



A feladat megoldásaként a teljes solution mappát betömörítve a Moodle rendszerben kell leadni. A beadandó solution elnevezése a féléves feladat azonosítója és a saját neptunkódja legyen alulvonással elválasztva, nagybetűkkel: **AZONOSÍTÓ_NEPTUNKOD.zip**.

A feladat akkor kerül elfogadásra, ha a hallgató azt személyesen megvédte.

A feladattal kapcsolatos további információk az utolsó oldalon találhatóak (ezen ismeretek hiányából adódó reklamációt nem fogadunk el!).

Készítsen játékot a klasszikus szerencsekerék játékra. A játék során magyar közmondásokat kell kitalálni. A számítógép a játék elején egy szövegfájlból olvasson be magyar közmondásokat és ezek közül véletlenszerűen válasszon egyet.

- - - - -

A játék elején csak a szavak hossza látszik. A játék során a játékosok mássalhangzókat mondhatnak (minden egyéb esetben figyelmeztesse a program a felhasználót a helytelen karakter megadása miatt). Amennyiben van találat a mássalhangzók bekerülnek a megfelelő helyre. A találatok száma szorozódik a szerencsekerék kipörgetett értékével (ezeket véletlenszerűen adja a gép 1000 és 20000 Ft között):

1. Játékos? M

- - M M - - - - -

Kipörgetett érték: 5000 Ft $2 \cdot 5000 = 10000$ Ft

2. Játékos? S

- - M M - - - - - S - - - - -

Kipörgetett érték: 8000 Ft $1 \cdot 8000 = 8000$ Ft

Megkötések

- A játék során 3 játékos játszik. A játék indulásakor dönthet úgy a felhasználó, hogy a másik két játékos gépi játékos legyen, akik véletlenszerűen választanak mássalhangzókat, illetve vesznek magánhangzókat.
- A képernyőn jelenjen meg az aktuális állás.
- A játék 3 fordulóból álljon.
- Amennyiben egy játékos ki szeretné találni a megoldást, megmondhatja azt. Ha a válasz helyes megkapja az addig összegyűjtött pontjait.
- A játékban legyen lehetőség a folyamatban lévő játékot menteni a felhasználó által megadott néven, és korábban mentett játékot betölteni és folytatni.

N E M M I N D E G Y H O G Y H O L S Z O R Í T A C I P Ö

A megoldás során tartsa be a tanult OOP alapelveket (egységbezárás, láthatóság, öröklés), törekedjen saját osztályok létrehozására. Ahol lehetséges alkalmazza a megtanult programozási tételeket, illetve használja a tanult technikákat.

Tájékoztató

A feladattal kapcsolatosan általános szabályok:

- A feladat megoldását egy Console Application részeként kell elkészíteni.
- A feladat megoldásaként beadni vagy a betömörített solution mappa egészét vagy a Program.cs forrásfájlt kell (hogy pontosan melyiket, azt minden feladat külön definiálja), melynek elnevezése a feladat azonosítója és a saját neptunkódja legyen alulvonással elválasztva, nagybetűkkel:
AZONOSÍTÓ_NEPTUNKOD[.zip|.cs]
- A megvalósítás során lehetőség szerint alkalmazza az előadáson és a laboron ismertetett programozási tételeket és egyéb algoritmusokat figyelembe véve a *Megkötések* pontban definiáltakat, ezeket leszámítva viszont legyen kreatív a feladat megoldásával kapcsolatban.
- Az alkalmazás elkészítése során minden esetben törekedjen a megfelelő típusok használatára, illetve az igényes (*formázott, felesleges változóktól, utasításoktól mentes*) kód kialakítására, mely magába foglalja az elnevezésekkel kapcsolatos ajánlások betartását is (*bővebben*).
- **Ne másoljon vagy adja be más megoldását!** Minden ilyen esetben az összes (felépítésben) azonos megoldás duplikátumként lesz megjelölve és a megoldás el lesz utasítva.
- **Idő után leadott vagy helytelen elnevezésű megoldás vagy a kiírásnak nem megfelelő megoldás vagy fordítási hibát tartalmazó vagy (helyes bemenetet megadva) futásidejű hibával leálló kód nem értékelhető!**
- A feladat leírása az alábbiak szerint épül fel (* - opcionális):
 - *Feladat leírása* - a feladat megfogalmazása
 - *Bemenet* - a bemenettel kapcsolatos információk
 - *Kimenet* - az elvárt kimenettel kapcsolatos információk
 - *Megkötések* - a bemenettel, a kimenettel és az algoritmussal kapcsolatos megkötések, melyek figyelembevétele és betartása kötelező
 - **Megjegyzések* - további, a feladattal, vagy a megvalósítással kapcsolatos megjegyzések
- A leadott megoldással kapcsolatos minimális elvárás:
 - Nem tartalmazhat fordítás idejű figyelmeztetést.
 - Nem tartalmazhat fordítási hibát.
- A feladat megoldásához minden esetben elegendő a **.NET Framework 4.7.2**, illetve a **C# 7.3**, azonban megoldását elkészítheti **.NET 5**-öt, illetve a **C# 9**-et használva is, viszont a nyelv újjításait nem használhatja. További általános, nyelvi elemekkel való megkötés, melyet a házi feladatok során nem használhat a megoldásában:
 - Methods: `Array.Sort`, `Array.Reverse`, `Console.ReadKey`, `Environment.Exit`
 - LINQ: `System.Linq`
 - Attributes
 - Collections: `ArrayList`, `BitArray`, `DictionaryEntry`, `Hashtable`, `Queue`, `SortedList`, `Stack`
 - Generic collections: `Dictionary<K,V>`, `HashSet<T>`, `List<T>`, `SortedList<T>`, `Stack<T>`, `Queue<T>`
 - Keywords:
 - Modifiers: `protected`, `internal`, `abstract`, `async`, `event`, `external`, `in`, `out`, `sealed`, `unsafe`, `virtual`, `volatile`
 - Method parameters: `params`, `in`, `out`
 - Generic type constraint: `where`
 - Access: `base`
 - Contextual: `partial`, `when`, `add`, `remove`, `init`
 - Statement: `checked`, `unchecked`, `try-catch-finally`, `throw`, `fixed`, `foreach`, `continue`, `goto`, `yield`, `lock`, `break` - *in loop*
 - Operator and Expression:
 - Member access: `^` - *index from end*, `..` - *range*
 - Type-testing: `is`, `as`, `typeof`
 - Conversion: `implicit`, `explicit`
 - Pointer: `*` - *pointer*, `&` - *address-of*, `*` - *pointer indirection*, `->` - *member access*
 - Lambda: `=>` - *expression, statement*



- Others: `?:` - *tenary*, `!` - *null forgiving*, `?.` - *null conditional member access*, `?[]` - *null conditional element access*, `??` - *null coalescing*, `??=` - *null coalescing assignment*, `::` - *namespace alias qualifier*, `await`, `default` - *operator, literal*, `delegate`, `is` - *pattern matching*, `nameof`, `sizeof`, `stackalloc`, `switch`, `with` - *expressiong*, *operator*
- Types: `dynamic`, `interface`, `object`, `Object`, `var`, `struct`, `nullable`, `pointer`, `record`, `Tuple`, `Func<T>`, `Action<T>`,