Feladat

- A feladat elkezdéséhez a mellékelt projekt tartalmaz kódokat. A megoldás bizonyos feladatoknál ezekhez a kódokhoz is hozzá kell nyúlni, és a teszteléshez a **main** függvényben a nem használt kódokat ki lehet kommentezni.
- A megadott osztály a **Color** osztály, amely nem tárol adattagot, viszont egy absztrakt ősosztály lesz, aminek a gyerekosztályai színeket tárolnak különböző módokon. A színkomponensek kezeléséhez segítséget nyújtó táblázatok a feladat leírás után találhatóak.
- Készíts a **Color** osztályba három absztrakt metódust *R*, *G*, illetve *B* néven, ami majd a szín három komponensét adják vissza **int** típusként. Ezen értékek 0 és 255 között változnak.
- Származtass a **Color** osztályból egy **RGBColor** osztályt, amely három **int** típusként tárolja a szín három (r, g, b) komponensét. A három adatot a konstruktor várja, és állítsa be. Legyen az osztályban egy paraméter nélküli konstruktor is, amely a fekete színt állítja be.
- Fejtsd ki az *R*, *G*, és *B* metódusokat az **RGBColor** osztályban úgy, hogy visszaadják a tárolt értékeket.
- Származtass egy **BlackOrWhite** osztályt a **Color** osztályból, ami egy logikai értéket tárol. Az érték igaz, ha a szín fekete, illetve hamis, hogyha fehér. Az osztály konstruktora is ezt az értéket várja. Fejtsd ki az *R*, *G*, és *B* metódusokat az osztályban.

- Származtass egy **GrayScale** osztályt a **Color** osztályból, amely lebegőpontos értéket tárol, 0 és 1 között. Ez azt mondja meg, hogy a szürke szín mennyire világos (0-teljesen fekete, 1-teljesen fehér, közte egyenletesen változik). Fejtsd ki az R, G, és B függvényeket az osztályban.
- Készíts a **Color** osztályba egy statikus *NumberToHex* metódust, amely megkap egy egész értéket (0 és 15 között), és visszaadja az értéket hexa karakterré konvertálva (0-F). A metódus feltételezheti, hogy az érték tényleg 0 és 15 közötti. Készíts egy hasonló *HexToNumber* metódust is, amely a fordított átalakítást végzi el.
- A **Color** osztály írja felül a *ToString* metódust úgy, hogy adja vissza a színt szövegesen, hexadecimális formában (pl. „#E4CB27”).
- Származtass egy **TextColor** osztályt az **RGBColor** osztályból. Ez tároljon még egy szöveget, ami a szín szöveges megnevezése („black”, „green”, stb.). A konstruktora csak ezt a szöveget várja, és állítsa be a szín komponenseit a megfelelő értékre. Ha a megkapott szöveg nincs a lehetőségek között (lásd a táblázatot a feladat után) akkor a fekete színt tárolja el.
- A **TextColor** osztály konstruktorát módosítsd úgy, hogy a hexadecimális kódban megadott színt is elfogadja (’#’ az első karaktere), és az alapján is be tudja állítani a komponenseket.
- Írd felül a *ToString* metódust a **TextColor** osztályban úgy, hogy a hexadecimális forma helyett a szín eltárolt megnevezését adja vissza.
- A **Color** osztály implementálja az **IEquatable** interfészt. Két szín akkor egyezik meg, ha mindhárom komponensük megegyezik.
- Az **RGBColor** osztálynak készíts egy konstruktort, amely paraméterben egy tetszőleges színt vár. A komponensek értékeit úgy állítsa be, hogy a paraméterben kapott színével megegyezzenek.

Segítség a színek kezeléséhez:

Az r, g, b értékek 0 és 255 között változnak.

Hexadecimális kód: mindig ’#’-vel kezdődik, és 16-os számrendszerben tárolja először az r, g, b értékeket ilyen sorrendben, minden komponensnek kettő karaktert hagyva.

Színek, amiket a **TextColor** osztály felismerhet:

Név	**Értékek (R,G,B)**	**Hexadecimális**
black	(0,0,0)	#000000
white	(255,255,255)	#FFFFFF
blue	(0,0,255)	#0000FF
green	(0,255,0)	#00FF00
red	(255,0,0)	#FF0000
cyan	(0,255,255)	#00FFFF
magenta	(255,0,255)	#FF00FF
yellow	(255,255,0)	#FFFF00

Egyéb hexadecimális példák:

Értékek (R,G,B)	**Hexadecimális**
(120,65,234)	#7841EA
(6,54,15)	#06360F
(212,200,253)	#D4C8FD
(128,96,177)	#8060B1