

Szín-kép

Egy digitális kép tárolásánál minden egyes képpont színét tároljuk. A képpontok színét az RGB kód adja. Az RGB kód a vörös (R), zöld (G) és a kék (B) színösszetevő értékét határozza meg. Ezen színösszetevők értéke 0 és 255 közötti egész szám lehet.

A `kep.txt` fájlban egy 50×50 képpontos kép képpontjainak RGB kódjai vannak a következő formában. Az állomány a képet sorfolytonosan, a képpontok RGB kódját szóközzel elválasztva tartalmazza, minden képpontot egy újabb sorban:

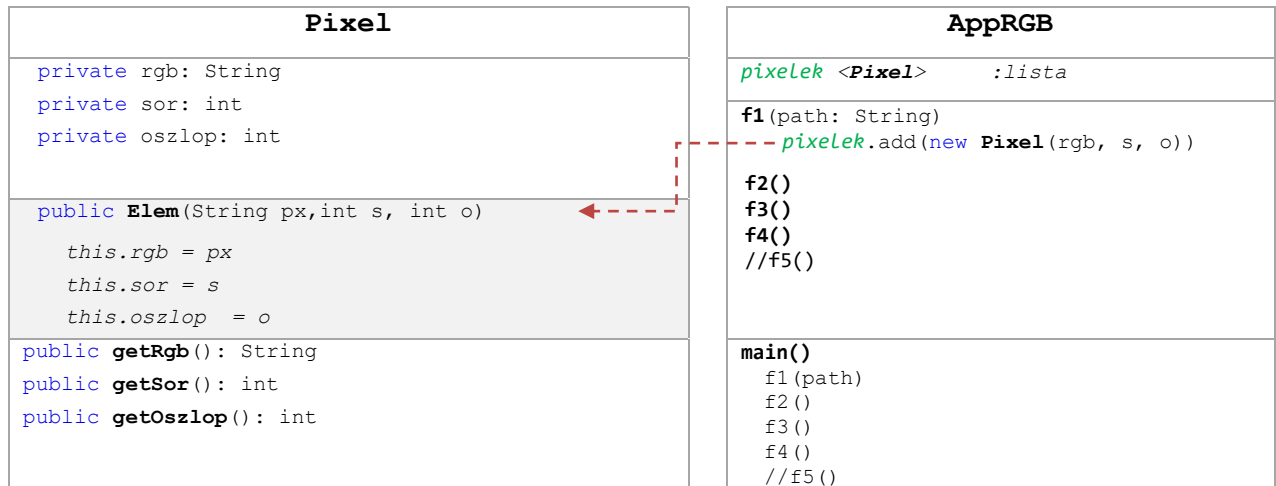
```
200 96 64
200 96 64
200 96 64
200 96 64
200 96 64
```

1. Olvassa be a fájlból egy megfelelő adatszerkezetbe az egyes képpontok RGB kódját!
2. Kérjen be a felhasználótól egy RGB kódot! Állapítsa meg a program segítségével, hogy a bekért szín megtalálható-e a képen, ha igen akkor írja ki az első ilyen színű pixel koordinátáit!
3. Határozza meg, hogy a kép 35. sor 11. képpontjának - `px(35, 11)` - színe milyen RGB kódú és hányszor szerepel a 35. sorban, illetve a 11. oszlopban.
4. Állapítsa meg, hogy a vörös, kék és zöld színek közül melyik szín fordul elő legtöbbször a képen! Az (egyik) legtöbbször előforduló szín nevét írja ki a képernyőre!

A színek kódjai:

Vörös	255, 0, 0
Zöld	0, 255, 0
Kék	0, 0, 255

5. Az képen egy sárga RGB (255, 255, 0) színű téglalap van. Határozza meg a program segítségével a bal felső és a jobb alsó sárga képpontnak a helyét (sor, oszlop), majd határozza meg, hogy a sárga téglalap hány képpontból áll! A képpontok helyét és a sárga alakzat méretét írassa ki a képernyőre.



```
1. feladat: az adatok beolvasása a(z) kep.txt fájlból.
    A fájl beolvasása ..... kész!

2. feladat: van-e ilyen színű pixel?
    Kérem az RGB kódot szóközzel elválasztva (pl.: 255 255 0): 255 0 0
    Az első ilyen színű pixel helye: px(10, 10)

3. feladat: a 35. sor 11. képpontjának kódja, hányszor szerepel ez a kód az adott sorban ill. oszlopban?
    pixel [200 96 64]
    a 35. sorban: 29 db
    a 11. oszlopban: 39 db

4. feladat: a vörös, kék és zöld színek közül a leggyakoribb
    85 220 231
    a leggyakoribb a kék (231)

+5. feladat: A sárga téglalap adatai.
    bal felső sarok: px(31,12)
    jobb alsó sarok: px(41,21)
    pixelek száma: 110

<<Kilépéshez: ENTER>>
```

1. feladat

```
eljárás f1(path: String)
    f = fájl megnyitása olvasásra (path)
    px: String
    ciklus (s = 1-től, s < 50-ig egyesével)
        ciklus (o = 1-től, o < 50-ig egyesével)
            px = f.sorbeolvasása()
            pixelek.add( new Pixel(px, s, o) ) // konstruktor hívása
        ciklus vége
    ciklus vége
    f.zárása()
eljárás vége
```

2. feladat

```
eljárás f2()
    N = pixelek.mérete()
    Be: px
    ciklus amíg ( i<N és !(pixelek(i).rgb ⇔ px) )
        i++
    ciklus vége
    ha (i<N)
        akkor:      Ki: "Pixelhelye: " pixelek(i).sor, pixelek(i).oszlop
        különben:   Ki: " A pixel nincs a képen."
    ha vége
eljárás vége
```

3. feladat

```
eljárás f3()
    N = pixelek.mérete()
    ciklus amíg ( !(pixelek(i).sor = 35 és pixelek(i).oszlop = 11) )
        i++
    ciklus vége
    px = pixelek(i).rgb; sDB = 0; oDB = 0
    ciklus (i=0-től i<N-ig egyesével)
        ha (pixelek(i).sor = 35 és pixelek(i).rgb ⇔ px) akkor: sDB++
        ha (pixelek(i).oszlop = 11 és pixelek(i).rgb ⇔ px) akkor: oDB++
    ciklus vége
    Ki: px, sDB, oDB
eljárás vége
```

4. feladat

```
eljárás f4()
    N = pixelek.mérete()
    rgbDB [] = {0, 0, 0}
    ciklus (i=0-től i<N-ig egyesével)
        elágazás (pixelek(i).rgb szerint)
            ha (pixelek(i).rgb ⇔ "255 0 0") akkor: rgbDB[0]++
            ha (pixelek(i).rgb ⇔ "0 255 0") akkor: rgbDB[1]++
            ha (pixelek(i).rgb ⇔ "0 0 255") akkor: rgbDB[2]++
        elágazás vége
    ciklus vége
    color[] = {"vörös", "zöld", "kék"}
    max = 0
    ciklus (i=0-től i<3-ig egyesével)
        ha (rgbDB[i] > max)
            akkor: max = rgbDB[i]; c = color[i]
        ha vége
    ciklus vége
    Ki: c, max
eljárás vége
```