-nek, vagy tedd a repository-t publikussá, és küldd el a linkjét az teszt@progmatic.hu címre, a levélben a neveddel!