

Архитектурни стилове на софтуерни системи- Резюме

- **Software Architecture:** Колекция от структури, които представят различни гледни точки върху системата.
- **Architectural Styles:** Определят семейство от системи чрез модел на структурна организация.

Основни архитектурни стилове

1. Pipe-and-Filter

- **Компоненти (Filters):** Прехвърлят данни последователно към следващия компонент.
- **Конектори (Pipes):** Представяват механизмите за прехвърляне на данни.
- **Предимства:** Лесен за разбиране, гъвкавост при поддръжка и повторна употреба, лесно имплементиране на паралелизъм.
- **Недостатъци:** Трудно за интерактивни приложения, ниска производителност, трудност при споделяне на глобални данни.

2. Shared-Data (Repository/Blackboard)

- **Компоненти:** Прехвърлят големи количества данни чрез споделени данни.
- **Предимства:** Скалируемост, паралелна работа на компонентите, централизирано управл. на данните.
- **Недостатъци:** Трудно при разпределени среди, зависим. между компонентите и споделените данни.

3. Client-Server

- **Система:** Състои се от сървъри, които предлагат услуги, и клиенти, които ги използват.
- **Предимства:** Централизация на данните, сигурност, лесно имплементиране на резервни копия и възстановяване.

- **Недостатъци:** Натоварване на сървъра при голям брой клиенти, нужда от резервираност.

4. Layered

- **Система:** Организирана в йерарх. подредени слоеве.
- **Предимства:** Скриване на вътрешната структура на слоевете, абстракция, по-добра кохезия.
- **Недостатъци:** Трудност при разграничаване на слоевете, компромис с производителността.

5. Object-Oriented

- **Обекти:** Представяват изчислителни единици, отговорни за своята вътрешна репрезентация.
- **Предимства:** Енкапсулация на данни и логика, системна декомпозиция.
- **Недостатъци:** Обектите трябва да знаят идентичността на другите обекти, странични ефекти при методите.

6. Implicit Invocation (Event-Based)

- **Компоненти:** Взаимод. чрез излъчване на събития.
- **Предимства:** Слаба свързаност, лесна подмяна и повторна употреба на компоненти, ефективност при разпределени системи.
- **Недостатъци:** Неясна структура на системата, трудност при дебъгване, проблеми с надеждността.

Заклучение

- Архитектурните стилове представляват полезни знания за решения на често срещани проблеми в софтуерните системи.
- Стиловете не могат да се прилагат изолирано – системите се проектират чрез комбинация от различни стилове, за да отговорят на максимално много потребителски изисквания.