2023.10.19. C# dolgozat

1. **Hogyan csoportosíthatjuk a programnyelveket 4p**

**Gyengén típusos**

* 1. BASIC b. JavaScript

**Erősen típusos**

1. Java d. Python
2. **Miért nem mindegy C#-ban, hogy milyen változó típust használunk egy szám letárolásához? Miben térnek el a szám típusok? Milyen előny – hátrányaik vannak az egyes típusoknak? 4p**

Különböző típusok eltérő memóriaigényekkel rendelkeznek. Például egy byte típus csak 8 bites memóriát használ, míg egy double típus 64 bitet használ. Ha feleslegesen túl nagy típust használunk egy egyszerű szám tárolásához, akkor feleslegesen sok memóriát pazarolunk.

Előnyök és hátrányok:

int: Hatékony és gyors, de korlátozott tartományú.

long: Szélesebb tartományú, de több memóriát igényel.

float és double: Tizedes tört számok tárolására alkalmasak, de pontossági problémák előfordulhatnak.

decimal: Nagy pontosságot nyújt, de lassabb és több memóriát használ.

1. **Írjunk szám kitalálós programot 20p**

Feladat: Generáljunk egy kétjegyű véletlenszerű számot, majd azt próbáljuk meg kitalálni. Írjuk ki, hogy a próbánk kisebb vagy nagyobb volt, mint a szám. Maximum 5-ször lehet próbálkozni, mielőtt vége a programnak.

A megoldásnál egy osztályba kerüljön a generált véletlen szám, és a szám összehasonlító logika, de csak ez! A szám generálása és a próbálkozások kezelése NE az osztályban legyenek, hanem a Program.cs-ben.

Github-on kérem a feladatot beadni, az 1 és 2-es feladat megoldása mehet a Readme-be.

Ha:

* következetes, jól olvasható, értelmezhető a kód 4p
* szépen követi az órán tanult névkonvenciókat 4p
* minden változó név angol 4p

akkor kapja meg a maximális 20 pontot.

**Bónusz: 8p**

A szám kitalálós játékot több játékos játssza, ahol egy játékos a Player osztály egy példánya. A játékosnak van neve, és próbálkozás száma. A játékosok felváltva találgatnak, maximum 5-ször. Addig tart a játék, amíg valamelyik játékos ki nem találja. Ekkor írjuk ki, hogy ki a nyertes, illetve, hogy a többiek hány próbálkozásnál tartanak.