SOLID

- Single Responsibility Principle (Principiul responsabilitatii unice)
- Open Closed Principle (Principiul deschis/închis)
- Liskov Substitution Principle (Principiul Liskov al substitutiei)
- Interface Segregation Principle (Principiul separarii interfetelor)
- Dependency Inversion Principle (Principiul inversarii dependentelor)

În programarea orientată pe obiect, termenul SOLID este un acronim mnemonic pentru cinci principii de proiectare menite să facă design-ul de software mai ușor de înțeles, flexibil și întreținut. Este strâns legată de principiile de proiectare a software-ului GRASP. Principiile sunt un subset al multor principii promovate de Robert C. Martin. Deși se aplică în cazul oricărui proiect orientat pe obiecte, principiile SOLID pot, de asemenea, să formeze o filozofie de bază pentru metodologii cum ar fi dezvoltarea agilă sau dezvoltarea software-ului adaptiv. Teoria principiilor SOLID a fost introdusă de Martin în principiile de proiectare și modele de design ale lucrării din 2000, deși acronimul SOLID însuși a fost introdus mai târziu de Michael Feathers

Principiul responsabilitatii unice:

- Nu trebuie sa existe niciodata mai mult de un motiv pentru ca o clasa sa sufere modificari
- Nu trebuie sa existe mai mult de o responsabilitate per clasa
- SRP este unul dintre cele mai simple principii și unul dintre cele mai grele de obținut. Îmbinarea responsabilitățile este ceva ce facem în mod natural. Găsirea și separarea acestora reprezintă o mare parte din proiectarea software-ului.

Principiul deschis/inchis

- Entitatile software (clase, module, functii) trebuie sa fie deschise pentru extindere, dar închise pentru modificare

Principiul separarii interfetelor

- Clientii nu trebuie constrânsi sa depinda de interfete pe care nu le utilizeaza

Principiul inversarii dependentelor

- Modulele de nivel înalt nu ar trebui sa depinda de module de nivel jos, ambele ar trebui sa depinda de abstractizari
- Abstractizarile nu ar trebui sa depinda de detalii, detaliile ar trebui sa depinda de abstractizari

Principiile nu vor transforma un programator rău într-un bun programator. Principiile trebuie aplicate cu judecata. Dacă acestea sunt aplicate de către rote este la fel de rău ca și cum nu ar fi aplicate deloc.