**IskolaProject Dokumentáció**

(cli xml crud)

Fájlok: SchoolManagement.xml, pom.xml, App.java, Main.java, Schoolmanager.java, Person.java, School.java, Student.java, Teacher.java

Téma: Iskolákat rendező program. Tartalmazhat Iskolákat, az iskolákhoz tartozó tanárokat, illetve diákokat. Az iskolák a diákok és a tanárok egy xml-fileban vannak tárolva. A tanárok és a diákok az iskolákon belül helyezkednek el. Az iskolákat ki lehet listázni, lehet iskolát hozzáadni, illetve törölni. Ez a diákokra és a tanárokra is szintén érvényes. Az iskolákat véletlenszerű id-kkal kezeljük le, míg a tanárok és a diákok 1-től sorszámozzuk. A projectet a jaxb-s xml kezelő-vel csináltam.

1. Schoolmanagement.xml:

xml root element: <SM>:

xml element<schools>: ebben vannak az iskolák

<entry>

<key>ez tartalmazza az iskolák random id-jeit, amikkel azonosítom őket uuid-val jobb lenne, de azt nehezebb begépelni a usernek (gui hiányában döntöttem a random school id-k mellett) </key>

<value>

<name> itt van az iskola neve</name>

xml element<students> itt találhatóak a diákok adatai elemenként </students>

xml element<teachers> itt találhatóak a tanárok adatai elemenként </teachers>

</value>

</entry>

</schools>

</SM>

1. School.java

class tagváltozói: String name

Itt Arraylistetkben tárolom a diákokat és a tanárokat is, majd ezeket és az iskola nevét is hozzárendelem egy school object-hez. Ezek is elemei az xml filenak. Az itt található functionökről nem nagyon lehet sok mindent mondani, ami a nevük az történik: addTeacher, addStudent, removeTeacher, removeStudent, getTeachers, getStudents

Illetve létrehozunk egy school objectumot és adunk neki egy nevet, illetve hozzárakjuk a diákok arraylistét és a tanárok arraylistét)

1. Person.java

Minden tanár és diák egy Person

class tagváltozói: String name, int age

1. Student.java extends Person

class tagváltozói: Person tagváltozói + float average (a diák átlaga)

Létrehozunk egy student objectumot és adunk neki nevet, kort és átlagot)

1. Teacher.java extends Person

class tagváltozói: Person tagváltozói + int salary (a tanár fizetése)

Létrehozunk egy teacher objectumot és adunk neki nevet, kort és fizetést)

1. Schoolmanager.java

itt van a root (SM)

Hasmapbe tárolom a schoolokat a key az id és a value az iskola

addSchool() = iskolahozzáadás véletlenszerű id létrehozással (0-tól 100.000-ig)

addig generál random id-t amíg nem talál „üreset”

getSchools()= iskolák lekérése

1. App.java

Itt van a cli, illetve a kezelő functionok.

XmlManagger

létrehozza a SchoolManagment xml fájlt xmlFile néven.

xmlWrite()

kiír az xml.be a marsh.marshal-el és hibát kezel

xmlLoad()

betölti az xml-t az unmarshaller.unmarshal( segítségével

és hibát kezel

ezek a marshallerek a jaxb xml kezelő parancsai (csak mert gondolom a többi dependency-nél más a nevük)

class Menu()

A cli-ről annyit érdemes megemlíteni, hogy rekurzív, illetve a hibákat helyesen kezeli(catch) le. Külön sub-okra van szedve, így kevesebb menüpontból állnak a külön menük, nem egyszerre kell kiválasztani egy menüpontot a sok közül.

Ahogy kódba is beleírtam kikommentezve, ezek után jönnek a tényleges add, remove, choose, print metódusok, illetve a user által használt cli-ben lévő sub metódusok, amik meghívják a tényleges metódusokat és segítik a könnyebb konzolos kommunikációt. Mindegyik metódus hibát kezel és rekurzív.

1. Main.java

Itt csak futtatom az App class-t

1. pom.xml

Hozzáadom a jaxb dependency-t (azt találtam, hogy ez a leghatékonyabb az összes xml kezelők közül azért csináltam ebbe)