Object-Oriented Design Principles (OODP)

Object-Oriented Design Principles jsou souborem osvědčených postupů a pravidel, které pomáhají vývojářům vytvářet software, jenž je snadno udržovatelný, rozšiřitelný a srozumitelný. Tyto principy tvoří základ pro pokročilé návrhové vzory.

Základní pilíře (SOLID)

Nejdůležitější sadou principů v OOD je zkratka **SOLID**, kterou zpopularizoval Robert C. Martin („Uncle Bob“).

Zkratka	Princip	Stručný popis
S	**SRP** (Single Responsibility)	Třída by měla mít pouze jednu odpovědnost (jeden důvod ke změně).
O	**OCP** (Open/Closed)	Entity by měly být otevřené pro rozšíření, ale uzavřené pro modifikaci.
L	**LSP** (Liskov Substitution)	Podtřídy musí být nahraditelné svými nadtřídami, aniž by se změnilo chování programu.
I	**ISP** (Interface Segregation)	Klienti by neměli být nuceni záviset na rozhraních, která nepoužívají.
D	**DIP** (Dependency Inversion)	Záviset by se mělo na abstrakcích, nikoliv na konkrétních implementacích.

Další důležité principy

Kromě SOLID existují další pravidla, která jsou pro kvalitní design klíčová:

1. DRY (Don't Repeat Yourself)

Každá znalost nebo logika musí mít v rámci systému jedinou, jednoznačnou reprezentaci. Duplicita kódu vede k chybám při aktualizacích.

2. KISS (Keep It Simple, Stupid)

Většina systémů funguje nejlépe, pokud jsou jednoduché. Je třeba se vyhýbat zbytečné komplexitě a „předimenzovanému“ inženýrství.

3. Composition over Inheritance

Dává přednost skládání objektů (**kompozici**) před dědičností. Kompozice je flexibilnější a snižuje pevnou vazbu mezi třídami.

Ukázka implementace

Příklad principu **SRP** v jazyce Java:

```java
// ŠPATNĚ: Třída řeší data i uložení
public class User {
    private String name;
    public void saveToDatabase() { /* ... */ }
}

// SPRÁVNĚ: Odpovědnosti jsou oddělené
public class User {
    private String name;
}

public class UserRepository {
    public void save(User user) {
        // Logika pro uložení do databáze
    }
}
```

Související témata

- Návrhové vzory (Design Patterns)
- Základy OOP (Zapouzdření, Polymorfismus)
- Refaktorování kódu

Tagy: #vývoj #architektura #OOP #SOLID

From:
https://serviceit.cz/ - **IT ENCYKLOPEDIE**

Permanent link:
https://serviceit.cz/doku.php?id=solid

Last update: **2026/01/02 11:31**