Priprema za drugi kolokvijum

Neophodno je implementirati klijent-server arhitekturu. Učesnici u ovoj komunikaciji su ***UploadClient***, ***Server*** i ***Client***.

U okviru deljene biblioteke **Common** treba implemetirati klase ***Book***, ***CustomException, BookEventArgs, LibrarySubscriber*** i ***FileManipulationOptions*** koja treba da implementira interfejs *IDisposable*.

Klasa ***Book*** treba da sadrži polja ***title*** (tipa string) i ***score*** (tipa int) čija je inicijalna vrednost postavljena na 0. Klasa ***CustomException*** sadrži polje ***message*** (tipa string) i odgovarajući property. Klasa ***FileManipulationOptions*** sadrži polja ***memoryStream*** (tipa MemoryStream) i ***fileName*** (tipa string). Klasa ***BookEventArgs*** sadrži polja ***title*** (string), ***newScore*** (int) i ***oldScore*** (int). Za svako polje obezbediti odgovarajući property. ***LibrarySubscriber*** klasa predstavlja pretplatnika na događaj koji se okine kada se promeni ocena knjige.

Sa strane servisa neophodno je da postoji kolekcija knjiga.

Deljeni interfejs treba da sadrži sledeće metode:

- metoda za slanje fajlova (slanje fajlova inicira *UploadClient* dok se fajlovi preuzimaju na *Serverskoj* strani)
- metoda za dobavljanje svih knjiga (Lista knjiga se prikazuje u konzoli od *Client*-a)
- metoda za izmenu ocene knjige (poziva je *Client*)
- *Subscribe* i *Unsubscribe* metode koje simuliraju pretplatu odnosno odjavu sa događaja

```
[OperationContract]
[FaultContract(typeof(CustomException))]
void AddBookRecommendation(FileManipulationOptions options);

[OperationContract]
[FaultContract(typeof(CustomException))]
List<Book> GetAllBooks();

[OperationContract]
void ChangeScore(string title, int newScore);

[OperationContract]
void Subscribe();

[OperationContract]
void Unsubscribe();
```

AddBookRecomendation – metoda za slanje fajlova preko mreže. Korisnik (*UploadClient*) preko konzole unosi naziv knjige u obliku *Title.txt* i na osnovu unetog naziva se formira istoimeni fajl na putanji koja se čita iz konfiguracionog fajla (naziv direktorijuma: *BookDirectory*). Datoteke na datoj putanji nadgleda ***FileSystemWatcher*** (potrebno je proveriti da li direktorijum postoji i ukoliko ne postoji kreirati ga). Nakon što korisnik završi sa kreiranjem fajlova potrebno je da izmeni odnosno

doda sadržaj u novokreirane fajlove. Promena sadržaja okida događaj ***Changed*** koji treba da pozove metodu *AddBookRecommendation* koja šalje izmenjeni fajl *Serveru* i smešta ga u novi direktorijum (putanju je potrebno pročitati iz konfiguracionog fajla i ukoliko ne postoji kreirati ga). Naziv fajla kao i vreme izmene ispisati u konzoli UploadClient-a.

GetAllBooks – Metoda treba da vrati sve knjige sa Serverske strane i prikaže ih u konzoli od *Client*-a. Nakon što su svi fajlovi smešteni kod Servera potrebno je dobaviti sve nazive fajlova na toj putanji i na osnovu tih naziva kreirati instance knjiga koje se čuvaju u kolekciji sa serverske strane.

ChangeScore **-** Metoda treba da omogući definisanje nove vrednosti za ocenu knjige. Ukoliko je moguće izvršiti promenu, pozvati događaj ***BookScoreChanged*** koji putem dodatnih informacija o događaju (***BookEventArgs***) obaveštava pretplatnike koji treba u konzolu da loguju nastalu promenu.