Kocsmaszimulátor part 2:

A meglévő Kocsmáros osztályunkat örököltessük az Ember-ből, mivel a kocsmárosok is emberek. Ettől automatikusan megkapja az Ember osztályban deklarált adattagokat (változókat) is. A Kocsmároshoz írjunk egy új paraméteres konstruktort, ami az Ember (tehát az ős) konstruktorát is meghívja (super kulcsszó).

Az Ember osztályunkból örököltessünk egy Diákot is, aki egy különleges ember lesz. A diák rendelkezzen privát ösztöndíj változóval, ami megadja, hogy mennyi pénzt keres a tanulással. Legyen egy tanul() metódusa, amitől az ösztöndíja mennyiségével nő a pénze. Legyen egy paraméteres tanul(int mennyit) metódusa is, amivel be lehet állítani, mennyit tanuljon, és ennek megfelelő pénzt kap.

Az Emberből származzon még egy Kidobó is. Neki legyen egy privát boolean dolgozik adattagja. Legyen egy static változója is, ami azt fogja megadni, hogy hány olyan kidobó van, aki dolgozik éppen. Ez alapból 0. Új kidobó példányok konstruktorába rakjuk ennek a változónak a növelését, és változtassuk mindig megfelelően, ha a dolgozik változó értéke változhat (setter, szolgalatbaAll(), szolgalatbolKilep()). Legyen tehát szolgalatbaAll(), és szolgalatbolKilep() metódusa is, amelyek a dolgozik értékét állítják. Írjuk felül az iszik függvényét úgy, hogy ellenőrizze le, az adott kidobó éppen dolgozik-e, mivel szolgálatban nem ihat alkoholtartalmú italt, azonban 0 alkoholtartalmút igen. Ha nincs szolgálatban, hívjuk meg az Ember (tehát super) iszik függvényét.

A Diák, és a Kidobó is rendelkezzen paraméteres konstruktorral.

A Diák, Kocsmáros, és Kidobó toString()-jét írjuk felül, hívjuk meg benne az Ember toString()-jét is, valamint írjuk ki az adott osztályra jellemző egyéb adatokat is.

Az Ember rendelkezzen egy kötözködik(Ember kivel) metódussal, ami egy másik embert vár paraméterül. Le kell ellenőrizni, hogy van-e szolgálatban lévő kidobó, ha igen, az első embert hazaküldjük. Ha nincs, kiírjuk, hogy jót kötözködött.

**Italok**

Az Ital osztályból örököltessünk egy Sör, Bor, és egy Kevert osztályt.

A Bor rendelkezzen egy privát int évjárat változóval. A Sörnek ne legyen új változója. Ezek kapjanak megfelelő konstruktort.

A Kevert osztálynak legyen 2, 3, 4 paraméteres konstruktora is, ami italokat vár paraméterül, például: public Kevert(Ital it1, Ital it2, Ital it3, Ital it4)

Ezek az italok lehetnek sima Ital, Sör, vagy Bor tipusúak, ezt nem kell külön lekezelnünk, mert erre is jó az öröklődés, mivel ezek mind kezelhetőek Italként.

A konstruktor a Kevert ital alkoholtartalmát a megadott italok átlag-alkoholtartalmára állítsa be. Egy példa a konstruktor hívására:

Kevert kevert1 = new Kevert(sor1, sor2, bor1, ital1);

Minden osztály privát adattagjainak legyenek getterei és setterei.