NÉV:		NEPTUN KÓD:						

A beadandó *solution* elnevezési mintája: **NEPTUNKOD_VEZETEKNEV** (ékezetek nélkül, nagybetűkkel)

A beadott dolgozat fordítási hiba esetén nem értékelhető, a kommentezett részre pont nem szerezhető!

A munka végeztével a **teljes Solution mappát** tömörítse be egyetlen **ZIP** állományba és az interneten keresztül töltse fel a megfelelő kurzust kiválasztva! A feltöltés után kérje meg oktatóját, hogy ellenőrizze le a fájl megérkezését!

Készítsen hallgatói tankört kezelő alkalmazást!

A hallgatók számára a tanulmányi osztály („TO”) egy adott hétre vonatkozó vizsgák eredményeit egy szöveges állományban (**`INPUT.TXT`**) küldi el. A szöveges állomány egyetlen sort tartalmaz, melyre egy példa a következő:

```
Kiss József#Fizika!Szerda:1@Nagy Imre#Programozás1!Kedd:2@Kiss József#Elektronika!Kedd
```

- A hallgató neve előtt a `@` karakter látható (kivétel az első hallgató neve).
- A tárgynév előtt a **`#`** (kettős kereszt) karakter található.
- A nap előtt **`!`** (felkiáltójel) szerepel.
- Az eredmény előtt **`:`** (kettőspont) található.

Egy hallgatóra vonatkozó adat a `@` karaktertől (kivéve az első hallgató) a következő `@` karakterig, vagy az üzenet végéig tart. A név mindig az adat elején van, majd ezt követi a vizsga neve, a nap neve (Hétfő, Kedd, Szerda, Csütörtök, Péntek) és az eredmény. Lehetséges, hogy az eredményt nem adta meg a TO, ilyenkor ez az eredmény véletlenszerűen 1 és 5 közötti egész érték. Az biztos, hogy a TO helyes formátumot használ, így ezt nem kell ellenőrizni.

Az elkészítendő programnak fel kell tudnia dolgozni a TO-tól érkező fájlban található üzenetet, illetve a feldolgozott üzenet alapján információkat kell tudnia szolgálni az egyes hallgatók heti eredményeiről, amiket fájlba kell kiírni (**`OUTPUT.TXT`**):

- a legjobb eredményt elérő hallgató(k) neve,
- a legjobb átlageredményt elérő hallgató(k) neve,
- annak a napnak a nevét, amelyiken a legtöbb vizsga történt.

A `INPUT.TXT` állományra példa:

```
Kiss József#Fizika!Szerda:1@Nagy Imre#Programozás1!Kedd:2@Kiss József#Elektronika!Kedd
```

A `OUTPUT.TXT` állományra példa:

```
Legjobb eredményű hallgató(k) [érdemjegy:3]:   Kiss József,
Legjobb átlagú hallgató(k) [átlag:2]:          Kiss József, Nagy Imre,
A legtöbb vizsga a következő napon történt:    Kedd (2)
```

Az alkalmazás indulását követően ne kérjen be adatot a felhasználótól és ne írjon ki számára semmit. A beolvasandó állomány a futtatandó .exe mellett szerepel INPUT.TXT névvel, ahova a kimeneti állományt kell elkészíteni OUTPUT.TXT névvel. Az alkalmazás teszteléséhez készítsen egy INPUT.TXT állományt. **1 pont**

A FELADAT MEGOLDÁSA SORÁN A KÖVETKEZŐ STRUKTÚRÁT VALÓSÍTSA MEG

Az alábbi felépítést követve készítsen osztályokat, és valósítsa meg a leírás alapján az alkalmazás működését. Ügyeljen rá, hogy betartsa az objektum-orientált programozásban használatos egységbezásári és adatrejtési elveket. Ahol szükséges az adatmezőkhöz hozzon létre tulajdonságokat, ha elegendő csak olvashatót, illetve a példányhoz nem kötődő metódusokat, és adatmezőket tegye statikussá. **3 pont**

`static class ÜzenetFeldolgozó` **10 pont**

- Adatmező, amiben a feldolgozatlan üzenetet tároljuk: `üzenet[]`.
- Csak írható tulajdonság, mely az `üzenet`hez hozzáadja az új üzenet értéket: `Üzenet`.
- `VanÜzenet()` metódus, ami megadja, hogy van-e még feldolgozatlan `üzenet`.
- `Feldolgoz()` metódus, mely visszaadja az `üzenet[]` legelső elemét elhagyva azt az `üzenet[]` elejéről.

`class Vizsga` **10 pont**

- Adatmezők: `tantárgynév, nap, eredmény`.
- Csak olvasható tulajdonságok: `Tantárgynév, Nap, Eredmény`.
- Konstruktor, ami egy karakterláncot kap bemenetére és meghívja a `Feldolgoz(…)` metódust.
- `Feldolgoz(string)` az osztályon kívülről nem látható metódus, ami a bemeneti paramétert (pl: *`Fizika!Szerda:3`*) feldolgozza úgy, hogy értéket kapjanak az osztály adatmezei. Ha a bemeneti karakterlánc nem tartalmaz eredményre vonatkozó információt (pl: *`Fizika!Péntek`*), úgy 1 és 5 közötti véletlen érték legyen az eredmény.

`class Hallgató` **10 pont**

- Adatmezők: `név, vizsgák[]`.
- Csak olvasható tulajdonság: `Név`.
- Konstruktor, ami a név értékét kapja meg.
- `ÚjVizsga(Vizsga)` metódus, ami a `vizsgák` tömbbe felvesz egy új `Vizsga` elemet.
- `HetiÁtlagEredmény()` metódus, ami meghatározza, hogy mennyi a hallgató heti átlageredménye.
- `HetiMaxEredmény()` metódus, ami meghatározza, hogy mennyi a hallgató heti legjobb eredménye.
- `HetiMaxVizsga()` metódus, ami meghatározza, hogy melyik napon vizsgázott a legtöbbet a hallgató.

`class Tankör` **10 pont**

- Adatmezők: `hallgatók[]`.
- Csak olvasható tulajdonság: `Név`.
- Konstruktor, ami a név értékét kapja meg.
- `VanHallgató(string)` az osztályon kívülről nem látható metódus, ami meghatározza, hogy az adott hallgató tagja-e már a tanulókörnek.
- `HallgatóIndex(string)` az osztályon kívülről nem látható metódus, ami meghatározza, az adott hallgató hanyadik eleme a `hallgatók[]` tömbnek.
- `ÚjHallgató(string)` metódus, mely egy üzenet darabot (pl: *`Kiss_József#Fizika!Szerda:1`*) kap bemenetén és hozzáadja a hallgatót a tömbhöz ha még nincsen benne, illetve a hallgatóhoz hozzáadja az üzenetben található vizsgát.
- `LegjobbEredmény()` metódus, mely a legjobb eredményű hallgató(k) nevét és eredményét adja vissza.
- `LegjobbÁtlag()` metódus, mely a legjobb átlagú hallgató(k) nevét és átlagát adja vissza.
- `LegtöbbVizsgaNap()` metódus, mely annak a napnak a nevét adja vissza, amelyiken a legtöbb hallgató vizsgázott a vizsgázó hallgatók számával. Ha több napon is azonos számú hallgató vizsgázott, akkor a „*`Több napon is egyezik a vizsgázó hallgatók száma`*!” üzenetet adja vissza.

`class Program` **6 pont**

- `Main(…)` metódus, ami a fájl betöltését követően létrehozza a tankört egy tetszőleges névvel, majd feldolgozza a fájl sorait, végül a kérdésekre a válaszokat fájlba írja.
- `Betölt()` metódus, ami beolvassa a fájl sorait és feltölti az `ÜzenetFeldolgozó` osztály üzenet adatmezejét.
- `Kiír(Tankör)` metódus, ami fájlba írja a kérdésekre a válaszokat.

ÉRTÉKELÉS: **1**: 0 – 24,5P **2**: 25 – 30,5P **3**: 31 – 36,5P **4**: 37 – 42,5P **5**: 43 – 50P