Regionális forduló 2020. november 21.

Verziókövetés

A "verziókövetés" a programozás világában egy projekt életében történő változások végig követését, tárolását, strukturálását jelenti. Használatával a forráskódban minden egyes sorról tudhatjuk, hogy ki miért és mikor változtatta, és emellett a mappánkat bármikor egy tetszőleges verzióra hozhatjuk. Az ilyen rendszerek olyan hasznosnak bizonyultak, hogy verziókövetetlen professzionális projekt manapság már gyakorlatilag nem létezik. A legelterjedtebb verziókezelő rendszer a világon a *git*.

Működés

Verziókövetni egy mappát lehet, egy verzió mentése pedig felhasználói utasításra történik.

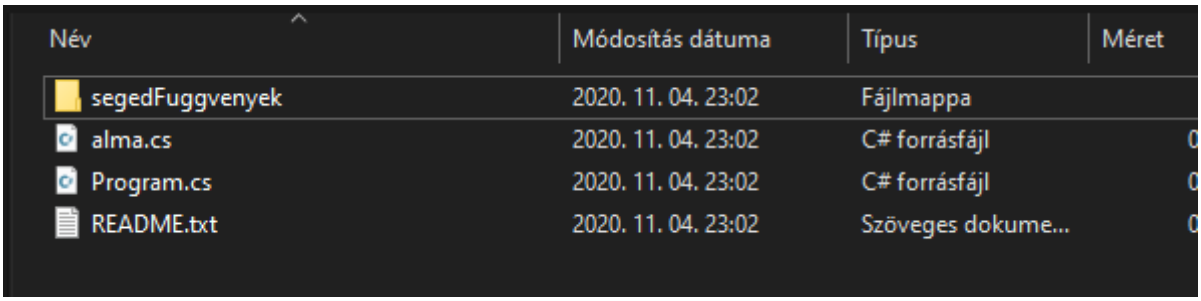
A verziókövetés legfontosabb funkciói:

1. Egy létező mappa verziókövetésre alkalmassá tétele (inicializálás)
2. Egy verzió mentése (commit)
3. A verziótörténet megjelenítése
4. A mappa állapotának beállítása egy kiválasztott verzióra

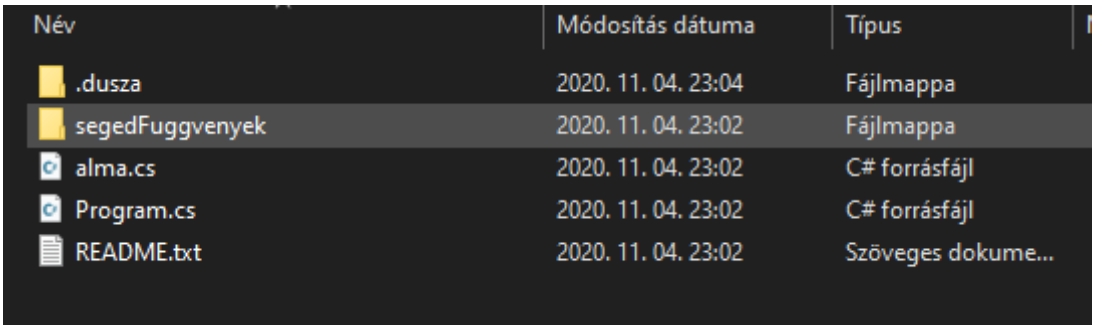
Egy inicializált mappát (repository) az különböztet meg egy szimpla hétköznapi mappától, hogy a benne eredetileg meglévő almappák és fájlok mellett tartalmazza (általában rejtve) az adott mappa korábbi változatait is. Amikor a felhasználó elkészít egy commitot (azaz verziót), a mappa aktuális állapotát menti, mint egy lenyomatot egy rejtett mappába, megcímezve a megfelelő adatokkal (mi változott, ki mikor miért változtatta).

Pl.:

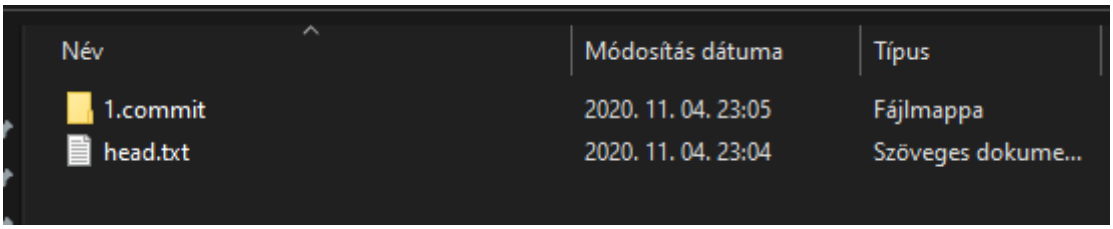
Tegyük fel, hogy egy mappa tartalma a következőképpen néz ki az inicializálás előtt (almappák is lehetnek, melyekben szintén lehetnek további elemek):



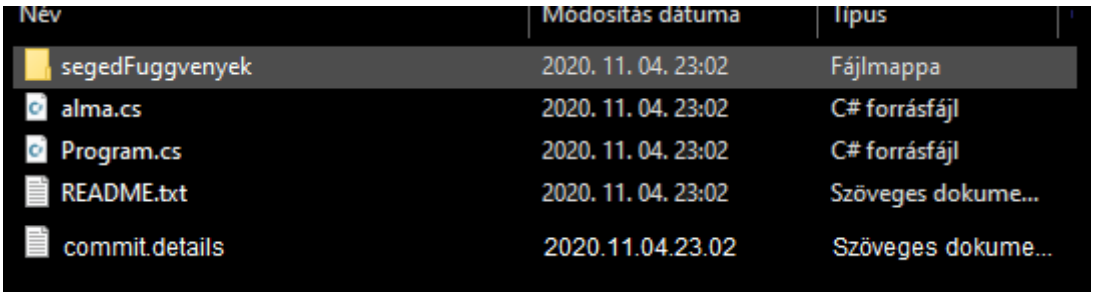
Inicializálás után létrejött a rejtett mappa (.dusza), benne pedig az 1. commit:



A „ .dusza ” mappa tartalma:



Az “1.commit” mappa tartalma az első lenyomat, azaz minden fájl és almappa, ami mentéskor a mappában volt (kivéve persze a .dusza mappát), vagyis ami az 1. képen látható, kiegészítve egy fájllal (neve „commit.details”), mely a commithoz tartozó adatokat tartalmazza:



1.commit/commit.details:

```
Szulo:-
Szerzo: Gipsz Jakab
Datum: 2020.11.04 14:37:56
Commit leiras: A projekt elso valtozata, mely tartalmazza az
alapveto funkciokat.
Valtozott:
uj alma.cs 2020.11.03 15:23:12
uj Program.cs 2020.11.03 15:23:13
uj README.txt 2020.11.03 15:23:14
```

A rendszer alkalmas arra, hogy megjelenítse a mappa összes előforduló verzióját, és arra, hogy közülük bármelyiket aktuálissá tegye. (Az almappákat és az abban lévő fájlokat az egyszerűség kedvéért most nem követjük.)

Feladat

A feladatotok egy, a legalapvetőbb funkciókat kezelni képes verziókövető rendszer létrehozása.

Egy mappa inicializálása (verziókövethetővé tétele)

Az elkészített programnak alkalmasnak kell lennie egy, a fájlrendszeren létező mappa inicializálására (függetlenül annak tartalmától). Ez a gyakorlatban azt jelenti, hogy létrejön a verziókövetéshez szükséges adatok tárolására használt „ **.dusza** ” mappa, és benne az első commit (az **1.commit** mappában), valamint a **head.txt**.

Amikor a felhasználó kiválaszt egy mappát, a program jelzi, hogy egy már inicializált mappáról van-e szó. (Van-e benne „ .dusza ” mappa?) Amennyiben nem, az inicializáláson kívül semmilyen funkció nem hajtható végre.
A **head.txt** tartalma egy szám, ami a verzió sorszámát jelenti.

Commit létrehozása

Egy commit a „ .**dusza** ” mappában lévő ***almappa*** egy egyedi azonosítóval (a programotokban kövesse az N.commit (N>0) formát, folyamatosan növekedő sorszámmal függetlenül a szülő commit sorszámától), mely tartalmazza a commit létrehozásának pillanatában az inicializált mappában található összes fájlt, valamint egy „***commit.details***” fájlt-t, melynek struktúrája a következő:

Szulo: <Szülő commit egyedi azonosítója>
Szerzo: <Szerző neve>
Datum: <A commit létrehozásának időpillanata ÉÉÉÉ.HH.NN ÓÓ.PP.MM formátumban>
Commit leiras: <A szerző által megadott leírás>
Valtozott:
torolt/uj/valtozott <fájlnév> <A fájl utolsó mentésének pillanata ÉÉÉÉ.HH.NN ÓÓ.PP.MM formátumban>
torolt/uj/valtozott <fájlnév> <A fájl utolsó mentésének pillanata ÉÉÉÉ.HH.NN ÓÓ.PP.MM formátumban>
..

- Szülő commitnak azt tekintjük, amelyik az új commit létrehozásának pillanatában aktív volt (az első commit esetében: −)
- Mivel ez az alkalmazás csak egy szerző módosításait követi, ezért a program indulásakor bekéri a szerző nevét, és minden alkalommal azt használja.
- A dátum a mentés időpontját tartalmazza a megadott formátumban. (Lehetőség szerint ne a billentyűzetről bevitt adat legyen!)
- A leírás a commit elkészítésekor megadott szöveg.

- A "*Valtozott:"* után a változás típusa, a megváltozott fájl neve (szóköz nem lehet benne), valamint az utolsó mentésének pillanata a megadott formátumban. A változás típusát, a fájl nevét, a dátumot és az időt szóköz választja el.

  Minden változás egy sor.

  Típusok:
  a. "torolt" amennyiben a fájlt törölték
  b. "uj" amennyiben egy fájl létrejött
  c. "valtozott" amennyiben a fájl változott

Az inicializált mappa almappáinak és a bennük lévő fájloknak a változását nem követi ez a rendszer.

Üres commitot nem szabad készíteni! A program jelezze a felhasználó számára, ha olyan pillanatban commitál, amikor az aktuális verzióhoz képest nem történt változás!

Az aktív commit megváltoztatása

Amennyiben egy mappa inicializálás után verziókövetetté vált (repositoryvá vált), az aktív commit azt jelzi, hogy a mappa melyik verziója látható éppen a mappában. Az aktív commit egyedi azonosítója (commit azonosító) a „ ***.dusza/head.*txt** "-ben található.

A programnak alkalmasnak kell lennie arra, hogy a felhasználó az aktív commitot megváltoztassa a commit azonosító megadásával. Ebben az esetben a **head.txt** fájl tartalma értelemszerűen megváltozik, valamint az eredeti mappáé is. Minden nemcommitált változás elveszik, és a mappa állapota pontosan azzá válik, ami a megadott commitban elmentésre került.

Az aktív commit megváltoztatásánál a program kérjen megerősítést!

A verziók listázása

A programnak alkalmasnak kell lennie a már létező commitok listázására időrendben a sorszámuk megjelenítésével. Ezek közül bármelyik kiválasztása esetén meg kell jelennie a hozzá tartozó jellemzőknek, azaz a megfelelő **commit.details** tartalmának.

Kiegészítő feladat

Az alapfunkciókat kiegészíthetitek a commit elágazások megjelenítésével (vizualizációjával). Ennek módját megválaszthatjátok. Elágazásról beszélünk, ha egy commitnak nem egy gyereke van, hanem több.

Figyeljetek a jól szervezett, áttekinthető képernyőképre! Elvárás, hogy a program csak egy speciális parancs esetén álljon le, a különböző műveletek egymás után legyenek elvégezhetők, és a kijelzőn megjelenő utasítások egyértelműek legyenek!

Beadandó

A program **forráskódja** (a programozási környezettől függően a forráskód több fájl is lehet, esetleg mappák is tartozhatnak hozzá), és a **lefordított fájl** (git.exe), ha a programozási környezet a fordítást támogatja.

A program által használt külső fájlokat mindig az aktuális könyvtárban kell elhelyezni (A program ne tartalmazzon abszolút elérési útvonalat!)

Kódolási alapelvek

A forráskód minőségét is értékeljük.

Irányelvek, szempontok:

- Egységes kódolási szabályok az azonosítókra:
  - a változók egységes elnevezése (kis- és nagybetűk vagy más speciális karakterek használata),
  - az osztályok egységes elnevezése (objektum-orientált programnyelv esetén),
  - a függvények és eljárások tartalomra utaló elnevezése,
  - a programkód egységes strukturáltsága, tagoltsága.
- A kód minősége (könnyen – emberek számára – érthető illetve karbantartható kód):
  - áttekinthető, lehetőség szerint rövid eljárások, függvények, fájlok,
  - beszédes, tömör elnevezésű azonosítók,
  - objektum-orientált nyelveknél globális változók mellőzése.
- Törekedjetek arra, hogy a kódotok minél hatékonyabb legyen!
- Kommentezés:
  - A kommentezés elsődleges célja, hogy a programban a miért? kérdésre adjon választ. (A mit? kérdésre az azonosítók megfelelő elnevezése és a megfelelően strukturált kód, a hogyan? kérdésre pedig az áttekinthető forráskód ad választ.)
  - A túlzásba vitt kommentezés csökkenti az áttekinthetőséget, a túl kevés komment nehezíti a megértést.
  - Elvárás a változók, osztályok, függvények és eljárások szerepének rövid, értelemszerű kommentezése.

Elérhető pontszám: 120 pont (Ebből a helyes dokumentálás – kommentezés – 20 pont)

Jó munkát kíván a versenybizottság!