

Descrierea proiectului

1. Introducere

Scopul proiectului este acela de a aplica cunoștințele de POO acumulate în proiectarea și implementarea unei aplicații ce utilizează:

- Baze de date;
- Fire de execuție;
- Procesare XML;
- Încapsulare;
- Interfețe;
- Moștenire;
- DLL.

Aplicația va trebui să asigure sortarea numerelor citite dintr-o bază de date, fără stocarea rezultatului sortării. Sortarea se va realiza în paralel de mai multe fire de execuție, fiecare fir implementând un alt tip de sortare.

2. Arhitectura aplicației

Arhitectura propusă a aplicației este ilustrată în figura 1. La pornire, modulul *INIT PARSER* va citi dintr-un fișier de configurare XML configurația de rulare. Pentru claritate, această configurație este dată sub forma unui XML. Aceasta conține o listă cu modulele care sunt instanțiate, și parametrii acestora. Un exemplu de asemenea fișier este dat în continuare:

```
<?xml version="1.0"?>
<module name="Generator">
  <param name="automatic dataset count" value="500"/>
  <param name="automatic dataset size" value="1000"/>
  <param name="dataset" value="2000"/>
  <param name="dataset" value="1500"/>
</module>
<module name="BubbleSorter"/>
<module name="SelectSorter"/>
<module name="InsertSorter"/>
```

După cum se poate observa, există două tipuri de module: *Generator* și *XSorter*. Modulul *Generator* asigură generarea seturilor de numere care sunt stocate în baza de date, cu posibili parametrii:

- *automatic dataset count*: prin care se asigură specificarea numărului de seturi de date ce trebuie generate automat;

- *automatic dataset size*: prin care se asigură specificarea dimensiunii seturilor de date generate automat. Dacă acest parametru nu este dat, dimensiunea implicită este 1000;
- *dataset*: prin care se specifică un set de date care să se genereze.

Numerele sunt generate de aplicația implementată la specificarea unei opțiuni speciale în fișierul de configurare.

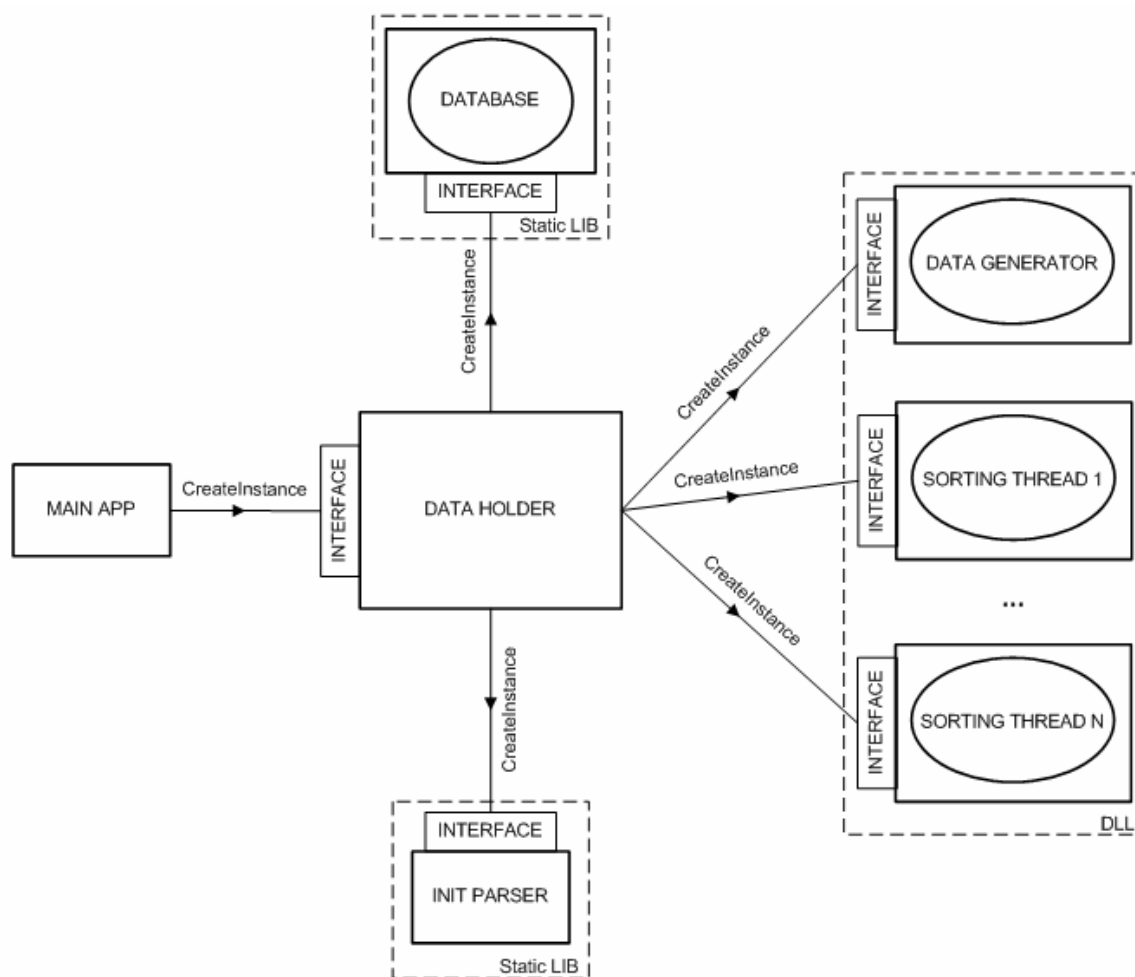


Figura 1. Arhitectura aplicației

Modulele *XSorter* asigură implementarea unui anumit tip de sortare. Acestea nu au parametri, fiecare tip de sortare fiind implementat într-un modul diferit.

În XML-ul dat ca și exemplu, au fost incluse atât generatorul cât și firele de sortare. De regulă, acestea nu rulează deodată: mai întâi se rulează generatorul care asigură generarea datelor și transmiterea lor la baza de date, după care aplicația este oprită și este repornită cu configurația firelor de sortare.

Din punctul de vedere al implementării folosirea firelor de execuție este ilustrată în figura 1 printr-o elipsă. Următoarele module sunt încapsulate într-un DLL:

- *DATA GENERATOR*;
- *SORTING THREAD X*.

Modulele *DATABASE* și *INIT PARSER* sunt incluse într-o bibliotecă statică.

3. Implementare pe echipe

Aplicația va fi implementată pe echipe. Fiecare echipă este formată din maximum 2 studenți, cu posibilitatea alegerii unui coordonator dintr-o altă echipă. Studentul ales ca și coordonator trebuie să-și dea acceptul. Atât echipele cât și coordonatorii trebuie aprobați de subsemnatul.

Rolul coordonatorului nu este acela de a implementa aplicația ci acela de a superviza procesul de dezvoltare, de a asigura o împărțire și planificare eficientă a task-urilor. Coordonatorul va primi în plus 0.5p pentru fiecare proiect coordonat. Un student nu poate coordona mai mult de 3 proiecte. Pentru fiecare coordonare ineficientă studentul coordonator poate fi penalizat cu pana la 1p.

La sfârșitul semestrului, echipele trebuie să:

- Susțină proiectul realizat cu precizarea clară a modulelor implementate de fiecare membru al echipei;
- Predea un document descriptiv al proiectului în limba română/engleză de minimum 7 pagini.