

Tema 1 – PSSC

Principiul responsabilității unice este un principiu de programare a calculatorului care prevede că fiecare modul sau clasă trebuie să aibă responsabilitatea asupra unei singure părți a funcționalității furnizate de software și că această responsabilitate ar trebui să fie în întregime încapsulată de clasă.

Principiul deschis / închis a fost folosit în două moduri. Ambele moduri folosesc generalizări (de exemplu, funcții de moștenire sau delegare) pentru a rezolva dilema aparentă, dar obiectivele, tehnicile și rezultatele sunt diferite.

Noțiunea lui Liskov despre un subtip de comportament definește noțiunea de substituibilitate a obiectelor; adică, dacă S este un subtip de T, atunci obiectele de tipul T dintr-un program pot fi înlocuite cu obiecte de tip S fără a altera oricare dintre proprietățile dorite ale acelui program

Principiul de segregare a interfeței (ISP) prevede că niciun client nu ar trebui să fie obligat să depindă de metodele pe care nu le utilizează [1]. ISP împarte interfețe care sunt foarte mari în mai mici și mai specifice, astfel încât clienții vor trebui doar să știe despre metodele care le interesează.

În proiectarea orientată pe obiecte, principiul inversării dependenței se referă la o formă specifică de separare a modulelor software.

Atunci când se respectă acest principiu, relațiile de dependență convenționale stabilite de la module la nivel înalt, de stabilire a politicilor, la module de dependență de nivel inferior sunt inversate, ceea ce face ca modulele de nivel înalt să fie independente de detaliile implementării modulelor de nivel inferior. Principiul prevede:

A. Modulele de nivel înalt nu ar trebui să depindă de modulele de nivel inferior. Ambele ar trebui să depindă de abstractizări.

B. Abstractizările nu trebuie să depindă de detalii. Detaliile ar trebui să depindă de abstractizări.