**2018. Középszintű érettségi (május)**

**Matematika**

**16. feladat feldolgozása Java programnyelven**

**Feladat**

Anna dominókészletében a dominókövek egyik oldala egy vonallal két részre van osztva. Az egyes részeken a pöttyök száma 0, 1, 2, 3, 4, 5 vagy 6 lehet. A készletben minden lehetséges pöttyözésű dominóból pontosan egy darab van. Az ábrán a 2-6-os (6-2-es) dominó látható.

1. Hány olyan dominó van a készletben, amelyen a két részen lévő pöttyök számának szorzata prímszám?
2. Rajzolja be a tíz üres részre a hiányzó pöttyöket az összekötésnek megfelelően! A játékban két dominó akkor csatlakozhat egymáshoz, ha a két érintkező részen ugyanannyi pötty van. Anna egy lapra elhelyezte dominókészletének azt a hat dominóját, amelyek mindkét részén van legalább 1, de legfeljebb 3 pötty. Ezután összekötötte azokat a dominókat, amelyeket a játékban csatlakoztatni lehetne egymáshoz. Az alábbi ábra a hat dominót és az összekötő vonalakat mutatja, de csak két részen adtuk meg a pöttyöket.
3. Anna a teljes 28 darabos készletből kihúzta a 2-6-os dominót. Ezután véletlenszerűen kihúz még egy dominót.

Számítsa ki annak a valószínűségét, hogy a másodiknak kihúzott dominót csatlakoztatni tudja az elsőhöz!

1. Egy játékbemutatóra Anna és Balázs 1800 dominót szeretne felállítani a földre úgy, hogy a legelsőt meglökve az összes dominó sorban eldőljön. Anna egyedül 6 óra alatt, Balázs pedig 9 óra alatt építené meg a dominóláncot. Ha Anna és Balázs – tartva a saját tempójukat – együtt dolgozna, akkor hány óra alatt végeznének az 1800 dominó felállításával?

**Tervezés**

A feladat értelmezése és a feltett kérdések alapján a tervezés során a következőkre jutottam. Létre kell hozni egy POJO-t, ami az alábbi adatokat fogja össze egy egységbe: egy dominó oldalait(két int típusú változó), amikhez később külön-külön is hozzá szeretnék férni, emiatt az ezekhez tartozó getter-k. Itt szükségem lesz az equals() metódus felülírására. Emellé létrehozok egy másik osztályt, amiben a feladat megoldásához szükséges adatszerkezetet tárolom (ArrayList vagy TreeSet), mely hozzá tud férni az előbbi POJO-hoz.

POJO osztályom neve: Domino, tulajdonságai: oldal1, oldal2 (int típusú változók). A konstruktor ezeket veszi át paraméterként, itt ellenőrzést is végzek:0 és 6 közötti értékük lehet csak.

Adatszerkezetet létrehozó osztályom neve: DominoDoboz , melynél ArrayListet választottam. Az itt lévő int típusú tömböt használom fel a listában tárolandó objektumok oldalainak a generálásához.

Konstruktora átvesz egy metódust (dobozKeszit() ), ami az objektumokat tartalmazó ArrayListet állítja elő.

Az outputhoz a Domino osztályon belül a toString() metódust felülírtam, úgy hogy reprezentálja egy dominó kiírását oldalak szerint például: 3|6 jelenti egy 3 és 6 pöttyű dominó kiírtatását. A DominoDobozon belül szükségem van arra, hogy néha az egész lista, néha csak a mérete legyen kiírva, így eszerint jártam el. Ha az egész listát szeretném látni, az objektumaimat “kirajzolva” a Domino osztály toString() metódusát hívom meg.

A belépési pont és a vezérlés egy másik osztályban történik. Itt a feladatok megoldásához két DominoDoboz objektumot használok, melyekben a másolás, keresés, kiválasztás programozási tételeket alkalmazom, külön a feladatra átírt metódusok által.

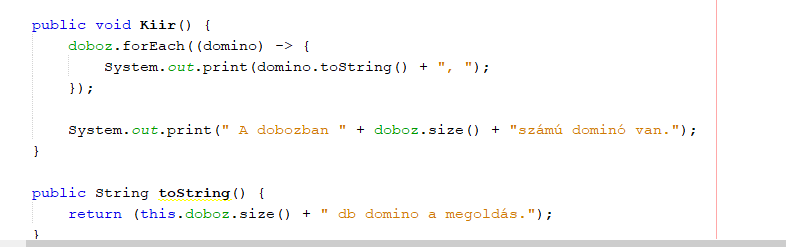
**A POJO osztály forráskódja:**



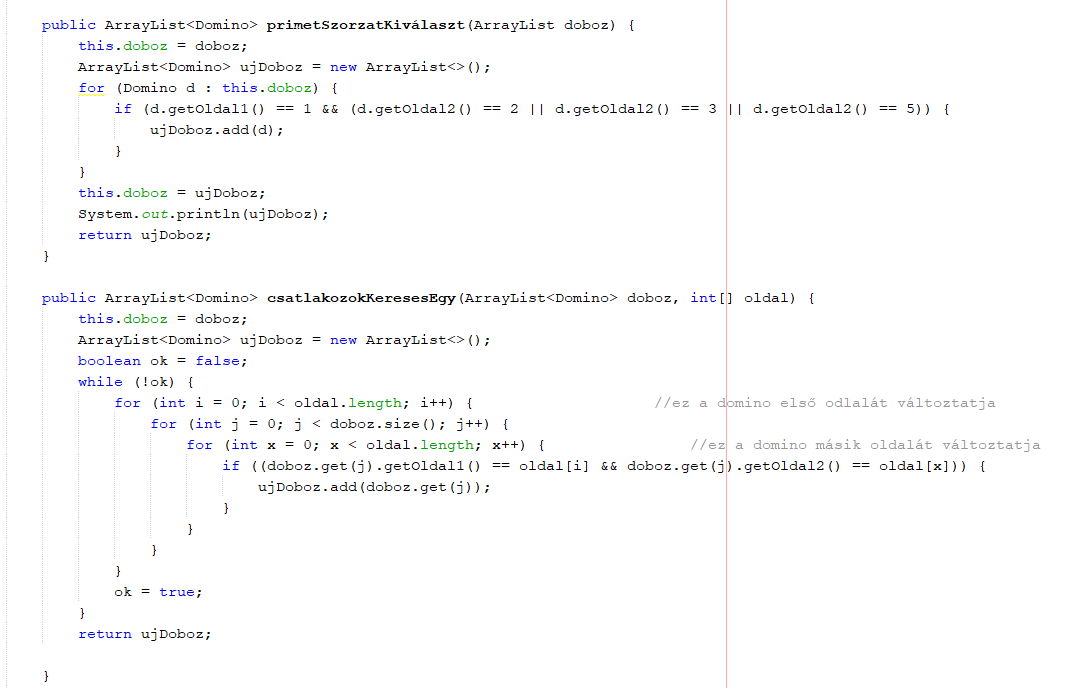
Hozzá tartozó két “összehasonlító” metódus:

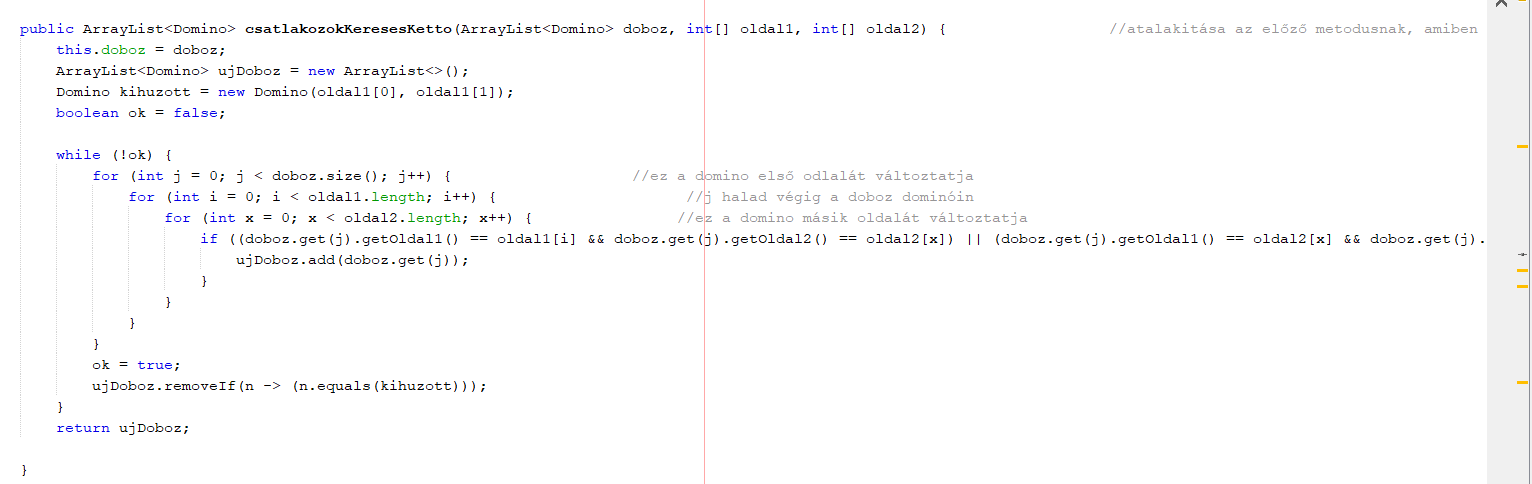


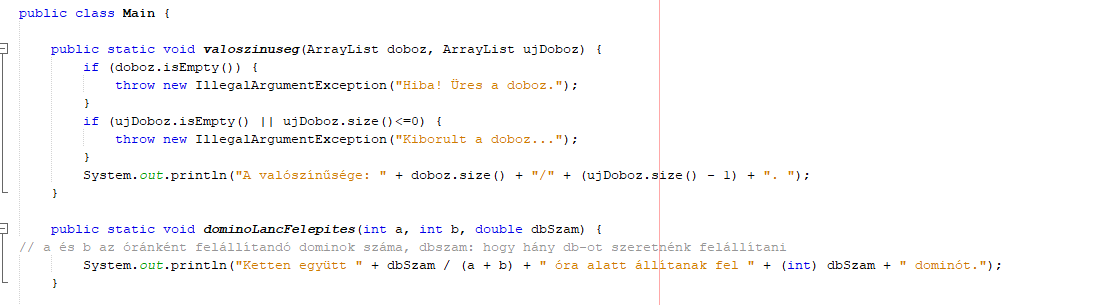
**DominoDoboz osztály:**

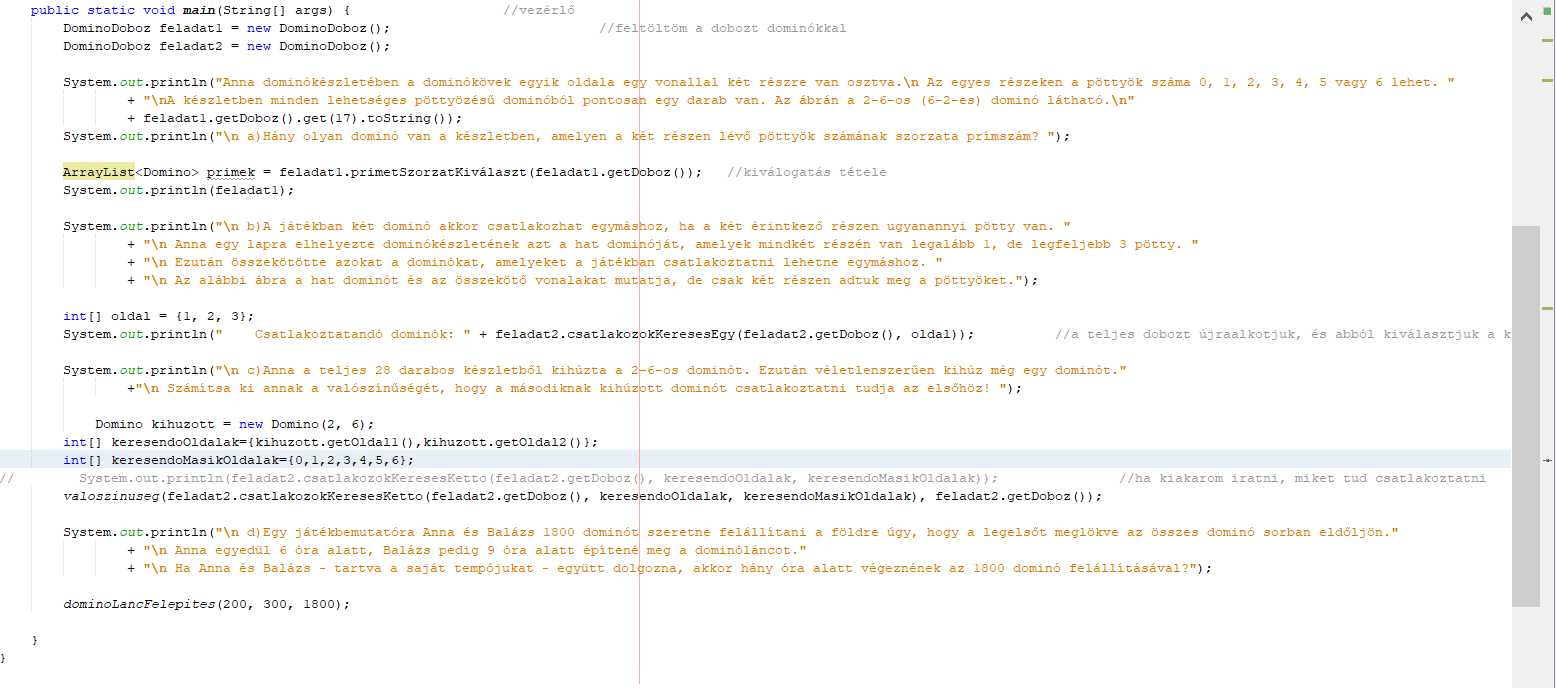


**Ezen belül a feladatokhoz írt metódusok:**





**Vezérlő**: 

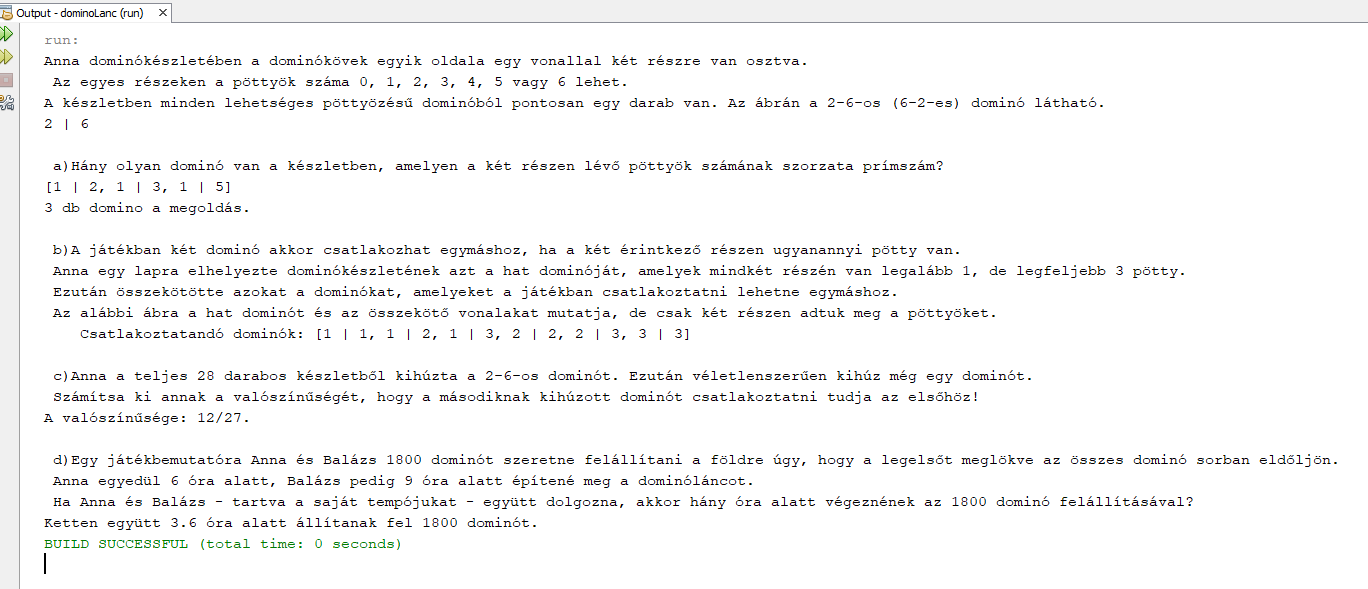


A main() metódus feltölti a generikus lista adatszerkezetet az alkalmazott modellben létrehozott domino objektumokkal. Ez a feladatban egy doboz dominónak felel meg. A felülírt equals() metódusban specifikálni kellett, hogy a listában nem lehet két ugyanolyan oldallal rendelkező domino(pl.:2|6=6|2). Mivel ArrayList-t használok, ami magától nem rendezi sorba az objektumokat, az áttekinthetőbb ellenőrzések miatt sorba rendezést is véghez vittem, compareto() metódust is alkalmazok.

Az a, feladatban a választ kiválasztás programozási tétel felhasználásával kapjuk meg. Itt nem kérdés, hogy van-e a listában ilyen. Erre külön létrehozok egy új ArrayListet, illetve a primetSzorzatKiválaszt() metódust, ami paraméterként fogadja a domino objektumokkal feltöltött teljes listát.

A b, feladathoz létrehozok egy másik dominókat tartalmazó teljes listát. A továbbiakban ezt használom a feladatokhoz. Itt azon objektumokat kell kiválasztani, amiknek 1-3 pötty van az oldalukon. Ehhez létrehozok egy metódust, ami ezeket keresi meg. Paraméterként fogadja a listát és azt az int típusú tömböt ami az 1-3 számokat tartalmazza. A c, feladatban a b,-ben felhasznált metódust fejlesztem tovább. Két int típusú tömböt fogad a megírt metódus és a listát, amiből ki kell választania az adott objektumukat. Végül a kapott lista méretét használom fel és adom át a vezérlő osztályban található valószínűségszámítas() nevű metódusnak. Itt kivételkezelést is alkalmazok.

**Eredmény:**

****

A program végül szövegesen jeleníti meg a feltett kérdéseket és válaszokat.