

Tema 2

Termen de predare: laboratorul din săptămâna a XV-a.

Să se proiecteze și să se implementeze un program care să permită gestionarea unei *agende* telefonice.

Descriere generală:

O *agendă* telefonică este o colecție de *contacte*. Un *contact* este format dintr-un *numar telefonic* și un set asociat de informații ce descriu proprietarul acelui număr telefonic (de exemplu, *nume*, *prenume*, *adresa*, *oras*, *judet*).

Programul trebuie să permită următoarele:

- *adăugarea* unui nou *contact* în *agendă*;
- *editarea* unui *contact* existent în *agendă*;
- *ștergerea* unui *contact* din *agendă*;
- *afișarea* tuturor *informațiilor* existente în *agendă* despre un *contact*;
- *crearea* unei noi *agende*;
- *salvarea* unei *agende* într-un fișier;
- *încărcarea* unei *agende* dintr-un fișier;
- *închiderea* unei *agende*;

Programul va fi capabil să gestioneze *o singură agendă la un moment dat*; din acest motiv, crearea sau încărcarea unei noi *agende* va implica salvarea și închiderea *agendei* curente.

Programul va ține evidența modificărilor suferite de *agenda* curentă și care nu sunt salvate. Închiderea explicită sau implicită a *agendei* curente va genera interogarea utilizatorului în privința salvării respectivei modificări.

Salvarea unei *agende* se va realiza în același fișier din care a fost încărcată; în cazul creării unei noi *agende*, salvarea va implica și interogarea utilizatorului în privința căii și a numelui noului fișier.

Cerite:

1. Programul va fi proiectat utilizând șablonul arhitectural MVC (Model-View-Controller, <http://en.wikipedia.org/wiki/Model-View-Controller>).
2. *Agenda* va conține diverse tipuri de *contacte*, ce posedă atât informații comune, cât și informații specifice.
3. Utilizatorul va fi capabil să vizualizeze, în *format grafic*, informația existentă în *agendă*.
4. Utilizatorul va fi capabil să introducă *contacte* noi în *agendă*.
5. Utilizatorul va fi capabil să elimine din *agendă* *contacte* existente.
6. Utilizatorul va fi capabil să actualizeze informația păstrată de un *contact*.
7. Utilizatorul va fi capabil să întrogheze conținutul *agendei*, putând căuta un *contact* după *nume* sau după alte criterii.
8. Utilizatorul va fi capabil să filtreze conținutul *agendei*, pentru a selecta doar *contacte* de un anumit tip.
9. Utilizatorul va fi capabil să salveze într-un fișier conținutul *agendei*.
10. Utilizatorul va fi capabil să restaureze, dintr-un fișier, conținutul *agendei*.

Specificații pentru implementare:

1. Funcționalitatea anterior realizată a aplicației trebuie păstrată..
2. Gestionarea seturilor de contacte se va realiza prin intermediul containerelor STL.
3. Codul sursă trebuie restructurat (*refactoring*, <http://en.wikipedia.org/wiki/Refactoring>) a.î. să fie integrate în program diverse șabloane de proiectare (*design pattern*, [http://en.wikipedia.org/wiki/Design_patterns_\(computer_science\)](http://en.wikipedia.org/wiki/Design_patterns_(computer_science))); de exemplu:
 - a. Factory Method (http://en.wikipedia.org/wiki/Factory_method_pattern)
 - b. Composite (http://en.wikipedia.org/wiki/Composite_pattern)
 - c. Observer (http://en.wikipedia.org/wiki/Observer_pattern)
 - d. Strategy (http://en.wikipedia.org/wiki/Strategy_pattern)
 - e. Visitor (http://en.wikipedia.org/wiki/Visitor_pattern)
4. Se poate utiliza orice framework UI bazat pe C++; de exemplu:
 - a. MFC (http://en.wikipedia.org/wiki/Microsoft_Foundation_Class_Library)
 - b. Qt (<http://qt.nokia.com/products>)
 - c. wxWidgets (<http://www.wxwidgets.org/>)

De reținut:

Rezultatul evaluării va fi determinat de numărul și de modalitatea de integrare a design pattern-ilor în structura aplicației.

Bonus:

Nota finală maximă poate fi mărită cu 1 punct (i.e. nota 11) dacă se integrează în aplicație un framework de baze de date relaționale; de exemplu, Mini Servers 11 (http://wiki.uniformserver.com/index.php/Mini_Servers:_MySQL_5.0.67_Portable) accesat prin Connector/C++ (<http://www.mysql.com/downloads/connector/cpp/>). Succes!