



Visual C#

Programozás

gyakorlat – 14. hét

December 2

2019

Metódus

Feladatlap

Tartalom

01.	Feladat - Nagyobb	2
02.	Feladat - Dátum.....	2
03.	Feladat - Szorzótábla	2
04.	Feladat - Hatvány	2
05.	Feladat - Véletlen	2
06.	Feladat - Faktoriális	2
07.	Feladat – Háromszög kerület	3
08.	Feladat – A hét adott napja	3
09.	Feladat - Számrendszer	3
10.	Feladat – Kitalálós játék	3
11.	Feladat – Tökéletes szám	4
12.	Feladat – Hiányos szám	4
13.	Feladat – Bővelkedő szám	4
14.	Feladat – Barátságos számpár.....	4
15.	Feladat – Fájl kezelés.....	4

01. Feladat - Nagyobb

Írj függvényt, amely két valós szám közül a nagyobbikat (nem kisebbet) adja vissza!

02. Feladat - Dátum

Írj függvényt, amely egy rövid dátumot hosszú dátummá alakít!

- Pl. a `getHosszuDatum(2008,10,13)` függvényhívás a '2008. október 13.' értéket adja vissza.

Készíts `isSzokoEv()` néven metódust, amely eldönti egy évről, hogy az szökőév-e?

Készíts egy `isHelyesDatumE(2018,10,13)` logikai értéket visszaadó metódust, amely megállapítja a kapott egész értékekről, hogy azok helyes tartományból valók. Az elkészült metódust használd fel a `getHosszuDatum()` metódusban. A `getHosszuDatum()` csak akkor írjon ki eredményt, ha helyes értékeket adtak meg.

- Az év csak 1900 és 2100 közötti lehet.
- A hónap 1-12 közötti lehet.
- A nap az egyes hónapokhoz tartozóan (31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31) értékű lehet. Ügyelj arra az esetre is, ha szökőévet adnak meg, mert akkor a február lehet 29-es értékű is, így használd fel a korábban megírt szökőév metódust is.

03. Feladat - Szorzótábla

Írjunk metódust, ami egy paraméterül kapott n szám esetén egy $n \times n$ -es szorzótáblát rajzol a képernyőre.

- A kiírás formázott, esztétikus legyen, oszlopszerűen megjelenítve az értékeket.
- tartalmazzon egy fejléctet, és egy 0. oszlopot is, ahol megjelenítésre kerülnek a szorzandó értékek is

04. Feladat - Hatvány

Készíts hatvány függvényt és teszteld konstans paraméterrel!

- Írj programot, amely kiszámolja a felhasználó által megadott alapszám megadott kitevőjű hatványát!
- Írj programot, amely táblázatszerűen kiírja a felhasználó által megadott alapszám első 10 hatványát!
- Módosítsd az előző hatvány függvényt, amely kiszámolja egy valós szám tetszőleges (pozitív, negatív, vagy 0) kitevőjű hatványát!

05. Feladat - Véletlen

Írjunk véletleneket generáló metódusokat:

- Nem
- Monogram (Gondoskodj a két-három karakteres betűkről is.)
- Születési idő
- Telefonszám (A formátum mobil telefonszámok esetén +36 00 000 0000, míg a vezetékes számok esetén +36 00 000 000 legyen.)
- Rendszám (Formátuma AAA-000)

06. Feladat - Faktoriális

Írj programot, amely kiszámolja a felhasználó által megadott szám faktoriálisát!

- Ehhez készíts egy faktoriális függvényt! (Emlékeztető: $4! = 1 \cdot 2 \cdot 3 \cdot 4$)
- Írj programot, amely táblázatszerűen kiírja az első 5 pozitív egész szám faktoriálisát!

07. Feladat – Háromszög terület

Írjunk programot, ami kiszámolja két - a felhasználótól bekért sugarú kör területének a különbségét.

A sugarak bekéréséhez írjunk metódust beker néven. A metódusnak paramétere nincs, visszatérési értéke pedig az ellenőrzött bekért (0 és 10 közötti) valós érték.

Írjunk metódust a terület kiszámításához. A metódus neve korTerulet legyen. Paramétere egy valós szám, a kör sugara, visszatérési értéke is valós szám, a kör területe legyen.

A metódusokat felhasználva oldjuk meg a feladatot.

08. Feladat – A hét adott napja

Írjunk programot, ami ellenőrzött bekéri a felhasználótól, hogy a hét hányadik napja van most, ezután kiírja szövegesen, hogy milyen nap van ma, és milyen nap lesz holnap.

A main metódusban ez szerepeljen:

```
int nap = Beker();
Console.WriteLine("Ma {0} van, holnap {1} lesz", Hetnapja(nap), Hetnapja(nap+1));
```

09. Feladat - Számrendszer

Írjunk programot, ami számrendszerek között tud átváltani.

10. Feladat – Kitalálós játék

Írjunk számkitalálós játékot!

A játék kezdete előtt lehessen szintet választani. Három fokozat legyen, könnyű, közepes és nehéz. A szint kiválasztásához írjunk metódust, melynek paraméterei nincsenek, visszatérési értéke pedig 1, 2 vagy 3, a választott szint.

Készítsünk egy metódust, ami megadott intervallumból generál egy véletlen számot. A metódus neve legyen veletlen, paramétere a választott szint, visszatérési értéke pedig egy egész szám, a generált véletlen.

Ezzel a metódussal generáljuk le a véletlen számot, ha a választott szint 1, akkor a [0,10[intervallumból, ha 2, akkor a [0,100[intervallumból, ha pedig 3, akkor a [0,1000[intervallumból.

Írjunk egy olyan metódust, ami bekér a felhasználótól egy egész számot, egy megadott intervallumból.

A metódus neve legyen beker, paramétere a választott szint, visszatérési értéke pedig egy pozitív egész szám, amit a felhasználó adott meg. A bekérés legyen ellenőrzött.

Ezzel a metódussal kérjük be a felhasználó tippjeit, egészen addig, míg el nem találja a gondolt számot.

A tipp ellenőrzéséhez írjunk egy metódust. A metódus neve legyen ellenoriz, paraméterei a gondolt szám és a tipp. Visszatérési értéke logikai típusú legyen, igaz, ha a két szám megegyezik, és hamis ha nem. Utóbbi esetben a

metódus írja ki a képernyőre az alábbi szövegek egyikét is: "Kisebb számra gondoltam!" , "Nagyobb számra gondoltam!".

Játék közben számoljuk, hogy hányat tippelt a felhasználó, mire eltalálta a gondolt számot, és ezt a játék végén írjuk is ki neki.

Végezetül írjunk egy olyan metódust, ami csak i vagy n karaktert fogad el a felhasználótól. A metódus neve legyen `kilep`, paraméterei nincsenek, visszatérési értéke pedig egy karakter. Ennek a metódusnak a segítségével kérdezzük meg a felhasználót, hogy szeretne-e még egyet játszani. Ha igen, indítsuk újra a játékot a szintválasztástól kezdve, ha nem, lépünk ki a programból.

11. Feladat – Tökéletes szám

Írjon programot, amely eldönti egy számról, hogy tökéletes szám-e! Tökéletes számnak nevezzük azokat a természetes számokat, amelyek megegyeznek osztóik összegével (az 1-et beleértve, önmagukat kivéve).

Írjon programot, amely kiírja n -nél kisebb tökéletes számokat!

12. Feladat – Hiányos szám

Írjon programot, amely eldönti egy számról, hogy hiányos szám-e t ! Hiányos számnak nevezzük azokat a természetes számokat, amelyek nagyobbak az osztóik összegénél (az 1-et beleértve, önmagukat kivéve).

Írjon programot, amely kiírja n -nél kisebb hiányos számokat!

13. Feladat – Bővelkedő szám

Írjon programot, amely eldönti egy számról, hogy bővelkedő szám-e! Bővelkedő számnak nevezzük azokat a természetes számokat, amelyek kisebbek az osztóik összegénél (az 1-et beleértve, önmagukat kivéve).

Írjon programot, amely kiírja n -nél kisebb bővelkedő számokat!

14. Feladat – Barátságos számpár

Írjon programot, amely eldönti egy számpárról, hogy barátságos számok-e! Azokat a számpárokat, amelyekre igaz, hogy az egyik szám önmagánál kisebb (az 1-et is beleértve) osztóinak összege a másik számmal egyenlő (és fordítva), barátságos számoknak hívjuk.

Írjon programot, amely kiírja n -nél kisebb barátságos számpárokat!

15. Feladat – Fájl kezelés

Készíts `FajlKeszit` nevű metódust, amely egy `bemenet.txt` fájlt fog előállítani. Az előállítandó fájl nevét paraméterként kapja majd meg.

Készíts egy `VeletlenEgesz` metódust, amely egész számokat fog generálni az $[a,b]$ intervallumból. A metódus felhasználásával -100 és 100 közötti számokkal töltsd fel a fájl sorait.

Olvasd be a fájlt, majd anélkül, hogy a fájl elemeit eltárolnánk, írjuk ki őket a képernyőre az egyes értékeket.

Számoljuk meg, hogy hány elem van a fájlban, majd hozzunk létre nekik egy tömböt, és másoljuk ki a számokat a tömbbe.