

Tehnici Avansate de Programare

Modularizarea codului-sursă

Petru Rebeja, Marius Apetrii

19 Martie 2020

Facultatea de Matematică

Universitatea Alexandru Ioan Cuza, Iași

Introdurre

Recapitulare

- Fluxul de lucru Git permite să implementăm cerințe noi fără să afectăm aplicația în vreun fel.
- Ecosistemul NuGet ne permite să integrăm biblioteci terțe pentru a ne concentra asupra logicii aplicației.
- Principiile SOLID ne ajută să scriem cod de calitate ridicată.

- Cum organizăm codul-sursă în soluție

Concepte de bază în Visual Studio

- O mulțime de resurse necesare pentru crearea unei aplicații.
- Resursele sunt grupate (de obicei) în proiecte.

- O submulțime de resurse menite să fie compilate împreună în:
 - O bibliotecă software (.dll),
 - O aplicație executabilă (.exe),
 - O aplicație Web.

Spațiu de nume (namespace)

- Denotă un domeniu restrâns căruia îi aparține o mulțime de tipuri.
- Este folosit pentru a diferenția tipurile cu același nume.

Șablonul recomandat pentru denumirea spațiilor de nume noi este¹:

`<Company>.(<Product>|<Technology>)[.<Feature>][.<Subnamespace>]`

¹<https://docs.microsoft.com/en-us/dotnet/standard/design-guidelines/names-of-namespaces>

Modularizarea codului sursă

Recapitulare: Acuplarea

Acuplarea este o măsură a gradului de interdependență dintre modulele unui produs software².

²[https://en.wikipedia.org/wiki/Coupling_\(computer_programming\)](https://en.wikipedia.org/wiki/Coupling_(computer_programming))

Coeziunea este măsura în care elementele unui modul aparțin unul de celălalt³.

³[https://en.wikipedia.org/wiki/Cohesion_\(computer_science\)](https://en.wikipedia.org/wiki/Cohesion_(computer_science))

De ce modularizăm codul-sursă?

Separarea funcționalităților

- Fiecare modul este responsabil de o singură funcționalitate.
- Gradul ridicat de coeziune și gradul redus de acuplare rezultă în cod ușor de întreținut.

- Modulele pot fi combinate pentru a obține aplicații noi (ex: pachete NuGet).
- Aceleași module pot fi reutilizate pentru publicarea aplicației pe diferite platforme.

- Modulul expune doar interfața publică și ascunde detaliile de implementare.
- Ulterior detaliile de implementare pot fi modificate fără să afecteze funcționalitatea consumatorilor.

- Este mai ușor să scrii teste pentru module cu acuplare mică și coeziune mare.
- La rândul lor și testele devin suficient de mici și modulare.

Înceiere

Modularizarea codului-sursă ne permite să:

- Separăm funcționalitățile/responsabilitățile,
- Încapsulăm și reutilizăm codul-sursă și
- Să testăm mai codul mai ușor/rapid.

Vă mulțumesc!

Mulțumesc pentru atenție!