mfp05a − 4. házi feladat (haskell)

"Map (leképezés) modul írása − interfész és implementáció"
==========

Interfész: a map kulcsok és adatok közti leképezést biztosít.

Kezdetben vala az üres map.

Létrehozni a "map_new :: key −> data −> map" konstruktorfüggvénnyel lehet, amely tehát egy kulcsot és egy adatot kap. A kulcsot nem használja semmire (a típuslevezetés miatt kell), az adatot a default leképezéshez használja. A leképezés ugyanis kezdetben minden kulcsra a default adatot adja vissza. Amikor a map−be adatot teszünk be, olyankor tulajdonképpen a default leképezéstõl való eltérést mondjuk ki az adott kulcsra.

A leképezés megváltoztatása: "map_set :: map −> key −> data −> map".

Lekérdezése: "map_get :: map −> key −> data".

Egy (a defaulttól eltérõ) leképezés törlése úgy történik, hogy az adott kulcshoz újra a default adatot rendeljük hozzá. (Ilyenkor elvárjuk az implementációtól, hogy valamilyen erõforrást fel is szabadítson.)

Implementáció: lehet szimplán egy listában tárolni a kulcs−adat párokat, és benne lineárisan keresni. Ehhez a kulcs−típusnak meg kell valósítania az "Eq" osztályt. Kicsit jobb implementáció, ha a listában rendezve tárolunk, ehhez a kulcsnak a rendezést is implementálnia kell. A kettõ közül az egyiket mindenkinek implementálni kell. Bátrabbak próbálkozhatnak alternatív implementációval, amit a fantáziátokra bízunk (hash−tábla, bináris fa, B* fa, mittudoménmilyen−önszervezõdõ−fa, satöbbi), és nyugodtan lehet copy−pastelni interneten talált kódot, mert nem ez a része a lényeg.

A gyakorlatban szükség lehet még egy használati módra: ha szükségünk van az összes (defaulttól eltérõ) leképezésre, azt valahogy fel kell soroltatni a mappel. (A default implementáció esetén ez nem túl sok munka, azonban szükséges, mert a map belsejébe, ugye, nem láthatunk be.)

Végezze ezt a "map_list :: map −> [(key, data)]" függvény, amely egy map−bõl (kulcs, adat) párok (nem definiált sorrendû) listáját adja vissza.

A map−nek továbbá meg kell valósítania a show interfészt. Az az implementáció magánügye, hogy mit ír ki, annyit kérünk, hogy legyen szemmel olvasható (ezen azt értjük, hogy a kiírás legyen szebb, mint amit a map_list eredményére meghívott default show függvény produkál).

Ezzel a feladattal kimerítjük a modulok, típusosztályok gyakorlását. (A lusta nyelv lehetõségeivel való kísérletezést most elhagyjuk.) Az I/O viszont fontos, hiszen lefordítható, önállóan futtatható programokat akarunk továbbra is. Azonban az I/O csak minimális nehézségû feladat lesz: a stdin−en érkeznek sorok, amelyeket be kell pakolni egy map−be, méghozzá úgy, hogy az elsõ szó legyen a kulcs (string), a sor többi része pedig az adat (string). Az utolsó sor (EOF) után a program írja ki a map−be menet közben begyûjtött adatokat (magyarul: írja ki a show map eredményét).

(Szerzõ: Békés András György)