

به نام خدا

تمرین سری چهارم
مهندسی نرم افزار ۲

نعیم اصفهانی
۸۴۲۰۱۰۰۳

۱. عامل‌گرایی

۱،۱- مقایسه

روش عامل‌گرا اصولاً بر مبنای روش شیء‌گرا می‌باشد:

- هردو به ارتباط بین موجودیت‌ها اهمیت می‌دهند.
- هردو داده و حالت را کیسوله می‌کنند.

ولی این تفاوت‌ها را با آن دارد:

- اشیاء موجودیت‌های منفعل هستند در حالی که عامل‌ها عناصری فعال هستند.
- عامل‌ها ساختاری برای ارائه‌ی عقاید و نظرات نیز دارند.
- کنترل اشیاء توسط یک عامل بیرونی است؛ در حالی که عامل‌ها عناصری خودمختارند.
- عامل‌ها امکان فعال‌سازی رفتار را نیز دارند (دقیقاً کار خواسته شده را انجام ندهند).

۱،۲- GAIA

مدل عامل:

مدلی است که در آن گونه‌ها و نمونه‌های زمان اجرای عامل‌ها را شناسایی می‌کنیم. گونه‌ی عامل مجموعه‌ای از نقش‌ها در سیستم است. بین یک نوع و یک گونه، ارتباطی یک به یک برقرار است. اما گاهی به خاطر در نظر گرفتن کارایی بیش از یک نقش را به یک گونه از عامل اختصاص می‌دهند. ارتباط بین نمونه عامل‌ها و گونه‌ی آن‌ها نیز توسط این مدل مشخص می‌شود.

مدل خدمت:

هدف اصلی این مدل مشخص نمودن خدمات هرکدام از نقش‌های در سیستم است. این مدل خصوصیات اصلی هر خدمت را مشخص می‌کند. برای هر خدمت در این مدل ورودی‌ها، خروجی‌ها، پیش‌شرط‌ها و پس‌شرط‌ها را مشخص می‌کنیم ولی اطلاعاتی در مورد جزئیات پیاده‌سازی آن‌ها ارائه نمی‌کنیم.

۲. معماری نرم افزار

۲،۱- ویژگی های کیفیتی

				×	تغییر پذیری
			×	✓	کارایی
		×	✓	×	امنیت
	×	×	✓	×	آزمون پذیری
✓	×	✓	×	✓	استفاده پذیری
آزمون پذیری	امنیت	کارایی	تغییر پذیری	دسترسی	

در زیر برخی از موارد که بدیهی به نظر نمی رسند توضیح داده می شوند:

۱. آزمون پذیری و دسترسی
باید بتوان دسترسی را نیز آزمود و این هزینه ی آزمون را زیاد می کند.
۲. امنیت و تغییر پذیری
اگر تغییر پذیری بالا باشد امکان رفع سریع مشکلات امنیتی فراهم می شود.
۳. استفاده پذیری و تغییر پذیری
استفاده از فایل های پیکربندی و محدود کردن انتخابها باعث کم شدن استفاده پذیری می شود.
۴. آزمون پذیری و کارایی
کدی که کارا شده باشد قابلیت فهم و آزمون پذیری آن کاهش می یابد.
۵. استفاده پذیری و کارایی
با بالا رفتن کارایی کاربر به دلیل پاسخگویی بهتر راحت تر از سیستم استفاده می کند.
۶. استفاده پذیری و آزمون پذیری
آزمودن سطح بالای سیستمی که کاربر با آن راحت کار می کند هم راحت است.

۲،۲- تعاریف

معماری قابل اجرا:

معماری نرم افزار که هسته‌ی نرم افزار را تشکیل می‌دهد و به مرور زمان در طی فرآیند تولید سیستم کامل می‌شود. این معماری که جنس آن از جنس طراحی است امکان اجرا را دارد و به ما کمک می‌کند که تشخیص دهیم آیا طراحی نرم افزار قابلیت (معمولا غیر عملیاتی) مورد انتظار را دارد یا نه. به طور مثال در طراحی یک سیستم که قرار است یک پروتکل را استفاده کند و حد خاصی از کارایی را ارائه دهد بسیار مناسب است این پروتکل طراحی شده در قالب معماری ابتدا آزموده شود.

زبان توصیف معماری:

زبان‌بست (گرافیکی و یا متنی) برای توصیف یک سیستم نرم افزاری بر مبنای عناصر معماری و روابط آن‌ها. این زبان به ما امکان توصیف بدون ابهام، قابل فهم و قابل استفاده‌ی مجدد از معماری را می‌دهد. مثال‌های آن عبارتند از:

- Aesop
- Rapide
- Wright
- MetaH
- LILEAnna
- UniCon

موثرهای بر معماری:

مجموعه‌ای از نیازمندی‌های عملیاتی، کیفیتی و تجاری که معماری یا یک عنصر را تحت تاثیر قرار می‌دهند. این عوامل عبارتند از:

- چشم‌انداز معماری
- نیازمندی‌های معماری
- مفروضات و ضروریات و تمایلات معماری
- اصول معماری
- روح سازمان دهی