

Șabloane de proiectare (Design Patterns)

Nivel: mediu / avansat

Durata: 35 ore

Obiectivul cursului: exersarea principiilor de design aplicabile în programarea orientată pe obiecte prin studiul șabloanelor de proiectare, modalitățile de exprimare într-un limbaj orientat obiect

Ce veți învăța

- Principiile de design OO, cum se aplică
- Locul și rolul șabloanelor de proiectare în dezvoltarea de software
- Studiul unor șabloane de proiectare importante, cum se aplică
- Partea opțională este dedicată șabloanelor de proiectare ce apar în programarea cu fire de execuție
- Cum se rezolvă diverse probleme practice

Cine poate participa: programatori care doresc să aplice în munca lor zilnică experiența înglobată în șabloanele de proiectare.

Cerințe pentru participanți: cunoașterea unui limbaj orientat pe obiecte cel puțin la nivel mediu, noțiuni de bază în proiectarea orientată pe obiecte, noțiuni de bază UML. Pentru partea opțională, legată de concurență, este necesară în plus cunoașterea prealabilă a suportului oferit de limbaj pentru concurență (Java, C#, etc.) sau de o bibliotecă / framework extern ce oferă suport pentru programare multi-threading (de exemplu thread-uri POSIX).

Infrastructura folosită: proiector, tablă, calculator personal pe care este instalat un mediu de dezvoltare pentru un limbaj orientat obiect (C++, Java, C#)

Bibliografie (minimală): Design Patterns – Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Addison-Wesley, ISBN 0-201-63361-2

Cursuri conexe: Fundamentele UML, Aplicarea programării orientate pe obiecte, UML și a șabloanelor de proiectare

Notă: cursul se va personaliza pe baza profilului tehnic al participanților (limbajul de programare folosit de aceștia, experiența lor practică)

Descriere

Chiar dacă parte din subiectele tratate sunt cunoscute din școală sau din activitatea practică va fi o ocazie excelentă pentru reîmprospătarea și aprofundarea cunoștințelor; cursul abordează un set de șabloane de proiectare care include pe cele definite inițial de GoF (vezi bibliografia) prin prezentarea de exemple și rezolvarea unor studii de caz. Pe lângă înțelegerea lor teoretică este esențială aplicarea lor: recunoașterea contextului de aplicare, variante de aplicare, particularități de implementare ce țin de un limbaj de programare.

Chiar dacă cursul acoperă un număr mare de pattern-uri unele sunt foarte simple și se vor trata rapid, altele fiind mai complexe (sau mai interesante) vor primi mai multă atenție în principal prin rezolvarea unor probleme practice.

Cursul are un pronunțat caracter practic și interactiv, este un prilej excelent de exersare și învățare a tehnicilor de proiectare orientat obiect.

Noțiunile teoretice se vor fixa prin rezolvarea unor probleme practice concrete sau prin implementări în Java, C# sau C++.

Șabloanele legate de concurență înglobează construcții și soluții particulare programării concurente.

Conținut

1. Introducere: definire, organizare, probleme specifice
2. **Principii ale proiectării claselor:** responsabilitate unică, deschis/închis, substituie Liskov, inversarea dependenței, segregare prin interfață
3. **Șabloane de proiectare fundamentale:** delegare, interfața, superclasa abstractă, obiecte imutabile, interfețe marker, proxy
4. **Șabloane creaționale:** method factory, abstrat factory, builder, prototype, singleton, object pool
5. **Șabloane de partiționare:** filtru, compozit, interfața read-only
6. **Șabloane structurale:** adaptor, iterator, bridge, facade, flyweight, dynamic linkage, virtual proxy, decorator, cache management
7. **Șabloane comportamentale:** chain of responsibility, command, interpreter, mediator, snapshot, observer, state, strategy, null object, template method, visitor
8. **Șabloane de legate de concurență** (optional): zone critice, lock object, guarded suspension, balking, scheduler, read/write lock, producător – consumator, terminare în doi pași, buferizare dublă, procesare asincronă, viitor, thread pool, double check locking, active object, monitor object, thread specific storage, leader/followers