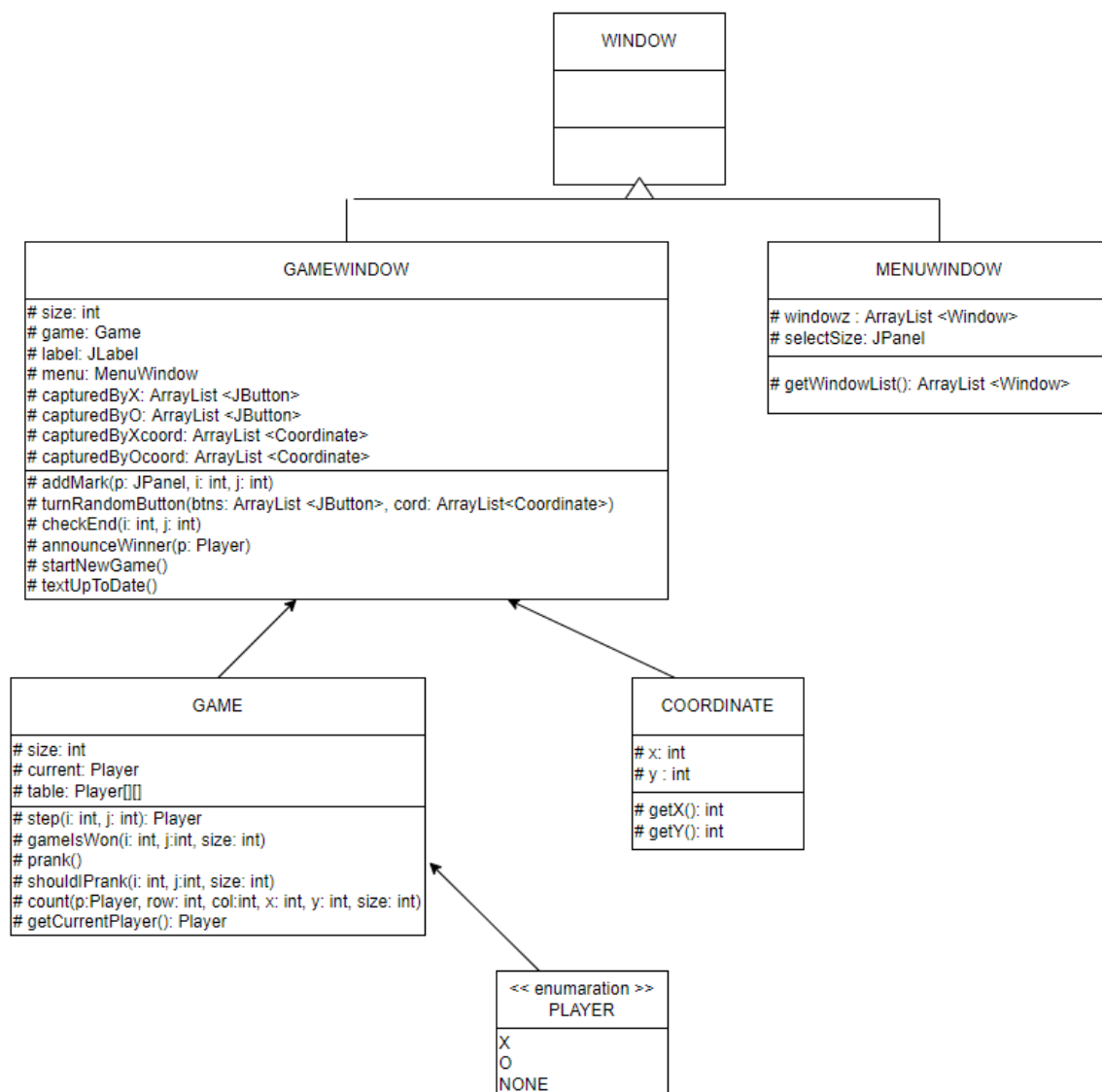


## Feladat

Készítsünk programot, amellyel a közismert amőba játék következő változatát játszhatjuk. Adott egy  $n \times n$ -es tábla, amelyen a két játékos felváltva X, illetve O jeleket helyez el. Csak olyan mezőre tehetünk jelet, amely még üres. A játék akkor ér véget, ha betelik a tábla (döntetlen), vagy valamelyik játékos kirak 5 egymással szomszédos jelet vízszintesen, függőlegesen vagy átlósan. A program minden lépésnél jelezze, hogy melyik játékos következik, és a tábla egy üres mezőjét kijelölve helyezhessük el a megfelelő jelet.

A kiszúrás a játékban az, hogy ha egy játékos eléri a 3 egymással szomszédos jelet, akkor a program automatikusan törli egy jelét egy véletlenszerűen kiválasztott pozícióról (nem biztos, hogy a hármastól), ha pedig 4 egymással szomszédos jelet ér el, akkor pedig kettőt. A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával (6×6, 10×10, 14×14), és ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, hogy melyik játékos győzött (ha nem lett döntetlen), majd kezdjen automatikusan új játékot.

## Terv



A Window űrsztály egy származtatott JFrame osztály, ahol semmi jelentős nem történik. Konstruktórában meg van adva az ablak neve és mérete, valamint a program ikonja.

A MenuWindow osztály a Window osztály leszármazottja. Ezen az ablakon lehet kiválasztani a pálya méretét, ami lehet 6x6, 10x10 vagy 14x14.

A GameWindow osztály szintén a Window osztály leszármazottja. Megjeleníti az adott méretű pályát. A játékosok sorra foglalhatják el a négyzeteket. Bármikor új játékot lehet indítani, valamint a menüből ki lehet választani más méretű pályát is. Az addMark() metódu s nézi hova kattint a felhasználó és megjelöli a mezőt, amire rákattintottak. A turnRandomButton() metódu s a kisorsol egy random gombot, amire már az adott játékos korábban rákattintott és reseteli azt. A checkEnd() metódu s megnézi, hogy vége van-e már a játéknak tehát betelt-e a pálya vagy nyert-e már valaki. Az announceTie() a döntetlent, az announceWinner() a nyertest hírdeti meg. A startNewGame() új játékot indít új ablakban, a régit megsemmisíti. A textUpToDate() metódu s a label szövegét frissíti minden lépés után, jelezve, hogy ki következik épp.

A Coordinate osztály egy olyan pár, amiben a gombok koordinátáit lehet tárolni.

A Player osztály egy enum, ahol fel vannak sorolva a játékos opciók: X – player 1, O – player 2, NONE – ha épp üres a mező.

A Game osztályban szimulálódik le a játék. Létrehozunk egy Player mátrixot, amit feltöltünk NONE-al. A step() metódu s megnézi, hogy a kattintott mező NONE-e és ha igen, akkor a játékos elfoglalja. A gamelsWon() metódu s azt nézi, hogy az adott játékosnak az előző lépésével sikerült-e 5 mezőt összekötnie. A prank() visszaállítja NONE-ra az adott mezőt. A shouldIPrank() megnézi, hogy 4 vagy 3 egymás melletti mező van, ha 4 akkor 2-vel, ha 3 akkor 1-el és ha kevesebb 0-val tér vissza. A count() metódu s végig számolja minden irányban hogy hány összeköthető mező van. A getCurrentPlayer() visszaadja az aktuális játékost, a getEmpty() meg visszaadja hány üres mező van.

## Tesztelési Terv

Kísérő Amőba tesztelés		
MenuWindow gombok szerint	6x6	létrehoz egy 6x6-os pályát
	10x10	létrehoz egy 10x10-os pályát
	14x14	létrehoz egy 14x14-os pályát
MenuWindow menu szerint	EXIT	bezárja a programot
GameWindow menu szerint	EXIT	bezárja a programot
	6x6	létrehoz egy 6x6-os pályát
	10x10	létrehoz egy 10x10-os pályát
	14x14	létrehoz egy 14x14-os pályát
felhasználó szerint	NEW GAME	új játékot indít ugyanakkora méretben
	rákattint üres mezőre	megjelenik a mezőn a játékos jele(X/O)
	rákattint egy nem üres mezőre	nem történik semmi, még mindig ő jön
	kigyúl 2 egymás mellett	nem történik semmi
	kigyúl 3 egymás mellett	visszafordul 1 mező
	kigyúl 4 egymás mellett	visszafordul 2 mező
	kigyúl 5 egymás mellett	a játékos nyer
	betelik a pálya	döntetlen lesz