

Laboratorul 9: I/O

Operații de intrare/ieșire în Haskell

Vom începe prin a exersa operațiile de citire și scriere.

Funcțiile pe care le vom folosi sunt:

```
getChar :: IO Char -- citește un caracter de la tastatura

getLine :: IO String -- citește o linie de la tastatura

putStrLn :: String -> IO () -- afișează un șir urmat de sfârșit de linie

putStr :: String -> IO () -- afișează un șir fără a fi urmat de sfârșit de linie

putChar :: Char -> IO () -- afișează un caracter

print :: Show a => a -> IO () -- afișează orice element de un anumit tip de date
                                -- care este instanța a clasei Show

readLn :: Read a => IO a -- citește un element și îl transformă din String
                        -- într-un tip de date a

readFile :: FilePath -> IO String
writeFile :: FilePath -> String -> IO ()
appendFile :: FilePath -> String -> IO ()

-- unde FilePath = String
```

Exemplul 1

Citirea de la tastatură a unui șir și afișarea rezultatului obținut după prelucrare.

```
prelStr strin = map toUpper strin

ioString = do
```

```

    strin <- getLine
    putStrLn $ "Intrare\n" ++ strin
    let strout = prelStr strin
    putStrLn $ "Iesire\n" ++ strout

```

Exemplul 2

Citirea de la tastatura a unui număr și afișarea rezultatului obținut după preluare.

```

prelNo noin = sqrt noin

ioNumber = do
    noin <- readLn :: IO Double
    putStrLn $ "Intrare\n" ++ (show noin)
    let noout = prelNo noin
    putStrLn $ "Iesire"
    print noout

```

Observati utilizarea functiilor “readLn”, “show” si “print”.

Exemplul 3

Citirea din fișier de intrare și afișarea rezultatului într-un fișier de ieșire.

```

inoutFile = do
    sin <- readFile "Input.txt"
    putStrLn $ "Intrare\n" ++ sin
    let sout = prelStr sin
    putStrLn $ "Iesire\n" ++ sout
    writeFile "Output.txt" sout

```

Atenție! Funcția `readFile` întoarce un rezultat de tipul `IO String`.

Exercițiul 1

Scrieți un program care citește de la tastatură un număr n și o secvență de n persoane, pentru fiecare persoană citind numele și vârsta. Programul trebuie să afișeze persoana (sau persoanele) cu vârsta cea mai mare. Presupunem că vârsta este exprimată printr-un `Int`.

Exemplu de intrare:

```

3
Ion Ionel Ionescu
70
Gica Petrescu
99
Mustafa ben Muhamad
7

```

Exemplu de iesire:

Cel mai in varsta este Gica Petrescu (99 ani).

Exercițiul 2

Aceeași cerință ca mai sus, dar datele se citesc dintr-un fișier de intrare, în care fiecare linie conține informația asociată unei persoane, numele si varsta fiind separate prin virgulă (vedeți fișierul `ex2.in`).

Indicație: pentru a separa numele de varsta puteti folosi funcția `splitOn` din modulul `Data.List.Split`.

Exercițiul 3

- a) Să se scrie o funcție care citește de la tastatură un număr întreg si verifică dacă acesta este palindrom. Funcția afișează un mesaj corespunzător.
- b) Folosind funcția de la a) să se scrie o funcție care are ca parametru un nr întreg `n` și citește `n` numere de la tastatura, verificând dacă acestea sunt numere palindrom.