

S.O.L.I.D Principiile programării orientate pe obiecte

Principiile de proiectare și programare orientată pe obiect s-au dezvoltat în timp, pe baza problemelor practice și a experienței dobândite de către alți programatori.

Principiile SOLID sunt un subset al multor principii promovate de Robert C. Martin și introduse în lucrarea sa *Agile Software Development: Principles, Patterns, and Practices*.

În programarea orientată pe obiecte, termenul SOLID este un acronim pentru cinci principii de proiectare menite să facă design-ul de software mai ușor de înțeles, flexibil și întreținut.

Single responsibility principle (Principiul responsabilității unice)

- ✓ Orice context (clasă, funcție, variabilă etc.) trebuie să aibă o unică responsabilitate, iar această responsabilitate trebuie să fie în întregime încapsulată în context. Toate serviciile sale trebuie să fie orientate pentru a servi această unică responsabilitate.

Open-closed principle (Principiul deschis/închis)

- ✓ Acest principiu implică ca entitățile software (clase, module, metode) să fie deschise pentru extensie dar închise pentru modificare. Astfel, o entitate își poate extinde comportamentul fără modificări în codul sursă.

Liskov substitution principle (Principiul Liskov al substituției)

- ✓ Dacă S este un subtip al lui T, atunci obiectele de tip T pot fi substituite cu obiecte de tip S fără a afecta niciuna din proprietățile programului (corectitudine, realizarea execuției etc.).

Interface Segregation Principle (Principiul separării interfețelor)

- ✓ Acest principiu afirmă că niciun client nu trebuie să fie forțat să depindă de metode pe care nu le utilizează. Interfețele trebuie separate în alte interfețe mai mici și mai specifice.

Dependency Inversion Principle (Principiul inversării dependențelor)

- ✓ Modulele de înalt nivel nu ar trebui să depindă de modulele de nivel scăzut. Ambele ar trebui să depindă de abstracții.

Programarea orientată pe obiect pornește de la o serie de principii de bază precum:

1. posibilitatea de reutilizare a codului,
2. capacitatea de a-l extinde cu ușurință,
3. încapsularea proprietăților și comportamentului obiectelor,
4. mentenabilitate.

Totuși, aceste lucruri nu vin de la sine, iar un software developer bun trebuie să urmărească o serie de reguli și principii pentru a se asigura că programul său respectă criteriile de mai sus. Pentru acest lucru sunt folosite cele cinci principii SOLID.