

**Objektum orientált programozás**  
**8. gyakorlat**  
**Abstract osztályok, interfészek, interfész referenciák**

1. Készítsen saját csomagban **Hasáb** absztrakt osztályt.

Adattagja: magasság (*int*)

Konstruktor: a paraméterben kapott értékkel inicializálja a magasságot

Metódusok:

- Getter metódus, amely visszaadja a magasságot.
- Abstract metódus az alapterület visszaadására.
- Metódus a térfogat visszaadására.
- Összehasonlító metódus, amely igazat ad ha a hasáb nagyobb térfogatú, mint egy paraméterben kapott másik hasáb térfogata.

Ugyanebben a csomagban készítsen **Henger** osztályt, amely a Hasáb leszármazottja.

Adattagja: sugár (*double*)

Konstruktor: a paraméterben kapott sugár és magasság értékekkel inicializálja az objektumot

Metódusok:

- Definiálja felül az örökölt alapterületet visszaadó metódust.
- Definiálja felül a *toString()* metódust: ez összefűzve adja vissza a henger sugarát és magasságát.

Készítsen ugyanebben a csomagban **Téglatest** osztályt, amely a Hasáb leszármazottja.

Adattagjai: az alap két oldala (*double*)

Konstruktor: a három adattagot a paraméterben megkapott értékkel inicializálja

Metódusok:

- Definiálja felül az örökölt alapterületet visszaadó metódus.
- Definiálja felül a *toString()* metódust: ez összefűzve adja vissza a téglatest alapjának oldalait és magasságát.

Alcsomagban készítsen futtatható osztályt, amelyben létrehoz egy hengert majd kiírja az adatait és a térfogatát. Hozzon létre egy téglatestet és írja ki az adatait és a térfogatát, majd azt hogy melyik a nagyobb térfogatú.

\*\*\*

Készítsen új csomagban egy olyan osztályt, amely hasábokat képes tárolni.

Adattagja: Hasáb típusú tömb adattag

Konstruktor: paraméterben megkapja a hasábok max. számát és ezzel a mérettel hozza létre a tömböt.

Metódusok:

- Setter metódus, amely paraméterben megkap egy indexet és egy hasábot és a tömb adott indexű elemének értékül adja a kapott hasáb referenciáját.
- Getter metódus, amely visszaadja a tömb méretét, azaz a hasábok max. számát.
- Getter metódus, amely paraméterként egy indexet kap és visszaadja a tömb adott indexű elemét.
- Getter metódus, amely visszaadja, hogy hány darab nem null értékű tömbelem van.
- Getter metódus, amely visszaadja a hasábok átlagos térfogatát.
- Getter metódus, amely visszatér a Henger típusú hasábok számát.

Készítsen ugyanebben a csomagban futtatható osztályt, amelyben létrehoz egy 5 elemű hasáb-tömb objektumot. A tömb elemeinek adjon értéket úgy, hogy legyen null értékű tömbelem is. Két henger és egy téglatest adatait olvassa be. Írja ki az összes hasáb adatát, majd a hasábok átlagos térfogatát, és a hengerek számát.

2. Készítsen saját csomagban **Színezhető** néven **interfészt**.

Adattagja: alapértelmezett szín, értéke piros (használja a *java.awt.Color* osztályt).

Metódusok:

- Getter metódus a szín lekérdezésére.
- Setter metódus a szín megváltoztatására.

Készítsen másik csomagban **Pont** osztályt.

Adattagja: x és y koordináta (*int*)

Konstruktor: két megkapott paraméterrel inicializálja az adattagokat

Metódus:

- Definiálja felül a *toString()* metódust, amely Stringben összefűzve adja vissza a pont adatait.

Készítsen **Szinespont** osztályt a Pont osztályból származtatva.

Adattagja: szín

Konstruktor:

- Csak a szín paraméter megadásával inicializálja az adatokat.
- Három megkapott paraméterrel inicializálja az adatokat.

Metódus:

- Definiálja felül az Object osztálytól örökölt *toString()* metódust.
- Implementálja a Színezhető interfészt.

A 7. gyakorlat 1. példájában készített **Áru** osztállyal azonos csomagban készítsen **Toll** osztályt az Áru osztály leszármazottjaként.

Adattagjai: szín és márkanev

Konstruktor: minden adatot megkap paraméterként

Metódus:

- Definiálja felül az örökölt *toString()* metódust.
- Implementálja a Színezhető interfészt.

Készítsen egy futtatható osztályt, amelyben egy Színezhető típusú referenciának Szinespont példányt ad értékül, kiírja az adatait, majd átszínezi és újra kiírja az adatait. Ugyanezt próbálja ki egy Toll objektummal is.

Készítsen az osztályba egy statikus metódust, amely paraméterként egy Színezhető objektumot vár és az objektumot átszínezi alapszínre. A main-t is egészítse ki úgy, hogy használja a metódust egy Szinespont példánnyal és egy Toll példánnyal is.

Készítsen saját **Szín** osztályt, ami ugyanazt tudja, mint a *java.awt.Color* osztály, csak a *toString()* metódus a színek magyar nevét írja ki. Használja ezt az objektumok színének kiírásához.

## Házi feladat

1. A 7. gyakorlaton készített Áru osztályból legyen abstract osztály. Absztrakt metódusa az egységár kiszámító metódus. A korábbi gyakorlatokon készített Könyv osztály legyen az Áru leszármazottja. A könyv egységárát az ár/oldalszám képlettel számítsa.

Készítsen futtatható osztályt, amelyben beolvassa n db könyv adatait.

- Cím szerint rendezze a könyveket (használja a String osztály *compareToIgnoreCase(String str)* metódusát) és írja ki az egységárukat.
- Állapítsa meg, mely szerzőknek van 20 Ft/oldalnál drágább egységárú könyve.

2. Készítsen **Eladó** interfészt.

Adattagjai: alapértelmezett ár, értéke 1, és alapértelmezett devizanem, értéke Ft.

Metódusok:

- Getter metódus az ár és a devizanem lekérdezésére.
- Setter metódus az ár és a devizanem megváltoztatására.

Módosítsa az Áru osztályt úgy, hogy az implementálja az Eladó interfészt (legyen ár és devizanem adattagja). A Könyv osztály pedig az Áru leszármazottja.

Készítsen Ingatlan osztályt, amely implementálja az Eladó interfészt.

Adattagjai: ár, devizanem, méret (*int*), hely (város)

Konstruktor: négy paraméter megadásával inicializálja az adatokat

Metódusok:

- Az ingatlan méretéből és árából számítsa ki a négyzetméter árat.
- Számítsa ki az ingatlan árat euróban.
- Definiálja felül a *toString()* metódust.

Készítsen egy futtatható osztályt, amelyben egy Eladó típusú referenciának Ingatlan példányt ad értékül, kiírja az adatait, majd átszámítja az árat euróra és újra kiírja az adatait. Ugyanezt próbálja ki egy Könyv objektummal is.