

## I. PROCEDURÁLIS PROGRAMOZÁS

### 1. KIFEJEZÉSEK

1. Döntsd el egy egész számról, hogy páros-e.
2. Döntsd el egy tetszőleges számról, hogy egy adott intervallumba esik-e.
3. (a) Döntsd el egy koordinátákkal adott pontról, hogy az origó-e.  
(b) Döntsd el, hogy az egyik koordinátatengelyre esik-e.
4. Számítsd ki egy adott sugarú gömb térfogatát.
5. (a) Döntsd el két egész számról, hogy az első osztója-e a másodiknak.  
(b) Döntsd el, hogy bármelyik osztója-e a másodiknak.
6. Döntsd el három számról, hogy lehetnek-e egy háromszög oldalhosszai.
7. Döntsd el két számról, hogy megegyezik-e az előjelük.
8. (a) Add meg egy számtani sorozat első két elemének ismeretében a harmadik elemét.  
(b) Add meg az  $n$ -edik elemét. ( $n$ -et tetszőlegesen választhatod meg.)  
(c) Mértani sorozatra is add meg az  $n$ -edik elemét.
9. Számítsd ki egy háromszög területét az oldalhosszaiból.
10. Számítsd ki két térvektor vektoriális szorzatát (koordináták használatával).
11. (a) Add meg egy koordinátákkal adott pont távolságát az origótól.  
(b) Két tetszőleges, koordinátaival adott pont távolságát add meg.
12. Add meg egy másodfokú egyenlet megoldásait.
13. Számítsd ki egy síkbeli koordinátákkal megadott háromszög szögeit.
14. Döntsd el egy szövegről, hogy nagybetűvel kezdődik-e.
15. Döntsd el egy szövegről, hogy számjegyre végződik-e.
16. (a) Add meg egy tetszőleges szöveg első szavát.  
(b) Egy tetszőleges szövegnek töröld le az első szavát.  
(c) Egy tetszőleges szöveg első szavát cseréld le egy másik, adott szóra.

### 2. VEZÉRLÉSI SZERKEZETEK

17. (a) Írj ki  $n$  darab  $*$ -ot. ( $n$ -et tetszőlegesen választhatod meg.)  
(b) Rajzolj ki egy  $n \times n$ -es négyzetet  $*$ -okból.  
(c) Rajzolj ki egy  $n \times m$ -es téglalapot  $*$ -okból.  
(d) Rajzolj ki egy  $n$  hosszú befogójú, egyenlő szárú derékszögű háromszöget  $*$ -okból.  
(e) Rajzolj ki egy csúcsára állított  $n$  hosszú befogójú, egyenlő szárú derékszögű háromszöget  $*$ -okból.  
(f) Rajzolj ki egy  $n$  oldalhosszúságú, csúcsára állított rombuszt  $*$ -okból.  
(g) Rajzolj ki egy  $n \times n$ -es sakktáblát, a sötét mezőket  $**$ , a világosakat szóközők jelöljék.

18. Sorold fel két pozitív egész szám közös osztóit.
19. Sorold fel az első  $n$  négyzetszámot.
20. Sorold fel a  $n$ -nál kisebb négyzetszámokat. ( $k$ -t tetszőlegesen választhatod meg.)
21. Add meg az  $n$ . Fibonacci-számot. A Fibonacci sorozat egész számokból áll, az első két tagja 0 és 1, és minden további tagja az előző két tag összege.
22. Egész számhármassok tetszőleges sorozatát módosítsd úgy, hogy minden hármass növekvő sorrendben legyen.
23. Add meg egy tetszőleges számsorban az ismétlődő számokat.
24. Egy tetszőleges szövegben alakítsd át a kisbetűket nagybetűkké, a nagybetűket pedig kisbetűkké.
25. (a) Fordíts meg egy tetszőleges egész számsort.  
(b) Fordíts meg egy tetszőleges szöveget.
26. Egy tetszőleges szöveget módosítsd úgy, hogy a sorai elé írod az adott sor sorszámát.
27. Add meg egy tetszőleges szövegnek minden szavát külön-külön.
28. Adott egy szöveg, ami minden sorában szóközzel elválasztott egész számokat tartalmaz. Add meg minden sorhoz a benne található legnagyobb páros számot. (Vigyázz! Nem biztos, hogy minden sorban van páros szám!)

### 3. TÖMBÖK, ALPROGRAMOK

29. Valósítsd meg az **int kozos(int a, int b)** függvényt, ami a közös osztók számát adja vissza.
30. Valósítsd meg az **bool tokeletes(int a)** függvényt, ami visszaadja, hogy a paraméterül kapott érték tökéletes szám-e.
31. Valósítsd meg az **bool baratsagos(int a, int b)** függvényt, ami visszaadja, hogy a paraméterül kapott értékek barátságos számpárt alkotnak-e.
32. Valósíts meg függvényt, amely egy szöveget átalakít úgy, hogy ha több whitespace karakter (szóköz, tabulátor, újsor) van benne egymás után, azt egyetlen szóközzé alakítja.
33. Valósíts meg függvényt, ami egész számból szóveges változót csinál, ami előjelet és a megfelelő számjegyeket tartalmazza, felesleges karakterek nélkül.
34. Valósítsd meg a **void alahuzvakiir(string s)** függvényt, ami a paraméterül kapott szöveget új sorba kírja, és "=" karakterekkel aláhúzza.
35. Valósítsd meg az **int max(int[] t)** függvényt, ami a paraméterül kapott tömb elemei közül a legnagyobbat adja vissza.
36. Valósítsd meg az **bool vane(int[] t, int ez)** függvényt, ami eldönti, hogy a paraméterül kapott tömb elemei között van-e "ez".
37. Valósítsd meg a **float atlag(float[] t)** függvényt, ami a paraméterül kapott tömb átlagát adja vissza.
38. Valósítsd meg az **int hany(float[] t, double ez)** függvényt, ami a paraméterül kapott tömbben megszámolja, hogy hány "ez" van benne.
39. Valósítsd meg a **int szamol(string[] t, string ez)** függvényt, ami a paraméterül kapott tömbben megszámolja, hogy hány "ez" van benne.
40. Valósítsd meg az **bool vane(string[] t, string ez)** függvényt, ami eldönti, hogy a paraméterül kapott tömb elemei között van-e "ez".

#### 4. ELEMI PROGRAMOZÁSI TÉTELEK

41. Számítsd ki egy szám faktoriálisát.
42. Számítsd ki egy tetszőleges számsorozat átlagát.
43. Számítsd ki egy tetszőleges számsorozat szórását.
44. Add meg egy természetes szám valódi osztóinak összegét.
45. Add meg egy tetszőleges egész szám valódi osztóinak a számát.
46. Add meg egy természetes szám legnagyobb valódi osztóját.
47. Add meg két természetes szám legnagyobb közös osztóját.
48. (a) Sorold fel az első  $n$  tökéletes számot (olyan természetes számot, amely megegyezik a valódi osztóinak összegével).  
(b) Sorold fel a  $k$ -nál kisebb tökéletes számokat.
49. Add meg egy tetszőleges egész számsorról, hogy hány eleme nagyobb, ill. kisebb az átlagánál.
50. Egy pozitív egész számokból álló számsorban add meg, hogy hány páros szám van.
51. Egy tetszőleges számsorban add meg a legkisebb és a legnagyobb számot.
52. Add meg egy tetszőleges pozitív számsorozat elemeinek a négyzetgyök-összegét.
53. Számítsd ki két  $n$  dimenziós vektor skaláris szorzatát.
54. (a) Egy szigorúan növekedő egész számsorban add meg a legnagyobb ugrást (szomszédos elemek közötti legnagyobb előforduló különbséget).  
(b) Nem monoton számsorra is adj helyes eredményt.
55. Add meg egy tetszőleges egész számsorban a szomszédos elemek átlagos különbségét.
56. Tetszőleges sok számról dönts el, hogy növekedő sorrendben vannak-e.
57. Egy egész számról dönts el, hogy prímszám-e.
58. (a) Sorold fel az első  $n$  prímszámot.  
(b) Sorold fel a  $k$ -nál kisebb prímszámokat.  
(c) Sorold fel az  $a$  és  $b$  közé eső prímszámokat.
59. Egy tetszőleges szövegről add meg, hány kis "a" betű van benne.
60. Egy tetszőleges szövegről add meg, hány számjegy, hány nagybetű és hány kisbetű van benne.
61. Add meg egy tetszőleges szövegben, hogy melyik karakter fordul elő benne a legtöbbször.
62. Add meg egy tetszőleges szövegből a leghosszabb sort.
63. Döntsd el egy tetszőleges szövegről, hogy a sorai ábécé sorrendben vannak-e.
64. (a) Egy tetszőleges szövegben add meg a sorok számát.  
(b) Add meg a karakterek számát is.  
(c) Add meg a szavak számát is.
65. Add meg egy tetszőleges szöveg leghosszabb szavát.
66. (a) Egy tetszőleges szövegben számold meg, hány sor kezdődik azzal a betűvel, amivel az előző végződött.

- (b) Azt is számold meg, hogy hány szó kezdődik a megelőző szó utolsó betűjével.
67. (a) Egy tetszőleges szövegről add meg, hány mondat található benne. Mondatnak tekintünk minden olyan sort, ami nagybetűvel kezdődik, és ponttal, felkiáltójellel vagy kérdőjellel végződik.
- (b) A több mondatot tartalmazó sorokat és a többsoros mondatokat is kezeld helyesen.
68. (a) Egy tetszőleges szövegről add meg, hány szóból áll. Felteheted, hogy két szót mindig pontosan egy szóköz választ el.
- (b) Akkor is működjön, ha két szó között több szóköz is lehet.
69. Egész számhármások tetszőleges sorozatáról add meg, hogy a hármások közül hánynak vannak növekvő sorrendben az elemei.
70. (a) Háromszögek oldalhosszainak egy tetszőleges sorozatában (pozitív számhármások sorozata) add meg a legnagyobb kerületű háromszöget.
- (b) Add meg a legnagyobb területű háromszöget is.
- (c) Szűrd ki a sorozatból azokat a számhármásokat, melyekből nem építhető háromszög.
71. Tetszőleges, a csúcsai koordinátaival adott sokszögnek add meg a kerületét.
72. Tetszőleges, koordinátákkal adott pontsorozatból add meg, hogy mennyi esik az origó körüli  $r$  sugarú körön belülre.
73. Tetszőleges, koordinátákkal adott pontsorozatban add meg az origótól legtávolabb eső pontot.

## 5. ÖSSZETETT PROGRAMOZÁSI TÉTELEK

74. Add meg egy tetszőleges egész számsorban a prímszámok számát.
75. Add meg, hogy az  $a$  és  $b$  közötti egész számok közül melyiknek van a legtöbb valódi osztója.
76. Adott egy szöveg, ami minden sorában egész számokat tartalmaz. Add meg, hogy melyik sorban a legnagyobb a sor legkisebb száma (és azt is, hogy mi ez a szám).
77. (a) Egy több soros szövegben add meg, hány sorában található meg az "alma" szó.
- (b) Az "alma" helyett tetszőleges szöveget lehessen megadni.
- (c) Add meg a szó összes előfordulásának számát.
78. Mátrixban tároljuk egy osztály adatait, minden sora egy diák, minden oszlopa egy tantárgy, a mátrix értékei a jegyek. Add meg a következőket:
- (a) Osztályátlag.
- (b) Legjobb tanuló (átlag alapján).
- (c) Legnehezebb tantárgy (legtöbb bukás).
- (d) Van-e két hallgató, akiknek egyforma az átlaguk?
- (e) Hányan nem buktak meg semmiből?
- (f) A legjobb tanuló legrosszabb jegye.
- (g) Ki a legrosszabb átlagú hallgató azok közül, akik nem buktak meg semmiből?
- (h) Melyik hallgató jegyeinek a legnagyobb a szórása?
79. Egy vonalban szabályos távolságonként megmértük a tengerszint feletti magasságot egyik szárazföldi ponttól a másikig. Add meg a következőket:

- (a) Található-e hegycsúcs a területen (olyan pont, amely az előtte és utána lévőnél is magasabb).
  - (b) Hány sziget található a területen (sziget azon tengerszint feletti magasságok sorozata, amely előtt és után is tengerszint alatti magasság van).
  - (c) Milyen hosszú a legnagyobb (legtöbb mérésből álló) sziget.
  - (d) Van-e olyan sziget, amelyben völgy található.
  - (e) Hány hegycsúcs található a legnagyobb szigeten.
  - (f) Hány hegycsúcs alakú sziget van (ahol a hegycsúcs előtt monoton nőnek, utána monoton csökkenek az értékek).
80. Egy szöveges fájl minden sora egy mondatot tartalmaz, amelyek tetszőleges számú szóból állhatnak. Minden mondatnak van egy írásjel a végén, de ezen kívül nincs más írásjel a mondatban.
- (a) Hányadik mondatban található a legtöbb szó?
  - (b) Melyik a leghosszabb szó a teljes szövegben?
  - (c) Van-e olyan mondat a szövegben, amely legalább 10 szót tartalmaz?
  - (d) Hányadik mondatban található a legtöbb "s" betű?
  - (e) Van-e olyan mondat, amely évszámot (négyjegyű pozitív számot) tartalmaz?
  - (f) Mennyi a szavak hosszának átlaga a legtöbb szót tartalmaz mondatban?
  - (g) Hányadik mondatban található a legtöbb határozott névelő ("a", "az")?
81. Egy szöveges fájl minden sora egy egész számsorozatot tartalmaz (mindegyik sor tetszőleges hosszú, akár üres is lehet). Add meg a következőket:
- (a) Melyik sor összege a legnagyobb.
  - (b) Melyik sorban lévő számok abszolút értékeinek összege a legnagyobb.
  - (c) Melyik sorban található páros sok szám.
  - (d) Melyik sorban található a legnagyobb páros szám.
  - (e) Van-e olyan sor, amelyben csak egyféle érték fordul elő.
  - (f) Hány olyan sor van, amelyben a számok átlaga éppen nulla.
  - (g) Hányadik sorban van a legnagyobb szám.

## II. STRUKTÚRÁLT PROGRAMOZÁS

### 1. REKORDOK - struct

82. Valósítsuk meg a kör típust, ahol a kört a középpontjával és sugarával reprezentáljuk. Legyen lehetőség a terület és a kerület lekérdezésére, illetve annak megvizsgálására, hogy egy adott pont benne van-e egy körben. Ehhez valósítsuk meg a síkbeli pont típusát is. A főprogram rögzítsen egy síkbeli pontot, töltsön fel egy tömböt körökkel, majd minden olyan kör paramétereit írja ki, amely a rögzített pontot tartalmazza.
83. Valósítsuk meg a kör típust, ahol a kört a középpontjával és sugarával reprezentáljuk. Legyen lehetőség a terület és a kerület lekérdezésére, valamint a körök eltolására adott vektorral. Ehhez valósítsuk meg a síkbeli pont, valamint a vektor típusait is, amelyek biztosítják pont vektorral való eltolását. A főprogram rögzítsen egy síkvektort, töltsön fel egy tömböt körlemezekkel, majd minden körlemezt toljon el az adott vektorral, és írja ki az eltolt körök paramétereit.
84. Valósítsuk meg a háromszög típust, ahol a háromszöget a három oldal hosszával reprezentáljuk. Legyen lehetőség a terület és a kerület lekérdezésére. Legyen lehetőség továbbá konstruktorral véletlenszerű (0-100 közötti) oldalhosszúsággal rendelkező háromszöget generálni, odafigyelve, hogy a generált háromszög nem elfajuló kell legyen. A főprogram töltsön fel egy tömböt háromszögekkel, majd minden háromszög esetén írja ki a kerületet és a területet.
85. Valósítsuk meg a paralelogrammák típusát, amelyeket az oldalaik hosszával, és az egyik hegyes- (vagy derék) szögével reprezentálunk. Egy paralelogrammának le lehet kérdezni a szögeit, a területét és a kerületét. A főprogram töltsön fel egy tömböt paralelogrammákkal, és adja meg, hány négyzet található a paralelogrammák között.
86. Valósítsuk meg a derékszögű trapézok típusát, amelyeket három oldalhosszal reprezentálunk (a program rögzítheti, melyik három oldal értékeit használja minden trapézra). Egy trapéznek le lehet kérdezni az oldalainak hosszát, a szögeit, a területét és a kerületét. A főprogram töltsön fel egy tömböt trapézokkal, és adja meg, melyik trapéznak a legkisebb a legnagyobb szöge.
87. Valósítsuk meg a konvex deltoidok típusát, amelyeket az átlójuk hosszával, és azzal az aránnyal reprezentálunk, amely megmutatja, hogy hol metszi egyik átló a másikat. Egy deltoidnak le lehet kérdezni az oldalainak hosszát, a szögeit, valamint a területét és a kerületét. A főprogram töltsön fel egy tömböt deltoidokkal, és keresse meg, van-e négyzet a deltoidok között.
88. Valósítsuk meg a kockák típusát, ahol a kockát középpontjával, valamint oldalhosszával ábrázolunk. Ehhez valósítsuk meg a térbeli pont típust is. Egy kockának lekérdezhethetjük a felszínét, térfogatát, valamint megállapíthatjuk, hogy egy kocka tartalmaz-e egy másikat, illetve érintkezik-e vele (részben tartalmazza). A főprogram rögzítsen egy kockát, továbbá töltsön fel egy tömböt kockákkal, és adja meg, hány kocka tartalmazza a kijelöltet, mennyi érintkezik vele, illetve mennyit tartalmaz.
89. Valósítsuk meg a 3 dimenziós vektorok típusát, ahol a vektort három koordinátájával ábrázoljuk. Egy vektornak le tudjuk kérdezni a hosszát, negálhatjuk, valamint szorozhatjuk egy skalár értékkel (ekkor az a hosszát módosítja). Két vektort összeadhatunk, vehetjük a különbségüket, illetve lekérdezhethetjük a távolságukat. A főprogram töltsön fel egy tömböt vektorokkal, és adja meg az összesített vektort.
90. Valósítsuk meg a komplex számok típusát, ahol a komplex számokat algebrai alakjukkal (valós és a képzetes rész) ábrázoljuk. Implementáljuk a komplex számok összeadását:  $(a + bi) + (c + di) = (a + c) + (b + d)i$  és szorzását:  $(a + bi)(c + di) = (ac - bd) + (ad + bc)i$ . A főprogram töltsön fel egy tömböt komplex számokkal, amelyre ezután számolja ki az elemek összegét és szorzatát.

## 2. OSZTÁLYOK - class

91. Készíts alkalmazást a céllövészet bemutatására. A `Céltábla` típus minimálisan a következő tagokkal rendelkezzen:
- (a) Tulajdonságok: a céltábla középpontja (`Pont` típus) és sugara.
  - (b) Kontruktor: véletlenszerű középponttal és sugárral generált céltábla.
  - (c) Metódusok:
    - i. `bool Eltalálta( Pont lövés )`: megadja, hogy egy tetszőleges lövés eltalálta-e a céltáblát vagy sem.
    - ii. `double Távolság( Pont lövés )`: megadja, hogy egy tetszőleges lövés, mely eltalálta a céltáblát, milyen messze van a középponttól.
    - iii. `int Pontszám( Pont lövés )`: a középponttól való távolság alapján a találathoz pontszámot rendel.
92. Készíts alkalmazást a hallgatói élet bemutatására. A `Hallgató` típus minimálisan a következő tagokkal rendelkezzen:
- (a) Tulajdonságok: Neptunkód, név, energia (0 és 10 között), kedv (0 és 10 között).
  - (b) Kontruktor: kezdetben az energia és a kedv is 10.
  - (c) Metódusok:
    - i. `void Tanul()`: ekkor csökken az energiája 1-gyel, csökken a kedve 1-gyel.
    - ii. `void Vizsgál()`: ekkor csökken az energiája 3-mal, csökken a kedve 2-vel.
    - iii. `void Bulizik()`: ekkor csökken az energiája 1-gyel, nő a kedve 3-mal.
    - iv. `void Alszik()`: ekkor nő az energiája 5-tel.
93. Készíts alkalmazást az oktatói élet bemutatására. Az `Oktató` típus minimálisan a következő tagokkal rendelkezzen:
- (a) Tulajdonságok: Neptunkód, név, lelkerő (0 és 20 között).
  - (b) Kontruktor: kezdetben a lelkerő 20.
  - (c) Metódusok:
    - i. `void ŐrátTart( int részvételiArány )`: a részvételi arány függvényében változik a lelkerő. 50%-os részvétel felett 3-mal nő, alatta 3-mal csökken.
    - ii. `void Vizsgáztat( double átlag )`: ha a vizsgaeredmények átlaga több, mint 3.5, akkor a lelkerő 1-gyel nő, ha kevesebb, 2-vel csökken.
94. Készíts alkalmazást egy teherhajó bemutatására. A `Teherhajó` típus minimálisan a következő tagokkal rendelkezzen:
- (a) Tulajdonságok: név, kapacitás (hány konténer fér el), célállomás (a célkikötő neve), konténerek száma (aktuálisan hány konténer van a hajón).
  - (b) Kontruktor: adott névvel, kapacitással, konténerek számával generált teherhajó.
  - (c) Metódusok:
    - i. `void Kiköt( string kikötőNeve, bool engedélyezve )`: ha a kikötés engedélyezve, akkor a hajó kiköthet a kikötőben.

- ii. `void Kifut( string célkikötőNeve, bool engedélyezve )`: ha a kikötés engedélyezve, akkor a hajó elhagyja a régi kikötőt.
  - iii. `void Kipakol()`: ekkor a hajó megszabadul az összes konténertől.
  - iv. `void Felpakol( int konténerekSzama )`: ekkor a hajó új konténereket kap.
  - v. `void CélkikötőtBeállít( string célkikötőNeve )`: ha a hajó lepakolta a szállítmányt, akkor új célkikötőt állít be.
95. Készíts alkalmazást egy versenyautó bemutatására. A `Versenyautó` típus minimálisan a következő tagokkal rendelkezzen:
- (a) Tulajdonságok: versenyző neve, tank szintje (1 és 100 között), aktuális helyezés.
  - (b) Konstruktor: adott névvel generált versenyautó. Kezdetben a tank tele van, és az aktuális helyezés a rajtsorrendet tükrözi.
  - (c) Metódusok:
    - i. `void TeljesítEgyKört()`: ekkor 5 egységgel csökken a tank szintje.
    - ii. `void Előz()`: ekkor további 4 egységgel csökken a tank szintje, illetve baleset következhet be úgy, hogy 4% az esélye, hogy kiesik a támadó, és 2% az esélye, hogy mind a támadó, mind az előtte lévő kiesik.
    - iii. `void Tankol()`: ekkor tele lesz a tank, viszont 3 helyezéssel hátrébb kerül.
96. Készíts alkalmazást egy kaktusz bemutatására. A `Kaktusz` típus minimálisan a következő tagokkal rendelkezzen:
- (a) Tulajdonságok: azonosító, méret, túlburjánzás határa.
  - (b) Konstruktor: adott azonosítóval, kezdő mérettel és túlburjánzási határral generált kaktusz.
  - (c) Metódusok:
    - i. `void Táplálkozik( string víz|tápanyag|műtrágya )`: víz esetén a mérete eggyel csökken, tápanyag esetén eggyel nő, műtrágya esetén eggyel nő. Ha túlburjánzik, azaz a mérete nagyobb lesz, mint a kezdeti méret + túlburjánzás mértéke, akkor a növény kipusztul. Ha a mérete nullára csökken, akkor is kipusztul.
    - ii. `void Sorvad()`: a mérete eggyel csökken. Ha a mérete nullára csökken, akkor kipusztul.

### 3. ÖSSZETETT FELADAT

Készítsünk programot, mely egy kereskedelmi kikötő egyszerűsített működését követi.

A kikötőbe konténereket szállító hajók köthetnek ki, és futhatnak ki onnan. Egy hajó kikötését a kikötőnek engedélyeznie kell: ha a hajó célállomása megegyezik a kikötő nevével, és a hajón lévő konténerek száma kisebb vagy egyenlő, mint a kikötő kapacitásából levonva a kikötőben található konténerek számát, akkor a hajó kiköthet.

Miután egy hajó engedélyt kapott a kikötésre, kipakolja az összes konténerét a kikötőbe, és beállítja új célállomását, mely nem egyezhet meg annak a kikötőnek a nevével, amelyben éppen tartózkodik. Ezután a hajó konténereket pakol fel a kikötőből a következő feltételekkel: ha a konténer célállomása megegyezik a hajó célállomásával, és a hajó kapacitása még engedi, akkor kerül egy konténer a hajóra.

A hajó kifuthat a kikötőből minden további engedély nélkül, de csak akkor, ha nem üres.



**OBJEKTUMOK ÉS TULAJDONSÁGAIK**

## Kikötő

- név
- konténerek maximális száma (kapacitás)

## Hajó

- név az azonosításhoz, csak a józan ész alapján, mert a feladat leírásából nem derül ki
- célállomás
- konténerek maximális száma (kapacitás)

## Konténer

- célállomás
- *lehetne egyedi azonosítója, de felesleges, mert csak a konténerek száma és célállomása fontos*

**OBJEKTUMOK ÉS METÓDUSAIK**

## Egy kikötő

- engedélyt ad egy hajónak (kikötésre)

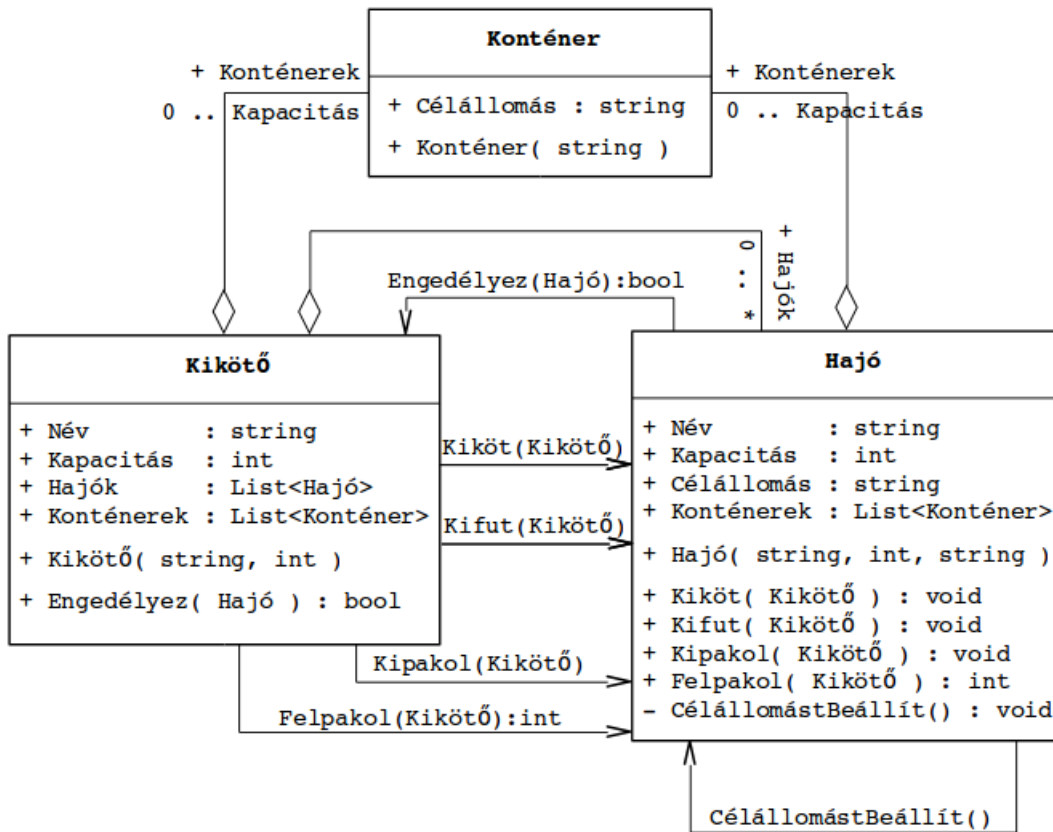
## Egy hajó

- kiköt egy kikötőbe (engedélyhez van kötve)
- kipakolja konténeit a kikötőbe
- új célállomást állít be
- felpakol konténereket a kikötőből
- kifut a kikötőből (nincs engedélyhez kötve)

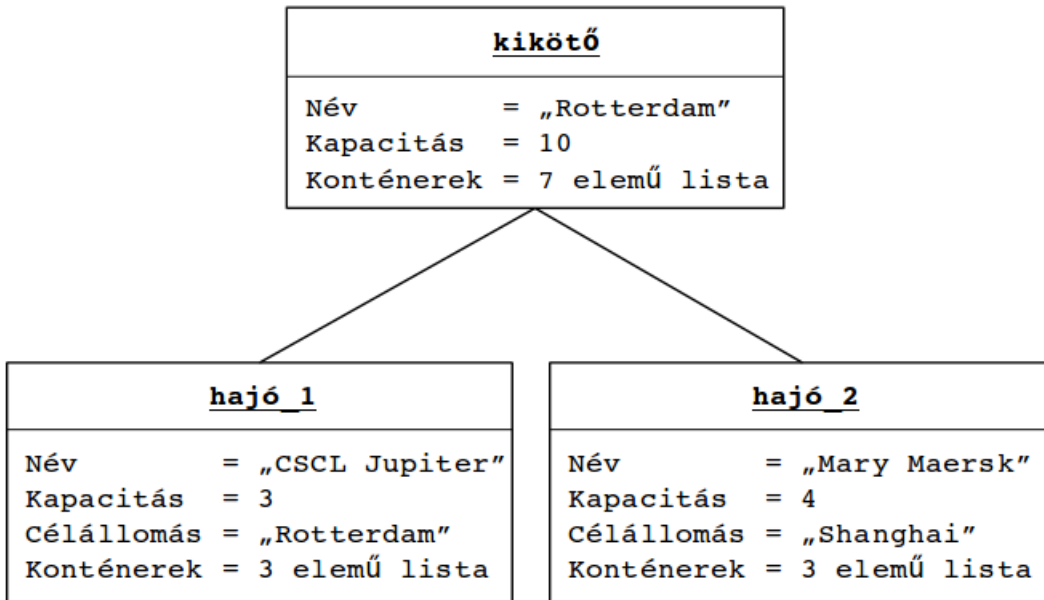
Egy konténer nem kezdeményez, passzív objektum. Önmagában nem létezik, kizárólag hajón vagy kikötőben fordul elő.

Lásd alább a tervezés diagramjait.

OSZTÁLYDIAGRAM



OBJEKTUMDIAGRAM



## KOMMUNIKÁCIÓS DIAGRAM

