Sebastian HIRTIE
Anul IV, AC-IS

Tema 1 – PSSC

Avand ca origine principiile promovate de Robert C. Martin, s-au format un set de reguli care au ca menire imbunatatirea arhitecturii aplicatiei. Prin respectarea lor se incearca evitarea unui cod greu de adaptat, fragil si cu multe dependente. Destinate programarii orientate pe obiecte, ele au ca scop proiectarea unui sistem robust, flexibil si reutilizabil.

Acesta ideologie a fost asociata ulterior cu acronimul S.O.L.I.D. de catre Michael Feathers si inglobeaza urmatoarele principii:

1. **Principiul singurei responsabilitati** –"O clasă ar trebui să aibă unul și numai unul motiv pentru a se schimba."

   Fiecare componenta, clasa ar trebui sa aiba o singura responsabilitate sau functionalitate. Aceasta inseamna ca are un singur motiv sa se schimbe. Daca ea ar avea mai multe responsabilitati, este probabil ca modificarea uneia dintre ele sa influenteze sau sa schimbe logica celorlalte.
2. **Principiul deschis / inchis** –" Clasa este deschisa pentru extindere, dar inchisa pentru modificare."

   Daca cerintele se schimba, ar trebui sa facem o clasa care sa aiba comportamentul cerut. Acesta se realizeaza prin mostenire, astfel codul clasei de baza nu este modificat.
3. **Principiul Substitutiei Liskov** --"Clasele derivate trebuie sa poate fii substituite cu cele de baza"

   Clasa derivate nu trebuie sa piarda sau sa modifice nici o functionalitate a clasei de baza, daca intr-un program se modifica un obiect al clasei derivate cu unul al clasei de baza, comportamentul ar trebui sa fie identic.
4. **Principiul Segregarii Interfetei** --" Clienții nu ar trebui să fie nevoiți să implementeze interfețele pe care nu le utilizează."

   Interfetele ar trebui sa fie mici si specifice, sa nu forteze clientul sa precizeze ceva ce nu foloseste.
5. **Principiul Inversarii Dependentei** –" Abstractizările nu trebuie să depindă de detalii. Detaliile trebuie să depindă de abstractizări."

   Modulele nu trebuie sa comunice direct unele cu altele, ci prin intermediul unei interfete, astfel daca un modul se schimba celelalte vor comunica la fel, doar interfata se va modifica.

Principiile SOLID ofera indrumare pentru obtinerea unui sistem scalabil, testabil si usor de intretinut.

https://team-coder.com/solid-principles/
https://www.vikingcodeschool.com/software-engineering-basics/solid-design-principles
https://hackernoon.com/solid-principles-made-easy-67b1246bcdf
https://blog.scottlogic.com/2018/06/26/solid-principles.html#liskov-substitution-principle