

Изгледи на софтуерната архитектура - Резюме

4 + 1 модел на софтуерната архитектура

1. **Логически изглед:** Показва основните абстракции в системата, като обекти, класове и компоненти.
2. **Изглед на процесите:** Показва системата като съвкупност от взаимодействащи си процеси по време на изпълнение.
3. **Изглед на кода:** Показва как отделните елементи на системата се разполагат във файлове код.
4. **Физически изглед:** Показва как софтуерните компоненти са разпределени между хардуерните възли в системата.
5. **Сценарии на употреба:** Допълнителен изглед, който показва как системата се използва в различни ситуации.

Архитектурни изгледи

- **Изглед:** Съвкупност от софтуерни елементи, техните външно видими свойства и връзките между тях.
- **Изглед (view)** – конкретно документирано представяне на дадена изглед;
- **Видове изгледи:**
 1. **Модулни изгледи:** Елементите са модули – единици работа за изпълнение.
 2. **Изгледи на компонентите и конекторите:** Елементите са компоненти и конектори, които се проявяват по време на изпълнението.
 3. **Изгледи на разположението:** Показват връзката между софтуерните елементи и елементите на околната среда.

Модулни изгледи - елементите в модулните изгледи са модули = единици работа за изпълнение

- **Декомпозиция на модулите:** Връзките между модулите са от вида "**Х е под-модул на Y**". Това се прави рекурсивно до момента, в който елементите станат достатъчно прости.
- **Употреба на модулите:** Връзките между модулите са от вида "**Х използва Y**". Това обуславя възможността за лесно добавяне на нова функционалност, обособяване самостоятелни подмножества от функционалност, както и позволява последователната разработка.
- **Изглед на слоевете:** Когато върху употребата са наложени стриктни правила, се обособяват слоеве. Модулите от слой номер N могат да се възползват само от услугите на модулите от слой номер N-1. Слоевете често са реализирани като виртуални машини или обособени подсистеми, които скриват детайлите относно работата си от следващия слой; Не е прието слоевете да се прескачат; Изгледът позволява без особени сътресения да бъде подменен цял един слой
- **Йерархия на класовете:** В терминологията на ООП, модулите се наричат "класове", а връзките между класовете са от вида "класът X наследява класа Y" и "обекта X е инстанция на клас Y"

Изглед на процесите/ Изгледи на компонентите и конекторите /

- **Елементи:** Процеси (или нишки), изпълнявани в C (**компоненти**) и комуникационни, синхронизационни или блокиращи **операции** между тях (**конектори**).
- **Въпроси:** Кои са основните изчислителни процеси и как те си взаимодействат? Кои са основните споделени ресурси? Как се развиват данните в системата? Кои части от системата могат да работят паралелно?

Изгледи на разположението

- **Изглед на внедряването (Deployment):** Показва как софтуерът се разполага върху хардуера и комуникационното

оборудване. Връзките са напр. “внедрен върху” или “мигрира върху”;

- **Файлов изглед:** Показва кой модул къде се помещава по време на различните фази на реализация.
- **Изглед на инсталацията:** Показва как софтуерните компоненти се инсталират и конфигурират.
- **Разпределение на работата:** Показва кой модул от кой екип се реализира. Елементите са модули и екипи, а връзките са кой модул от кой екип се разработва;

Избор на изгледи

- **Зависи от системата:** Различните системи изискват различни изгледи. Например, съгласно RUP (Rational Unified Process) -**iterative software development process framework** by the Rational Software Corporation, a division of IBM:
 - **Logical:** Елементите са ключови абстракции, а връзките – взаимодействие между тях.
 - **Process:** Адресира паралелното изпълнение и разпределението на процесите.
 - **Development:** Типичен изглед на разпределението на работата.
 - **Physical:** Показва кой процес на кой хардуер се изпълнява.

4 + 1 модел на софтуерната архитектура 1) **Логически изглед** – показва основните абстракции в С, като **обекти, класове и компоненти** 2) **Изглед на процесите** – показва С като съвкупност от **взаимодействащи си процеси по време на изпълнение** 3) **Изглед на кода** – Показва как отделните елементи на системата се **разполагат във файлове код** 4) **Физически изглед** – показва как софтуерните компоненти са **разпределени между хардуерните възли в С** +1) **Съответните сценарии на употреба**