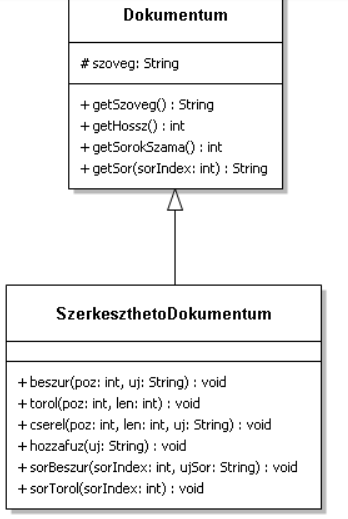
**16.1. Dokumentum**

**Szükséges fájlok**: Dokumentum.java, SzerkeszthetoDokumentum.java ([Letöltés](https://online.studicore.hu/mod/vpl/views/downloadrequiredfiles.php?id=543))  
**Feladat típusa**: Egyéni feladat

Programozzuk le a következő osztálydiagramnak megfelelő rendszert (a getterek, setterek és konstruktorok nem MIND szerepelnek!):  


A Dokumentum osztály értelmezése: egy szöveget tárol el String szoveg adattagjában

* getSzoveg(): String : visszaadja a szöveget.
* getHossz(): int : visszaadja a szöveg hosszát.
* getSorokSzama(): int : visszaadja a szöveg sorainak számát.
* getSor(sorIndex: int): String : visszaadja a sorIndex-edik sort.

A SzekeszthetoDokumentum osztály metódusainak értelmezése:

* beszur(poz: int, uj: String): void : a poz indexű karaktert hátrébbtolva beszúrja az uj Stringet a szoveg adattagba.
* torol(poz: int, len: int): void : a poz indexű karaktertől kezdve kitöröl len darab karaktert.
* cserel(poz: int, len: int, uj: String): void : mint az előző kettőt egymás után (fordított sorrendben) lefuttatnánk.
* hozzafuz(uj: String): void : a dokumentum végére fűzi hozzá az uj Stringet.
* sorBeszur(sorIndex: int, ujSor: String): void : a sorIndex-edik sort hátrébbtolva beszúrja az ujSor stringet a dokumentumba.
* sorTorol(sorIndex: int): void : törli a sorIndex-edik sort.

Minden sorIndex egy 0-val kezdődő indexet tartalmaz.

Ha az eddigiekkel készen vagyunk:

Adjunk hozzá a Dokumentum osztályhoz egy olyan metódust, amely visszaadja saját szerkeszthető  
változatát: public SzerkeszthetoDokumentum szerkeszthetoValtozat(). A szövegtartalom azonos, de abból egy másik objektumot készít.

**Fontos! A két osztálynak értelmetlen paraméterekre (null, rossz index) is helyesen kell működnie!**