

## 2. Karácsonyi díszek

40 pont

Egy kézműves termékeket készítő és áruló hölgy minden évben karácsony előtt 40 nappal kezd el a karácsonyi díszek készítését és árusítását. A hölgy háromféle horgolt karácsonyi díszet készít és árul: harangot, angyalkát és fenyőfát. Az idei év forgalmáról napi szinten statisztikát vezet, hogy a vállalkozása az elkövetkező évben ez alapján eredményesebb legyen.

Ebben a feladatban az elkészített és az eladott díszek adataival kell dolgoznia. A megoldás során vegye figyelembe a következőket:

- *Megoldását választása szerint Java vagy C# programozási nyelven kell elkészítenie! Az Ön által választott programozási nyelvet jelölje meg a feladatlap fedőlapján! A javítás során csak a megjelölt nyelven készült megoldás lesz értékelve!*
- *A képernyőre írást igénylő részfeladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például: 3.feladat:)!*
- *Az egyes feladatokban a kiírásokat a minta szerint készítse el!*
- *Az ékezetmentes kiírások is elfogadottak!*
- *Az azonosítókat kis- és nagybetűkkel is kezdheti!*
- *A program megírásakor az állományban lévő adatok helyes szerkezetét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek!*
- *A megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon!*

A nap során elkészített és eladott díszek számát minden nap végén a `díszek.txt` fájlba jegyzi fel a hölgy. A `díszek.txt` forrásállomány minden sorában egy-egy nap záróadatai találhatók:

- a nap sorszáma,
- az elkészített harangok száma,
- az eladott harangok száma,
- az elkészített angyalkák száma,
- az eladott angyalkák száma,
- az elkészített fenyőfák száma,
- és az eladott fenyőfák száma.

A soron belül az adatokat pontosvessző választja el egymástól. Az elkészített díszek száma mindig pozitív vagy 0 értékű, míg az eladott díszek száma mindig negatív vagy 0 értékű.

Például:

```
1;3;0;0;0;4;0
2;4;0;5;-2;1;0
3;1;-2;4;-3;0;0
4;0;0;3;-1;0;0
```

Az első napon a hölgy 3 harangot és 4 fenyőfát készített, de nem adott el egyetlen díszet sem. A második napon 4 harangot, 5 angyalkát és 1 fenyőfát készített, és eladott 2 angyalkát. Így a második nap végére 7 harang, 3 angyalka és 5 fenyőfa van készleten. A harmadik napon 1 harangot és 4 angyalkát készített, illetve eladott 2 harangot és 3 angyalkát. Ezért a harmadik nap végére 6 harang, 4 angyalka és 5 fenyőfa van készleten.

1. Készítsen konzolos alkalmazást a következő feladatok megoldására, amelynek projektjét `karacsonyCLI` néven mentse el!

*A feladat a következő oldalon folytatódik*

2. Forráskódjába tegye elérhetővé a `java.txt` vagy a `csharp.txt` állományból a `NapiMunka` osztályt definiáló kódrészletet! A beillesztett osztály tetszés szerint bővíthető további tagokkal, melyek a feladatok megoldása során felhasználhatók! Az osztály `NapiBevetel()` metódusa felhasználható arra, hogy megállapítsa az aktuális nap bevételét. Ezt a metódust a későbbiekben kell felhasználnia.
3. Olvassa be a `diszek.txt` állományban lévő adatokat, és tárolja el egy olyan adatszerkezetben, ami a további feladatok megoldására alkalmas!
4. Határozza meg és írja ki a képernyőre a minta szerint, hogy összesen hány karácsonyi dísz készített a hölgy!
5. Állapítsa meg, hogy volt-e olyan nap, amikor a hölgy egyetlen dísz sem készített! A keresést ne folytassa, ha választ meg tudja adni! A megállapítását írja a képernyőre!
6. Kérjen be a felhasználótól egy 1 és 40 közé eső számot (a határokat is beleértve)! Ismételje addig a nap számának bekérését, míg érvényes értéket nem ad meg a felhasználó! Ha nem tudta megoldani az adatbevitelt, akkor a feladat további részében dolgozzon a 15-ös számmal! Határozza meg, és írja a képernyőre, hogy az adott nap végén melyik díszből hány maradt készleten!
7. Határozza meg, és írja a képernyőre, hogy a 40 nap alatt melyik díszből sikerült eladni a legtöbbet! Az eladott mennyiséget is jelenítse meg! Ha több díszből is egyformán a legtöbb lett eladva, akkor mindegyiket jelenítse meg!
8. A `NapiMunka` osztály `NapiBevetel()` metódusának felhasználásával válogassa ki és írja ki a `bevetel.txt` fájlba azokat a napi bevételeket, melyek elérték a 10 000 forintot! Minden sorban jelenjen meg a nap száma és az aznapi bevétel egymástól kettősponttal elválasztva. A fájl utolsó sorában jelenítse meg, hogy hány olyan nap volt, amikor a bevétel elérte a 10 000 forintot! „*X napon volt legalább 10000 Ft a bevétel.*”
9. Készítsen grafikus alkalmazást, amelynek projektjét `karacsonyGUI` néven mentse el! A grafikus alkalmazást a hölgy arra szeretné használni, hogy a későbbiek során ne fájlba kelljen írnia az adatokat, hanem helyette egy könnyen kezelhető grafikus felületen tudja megadni a napi zárás adatait. A grafikus felületet csak az anygalkák számának kezeléséhez kell elkészítenie. Az alkalmazásban megadhatja, hogy melyik napon hány anygalka készült el és mennyit sikerült eladni. Ezek alapján az alkalmazásnak a készleten lévő anygalkák számát is kell számolnia, és meg kell jelenítenie.
10. A grafikus alkalmazásban a következő feladatokat végezze el:
  - a. Alakítsa ki a felhasználói felületet a következő minta szerint! Állítsa be az alkalmazás címsorában megjelenő feliratot! A beviteli mezők értéke a program indulásakor 0 legyen, a lenyíló lista 1 – 40 -ig tartalmazza a számokat! A két beviteli mező szám megadására szolgál, de ezt nem kell ellenőriznie, feltételezheti, hogy a felhasználó számot fog megadni.
  - b. A „Hozzáadás” parancsgomb lenyomásának hatására a minta szerinti formában jelenítse meg a választott napot, az elkészített és az eladott anygalkák számát, illetve a nap végén készleten lévő anygalkák darabszámát. (A mintában a megjelenített sorok elemei tabulátorral vannak tagolva, de eltérő tagolás használata is teljesértékű megoldásnak számít.)  
Hozzáadás előtt a megadott sorrendben végezze el a következő ellenőrzéseket! Ha valamelyik feltétel nem teljesül, akkor csak a megfelelő hibaüzenetet jelenítse meg, és ne folytassa tovább az ellenőrzés / hozzáadás folyamatát! A hibaüzenet minden esetben piros betűszínnel jelenjen meg! Feltételezheti, hogy a lenyíló listában mindig van kiválasztott érték.

*A feladat a következő oldalon folytatódik*

- i. A két beviteli mező egyikében sem szerepelhet negatív szám. Ha valamelyik esetben ez nem teljesül akkor a minta szerinti hibaüzenetet jelenítse meg!

Nap száma:	<input type="text"/>	Elkészített:	<input type="text" value="-3"/>	Eladott:	<input type="text" value="5"/>
Negatív számot nem adhat meg!					

- ii. Az eladott beviteli mezőbe került értéknek kisebbnek kell lennie, mint a most elkészített és a készleten lévő darabszámok összege! Ha ez nem teljesül, akkor a minta szerinti hibaüzenetet jelenítse meg!

Nap száma:	<input type="text" value="3"/>	Elkészített:	<input type="text" value="2"/>	Eladott:	<input type="text" value="5"/>										
Túl sok az eladott anygalka!															
<table border="1"> <tr> <td>1.nap:</td> <td>+0</td> <td>-0</td> <td>=</td> <td>0</td> </tr> <tr> <td>2.nap:</td> <td>+8</td> <td>-7</td> <td>=</td> <td>1</td> </tr> </table>						1.nap:	+0	-0	=	0	2.nap:	+8	-7	=	1
1.nap:	+0	-0	=	0											
2.nap:	+8	-7	=	1											

- c. Hozzáadás után a beviteli mezők értéke legyen 0, a lenyíló listából pedig legyen eltávolítva a kiválasztott nap és az azt megelőző napok!
- d. Ügyeljen arra, hogy a megjelenített sorokat a felhasználó ne tudja átírni, módosítani! Az összes sornak láthatónak kell lennie, ehhez használjon függőleges gördítősávot!

Nap száma:	<input type="text" value="3"/>	Elkészített:	<input type="text" value="3"/>	Eladott:	<input type="text" value="1"/>	Hozzáad										
<table border="1"> <tr> <td>3.nap:</td> <td>+3</td> <td>-1</td> <td>=</td> <td>2</td> </tr> </table>							3.nap:	+3	-1	=	2					
3.nap:	+3	-1	=	2												
Nap száma:	<input type="text" value="6"/>	Elkészített:	<input type="text" value="7"/>	Eladott:	<input type="text" value="3"/>	Hozzáad										
<table border="1"> <tr> <td>3.nap:</td> <td>+3</td> <td>-1</td> <td>=</td> <td>2</td> </tr> </table>							3.nap:	+3	-1	=	2					
3.nap:	+3	-1	=	2												
Nap száma:	<input type="text" value="11"/>	Elkészített:	<input type="text" value="0"/>	Eladott:	<input type="text" value="0"/>	Hozzáad										
<table border="1"> <tr> <td>3.nap:</td> <td>+3</td> <td>-1</td> <td>=</td> <td>2</td> </tr> <tr> <td>6.nap:</td> <td>+7</td> <td>-7</td> <td>=</td> <td>0</td> </tr> </table>							3.nap:	+3	-1	=	2	6.nap:	+7	-7	=	0
3.nap:	+3	-1	=	2												
6.nap:	+7	-7	=	0												

*A feladat a következő oldalon folytatódik*

**Konzol alkalmazás minta:**

```
4.feladat: Összesen 318 darab dísz készült.  
  
5.feladat: Volt olyan nap, amikor egyetlen dísz sem készült.  
  
6.feladat:  
Adja meg a keresett napot [1 ... 40]: 0  
Adja meg a keresett napot [1 ... 40]: 45  
Adja meg a keresett napot [1 ... 40]: 15  
    A(z) 15. nap végén 6 harang, 3 angyalka és 13 fenyőfa maradt készleten.  
  
7.feladat: Legtöbbet eladott dísz: 107 darab  
    Angyalka  
    Fenyőfa
```

**Grafikus alkalmazás minta:**

Angyalka

Nap száma:  Elkészített:  Eladott:

7.nap:	+0	-0	=	5
8.nap:	+0	-0	=	5
9.nap:	+0	-0	=	5
10.nap:	+12	-4	=	13
11.nap:	+0	-0	=	13
12.nap:	+0	-0	=	13
13.nap:	+0	-2	=	11
14.nap:	+0	-0	=	11
15.nap:	+0	-0	=	11
16.nap:	+0	-0	=	11
17.nap:	+14	-6	=	19
18.nap:	+0	-0	=	19
19.nap:	+0	-0	=	19
20.nap:	+0	-0	=	19
21.nap:	+0	-0	=	19