

Zasady OOP

1. **SRP - The Single Responsibility Principle:**

Zasada pojedynczej odpowiedzialności - Klasa powinna mieć tylko jedną odpowiedzialność (nigdy nie powinien istnieć więcej niż jeden powód do modyfikacji klasy).

2. **OCP - The Open-Closed Principle**

Zasada otwarte-zamknięte - Klasy (encje) powinny być otwarte na rozszerzenia i zamknięte na modyfikacje.

3. **LSP - The Liskov Substitution Principle**

Zasada podstawienia Liskov - Funkcje które używają wskaźników lub referencji do klas bazowych, muszą być w stanie używać również obiektów klas dziedziczących po klasach bazowych, bez dokładnej znajomości tych obiektów.

4. **DIP - The Dependency Inversion Principle**

Zasada odwrócenia zależności - Wysokopoziomowe moduły nie powinny zależeć od modułów niskopoziomowych - zależności między nimi powinny wynikać z abstrakcji.

5. **ISP - The Interface Segregation Principle**

Zasada segregacji interfejsów - Wiele dedykowanych interfejsów jest lepsze niż jeden ogólny.

6. **REP - The Release-Reuse Equivalency Principle**

Zasada równoważności zwolnienia i ponownego wykorzystania - Tylko komponenty wydane przez system śledzenia mogą być skutecznie ponownie wykorzystane.

7. **CCP - The Common Closure Principle**

Zasada wspólnego zamknięcia - Klasy w pakiecie powinny być zamknięte razem z tymi samymi rodzajami zmian. Zmiana wpływająca na zamknięty pakiet dotyczy wszystkich klas **tylko** w tym pakiecie.

8. **CRP - The Common Reuse Principle**

Zasada wspólnego ponownego użycia - Klasy w pakiecie są ponownie używane razem. Jeśli ponownie wykorzystasz jedną z klas w pakiecie, ponownie wykorzystasz wszystkie.

9. **ADP - The Acyclic Dependencies Principle**

Zasada acyklicznych uzależnień - to zasada, która stwierdza, że "wykres zależności pakietów lub komponentów nie powinien mieć żadnych cykli". Oznacza to, że zależności tworzą ukierunkowany wykres acykliczny.

10. **SDP - The Stable Dependencies Principle**

Zasada trwałych zależności - Zależności między pakietami oprogramowania powinny być zgodne ze stabilnością pakietów. Oznacza to, że dany pakiet powinien zależeć tylko od bardziej stabilnych pakietów.

11. **SAP - The Stable Abstractions Principle**

Zasada stabilnych abstrakcji - Abstrakcja pakietu powinna być proporcjonalna do jego stabilności. Klasa abstrakcyjna wydaje się być bardziej stabilna, ponieważ nie zależy od konkretnych klas.