LABORATOR 2 – Arhitecturi pentru sisteme software

Obiectiv – Înţelegerea stilului arhitectural pipe-and-filter. Experimentarea pe o anumită strategie de implementare a stilului pentru a înţelege problematica asociată transformării unui proiect arhitectural în cod.

Sistemul de bază – aplicaţie pentru înregistrarea studenţilor la cursuri. – dat sub formă de cod sursă.

Activitate:

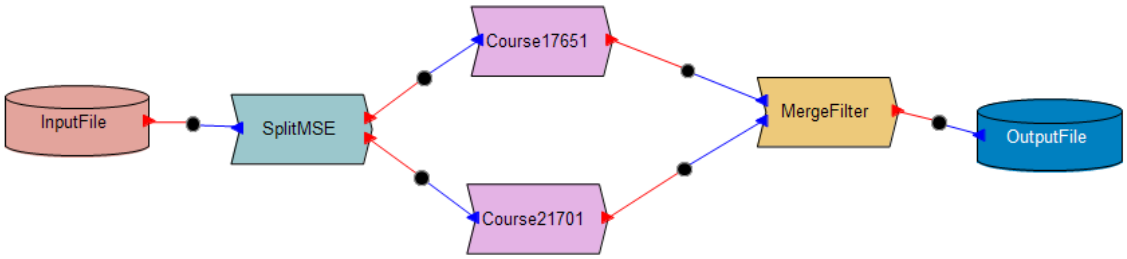
1. Modificarea codului existent şi analiza implicaţiilor.
2. Analizarea arhitecturii sistemului, a modificărilor acesteia şi a implicaţiilor.

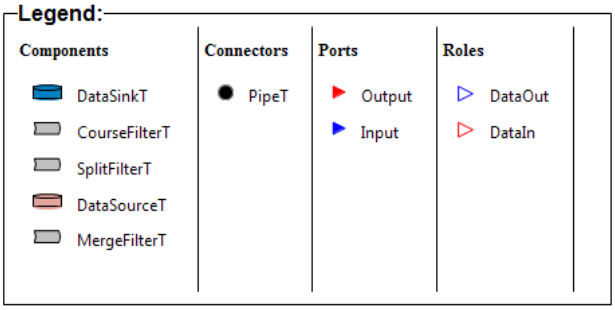
DETALII:

Arhitectura sistemului de bază.

Sistemul de bază este implementat şi asigură verificarea faptului că studenţii au absolvit cursurile necesare pentru a se putea înscrie la cursul de „Arhitecturi Software”. Acestea sunt „Inginerie Software” (cod curs 17651) pentru studenţii de la IS şi „Modelare şi simulare” (cod curs 21701) pentru studenţii de la alte specializări.

Sistemul constă din patru filtre conectate prin conducte.





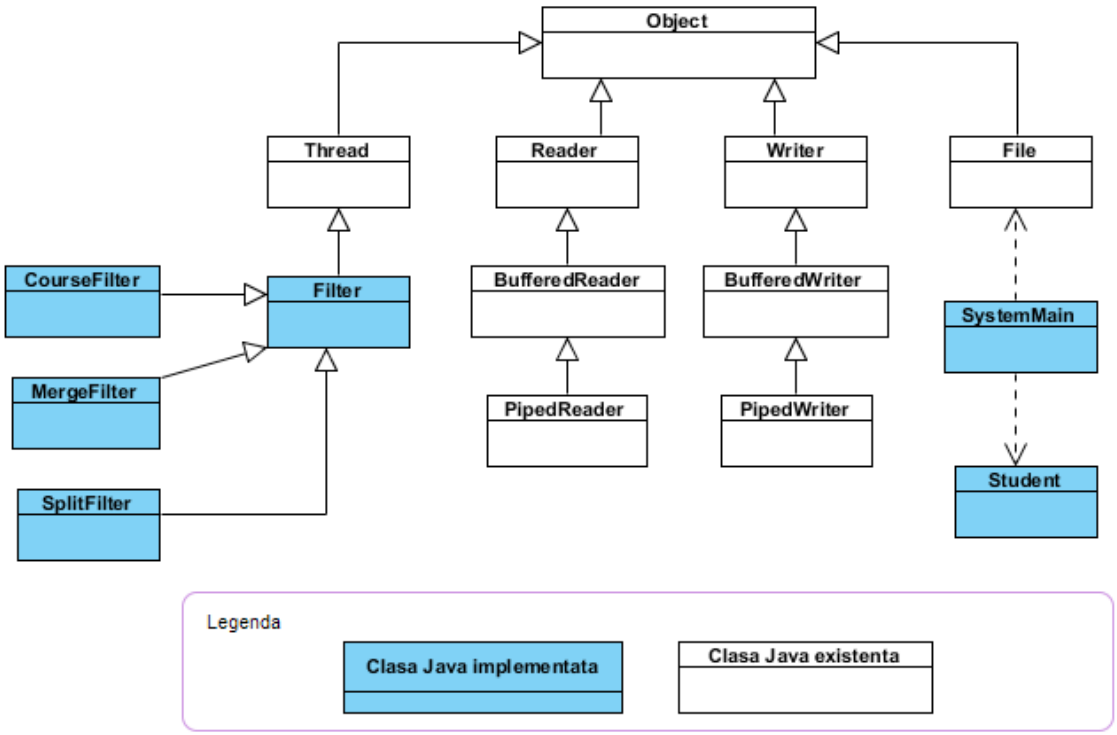
SplitFilter – citeşte înregistrările cu studenţii dintr-un fişier de intrare şi desparte informaţiile în studenţi la IS şi studenţi la alte specializări. Fişierul de intrare este orientat pe câmpuri şi foloseşte spaţiul ca separator. Fiecare linie conţine un singur student şi are următoarea structură:

ID student	Nume şi prenume	Specializarea	Lista cursurilor absolvite
100234009	Popescu Ion	IACD	13567 23324 21701

CourseFilter – două filtre, câte unul pentru fiecare curs necesar – cu responsabilitatea de a determina faptului că studenţii au absolvit cursul respectiv.
Ambele primesc intrare de la filtrul ***SplitFilter***. Ieşirile ambelor filtre sunt trimise la un filtru unificator.
MergeFilter – scrie rezultatul într-un fişier.

Fişierele aplicaţiei
Filter.java – superclasă pentru toate tipurile de filtre. Oferă o interfaţă ce permite determinarea faptului că filtrul funcţionează şi permite terminarea filtrului la apariţia unei întreruperi.
SplitFilter.java – citeşte câte o linie din fişierul de intrare şi verifică dacă studentul este la specializarea care este dată (ca parametru de intrare) la crearea filtrului. Înregistrarea studentului este scrisă pe portul de ieşire „accepted” dacă studentul este la specializarea pe care a fost setat filtrul. Altfel înregistrarea sa este scrisă pe portul de ieşire „rejected”.
CourseFilter.java – La instanţierea filtrului acesta primeşte ca parametru numărul unui curs. Dacă studentul a cărui înregistrare este primită pe portul de intrare a absolvit cursul respectiv, filtrul va transfera înregistrarea pe portul de ieşire. Altfel înregistrarea este eliminată.
MergeFilter.java – Preia intrări de la două porturi de intrare şi scrie toate liniile de text la portul de ieşire.
SystemMain.java – Instanţiază şi porneşte conductele şi filtrele. La executarea clasei utilizatorul va furniza numele fişierelor de intrare şi de ieşire.
Student.java – Reprezintă un student.

Structura codului (arhitectura modulelor)



Instrucţiuni

1. Compilarea şi executarea codului existent : `... > javac *.java`
2. Lansarea aplicaţiei :

```
...>java SystemMain [fişier studenţi] [fişier rezultat]
```

Obs. Folosiţi fişierul `datain.txt` ca fişier de intrare. Fişierul de ieşire va fi creat de aplicaţie.

Pentru a captura informaţiile de debugging furnizate în cursul funcţionării aplicaţiei redirectaţi-le într-un fişier astfel:

```
…>java SystemMain datain.txt results.txt > debug.txt 2>&1
```

Partea 1 : Modificări la sistemul existent

Utilizaţi sistemul existent ca bază pentru crearea a două sisteme noi.
Fiecare sistem nou creat va trebui să adere, de asemenea, la stilul arhitectural pipe-and-filter.
Va trebui să creaţi modelul arhitectural în Acme pentru fiecare sistem nou.
Va trebui să implementaţi cerinţele pentru fiecare sistem nou.

Obs.

1. Fiecare sistem nou va fi tratat independent în ce priveşte documentaţia şi implementarea sa.
2. Modelul arhitectural şi implementarea trebuie să fie sincronizate.
3. Codul va fi comentat cu informaţii referitoare la modificările aduse sistemului de bază.

Cerinţele pentru sistemele noi

Sistemul A : Va păstra întreaga funcţionalitate a sistemului de bază, la care vor adăuga următoarele extensii:

1. Modificarea formatului ieşirii finale a.î. să conţină doar numele studentului şi specializarea.
2. Sistemul va raporta intările rejectate[1]. Acestea vor fi scrise într-un fişier utilizând fie formatul intrărilor acceptate fie formatul fişierului de intrare.
3. Sortarea după nume a listei finale a studenţilor acceptaţi. Nu se cere sortarea intrărilor rejectate.

Sistemul B : Modificarea sistemului de bază pentru a verifica cursurile anterioare necesare pentru cursul „Metode Distribuite şi Tehnologii bazate pe XML". Acestea sunt „Tehnologii Workflow" (cod curs 13456) sau "Proiectarea Sistemelor Software" (cod curs 12333) dacă studentul este la specializarea IACD. Dacă studentul nu este la specializarea IACD, în plus faţă de „Tehnologii Workflow" sau „Proiectarea Sistemelor Software" va trebui să fi absolvit şi „Sisteme Distribuite" (cod curs 13222). Raportul de ieşire are acelaşi format ca cel din sistemul de bază.

Tema scrisă.

1. Descrierile arhitecturale, din perspectivă C&C, ale sistemelor noi : diagramele Acme şi legendele ataşate, codul Acme, explicaţiile textuale necesare înţelegerii corecte a arhitecturii.
2. Instrucţiuni cu modul de realizare şi cu modul de execuţie a sistemelor modificate. Aceste instrucţiuni trebuie să fie suficiente pentru ca urmărindu-le să pot ajunge la rezultatul cerut.

Partea 2 : Analiza sistemului – continuare la temă scrisă.

3. Descrieţi în detaliu arhitectura fiecăruia din sistemele A şi B.
4. Descrieţi modul în care fiecare sistem nou implementează fiecare din funcţionalităţile nou cerute.
5. Precizaţi care au fost deciziile arhitecturale majore şi care au fost problemele critice de proiectare pentru modificarea arhitecturii sistemului de bază. Discutaţi şi justificaţi deciziile de proiectare pe care le-aţi luat (utilizând conceptele arhitecturale şi vocabularul utilizate la curs).
6. Discutaţi atributele de calitate pe care le promovează şi pe care le inhibă arhitecturile acestor sisteme şi deciziile de proiectare pe care le-aţi luat cu implicaţii directe în promovarea sau inhibarea acestor calităţi.
7. Există alte soluţii posibile pe care le-aţi fi putut adopta? Ce v-a determinat să alegeţi soluţia voastră în raport cu aceste soluţii posibile?

[1] Intrările rejectate sunt cele corespunzătoare studenţilor care nu au absolvit cursurile necesare înainte de solicitarea înscrierii la cursul Arhitecturi Software.

Plecând de la proiectul şi implementarea voastă pentru sisteme pipe-and-filter:

8. Se modifică rezultatele calculului dacă se inversează oricare două filtre?
9. S-ar modifica rezultatele dacă filtrele ar fi executate pe calculatoare distribuite cu diferite frecvenţe de ceas?
10. În ce măsură implementările voastre diferă de cele ale noţiunii idealizate de arhitectură pipe-and-filter?
11. Evaluaţi şi discutaţi dificultatea realizării fiecărei modificări, pe baza experienţei voastre. Pentru fiecare modificare:
    - evaluaţi gradul de dificultate în raport cu celelalte modificări.
    - analizaţi în ce măsură gradul de dificultate este corelat cu faptul că stilul pipe-and-filter permite modificarea respectivă.
    - analizaţi în ce măsură gradul de dificultate este corelat cu faptul că infrastructura a fost mai potrivită pentru modificarea respectivă.

Să presupunem că înregistrările studenţilor sunt memorate într-o bază de date accesibilă fiecărui filtru. Pentru acest nou sistem intrarea va fi doar ID-ul studentului iar filtrele pot utiliza ID-ul studentului pentru a accesa restul informaţiilor din baza de date funcţie de necesităţi.

12. Schiţaţi arhitectura acestui sistem. Descrieţi-o astfel încât să poată fi înţeleasă în vederea implementării.
13. Discutaţi avantajele şi dezavantajele modificării sistemului pentru a interacţiona cu baza de date.
14. Deviază acest sistem de la stilul pipe-and-filter pur? Explicaţi şi justificaţi de ce credeţi că acest sistem deviază sau nu de la paradigma arhitecturală pipe-and-filter.

ANEXA

Model în detaliu al codului: