**Harry Potter karakter készítő program**

1. Osztályok:
   1. Készíts egy **Karakter** osztályt (kerezstnev, vezeteknev, felhasznalonev, kor, cim) amely a felhasználó keresztnevét, vezetéknevét, az elkészítendő karakter felhasználónevét, korát és Bagolyposta címét tartalmazza, illetve rendelkezzen a többi class-el is mint tulajdonsággal (Haz haz, Patronus patronus, Palca palca). Legyen benne egy felülírt **ToString()** metódus, amely képes alkalmazkodni a karakter kreálás szakaszaihoz.
   2. Készíts egy **Haz** osztályt (szin, element, napszak, segitseg, tudasvagy, haz) amely eltárolja a Teszlek Süveg folyamatában keletkezett választásokat a felhasználó által, majd végül a végeredményként kapott házat is. Legyen benne egy felülírt **ToString()** metódus, amely visszaadja csak az eredményül kapott házat.
   3. Készíts egy **Palca** osztályt (fa, hossz, core, szin) amely a pálca létrehozásának részleteit tárolja el, a fának a fajtáját, a pálca hosszát, a pálca magját és végül a pálca színét, ha a választott fa fajta eredeti színe nem lenne szimpatikus. Legyen benne egy felülírt **ToString()** metódus, amely visszaadva egy zárójelben kötőjellel elválasztva adja vissza a pálca tulajdonságait.
   4. Készíts egy **Patronus** osztályt (evszak, idojaras, erdoto, tamadved, vegpatronus) amely tartalmazza a patrónus választás folyamatának részeredményeit, majd a véglegesen megkapott patrónust is. Legyen benne egy felülírt **ToString()** metódus, amely a véglegesen kapott patrónust adja vissza.
2. Menü:
   1. Készítsen egy **ElsoMenu()** metódust, amely a főmenü lesz a programunknak, melyből elérhetjük a karakter készítés, teszlek süveg, pálcakészítés, patrónus idézés, karakter lekérdezés, és majd a későbbi 3 feladat menüpontját is. **A választásokat a felhasználó minden esetben számokkal fogja tudni megadni!**
   2. Ez után egy **MasodMenu()** metódust kell elkészíteni, ami egy külön metódus lesz az **ElsoMenu()** -ben rögzített input kezelésére, és egy switch case módszerrel fog elindítani minden szükséges metódust a menü működéséhez.
   3. A **KarakterMenu()** metódus fogja a karakterkészítést tartalmazni, bekérjük az adatokat a Karakter osztályhoz, amely folyamán legyen egy ellenőrzés, hogy ilyen felhasználónévvel létrejött-e már egy karakter, majd ezt egy objektummá alakítva egy **Veglegesites()** metóduson átadva eltároljuk egy univerzális Karakter típusú listában.
   4. **Veglegesites()** metódus arra szolgál, hogy esélyt adjunk a felhasználónak arra, hogy megnézhesse az adott menü kitöltése után karakterének az állapotát, majd az után egy inputtal jóváhagyhatja vagy pedig visszakerülhet a főmenüre mentés nélkül, hogy az adott almenüt újra csinálhassa, és ha már készített karaktert, akkor annak a karakternek az adott adattulajdonságát frissíthesse. Tartalmaznia kell egy kiíratást, hogy a felhasználó tudhassa, hogy amint lenyom egy gombot vissza fog kerülni a főmenüre.
   5. **TeszlekSuveg()** metódus fogja eldönteni a felhasználó házát, mely tartalmaznia kell legalább 5 db 4 választási lehetőségből álló quiz kérdéseket, melyeket inpulzus szerűen kell a felhasználónak kiválasztania, ezek közül elég ha csak 1 db 4 választási lehetőséges blokk befolyásolja a ház kimenetelét, viszont a többi választási lehetőség is kerüljön eltárolásra az objektumon keresztül! Ezek után a karakter kerüljön alá még egyszer a véglegesítési folyamaton.
   6. A **PalcaMenu()** metódus több almetódust fog tartalmazni, mivel egy varázspálca nem rendelkezik 1 db névvel, hanem le kell írni a fa típusát, hosszát, flexibilitását, és a pálca magját, ezért ezekhez egyesével érdemes létrehozni 1-1 metódust, melyeknek rendelkezniük kell a legtöbb és legismertebb választási lehetőségekkel a felhasználó számára. A fatípusokat egy txt fálban fogja megkapni, amelyhez ajánlott létrehozni egy külön metódust a beolvasására és majd adatainak eltárolására. Ezek után hasonlóan az eddigiekhez véglegesíteni kell a karaktert, ez által frissítve az adatokat.
   7. **PatronusMenu()** nagyon hasonló metódus lesz a teszlek süveghez, itt is választási lehetőségekből álló kérdésblokkok lesznek találhatóak, minimum 4 db kérdésblokk darabonként minimum 2 választási lehetőséggel, viszont itt a switch case résznek ki kell térnie az összes lehetőségre amit a felhasználó be tud írni, és minden esetre kell lennie egy patrónusnak amely különbözik a többitől. Itt is szükséges a véglegesítés.

**FIGYELEM:**

* Minden olyan menünél, ahol a menü után kisorsolás szerűen kapja meg a felhasználó a végeredményt lennie kell egy gratulálok feliratnak, és az után legyen csak a véglegesítés
* Minden menünél meg kell győződni arról, hogy csak létező karakternek tud a felhasználó kreálni patrónust, házat és pálcát
* Ezen kívül ügyelni kell a legtöbb kivétel kezelésre és input error-ra.

1. FELADATOK:
   1. Az első feladat egy txt fájlba írás lesz, melyhez a programon keresztül kell létrehoznia a txt fájlt. Ebbe a fájlba kell beleírnia az eddig manuálisan létrehozott karaktereket, soronként „(keresztnév) (vezetéknév) ((felhasználónév)): (életkor) – ((cím)) - (ház) – (pálca) – (patrónus)” formátumban.
   2. A második feladat egy lista létrehozása lesz, melynek univerzálisan ki kell választania a megadott ház alapján (jelenleg legyen ez Griffendél) a karaktereket, majd azokat egymás mellé ki kell tudni íratni a képernyőre egy „|” karakterrel elválasztva.
   3. A harmadik feladat egy szortírozás lesz, ahol meg kell találni és kiírni a konzolra az összes felhasználó nevét, akik a megadott pálca magjával rendelkeznek. Ez jelen esetben az Egyszarvúszőr legyen, de később lehessen ezt is megváltoztatni egyszerűen.

**JÓ MUNKÁT!**