

Rendszeresen gyűjtünk adatokat XML fájl formájában. A jólformázott és érvényes XML fájlok előállítását egy példa XML fájljal és egy XML schema fájljal (XSD) segítjük. (az XML fájl érvényessége az XSD fájl segítségével ellenőrizhető. Az ellenőrzés például a <http://www.utilities-online.info/xsdvalidation/> címen is megtehető.)

Ezúttal azonban az XSD fájlt véletlenül töröltük, csak a helyes XML példa áll rendelkezésünkre. Az XSD fájlt programunknak kell előállítania a lehető legpontosabb formában. Az egyes elemekhez a nevet, típust – és ha szükséges – az előfordulás számát kell bejegyezni.

A rendelkezésre álló XML fájlban az adatok legfeljebb négy szinten vannak egymásba ágyazva. (A négy szintbe a gyökérelemet is beleértjük.) Tudjuk, hogy az összetett típusok közül csak a szekvenciális szerepel, az elemi típusok közül pedig a következők fordulhatnak elő: xs:integer, xs:decimal (például 12.34), xs:date (formátuma: éééé-hh-nn, például 2014-01-01), xs:time (formátuma: óó:pp:mm, például 10:30:00). A fel nem ismert típust xs:string-ként adjuk meg. Ha egy elem többször is előfordul, az előfordulások maximális számát végtelenre, ha elmarad, akkor az előfordulások minimális számát pedig nullára állítjuk.

A témában a következő lapokon is tájékozódhatunk, de érdemes továbbiakat is felkutatni:

- http://www.inf.unideb.hu/~bodai/31-xml_alk/xmlSchemas.html;
- <http://soci.hu/articles/xml/XmlSema.pdf>.

A bemeneti fájl egy jólformázott XML fájl. A fájlban a <, >, / karakterek csak a tagekben találhatók meg. A kimeneti fájl egy XSD fájl, amely az első bekezdésben említett címen található úrlapon az XML fájlt érvényesnek mutatja. (A mintafájlok a weblapunkról tölthetők le.)

A program első parancssori argumentuma a bemeneti fájl neve, a második pedig a kimeneti fájl neve legyen.

Beküldendő egy tömörített i402.zip állományban a program forráskódja, valamint a program rövid dokumentációja, amely tartalmazza a megoldás rövid leírását, és megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.