*c# DOLGOZAT*

***Gyüjtemények és file kezelés***

***2024***

|  |  |  |
| --- | --- | --- |
| ***Osztályzat*** | ***Feltétel*** | ***Pontok bónusz pontok nélkül*** |
| ***1*** | ***0 – 39%*** | ***0 - 20*** |
| ***2*** | ***40 – 55%*** | ***21 - 28*** |
| ***3*** | ***56 – 75%*** | ***29 - 39*** |
| ***4*** | ***75 - 89%*** | ***40 - 49*** |
| ***5*** | ***90 -100%*** | ***49 - 54*** |

**A dolgozat kidolgozásához készítsen egy mappát az asztalon a saját neve (Vezetéknév Keresztnév). Ebbe a mappába másolja át a \_dolgozat.docx-t a \\optimus\h\Vastag Atila\Leadando hálózati meghajtóról. Ebben találja a kidolgozandó feladato-kat.**

**A feladat gyakorlati részéhez az asztalon elkészített mappába készítsen egy feladat nevű mappát. Ebben a mappában oldja meg a dolgozat gyakorlati részét.**

**A dolgozatot leadásakor az asztalon elkészített mappát másolja fel az \\optimus\h\Vastag Atila\Leadando hálózati meghajtóra.**

1. **A listák homogén statikus vagy dinamikus gyűjtemények-e? Indokold meg válaszod! (1)**

**válasz: dinamikus gyűjtemény, mert hossza nem előre meghatározott de homogén statikus is mert csak ugyanolyan típusú adat lehet benne, kivéve ha típusa dynamic**

1. **Mi a listák előnye a tömbökhöz képest? (1)**

**válasz: hosszuk nem előre meghatározott**

1. **Példányosítás után mekkora egy lista hossza? Ha meg tudja mondani, hogy példányosításkor mennyi hely foglalóik le a memóriában a lista elemeinek vagy mely utasítással lehet a *List* konstruktorának megadni a példányosításkor lefoglalandó memória mennyiséget bónusz pont jár! (1 + 1 + 1)**

**válasz: 0, és 10**

1. **Ha egy elemet adunk hozzá a listához, az a listán belül hova kerül? (1)**

**válasz: a végére**

1. **Lehetséges-e egy listában egy bizonyos helyre elemet írni? (1)**

**Ha igen mutassa be példán! (2)**

**Mi történik, azzal az elemel, amely azon a helyen volt addig? (1)**

**Beszúrhatunk e elemet például a lista 5 helyére egy 3 elemű listában? Indokolja meg a választ! (2)**

**válasz: igen lehet**

List<int> list = new List<int>

{

1,

2,

3,

4,

5

};

list.Insert(3, 6);

hátrább tolódik

nem mert ha a hossza három akkor csak annyi hely van a memóriában, system has run out of memory hiba lesz

1. **A lista példányosításának általános alakja: (2)**

***List<T> list = new List<T<();***

**Mit jelent a T?**

**válasz:**

**generikus típus, bármilyen lehet**

**string, int, structure, class, stb…**

1. **Adott két *string* típusú lista:**

***List<string> ferfiNevek = new List<string<();***

***List<string> noiNevek = new List<string<();***

**Melyik függvény segítségével:**

* **adunk hozzá elemet valamelyik listához: (1) -> ferfiNevek.Add(„Károly”);**
* **adunk hozzá elemeket valamelyik listához: (1) -> noiNevek.Add(„Móni”);**
* **szúrhatjuk be a 3. helyre a *Dalma* értéket a *noiNevek* listába: (1) -> noiNevek.Insert(2, „Dalma”);**
* **szúrunk be a *ferfiNevek* lista elejére elemet: (1) ->**
* **kérjük le valamelyik lista hosszát: (1) ->firfiNevek.Lenght;**
* **kérjük le az elem indexét valamelyik listában (feltételezzük a *noiNevek* listában szerepel a *Damla* név): (1) ->**
* **törlünk elemet valamelyik listából (feltételezzük a *noiNevek* listában szerepel a *Damla* név): (1) ->**

noiNevek.Remove("Dalma");

* **tartalmaz-e egy bizonyos elemet a lista (feltételezzük a *noiNevek* listában szerepel a *Damla* név): (1) ->**
* **noiNevek.Contains(„Dalma”);**
* **lehet a *noiNevek* lista végére fűzni a *ferfiNevek* listát: (1) ->**

**noiNevek.AddRange(ferfiNevek);**

**Válaszokban a teljes függvény használatát mutassa be!**

1. ***Dictionary működési elve? (1)***

**válasz:**

1. ***Miben jobb egy szótár, mint adatszerkezet a listától? (1)***

**válasz:**

1. **A program feladata kiírni 10 véletlenszerűen generált számot, ám hibás. Keresd meg a hibákat! A hibás kódrészeket húzza át és írja a helyére a jó választ! (5)**

*Random rnd = new Random();*

*int elemekSzama = 10;*

*for (int i = 0; i ~~>~~< elemekSzama; i~~--~~++)*

*{*

*intList[i] list = rnd.Next(-9, 9);*

*}*

*foreach (~~intList in int x~~int i in list)*

*{*

*Console.WriteLine($"{elemekSzama[i] }\t");*

*}*

**JOKER**

***A lista az referencia érték. Mit jelent ez? (5)***

**válasz**: indexek alapján lehet elérni az elemeit

**GYAKORLATI FELADAT** (26 + 2)

**FELADAT:**

A cégnél, ahol dolgozunk, minden héten, így ezen a héten is egy szöveges állományban kaptuk meg a dolgozóink nevét és azt hogy hétfőtől péntekig ki hány órát dolgozott és melyik projecten. Az adatok a következő képen vannak tárolva egy sorban:

Az első sorban a név található. Az azt követő sorban a project neve. Majd a következő sorban a ledolgozott órák *vesszővel* elválasztva (lásd az adatforrást), pl:

**Hapci**

**GIL**

**8,5,6,3,9**

Végezzük el az adatok beolvasását az **adatok.txt** állományból. **(5)**

feladat – 1: **(3)**

* Írjuk ki a munkások nevét, projectjét és napi ledolgozott órák számát a képernyőre a minta szerint:

*Feladat 1:*

*Dolgozók adatai:*

**Hérfő Kedd Szerda Csütörtök Péntek**

**Hapci GIL 8 5 6 8 9**

**Vidor KARMA 3 2 4 6 8**

**Tudor iiPay 6 7 6 8 5**

**.**

**.**

feladat – 2: **(2)**

* Milyen összeget kell kifizetni munkabérként, ha egy munkaóra 10.000HUF?

*Feladat 2:*

***Összesen 3,490,000 HUF munkabért kell kifizetni.***

feladat – 3: **(2 + 1 pont a megoldásért)**

* Ki dolgozta a héten a legtöbb munkaórát?

*Feladat 3:*

***A héten a legtöbb munkaórát Okoska dolgozta (37óra).***

feladat – 4: **(4)**

* Számítsuk ki a dolgozók fizetését, ha tudjuk, hogy 1 munkaóra az 10.000HUF és a **fizetes\_2024-03.txt** állományba mentsük (2024 a kifizetés éve, 03 a kifizetés hónapja) el a megadott minta szerint: **Hapci \t 360000 HUF).**

feladat – 5: **(5 + 2 pont a megoldásért)**

* Keresse ki csoportba bontva a dolgozókat ledolgozott órák után. Három csoport létezik: *rossz, átlagos* és *kiváló*. Rossznak az számít, aki nem érte el a heti 21 órát, átlagos az, aki 30 óráig teljesített és a kiváló csoportba az tartozik, aki a héten 30 óra feletti óraszámmal rendelkezik.

Az eredményt a mintának megfelelően írja ki a képernyőre. (a minta csak reprezentatív, nem a valós eredményt mutathatja).

*Feladat 5:*

*A dolgozók csoportba bontva:*

**rossz:**

* **Dundi (9)**
* **Lucifer (18)**

**átlagos:**

* **Vidor (23)**

**kiválló:**

* **Hapci (36)**
* **Tudor (32)**

feladat – 6: **(3)**

* Melyik projecten dolgozókra kell a kifizetés hetén a legtöbb forrást biztosítani?

A mintában a 2024 az elszámolás éve, 03 az elszámolás hónapja és a 04 az elszámolás hónapjának a hete. Az elszámolás hónap hetének meghatározására használhatja a megadott *GetWeekNumberOfMonth* függvényt!

*Feladat 6:*

**2024-03/04 heti kifizetéskor a KARMA csapatára kelet a legtöbb forrást biztosítani.**

int GetWeekNumberOfMonth(DateTime date)

{

date = date.Date;

DateTime firstMonthDay = new DateTime(date.Year, date.Month, 1);

DateTime firstMonthMonday = firstMonthDay.AddDays((DayOfWeek.Monday + 7 - firstMonthDay.DayOfWeek) % 7);

if (firstMonthMonday > date)

{

firstMonthDay = firstMonthDay.AddMonths(-1);

firstMonthMonday = firstMonthDay.AddDays((DayOfWeek.Monday + 7 - firstMonthDay.DayOfWeek) % 7);

}

return (date - firstMonthMonday).Days / 7 + 1;

}

bonus: **(4)**

* Ki/Kik és melyik nap dolgozta/dolgozták a legtöbb órát az elszámolás hetén? Az eredményt írja ki a képernyőre tetszőlegesen, de tartalmaznia kell a dolgozó nevét, ledolgozott óráinak számát és a napot!

Összesen elérhető bónusz pontok száma **13**.

**10** elért bónusz pont felett jár egy ajándék 5-s osztályzat!

Ügyeljünk az íratlan szabályok (konvenciók) betartására: változó nevek, függvény nevek, osztály nevek, könyvtárak nevének írására. Az egyes egészeket kötelező függvényekbe szervezni! Ügyeljünk arra, hogy egyes, a megoldás megkönnyítő segéd függvényeket logikailag a megfelelő helyen alkalmazzuk! **(2)**