Object Serialization

Cand se creaza un obiect el exista atat timp cat avem noi nevoie, dar sub nicio circumstanta el nu exista dupa terminarea executiei programului.

Serializarea obiectelor in Java permite preluarea oricarui obiect ce implementeaza interfata “Serializable”

Si transformarea lui intr-o serie de bytes ce poate fi mai apoi ‘regenerata’, recreeand obiectul original.

Aceasta metoda este functionala si intr-o retea, reprezentand faptul ca mecanismul de serializare compenseaza automat pentru posibila diferenta dintre sistemele de operare.

Serializarea unui obiect se face simplu atat timp cat acesta implementeaza interfata “Serializable”. Se creeaza un obiect de tip “OutputStream”, ce se incapsuleaza intr-un obiect de tip “ObjectOutputStream”. Se face apelul functiei “writeObject()”, obiectul fiind serializat si trimis in ‘OutputStream’. Pentru deserializare, incapsulam un obiect de tip “InputStream” intr-un obiect de tip “ObjectInputStream”, facandu-se apoi apelul functiei readObject() + downcast-ul de rigoare ( ce se returneaza este referinta catre ‘Object’ ).

- Cuvantul cheie “transient” : Cand controlam serializarea, pot exista subobiecte particulare pe care nu ati dori ca mecanismul de serializare sa le salveze si recompune automat. De obicei subobiectul reprezinta o informatie privata pe care nu o doriti serializata, o parola de exemplu. Indiferent de modificatorul de acces, odata serializat, exista posibilitatea ca informatia sa poata fi interceptata si descifrata intr-o transmisie in retea.

Pentru a controla aceasta situatie, putem opri serializarea la nivel camp-dupa-camp utilizand cuvantul cheie ‘transient’ care spune “ Nu te complica cu asta, se ocupa Varu’ ”

O metoda de prevenire a unei inconveniente de genul este implementarea clasei ca si “Externalizable”.

XML

O limitare importanta a serializarii obiectelor este aceea ca utilizarea reprezinta doar o solutie pentru Java: Doar programele Java pot deserializa astfel de obiecte. O solutie mult mai inteoperabila este convertirea datelor la format XML, permitand ‘consumul’ de o larga varietate de platforme si limbaje.

Datorita popularitatii, exista multiple metode de a programa cu XML, inclusiv javax.xml.\* distribuit cu JDK.

Am ales libraria open source XOM a lui Elliotte Rusty Harold( [www.xom.nu](http://www.xom.nu) ) deoarece pare cea mai simpla si directa metoda de a produce si modifica XML folosind Java, acesta punand accentul pe corectitudinea XML.