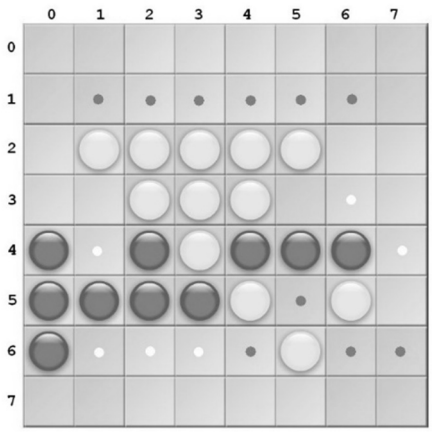
2. Reversi — 40 pont

A reversi játékot általában 8×8 mezőből álló négyzetrácsos táblán játsszák. Ebben a feladatban a tábla sorait és oszlopait is **0-tól 7-ig** azonosítjuk az ábra szerint. A játékot legjobb olyan korongokkal játszani, amelyek két oldala különböző színű (feladatunkban kék és fehér). A két játékos felváltva rakja le korongjait. A soron következő játékos csak olyan helyre rakhat, ahol meg tudja fordítani az ellenfél legalább egy korongját. Ez úgy lehetséges, hogy az éppen letett korong és a játékos másik korongja között egyenes vonalban vízszintesen, függőlegesen vagy átlósan kizárólag csak az ellenfél egy vagy több korongja található. Az egyes játékosok következő lehetséges lépéseit az ábrában kisebb körökkel szemléltetjük. Például a kék (sötét) játékos az 1;5 (sor;oszlop) mezőre azért rakhat szabályosan, mert a 4;2 mezőn lévő korongjával közrefog átlósan 2 db fehér korongot, így azok megfordulnak. A fehér játékos azért nem rakhat a 3;5 mezőre, mert nincs olyan korongja, amivel kizárólag csak kék (sötét) korongokat fogna közre, így erre a mezőre lépve nem tudna fordítani. Az ábra a forrásállomány adataival készült, így tanulmányozása a megoldáshoz segítséget nyújthat. A továbbiakban a reversi játékkal kapcsolatos feladatokat kell megoldania.



A megoldás során vegye figyelembe a következőket:

- *A program készítése során törekedjen az objektumorientált (OOP) megoldásra, amire a feladatsor ajánlásokat is tartalmaz. Amennyiben a programot ilyen módon nem tudja elkészíteni, akkor a feladatokat saját osztály létrehozása nélkül is megoldhatja, de így kevesebb pontot ér a megoldása. Ebben az esetben, ha a feladat jellemző vagy metódus létrehozását kéri, akkor Önnek saját alprogramot (függvényt, eljárást) kell készítenie, amely* ***paramétereken keresztül*** *kommunikál a hívó programmal.*
- *A képernyőre írást igénylő részfeladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például:* `5. feladat:`*)!*
- *Az egyes feladatokban a kiírásokat a minta szerint készítse el!*
- *Az ékezetmentes kiírás is elfogadott.*
- *Megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges input adatok mellett is megfelelően működjön!*

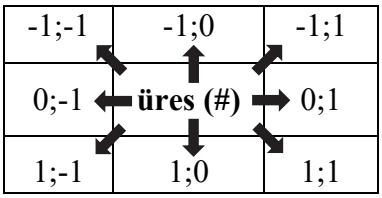
1. Készítsen programot a következő feladatok megoldására, amelyeknek a forráskódját `reversi` néven mentse el!
2. Hozzon létre saját osztályt `Tabla` azonosítóval és definiáljon benne egy karaktertípusú mátrixot (kétdimenziós tömböt) „`t`” azonosítóval, amelyben egy játék pillanatnyi állását tudja majd tárolni! A mátrix sorait és oszlopait 0-tól 7-ig indexelje!
3. Készítse el az osztály konstruktorát, ami a következő feladatokat hajtja végre:
   a. Inicializálja a „`t`” mátrixot 8×8-as mérettel.
   b. Feltölti a „`t`” mátrixot a „#”, „K” és „F” karakterekkel egy szöveges állományból. A feldolgozandó szöveges fájl nevét a konstruktor paramétereként adjuk át! A feladat megoldásához használandó `allas.txt` állomány 8 sora, soronként 8 karakterrel tárolja a játék állását. A tábla üres mezőit a „#” karakter, a játékosok korongjait a „K” (kék) és „F” (fehér) karakterek kódolják.
4. Hozzon létre egy `Tabla` típusú osztálypéldányt (objektumot), melynek a konstruktora az `allas.txt` forrásállomány nevét kapja aktuális paraméterként feldolgozásra!

5. Készítsen a `Tabla` osztályba `Megjelenit` azonosítóval metódust, ami a minta szerint megjeleníti a „t” mátrixban eltárolt játék állását!
6. Készítsen a játék állásáról összegzést a minta szerint! Az egyes karakterek („#”, „K”, „F”) megszámlálását a `Tabla` osztályban definiált paraméterezhető metódus segítségével végezze!
7. Definiáljon a `Tabla` osztályban metódust `VanForditas` néven a következő algoritmus kódolásával!
   (Ha nem a `Tabla` osztályban kódolja a metódust, akkor a „t” mátrix is a függvény paramétere legyen!)

```
Függvény VanForditas(jatekos: Karakter, sor, oszlop,
                     iranySor, iranyOszlop: Egész): Logikai
  Változó aktSor, aktOszlop: Egész
  Változó ellenfel: Karakter
  Változó nincsEllenfel: Logikai
  aktSor:=sor + iranySor
  aktOszlop:=oszlop + iranyOszlop
  ellenfel:='K'
  Ha (jatekos='K') akkor
      ellenfel='F'
  Elágazás vége
  nincsEllenfel:=igaz

  Ciklus amíg (aktSor>0 és aktSor<8 és
  aktOszlop>0 és aktOszlop<8 és t[aktSor, aktOszlop]=ellenfel)
      aktSor:=aktSor + iranySor
      aktOszlop:=aktOszlop + iranyOszlop
      nincsEllenfel:=hamis
  Ciklus vége

  Ha (nincsEllenfel vagy aktSor<0 vagy aktSor>7 vagy
  aktOszlop<0 vagy aktOszlop>7 vagy
  t[aktSor, aktOszlop]<>jatekos) akkor
      Térj vissza hamis
  Elágazás vége

  Térj vissza igaz
Függvény vége
```

A feladat a következő oldalon folytatódik.

8. A `VanForditas()` metódus hívásával határozza meg a minta szerint, hogy a megadott **üres cellába** korongot („F” vagy „K”) elhelyezve a megadott irányba történik-e fordítás! A metódus aktuális paramétereit egy karakterlánc típusú változóban (vagy konstansban) rögzítse programjában, az értékeket pontosvesszővel válassza el a következő sorrendben:
   `jatekos;sor;oszlop;iranySor;iranyOszlop` például: `"F;4;1;0;1"`
   Az `iranySor;iranyOszlop` paraméterek a következők szerint határozzák meg a feltételezett fordítás irányát:



   A metódushívás eredményét a minta szerint jelenítse meg!
9. Készítsen a `Tabla` osztályban logikai értékkel visszatérő metódust, ami meghatározza egy megadott játékos lépéséről, hogy szabályos lépés vagy nem szabályos lépés! Szabályosnak tekintünk egy lépést, ha a megadott cella üres, és a nyolc irány valamelyikéből (lásd előző feladat) a megadott játékossal történhet fordítás. Megoldásában felhasználhatja a korábban elkészített metódust is. A metódus aktuális paramétereit egy karakterlánc típusú változóban (vagy konstansban) rögzítse programjában, az értékeket pontosvesszővel válassza el a következő sorrendben: `jatekos;sor;oszlop` például: `"K;1;3"`
   A metódushívás eredményét a minta szerint jelenítse meg!

Minta:

```
5. feladat: A betöltött tábla
        ########
        ########
        #FFFFF##
        ##FFF###
        K#KFKKK#
        KKKKF#F#
        K####F##
        ########

6. feladat: Összegzés
        Kék korongok száma: 10
        Fehér korongok száma: 12
        Üres mezők száma: 42

8. feladat: [jatekos;sor;oszlop;iranySor;iranyOszlop] = F;4;1;0;1
        Van fordítás!

9. feladat: [jatekos;sor;oszlop] = K;1;3
        Szabályos lépés!
```