Student: IEREMIAȘ Viorel

Grupa: 244

Tema 1 – 5Programe

Tematică: I/O local

**Cerință**

*“Sa se implementeze rezolvarea problemei de mai jos in cinci variante, folosind respectiv limbajele: C#, Java, PHP, Python, NodeJs.*

*!!\*\* Daca se doreste, se pot inlocui, partial sau total: Java cu Kotlin, iar Python cu Go*

*Pe cat posibil se vor pastra: numele variabilelor, algoritmul de rezolvare, meniul / interfata grafica, parametrii de intrare, iesirile etc.*

*Se da la intrare un fisier text. Sa se dea la iesire un alt fisier text care sa contina aceleasi linii cu intrarea, dar fiecare grup de 10 linii din intrare se va depune in ordinea inversa a aparitiei liniilor.*

*Nu se va incarca in memorie intregul fisier de intrare si nu se va folosi un fisier suplimentar.*

*Numele fisierului se va da in linia de comanda.”*

**Soluție propusă**

În urma analizei problemei, se observă că aceasta poate fi descompusă în 3 pași independenți unul de celălalt:

1. Citirea liniilor din fișierul de intrare, în grupuri de câte 10
2. Inversarea ordinii liniilor dintr-o listă
3. Scrierea liniilor în fișierul de ieșire, în grupuri de câte 10

Ca urmare, fiecare din cele 5 programe implementează funcțiile de mai jos, a căror specificație este prezentată în pseudocod:

Function ReadBatchLines

Input: file\_reader - File Reader object

Output: List of 10 lines from the input file

Initialize lines\_count to 0

Initialize read\_lines to empty list

while lines\_count < 10

read\_line = file\_reader.read()

add read\_line to read\_lines

increment lines\_count

endwhile

Endfunction

Function ReverseBatchLines

Input: lines - List of character strings

Output: List of character strings in reversed order

Endfunction

Function WriteBatchLines

Input: file\_writer - File Writer object

Input: lines - List of lines to write to the file

Output: void

Initialize written\_lines\_count to 0

Initialize lines\_count to length of lines

while written\_lines\_count < lines\_count

line\_to\_write = lines[written\_lines\_count]

file\_writer.write(line\_to\_write)

increment written\_lines\_count

endwhile

Endfunction

Specificația pentru funcția care inversează ordinea liniilor într-o listă nu conține niciun detaliu de implementare deoarece utilizează funcția pusă la dispoziție de fiecare limbaj pentru a inversa elementele dintr-o listă sau dintr-un șir. Singurul motiv pentru care această funcție există este acela de a evidenția pașii principali ai problemei.

Cele 3 funcție sunt apelate secvențial în cadrul unei bucle, care repetă ciclul citire linii -> inversare linii -> scriere linii atât timp cât există conținut necitit în fișierul de intrare.

Function MainProgram

Initialize file\_reader to input file

Initialize file\_writer to output file

Initialize batch\_lines to empty list

while true

batch\_lines = ReadBatchLines()

if length(batch\_lines) > 0

reversed\_batch\_lines = ReverseBatchLines(batch\_lines)

WriteBatchLines(reversed\_batch\_lines)

else

break

endwhile

Endfunction

Implementarile în C#, Java, PHP și Python diferă doar prin detaliile de ordin de sintaxă a limbajului de programare, în rest fiind identice. Implementarea în NodeJS diferă atât prin felul în care se face citirea conținutului din fișierul de intrare, cât și prin felul în care se transmite controlul de la o metodă la alta.

În primul rând, metodele de citire puse la dispoziție de pachetul file system din Node.js permit citirea în întregime a fișierului sau citirea sub forma unui stream. Niciuna dintre modalități nu se potrivește problemei date. Ca urmare se folosește pachetul npm line-reader, care permite citirea linie cu linie dintr-un fișier.

În al doilea rând, asemenea funcțiilor clasice de lucru cu fișiere din Node.js, line-reader operează asincron, așadar rezultatul operațiunii de citire a unei linii din fișier poate fi procesat prin transmiterea unei funcții callback apelului de citire.

Prin urmare, soluția trebuie să cuprindă o parte care citește în mod continuu conținut din fișierul de intrare, și îl depune într-un container, și o parte care atunci când containerul conține un număr de 10 linii, extrage liniile, le inversează și le scrie în fișierul de output. Cele două părți trebuie să funcționeze în parallel, Acest lucru este realizat prin definirea unui obiect care conține o listă privată, un setter custom și posibilitatea de a atașa o funcție listener. De fiecare data când este apelată metoda setter, se verifică condiția ca numărul de linii depuse să fie egal cu 10, iar daca această condiție este adevarată, se apelează funcția listener. Funcția listener este inițializată la începutul programului și realizează inversarea liniilor și scrierea lor în fișierul de output.

**Instalare și rulare**

1. C#

* Se deschide soluția Assignment1Csharp.sln folosind mediul de dezvoltare Visual Studio
* Se ruleaza opțiunea Build Solution din meniul Build
* Se lansează în execuție fișierul executabil rezultat din folderul Assignment1CSharp\bin\Debug\netcoreapp3.1 folosind comanda .\Assignment1CSharp.exe numeFisierInput

1. Java

* Se ruleză comand gradle clean build
* Se lansează în execuție fișierul JAR autolansabil rezultat din fodlerul build/libs folosind comanda java -jar Assignment1Java.jar numeFisierInput

1. Node.js

* Se rulează comanda npm install
* Se rulează comanda node main.js numeFisierInput

1. PHP

* Se rulează comanda php main.php numeFisierInput

1. Python

* Se rulează comanda python main.py numeFisierInput