
UNIVERSITATEA POLITEHNICA BUCURESTI

FACULTATEA DE AUTOMATICA SI CALCULATOARE



DOCUMENT DE PROIECTARE ARHITECTURALA

ECHIPA CODEBANDITS :

BARBU MADALINA

DOBRE ANDREI-CIPRIAN

STAMATE STEFAN

INSTRUCTOR : BALAN OANA FLORINA

Document de proiectare arhitecturală

1. Introducere

1.1 Scopul sistemului

Sistemul are scopul de a permite utilizatorului comprimarea și incorporarea unor fișiere de diverse tipuri într-un singur fișier, de tip arhivă, cu intenția de a reduce semnificativ spațiul ocupat de acestea și a ușura astfel stocarea lor pe diverse dispozitive. Atunci când se dorește obținerea fișierelor incluse într-o arhivă se poate aplica operația de dezarhivare.

1.2 Obiectivele de proiectare

Proiectul are ca obiectiv principal realizarea compresiei de fișiere. Pentru a fi considerat un proiect de succes acesta trebuie să garanteze următoarele funcționalități:

1.2.1 Compresie lossless

Informația să nu fie pierdută, deci fișierele obținute în urma operației de dezarhivare să fie identice cu cele asupra cărora s-a aplicat operația de arhivare.

1.2.2 Spațiu economisit

În urma comprimării și arhivării unui grup de fișiere trebuie ca spațiul pe disc ocupat de acele fișiere să fie mai mic decât cel inițial.

1.2.3 Uz general

Sistemul trebuie să poată lucra cu fișiere de orice tip, într-un număr oricât de mare (eventual și cu directoare).

1.2.4 Arhive interactive

Aplicatia trebuie sa permita modificarea unei arhive existente prin operatii de adaugare/stergere de fisiere din arhiva.

1.2.5 Portabilitate

Sistemul trebuie sa poata rula pe mai multe sisteme de operare: Linux/Windows.

1.3 Definitii, acronime, abrevieri

Compresie - modificarea modului in care informatia este prezenta pe mediul de stocare pentru a ocupa un spatiu cat mai mic

Arhivare - combinarea mai multor fisiere intr-un singur fisier de tip arhiva

lossless - tip de compresie care se bazeaza pe eliminarea de informatie redundanta ce poate fi ulterior regenerata

zip - unul din cele mai generale formate pentru un fisier de tip arhiva, specific compresiei fara pierderi de date

Rata de compresie - valoarea care exprima cat de mult s-a redus dimensiunea fisierului realizand compresia, in raport cu dimensiunea initiala a acestuia; ea este folosita ca o masura a eficientei programului utilizat

LZW - Lempel–Ziv–Welch - algoritm de compresie de tip lossless bazat pe modele recurente

Huffman - algoritm de compresie de tip lossless bazat pe structurarea datelor sub forma unor arbori

1.4 Referinte

- [Documentul de specificatie a cerintelor](#)
- [File archiver](#)
- [Zip format](#)

2. Arhitectura unui sistem existent

În industrie există deja multe aplicații care servesc aceluiași scop - arhivarea de fișiere pentru o organizare mai bună a sistemelor informatice, pentru optimizarea spațiului utilizat, pentru diminuarea traficului din rețea etc.

Un exemplu clasic de arhivator folosit încă din 1979 în industrie pe platformele Unix este tar. Arhivele tar sunt deseori numite „tarball”, termenul din engleză pentru „bulgăre de smoală” sau „bulgăre de gudron” și exprimă faptul că de acest tip de fișier se poate „lipi” orice alt tip de fișier. Numele provine însă din inițialele cuvintelor (t)ape (ar)chive, programul având inițial rolul de a scrie date pe dispozitive care nu aveau un sistem de fișiere propriu. tar este un arhivator open source care permite și compresia de fișiere folosind diverși algoritmi de compresie. Acesta poate fi folosit fie prin intermediul unei interfețe grafice fie prin linia de comandă.

Formatul fișierului constă într-o serie de obiecte, fiecare obiect incluzând informația aferentă unui fișier și este precedat de un header de 512 octeți. Finalul fișierului arhivă este marcat de două blocuri de date nule (care conțin doar zerouri). Header-urile conțin metadate despre fișiere encodeate în format ASCII pentru a asigura portabilitatea pe o varietate cât mai mare de sisteme. Header-ul conține pe lângă numele fișierului și informații despre utilizatorul care a creat fișierul, permisiuni, dimensiune, data ultimei modificări și un cod de checksum.

Arhivatorul permite compresia fișierelor din arhivă cu diverși algoritmi ca gzip, bzip2, xz, lzip, lzma etc. În funcție de tipul compresiei aplicate arhivele rezultate au diverse formate ca tar.gz sau tar.bzip2.

O problemă remarcabilă a arhivelor tar este faptul că permit arhivarea mai multor fișiere care au același nume și aceeași cale în sistemul de fișiere. În momentul dezarhivării, pe majoritatea sistemelor doar ultimul fișier dezarhivat va rămâne în memorie.

Sursa:

https://en.wikipedia.org/wiki/Tar_%28computing%29

3. Arhitectura propusa

3.1 Prezentare generala a arhitecturii sistemului

Aceasta prezentare este descrisa in documentul de specificatie a cerintelor primit.

3.2 Decompozitia in subsisteme si responsabilitatile fiecarui subsistem

3.2.1 Interfata grafica

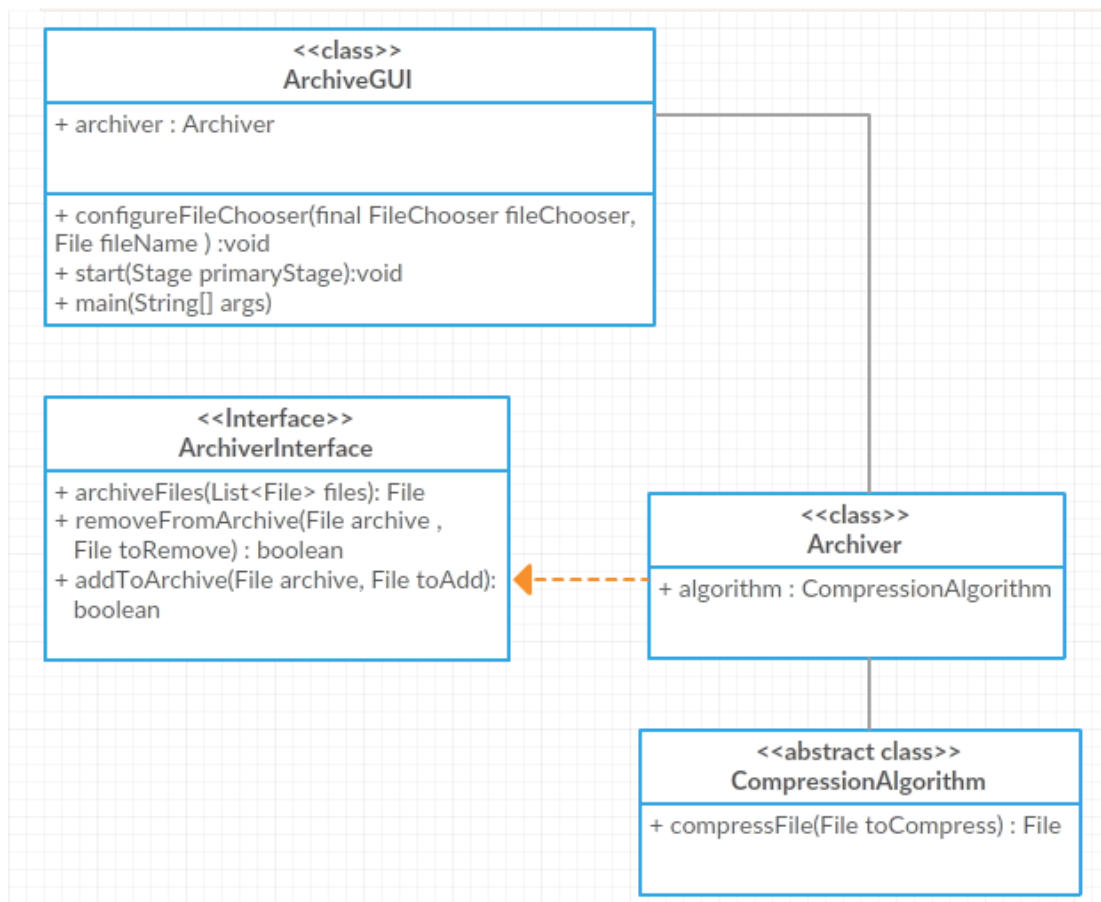
- Se va folosi biblioteca JavaFX
- Interfata va fi separata in doua module. In primul modul utilizatorul va putea alege fisierele dorite pentru arhivare, algoritmul de arhivare sau fisierele de adaugat/sters dintr-o arhiva. In cel de-al doilea modul se vor afisa statisticile dupa ce arhiva a fost creata. In timpul arhivarii, interfata va afisa cantitatea de informatie prelucrata pana la momentul respectiv.

3.2.2 Clasa CompressionAlgorithm

- Reprezinta o clasa abstracta ce ofera un model de implementat pentru fiecare algoritm de compresie.
- Responsabilitati: in extinderea acestei clase se va implementa metoda compressFile ce va intoarce un fisier comprimat folosind algoritmul respectiv.

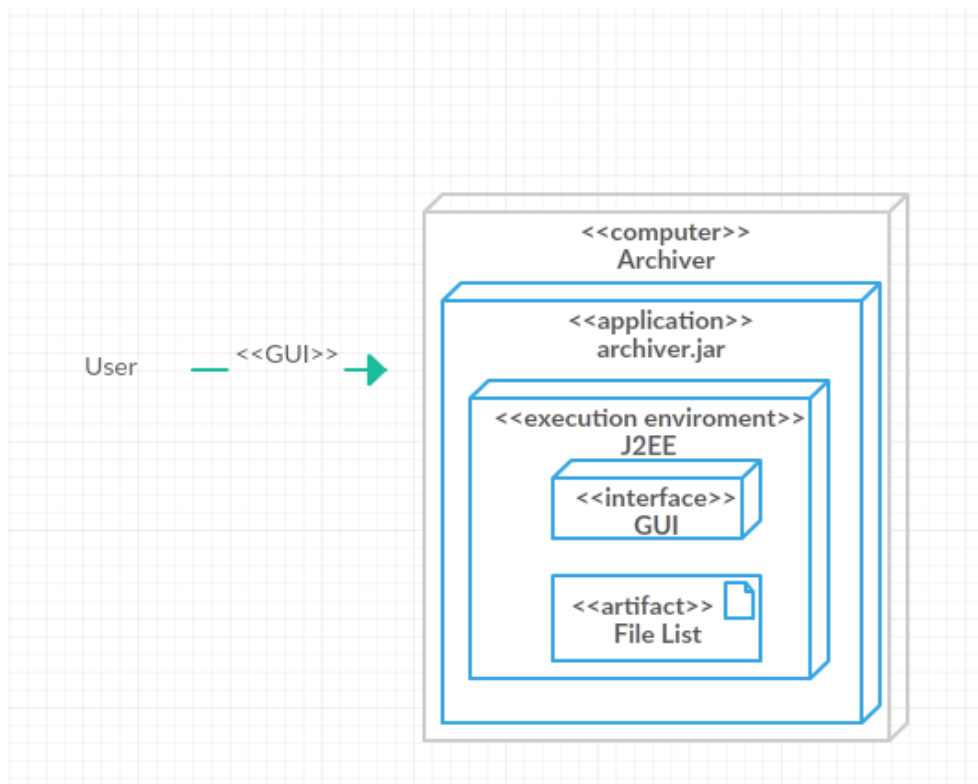
3.2.3 Clasa Archiver

- Implementeaza metodele din interface ArchiverInterface.
- Poate schimba algoritmul folosit pentru arhivarea fisierelor.
- Responsabilitati: realizeaza arhivarea propriu-zisa folosind metoda archiveFiles, realizeaza stergerea unui fisier dintr-o arhiva folosind removeFromArchive, realizeaza adaugarea unui fisier intr-o arhiva folosind addToArchive.



3.3 Distributia subsistemelor pe platforme hardware/software

Utilizatorul foloseste interfata grafica ce se foloseste de subsistemul reprezentat de Archiver ce se foloseste la randul sau de un CompressionAlgorithm.



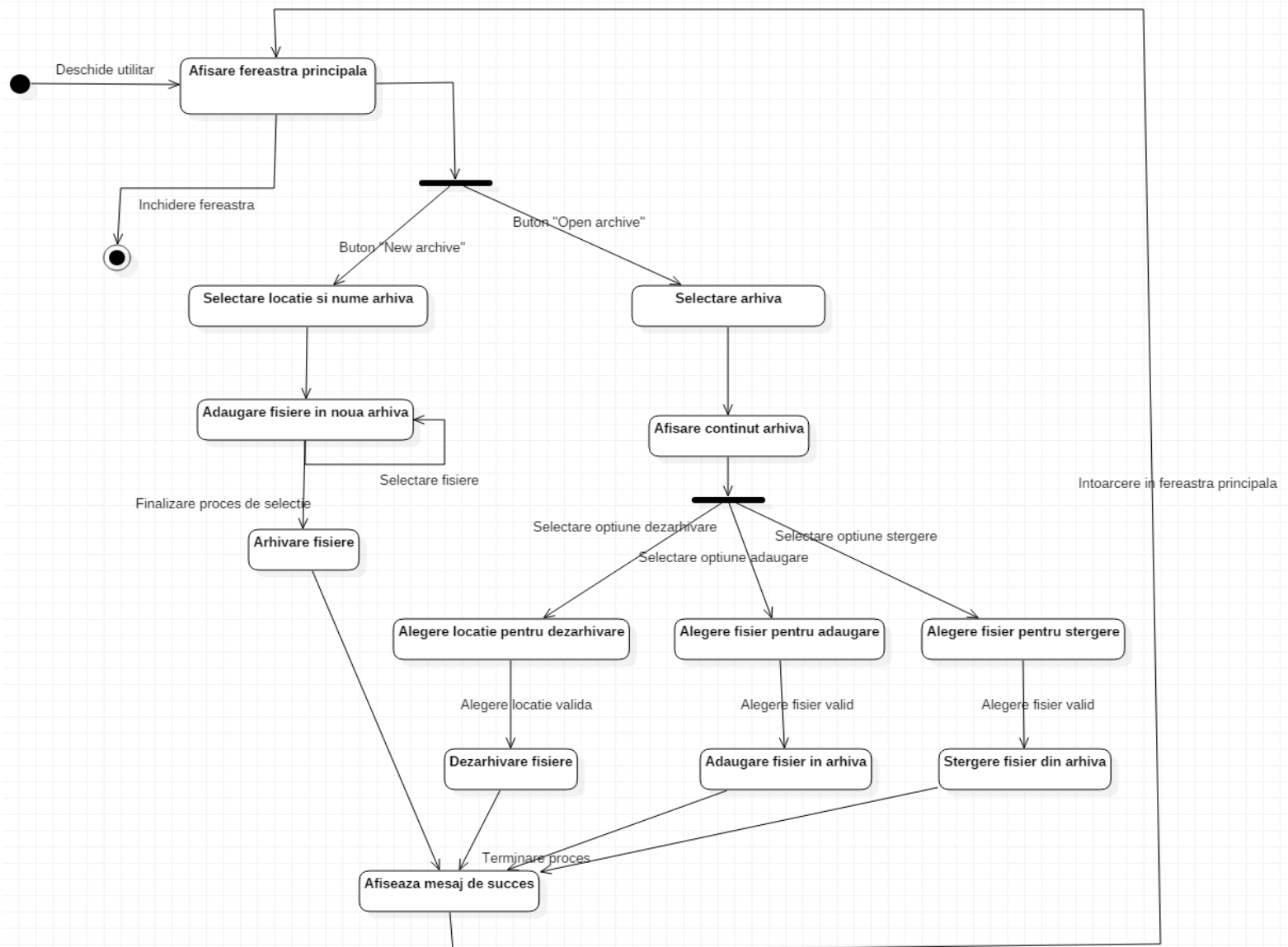
3.4 Managementul datelor persistente

Toate datele necesare unei dezarhivari ulterioare sunt salvate în cadrul fisierului de tip arhiva, care este salvat pe disc. Acelasi fisier va putea de asemenea fi deschis pentru a i se modifica continutul, printr-o operatie de adaugare sau de stergere a unui fisier.

3.5 Controlul accesului utilizatorilor la sistem

Arhivele create de catre sistem vor putea fi accesate si modificate de oricine utilizeaza calculatorul pe care se afla acestea. Totusi, se va urmări pastrarea permisiunilor fisierelor din arhiva.

3.6 Fluxul global al controlului



4. Sarcini interne:

Barbu Madalina: implementare GUI si interfatare cu biblioteca de compresie

Dobre Andrei: implementare algoritm compresie Lempel–Ziv–Welch.

Stamate Stefan: implementare algoritm compresie Huffman