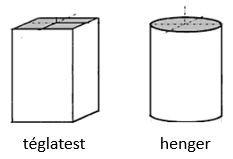
# 10. Absztrakt osztályok

## Feladat

A mai programban téglatestek és hengerek kezeléséhez készítünk osztályokat. Bár matematikailag ez nem teljesen helyes, hasáboknak fogjuk hívni őket.



A Hasab osztályban készítjük el a közös adattagokat és metódusokat. Ennek leszármazottjai lesznek a Teglatest és a Henger osztályok.

A két test alapjának területét különbözőképpen számoljuk. A térfogat viszont mindkettőnél az alapterület szorozva a magassággal.

Az alapterületet és a térfogatot számító metódusokat szeretnénk a szülő osztályban (a Hasab-ban) megadni. Az alapterület számítása viszont különbözőképpen történik a leszármazottakban. Ilyenkor alkalmazzuk az absztrakt metódusokat.

## Absztrakt metódusok

Az absztrakt metódus egy olyan metódus, amely az abstract módosítóval van ellátva és NINCS TÖRZSE. Például

public abstract double alapterulet();

Az absztrakt metódusokat a leszármazott osztályokban felül kell definiálni a törzsük megadásával. Ebből következik, hogy egy absztrakt metódusnak felüldefiniálhatónak kell lennie, azaz láthatónak kell lennie, nem lehet static, és nem lehet final.

Szintaktikailag tehát egyszerű dologról van szó, az érdekesebb inkább az, hogy mire jó egy ilyen metódus.

Az absztrakt metódusok szerepe, hogy az ősben rögzítse, hogy a leszármazottaiban kell lennie ilyen metódusnak, de még nem lehet tudni a megvalósítását (vagy azt lehet tudni, hogy minden leszármazottban másképpen kell majd megvalósítani).

Például a feladatban minden Hasab leszármazottban kell lenni alapterületet számító metódusnak, de minden leszármazottban másképpen kell azt kiszámolni.

De miért kell már az ősben deklarálni a metódust, miért nem elég ha minden leszármazott definiálja az alapterületet számító metódust?

Azért, mert így már ős típusú változóval is lehet használni ezt a metódust. Hiszen egy változóval csak olyan metódusra hivatkozhatunk, amely a referencia statikus típusában van.

Ha tehát definiálok egy Hasab típusú változót és értékül adok neki bármilyen Hasab leszármazottat, akkor használhatom az alapterület metódusát és ráadásul mindig a megfelelő leszármazott metódusa fog végrehajtódni, hiszen felüldefiniált metódus esetén a változó dinamikus típusa dönt.

Például:

Hasab a = new Henger(3,4);  
double t = a.alapterulet();

Mivel a példánymetódus hívásakor a dinamikus típus határozza meg, hogy melyik metódust hajtja végre, a második sorban a Henger osztály alapterulet() metódusát fogja lvégrehajtani.

Az absztrakt metódus már abban az osztályban is felhasználható, amelyben definiáltuk.

Például a feladatban az alapterületet számító metódust meghívhatjuk a térfogatot számító metódusban. A felüldefiniáláskor úgyis a változó dinamikus típusának megfelelő alapterületet számító metódus fog meghívódni.

## Absztrakt osztályok

Amelyik osztályban van akár csak egyetlen absztrakt metódus, az kötelezően absztrakt osztály. Az absztrakt osztályokat az abstract módosítóval kell ellátni.

Például

public abstract class Hasab { ... }

Az absztrakt osztályok nem példányosíthatók!!! Ha ugyanis lehetne példányosítani, akkor meghívható lenne az absztrakt metódusa.

Például a következő utasítások hibásak:

Hasab a = new Hasab(2);  
t = a.alapterulet();

De az absztrakt osztály természetesen lehet szülő osztály, így a gyermekeknek lehetőségük van az örökölt absztrakt metódusok felüldefiniálására.

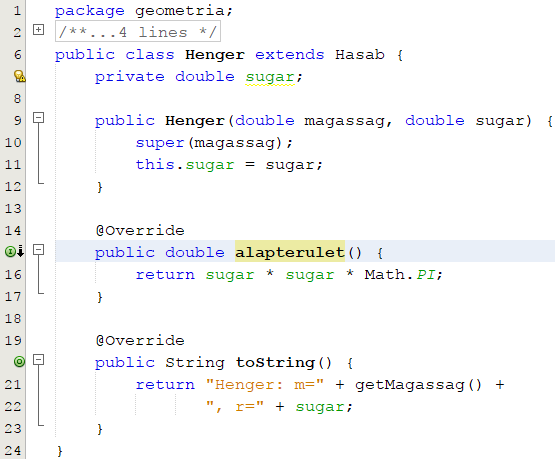
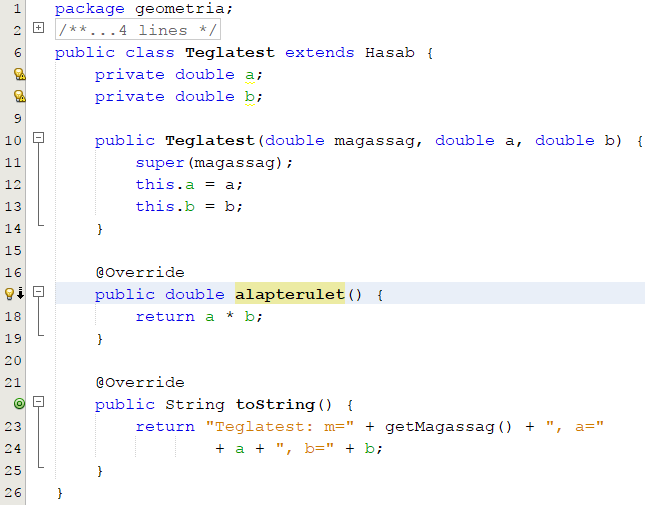
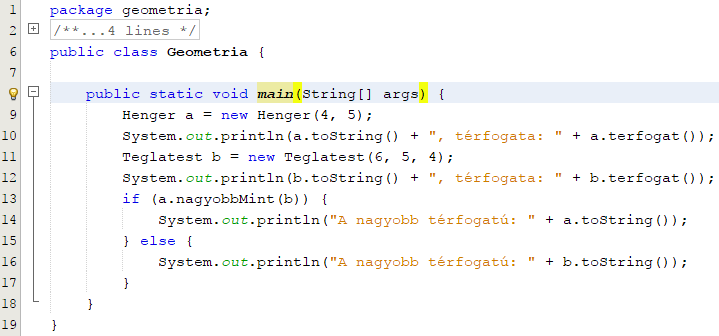
Ha egy gyermek nem definiálja felül az összes örökölt absztrakt metódust, akkor ő is absztrakt lesz, hiszen akkor van absztrakt metódusa.

Az absztrakt osztályoknak nem lehet final módosítója, hiszen az kizárná az utódokat.

## Feladat

Ebben a leckében is egy egyszerű feladatot oldunk meg, és közben kipróbáljuk az absztrakt osztályokról tanultakat.

## Hasab osztály

1. Indítsd el a NetBeans programot, és hozz létre egy új osztályt Geometria néven!
2. Hozz létre még egy osztályt a csomagon belül, Hasab néven, main metódus nélkül!
3. Készítsd el a Hasab osztály kódját az alábbiak szerint:  
      
   Figyeld meg, hogy az alapterulet() metódus és így az osztály is absztrakt, mert minden utódnál másképp kell majd kiszámítani!
4. Ezután készíts egy Henger osztályt is, az alábbi kóddal:  
     
   Figyeld meg, hogy felüldefiniáltuk az alapterulet() metódust (és a toString()-et is)!
5. Ezután készíts egy Teglatest osztályt is:  
      
   Figyeld meg, hogy ebben másképp definiáltuk felül az alapterulet() metódust (és a toString()-et is)!
6. Végül készítsd el a Geometria osztály kódját is, amelyben használod a korábban definált osztályokat:  
   
7. Futtasd a programot, és ellenőrizd az eredményeket:  
   