



تئوری و آزمایشگاه

تحلیل و طراحی شیء گرا

مرجعی برای دروس

تحلیل و طراحی سیستم‌ها ، تحلیل و طراحی شیء گرا و مهندسی نرم افزار

مؤلف:

دکتر محمدعلی ترکمانی

عنوان و نام بدیاور	سروشناسه
نکمانی، محمدعلی، - ۱۳۵۴ : تئوری و آزمایشگاه تحلیل و طراحی شیء‌گرا [کتاب] : مرجعی برای دروس تحلیل و طراحی سیستم‌ها و مهندسی نرم‌افزار / مؤلف: محمدعلی نکمانی.	
مشهد: ارسسطو، ۱۳۹۰، ۳۲۲ ص: مصور، جداول، نمودار.	مشخصات نشر
۹۷۸-۶۰۰-۴۳۲-۰۰۷-۸	مشخصات ظاهری
فیبا : کتابنامه.	شایع
تجزیه و تحلیل سیستم‌ها -- آزمون‌ها و تمرین‌ها (عالی)	وضع
نرم‌افزار -- مهندسی	موضوع
شیء‌گرایی (کامپیوتر)	موضوع
تجزیه و تحلیل سیستم‌ها -- آزمون‌ها و تمرین‌ها (عالی)	موضوع
نرم‌افزار -- مهندسی -- آزمون‌ها و تمرین‌ها (عالی)	موضوع
QA۴۰۲/۴۲۹۱۳۹۰	رده بندی کنگره
۵۳۱/۲۸۲۲۳	رده بندی دیوبی
۴۰۰۱۰۵	شماره کتابشناسی ملی

نام کتاب : تئوری و آزمایشگاه تحلیل و طراحی شیء گرا

موضوع: مهندسی نرم افزار

موضوع: تحلیل و طراحی شیء گرا

موضوع: UML

موضوع: Visual Paradigm

مؤلف : دکتر محمدعلی نکمانی

ناشر : ارسسطو (با همکاری سامانه اطلاع‌رسانی چاپ و نشر ایران)

صفحه‌آرایی ، تنظیم و طرح جلد : علی بیات

تیراژ : ۱۰۰۰ جلد

نوبت چاپ : دوم - ۱۳۹۸

تعداد صفحات: ۳۰۴ ص

چاپ : مدیران

قیمت : ۶۰۰۰ تومان

تلفن‌های مرکز پخش : ۰۹۱۷۷۱۶۴۹۴۰ - ۰۵۱۱ - ۰۵۰۹۶۱۴۶

این اثر مشمول قانون حمایت از مولفان و مصنفان و هنرمندان است. هر کس تمام یا قسمتی از این اثر را بدون اجازه مولف نشر یا پخش یا عرضه کند، مورد پیگرد قانونی قرار خواهد گرفت.

فهرست مطالب

فصل اول: اصول شیء‌گرایی ۱۹

۱۹	۱-۱-مقدمه
۱۹	۱-۲-مفاهیم اساسی
۲۰	۱-۳-نمایش کلاس در UML
۲۰	۱-۴-mekanizm arتباط بین اشیاء
۲۰	۱-۵-اصل شیء‌گرایی
۲۱	۱-۵-۱-تجزید یا انتزاع
۲۲	۱-۵-۱-۱-نقش تجزید در کنترل پیچیدگی
۲۲	۱-۵-۲-محصورسازی یا پنهانسازی جزئیات یا کپسوله سازی
۲۲	۱-۵-۲-۱-نقش محصورسازی در کنترل پیچیدگی
۲۳	۱-۵-۳- واحد بندی
۲۳	۱-۳-۵-۱-ویژگی‌های واحد بندی
۲۴	۱-۳-۵-۲-اصل مازولاتریتی
۲۵	۱-۳-۵-۳-نقش واحد بندی و تکنیک تجزیه در کنترل پیچیدگی
۲۵	۱-۳-۵-۴-مزایای طراحی پیمانه‌ای کارآمد
۲۵	۱-۴-۵-۴-سلسله‌مراتب
۲۶	۱-۴-۵-۱-نقش سلسله‌مراتب در کنترل پیچیدگی
۲۷	۱-۶-مزایای مدل شیء
۲۷	۱-۷-سؤالات تشریحی
۲۸	۱-۸-سؤالات چهارگزینه‌ای
۲۹	پاسخنامه:

فصل دوم: جایگاه UML و مدلسازی تصویری در توسعه نرم افزار..... ۳۱

۳۱	۱-۲- مقدمه
۳۲	۲-۲- مدل افزایشی (تکرار و توسعه تدریجی)
۳۳	۲-۳- مدل سازی تصویری نرم افزار
۳۴	۱-۳-۲- جایگاه مدلسازی تصویری در چرخه توسعه نرم افزار
۳۴	۴-۲- UML چیست؟
۳۵	۱-۴-۲- دیدهای مختلف UML
۳۶	۲-۴-۲- نمودارهای UML
۳۷	۵-۲- سوالات تشریحی
۳۷	۶-۲- سوالات چهارگزینه‌ای
۳۸	پاسخنامه:

فصل سوم: روش‌های یافتن و نمایش کلاس‌ها ۳۹

۳۹	۱-۳- مراحل یافتن و شناسایی کلاس‌ها
۴۱	۲-۳- روش‌های شناسایی کلاس‌ها
۴۳	۱-۲-۳- نمایش وراثت در کارت‌های CRC
۴۴	۲-۲-۳- فرایند مدلسازی به وسیله کارت‌های CRC
۴۵	۳-۳- انواع رابطه بین کلاس
۴۶	۱-۳-۳- رابطه انجمنی
۴۷	۲-۳-۳- سلسله‌مراتب Part-of یا رابطه تجمعی
۴۷	۳-۳-۳- رابطه عام/خاص یا رابطه وراثت
۴۹	۴-۳- سوالات تشریحی
۴۹	۵-۳- سوالات چهارگزینه‌ای

۵۰ پاسخنامه
۵۱ فصل چهارم: مدل سازی موارد کاربری
۵۱ ۱-۴- مفاهیم اساسی مدل مورد کاربری
۵۴ ۲-۴- سازمان دهی موارد کاربری
۵۴ ۲-۱- بسته بندی
۵۵ ۲-۲- رابطه عام / خاص (وراثت)
۵۵ ۲-۳- رابطه دربرداشتن
۵۶ ۲-۴- رابطه گسترش دادن
۵۶ ۴-۱- کاربردهای رابطه گسترش دادن
۵۶ ۴-۳- نحوه ایجاد مدل موارد کاربری
۵۸ ۴-۴- سوالات تشریحی
۵۸ ۴-۵- سوالات چهار گزینه‌ای
۵۹ پاسخنامه:
۶۱ فصل پنجم: مدل سازی کلاس‌ها
۶۱ ۱-۵- مقدمه
۶۱ ۲-۵- ایجاد نمودار کلاس
۶۲ ۲-۵- مدل تحلیل
۶۳ ۲-۵- مدل طراحی
۷۳ ۳-۵- مفهوم کلیشه در UML
۷۴ ۴-۵- سوالات تشریحی
۷۵ ۵-۵- سوالات چهار گزینه‌ای فصل ششم
۷۷ فصل ششم: مدل سازی تعامل و رفتار

۱-۶-ساختار پویای سیستم	۷۷
۲-۶-نمودار ترتیبی	۷۷
۳-۶-نمودار حالت	۷۹
۴-۶-نمودار فعالیت	۸۱
۵-۶-تحقيق و پژوهش	۸۲
۶-۶-سؤالات تشریحی	۸۵
۷-۶-سؤالات چهارگزینه‌ای	۸۷
پاسخنامه:	۹۰
فصل هفتم: مدل‌سازی مؤلفه و استقرار	۹۱
۱-۷-بسته‌ها	۹۱
۲-۷-واردکردن بسته‌ها	۹۱
۳-۷-قابلیت دستیابی بسته	۹۲
۴-۷-رابطه وراثت دربسته‌ها	۹۲
۵-۷-رابطه وابستگی بین بسته‌ها	۹۲
۶-۷-مدل‌سازی مؤلفه‌ها	۹۳
۷-۷-تفاوت‌های شیءگرایی و توسعه بر مبنای مؤلفه	۹۴
۸-۷-ویژگی‌های مشترک مؤلفه و شیء	۹۵
۹-۷-تفاوت‌های بین مؤلفه و کلاس	۹۵

۹۶	۳-۷- نقش واسط در مؤلفه‌ها
۹۷	۴-۷- جایگزین پذیری مؤلفه‌ها
۹۸	۵-۷- انواع مؤلفه‌های در UML
۹۹	۶-۷- مدل‌سازی استقرار
۱۰۱	۷-۷- سوالات تشریحی
۱۰۱	۸-۷- سوالات چهارگزینه‌ای
۱۰۲	: پاسخنامه

فصل هشتم: زیر سیستم‌ها

۱۰۳	۱-۸- زیرسیستم
۱۰۴	۲-۸- چه وقت زیرسیستم‌ها استفاده می‌شوند؟
۱۰۴	۳-۸- تکنیک‌های SPECIFICATION
۱۰۴	۱-۳-۸- رهیافت موارد کاربری
۱۰۵	۲-۳-۸- رهیافت ماشین حالت
۱۰۶	۳-۳-۸- رهیافت کلاس منطقی
۱۰۶	۴-۳-۸- روش Operation
۱۰۷	۴-۸- SPECIFICATION – REALIZATION
۱۰۷	۱-۴-۸- رابطه Realization
۱۰۸	۲-۴-۸- همکاری
۱۰۹	۱-۲-۴-۸- نمادگذاری Collaboration
۱۱۰	۵-۸- واسط زیرسیستم‌ها
۱۱۰	۱-۵-۸- واسط‌ها و اپراتورها
۱۱۱	۲-۵-۸- تعامل بین زیرسیستم‌ها

۱۱۱	۶-۸-وراثت زیرسیستم‌ها
۱۱۲	۷-۸-سؤالات تشریحی
۱۱۲	۸-۸-سؤالات چهارگزینه‌ای
۱۱۵	فصل نهم: مفاهیم طراحی
۱۱۵	۱-۹-طراحی در حیطه مهندسی نرم‌افزار
۱۱۵	۲-۹- انواع طراحی
۱۱۶	۳-۹-فرایند طراحی
۱۱۶	۴-۹-۱-دستورالعمل‌ها و صفت کیفیت نرم‌افزار
۱۱۷	۴-۹-۱-۱-صفات کیفیتی
۱۱۸	۴-۹-۲-تکامل طراحی نرم‌افزار
۱۱۸	۴-۹-۴-۴-۹-مفاهیم طراحی
۱۱۹	۴-۹-۱-۴-۹-انتزاع یا تجرید (Abstraction)
۱۱۹	۴-۹-۲-۴-۹-معماری (Architecture)
۱۲۰	۴-۹-۳-۴-۹-(patterns) الگوهای
۱۲۰	۴-۹-۴-۹-جداسازی دغدغه‌ها (Separation of concerns)
۱۲۰	۴-۹-۵-۴-۹-پیمانه بندی (Modularity)
۱۲۱	۴-۹-۶-۴-۹-پنهان‌سازی اطلاعات (Information Hiding)
۱۲۱	۴-۹-۷-۴-۹-استقلال عملیاتی (Functional Independence)
۱۲۲	۴-۹-۸-۴-۹-پالایش (Refinement)
۱۲۲	۴-۹-۹-۴-۹-جنبهای (Aspects)
۱۲۲	۴-۹-۱۰-۴-۹-بازآرایی (Refactoring)
۱۲۳	۴-۹-۱۱-۴-۹-کلاس‌های طراحی (Design classes)

۱۲۴	۵-۹- مدل طراحی (DESIGN MODEL)
۱۲۴	۹-۹- عناصر طراحی داده‌ها
۱۲۵	۹-۹- عناصر طراحی معماری
۱۲۵	۹-۹- عنصر طراحی واسطه‌ها
۱۲۶	۹-۹- عناصر طراحی در سطح استقرار
۱۲۶	۹-۹- سؤالات تشریحی
۱۲۷	۹-۹- سؤالات چهارگزینه‌ای
۱۲۷	پاسخنامه:
۱۲۹	فصل دهم: طراحی معماری
۱۲۹	۱۰-۱- مقدمه
۱۲۹	۱۰-۱-۱- معماری نرم افزار
۱۳۰	۱۰-۱-۲-۱- اهمیت معماری در چیست؟
۱۳۰	۱۰-۱-۲-۱- توصیف‌های معماری
۱۳۰	۱۰-۱-۲-۲- تصمیم‌گیری‌های معماری
۱۳۰	۱۰-۱-۳- سبک‌های معماری
۱۳۲	۱۰-۱-۳-۱- معرفی برخی از سبک‌های معماری
۱۳۲	۱۰-۱-۳-۱-۱- معماری‌های داده محور
۱۳۲	۱۰-۱-۳-۱-۲- سبک شیء‌گرا
۱۳۲	۱۰-۱-۳-۱-۳- معماری لایه‌ای
۱۳۴	۱۰-۱-۳-۱-۴- معماری مدل - نما - کنترل گر (MVC) برای برنامه‌های کاربردی تحت وب
۱۳۵	۱۰-۱-۴- تاکتیک‌ها، راهبردها و الگوهای معماری
۱۳۵	۱۰-۵- طراحی معماری

۱۳۶	۱-۵-۱۰-نمایش سیستم در حیطه کاری
۱۳۶	۲-۵-۱۰-تعریف نمونه‌های اولیه
۱۳۶	۳-۵-۱۰-پالایش معماری به مؤلفه‌ها
۱۳۶	۴-۶-۱۰-روش تحلیل توازن‌های معماری (ATAM)
۱۳۷	۵-۶-۱۰-خروجی‌های ATAM
۱۳۸	۶-۱۰-پیچیدگی معماری
۱۳۹	۷-۱۰-زبان‌های توصیف معماری
۱۳۹	۸-۱۰-پالایش طراحی معماری
۱۳۹	۹-۱۰-سؤالات تشریحی
۱۳۹	۱۰-۱۰-سؤالات چهارگزینه‌ای
۱۴۰	پاسخنامه:
۱۴۱	فصل یازدهم: طراحی در سطح مولفه ها
۱۴۱	۱-۱۱-مقدمه
۱۴۱	۲-۱۱-مؤلفه
۱۴۳	۳-۱۱-طراحی مولفه ها مبتنی بر کلاس
۱۴۳	۳-۱۱-۱-اصول پایه طراحی
۱۴۴	۳-۱۱-۲-دستورالعمل های طراحی در سطح مولفه
۱۴۵	۳-۱۱-۳-یکپارچگی
۱۴۵	۴-۱۱-۴-اتصال
۱۴۶	۴-۱۱-۴-اجرای طراحی در سطح مولفه ها
۱۴۹	۵-۱۱-۵-طراحی در سطح مولفه برای برنامه‌های تحت وب
۱۴۹	۵-۱۱-۱-طراحی عملیاتی در سطح مولفه ها

۱۵۰	۶-۶- طراحی مؤلفه های سنتی
۱۵۰	۷-۷- توسعه مبتنی بر مؤلفه ها
۱۵۰	۷-۸- مهندسی دامنه
۱۵۱	۷-۹- صلاحیت تطبیق و ترکیب
۱۵۲	۷-۱۰- تحلیل و طراحی برای استفاده مجدد
۱۵۳	۷-۱۱- طبقه بندی و بازیابی مؤلفه ها
۱۵۳	۸-۱- سوالات تشریحی
۱۵۴	۹-۱- سوالات چهار گزینه ای
۱۵۴	پاسخنامه:

۱۵۵	فصل دوازدهم: طراحی واسط کاربر
۱۵۵	۱۲-۱- طراحی رابط کاربر چیست؟
۱۵۵	۱۲-۲- قواعد طلایی MENDAL
۱۵۷	۱۲-۳- مدلهای تحلیل و طراحی واسط
۱۵۸	۱۲-۴- فرآیند تحلیل و طراحی واسط کاربر
۱۵۸	۱۲-۵- طراحی واسط
۱۵۹	۱۲-۶- اعتبار سنجی واسط
۱۵۹	۱۲-۷- تحلیل واسط
۱۵۹	۱۲-۷-۱- تحلیل کاربران
۱۵۹	۱۲-۷-۲- مدلسازی و تحلیل وظایف
۱۶۰	۱۲-۷-۲-۱- پرداختن به جزئیات وظایف
۱۶۰	۱۲-۷-۲-۲- پرداختن به جزئیات اشیاء
۱۶۰	۱۲-۷-۲-۳- تحلیل جریان کاری

۱۶۲	۱۲-۷-۳	- تحلیل محتوای صفحه نمایش
۱۶۲	۱۲-۷-۴	- تحلیل محیط کار
۱۶۲	۱۲-۸	- مراحل طراحی واسط
۱۶۳	۱۲-۸-۱	- به کارگیری مراحل طراحی
۱۶۳	۱۲-۸-۲	- مسائل طراحی
۱۶۴	۱۲-۹	- طراحی واسط برنامه تحت وب
۱۶۴	۱۲-۹-۱	- دستورالعمل و اصول طراحی واسط
۱۶۶	۱۲-۹-۲	- جریان کاری مقدماتی برای طراحی واسط برنامه‌ی تحت وب
۱۶۷	۱۲-۱۰	- ارزیابی طراحی
۱۶۷	۱۲-۱۰-۱	- چند ملاک برای ارزیابی
۱۶۸	۱۲-۱۱	- سوالات تشریحی
۱۶۸	۱۲-۱۲	- سوالات چهار گزینه‌ای
۱۷۰		پاسخنامه:
۱۷۱	فصل سیزدهم: الگوهای طراحی شیء‌گرا	
۱۷۱	۱-۱۳	- مقدمه
۱۷۱	۲-۱۳	- عوامل ایجاد الگوهای طراحی
۱۷۳	۲-۱۳	- تعریف الگو
۱۷۳	۴-۱۳	- تاریخچه الگوی طراحی
۱۷۴	۵-۱۳	- طبقه‌بندی الگوها
۱۷۵	۶-۱۳	- ساختار الگوی طراحی
۱۷۷	۷-۱۳	- بررسی چند الگوی طراحی
۱۷۸	۷-۱۳	۱-الگوی طراحی Abstract Factory

۱۸۱	۱۳-۷-۲-الگوی طراحی bridge
۱۸۴	۱۳-۷-۳-الگوی طراحی composite
۱۸۷	۱۳-۷-۴-الگوی طراحی decorator
۱۹۱	۱۳-۷-۵-الگوی طراحی Mediator
۱۹۵	۱۳-۷-۶-الگوی طراحی Observer
۱۹۷	۱۳-۸-ضد الگوها
۲۰۱	۱۳-۹-سوالات تشریحی
۲۰۱	۱۳-۱۰-سوالات چهارگزینهای
۲۰۲	: پاسخنامه
۲۰۳	فصل چهاردهم: معرفی ویژوال پارادایم
۲۰۳	۱۴-۱-مقدمه
۲۰۴	۱۴-۲-نصب نرم افزار
۲۰۴	۱۴-۳-نمای کلی رابط کاربر
۲۰۴	۱۴-۴-باز کردن / بستن قابها
۲۰۷	فصل پانزدهم: شروع کار با ویژوال پارادایم
۲۰۷	۱۵-۱-ایجاد پروژه جدید
۲۰۸	۱۵-۲-ذخیره سازی پروژه
۲۰۹	۱۵-۳-ارسال و دریافت فایلها
۲۰۹	۱۵-۴-تنظیمات فونت و رنگ
۲۱۱	۱۵-۵-تنظیم جعبه ابزار DIAGRAMS
۲۱۳	۱۵-۶-سوئیچ کردن بین دیاگرامهای مختلف یک پروژه
۲۱۷	فصل شانزدهم: مدل سازی موارد کاربری با ویژوال پارادایم

۱۶-۱-رسم نمودار موارد کاربری	۲۱۷
۱۶-۱-۱-روش اول	۲۱۷
۱۶-۱-۲-روش دوم	۲۱۹
۱۶-۲-مورد کاربری حرفه	۲۲۰
فصل هفدهم: طراحی نمودار کلاس با ویژوال پارادایم	۲۲۳
۱۷-۱-ایجاد نمودار کلاس	۲۲۳
۱۷-۱-۱-نحوه ساخت کلاس	۲۲۳
۱۷-۱-۲-ساخت رابطه همکاری	۲۲۴
۱۷-۱-۳-تجمیع	۲۲۶
۱۷-۱-۴-تعیین نوع رابطه بین کلاس‌ها	۲۲۶
۱۷-۱-۵-ایجاد رابطه وراثت	۲۲۹
۱۷-۱-۶-رابطه وابستگی بین دو کلاس	۲۳۰
۱۷-۱-۷-ایجاد صفت کلاس	۲۳۰
۱۷-۱-۸-ایجاد عملکردهای کلاس	۲۳۱
۱۷-۱-۹-مرتب‌سازی اعضای کلاس	۲۳۲
۱۷-۱-۱۰-کپی کردن یک عضو کلاس	۲۳۳
۱۷-۱-۱۱-انتقال یک عضو کلاس	۲۳۳
۱۷-۱-۱۲-تعیین رابطه وابستگی بین اعضای کلاس	۲۳۴
۱۷-۱-۱۳-رابطه use	۲۳۶
۱۷-۱-۱۴-مجموعه تعیین‌یافته	۲۳۶
۱۷-۲-ترسیم کلاس‌های تحلیلی	۲۳۸
فصل هجدهم: طراحی نمودار فعالیت با ویژوال پارادایم	۲۴۳

۲۴۳	۱-۱-۱۸- ترسیم نمودار فعالیت
۲۴۴	۱-۱-۱۸- ایجاد Swimlane
۲۴۵	۲-۱-۱۸- الحق پارتیشن به Swimlane
۲۴۵	۱-۱-۱۸- ایجاد گرههای اولیه
۲۴۶	۱-۱-۱۸- ایجاد عملیات (کنش)
۲۴۸	۱-۱-۱۸- تفاوت Action و Activity
۲۴۸	۲-۱-۱۸- تمرین

فصل نوزدهم: جستجوگر مدل در ویژوال پارادایم

۲۵۱	۱-۱-۱۹- جستجوگر مدل (MODEL EXPLORER)
۲۵۱	۲-۱-۱۹- بستن و باز کردن MODEL EXPLORER
۲۵۲	۳-۱-۱۹- ایجاد یک مدل
۲۵۲	۴-۱-۱۹- ساخت یک عنصر مدل
۲۵۶	۱-۴-۱۹- مرتب‌سازی عناصر مدل
۲۵۷	۵-۱-۱۹- ساخت نمودار در مدل

فصل بیستم: ترسیم نمودار توالی با ویژوال پارادایم

۲۵۹	۱-۲۰- توسعه نمودار توالی
۲۶۰	۱-۱-۲۰- ترسیم پیام بین Lifeline های موجود
۲۶۱	۲-۲۰- ارتباط بین نمودار کلاس و نمودار توالی
۲۶۳	۱-۲-۲۰- استفاده از کلاس‌ها در ساختن LifeLine ها

فصل بیست و یکم: مدیریت زیر نمودارها در ویژوال پارادایم

۲۶۷	۱-۲۱- افزودن زیر نمودار به نمودار موارد کاربری
۲۶۸	۲-۲۱- حذف زیر نمودار

۲۶۹	۳-۲۱- باز کردن یک زیر نمودار.....
۲۷۱	فصل بیست و دوم: نمودار بسته در ویژوال پارادایم
۲۷۱	۱-۲۲- ایجاد نمودار بسته.....
۲۷۲	۲-۲۲- تعیین کلیشه.....
۲۷۵	فصل بیست و سوم: ایجاد نمودار مؤلفه در ویژوال پارادایم
۲۷۵	۱-۲۲- ایجاد نمودار مؤلفه.....
۲۷۶	۲-۲۳- تعیین کلیشه برای مؤلفه.....
۲۷۶	۳-۲۳- تعریف واسطها.....
۲۷۷	۱-۳-۲۳- تعریف واسط ارائه.....
۲۷۷	۲-۳-۲۳- تعریف واسط موردنیاز.....
۲۷۸	۴-۲۳- ایجاد وابستگی بین مؤلفهها.....
۲۷۹	فصل بیست و چهارم: نمودار استقرار در ویژوال پارادایم
۲۷۹	۱-۲۴- ایجاد نمودار استقرار.....
۲۸۰	۱-۱-۲۴- ایجاد گرهها و ترسیم ارتباطات بین گرهها.....
۲۸۲	۱-۲-۲۴- ایجاد عنصر مدل گره.....
۲۸۳	۲-۲۴- ایجاد یک نمونه توصیف از گره.....
۲۸۴	۳-۲۴- انتخاب طبقه بنده یا رده.....
۲۸۵	۴-۲۴- ایجاد پیوند.....
۲۸۶	۵-۲۴- ایجاد نمونه از مؤلفه.....
۲۸۷	۶-۲۴- ایجاد وابستگی.....
۲۸۷	۷-۲۴- تمرین.....
۲۸۹	فصل بیست و پنجم: نمودار ماشین حالت در ویژوال پارادایم

۲۸۹	۱-۱-ایجاد نمودار حالت.....۲۵
۲۹۰	۱-۱-۱-ایجاد حالتها و انتقالها۲۵
۲۹۱	۱-۱-۲-اصافه کردن تابعیه به حالت۲۵
۲۹۲	۱-۳-۱-مدل سازی خصوصیات انتقال۲۵
۲۹۴	۲-نمایش محتوای حالت در نمودار حالت۲۵
۲۹۹	فصل بیست و ششم: ترسیم زیرسیستم در ویژوال پارادایم۲۶
۳۰۳	۱-ترسیم زیرسیستم۲۶
۳۰۴	۲-تمرین۲۶
	منابع:

مقدمه:

از بدو پیدایش نرم افزار تاکنون، تحويل به موقع نرم افزار، کاهش هزینه ها، مدیریت پیکربندی و اعمال تغییرات و تولید یک محصول باکیفیت و رضایت مشتری همواره از دغدغه های توسعه دهنده کان سیستم های نرم افزاری بوده است.

عوامل زیادی در موفقیت یا شکست نرم افزار دخیل هستند. علی از این عوامل طراحی نرم افزار است. حتی با در اختیار داشتن بهترین برنامه نویس ها، اگر نرم افزار طراحی مناسبی نداشته باشد، سرانجام خوبی در انتظار پروژه نخواهد بود. در این کتاب سعی شده است که مباحث تحلیل و طراحی شیء گرا با زبانی ساده و به صورت کاربردی بیان شود. همچنین آموزش نرم افزار ویژوال پارادایم که یک نرم افزار کامل برای مدلسازی پروژه های نرم افزاری است ارائه شده است.

امید است این اثر مورد توجه همکاران و دانشجویان گرامی قرار گیرد. از اساتید و دانشجویان گرامی تقاضا دارم دیدگاه های خود را از طریق ایمیل m.a.torkamani@gmail.com با اینجانب در میان بگذارند تا انشالله در ویرایش های بعدی اشکالات یا کاستی های احتمالی کتاب مورد تجدیدنظر قرار گیرد. در پایان وظیفه خود می دانم از خدمات آقای مهندس علی بیات به خاطر طراحی جلد کتاب و همچنین از مدیریت انتشارات ارسسطو و سامانه اطلاع رسانی چاپ و نشر ایران، جناب آقای حسین قبری، تشکر و قدردانی نمایم.

محمدعلی ترکمانی

تابستان ۱۳۹۵

فصل اول

اصول شیء‌گرایی

۱-۱- مقدمه

متخصصین مهندسی نرم‌افزار سعی نموده‌اند با ارائه رهیافت‌های مختلف بر مشکلات توسعه نرم‌افزار غلبه کنند. یکی از این رهیافت‌ها شیء‌گرایی است.

۱-۲- مفاهیم اساسی

۱-شیء^۱: یک مفهوم کلی است به گونه‌ای که دارای هویت بوده و قادر به بروز رفتار و ثبت حالات (وضعیت) خود باشد.

- هویت^۲: آن ویژگی از یک شیء است که آن را از بقیه اشیاء متمایز می‌سازد.
- حالت^۳: حالت یک شیء دربردارنده تمام خواص آن شیء و مقادیر جاری آن‌ها است.
- رفتار^۴: چگونگی عمل و عکس‌العمل یک شیء در مقابل دریافت و یا ارسال پیام.مثال: در سیستم کتابخانه، نام کتاب یک شیء است.
- هویت: کتاب
- حالت: اطلاعات فهرست‌نویسی، مکان نگهداری فیزیکی، وضعیت فعلی (امانت/رزرو/آزاد).
- رفتار: ثبت اطلاعات کتاب، جستجو، سفارش دادن برای خرید، ...

1 Object

2 Identity

3 State

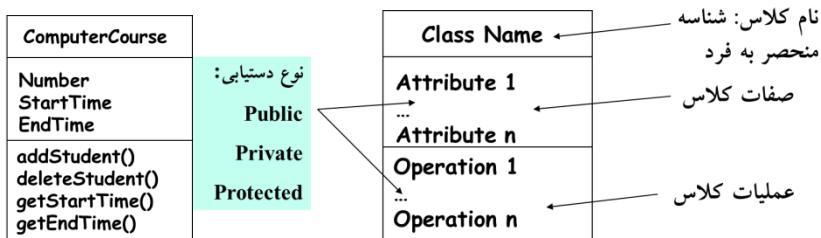
4 Behavior

۲-کلاس^۱: کلاس ابزار اصلی تجزیه در شیء‌گرایی است. کلاس به صورت قالبی که بر حسب نیاز نمونه‌هایی از آن برداشت می‌نماییم، عمل می‌کند.

۱-۳-نمایش کلاس در UML

مطابق با شکل ۱-۱ یک کلاس از دو قسمت تشکیل شده است:

- صفات:^۲ ویژگی‌هایی از کلاس که نگهداری آن برای ما مهم است.
- عملیات:^۳ رفتار کلاس در مقابل کلاس‌های دیگر



شکل ۱-۱: نمایش کلاس در UML

۱-۴-مکانیزم ارتباط بین اشیاء

ارتباط بین اشیاء و بهره‌مندی از سرویس‌های (عملیات) آن‌ها از طریق مکانیزم تبادل پیام (Message Passing) و فراخوانی متدها صورت می‌گیرد.

۱-۵-اصول شیء‌گرایی

مدل شیء‌گرا بر چهار اصل تجزیه،^۴ پنهان‌سازی جزئیات یا محصورسازی،^۵ واحد بندی^۶ و

1 Class
2 Instances
3 Attributes
4 Operations
5 Abstraction
6 Encapsulation
7 Modularity

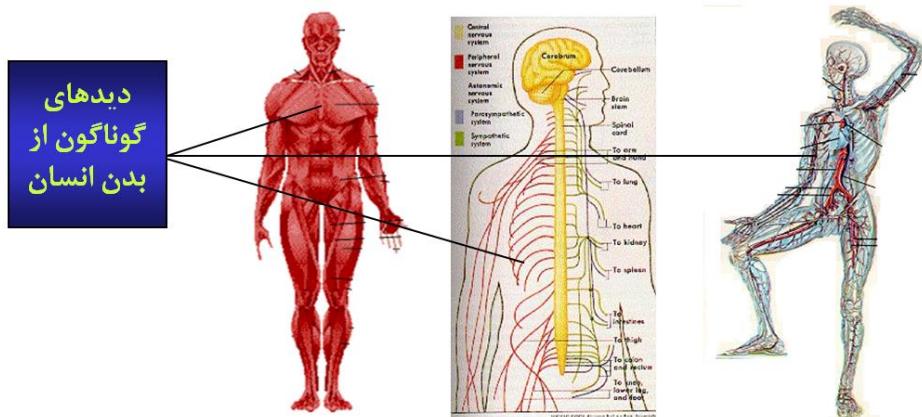
سلسله‌مراتب استوار است که در ادامه شرح داده می‌شود.

۱-۵-۱-تجزیه یا انتزاع

تجزیه عبارت است از فرایند متمرکز شدن روی ویژگی‌ها و رفتارهای اصلی یک پدیده، و نادیده گرفتن ویژگی‌های موقت و غیر مهم آن پدیده، از یک زاویه دید مشخص.

مثال‌هایی از تجزیه:

۱- دیدهای گوناگون از بدن انسان مانند سیستم گردش خون، استخوان‌بندی (شکل ۱-۱).



شکل ۱-۲: دیدهای گوناگون از بدن انسان

۲- نقشه‌های مختلف یک ساختمان (تأسیسات، برق، نقشه/سکلت و...)

۳- استفاده از نماد ACTOR (شکل ۱-۳) برای نمایش حضور موجودیت انسان در یک صحنه



شکل ۱-۳: نماد ACTOR

۴- نمایش گرافیکی رفتار یک سیستم

۱-۵-۱- نقش تجزیه در کنترل پیچیدگی

یکی از ابزارهای اصلی کنترل و تسلط بر پیچیدگی تجزیه است. به طور کلی جزئیات بی‌شماری که درباره یک پدیده مطرح است. به وسیله تجزیه تنها ابعاد اساسی پدیده مدنظر خواهد شد.

۱-۵-۲- محصورسازی یا پنهان‌سازی جزئیات یا کپسوله سازی

محصورسازی عبارت است از عدم پذیرش تأثیرات ناخواسته^۱ و یا کنترل نشده و محدود کردن طرق دسترسی به شیء یا استفاده از یک شیء. با توجه به این اصل هر شیء یا کلاس از دو قسمت زیر تشکیل می‌گردد:

۱- واسطه^۲: تو صیفی از سرویس‌هایی که یک شیء در اختیار مشتریان قرار می‌دهد. به طور کلی سه نوع واسطه وجود دارد:

- عمومی: برای همه قابل دسترس است.

- اختصاصی: برای خود کلاس و دوستان آن قابل دسترسی است.

- حفاظت‌شده: برای خود کلاس دوستان و زیر کلاس‌های آن قابل دسترسی است.

۲- پیاده‌سازی^۳ یا ساختار داخلی: شامل داده‌ها + اعمال رویداده‌ها.

مزایای محصورسازی عبارت است از:

- جدایی ساختار داخلی از نحوه استفاده از کلاس

- استقلال پیاده‌سازی

- پدیده انتشار تغییرات نخواهیم داشت

- نگهداری آسان تر

۱-۵-۲- نقش محصورسازی در کنترل پیچیدگی

1 Side Effects

2 Interface

3 Public

4 Private

5 Protected

6 Implementation

- کنترل و تسلط بر پیچیدگی به و سیله کنترل راههای دسترسی به یک شیء که باعث جلوگیری از خرابکاری‌های احتمالی و محلی کردن گستره خطاهای در خود شیء می‌شود.
 - چون از واسط استفاده می‌شود، می‌توان هر تغییری در پیاده‌سازی آن شیء انجام گیرد، بدون اینکه نیازی به تغییر در عنصر استفاده‌کننده باشد (مگر اینکه واسط تغییر کند)
 - مفهوم یک واسط برای چند پیاده‌سازی، امکان استفاده مجدد را بالا می‌برد
- نکته: تجربید مکانیزم تعیین جزئیاتی که باید پنهان شود، است. اما مخصوص‌سازی، فرایند پنهان‌سازی جزئیات و کنترل دسترسی به آن خواهد بود.

۱-۵-۳- واحد بندی^۱

سیستمی را واحد بندی شده می‌گویند که به مجموعه‌ای از مازول‌های (واحدهای) منسجم و معنی‌دار که وابستگی بین آن‌ها حداقل است، تجزیه شده باشد. یک مازول، واحد تشکیل‌دهنده ساختار فیزیکی سیستم نرم‌افزاری (شبیه مدارات مجتمع در سخت افزار) است. مثالی از واحدها:

- فایل‌ها در C++ و C
- واحدها در Object Pascal
- مولفه‌ها^۲ در استانداردهای COM ، .NET و Java Beans
- در جاوا Package

۱-۵-۳- ویژگی‌های واحد بندی

- ۱-اگر شرایط بیان شده در تعریف واحد بندی رعایت گردد آنگاه مازول‌های به دست آمده قابلیت استفاده مجدد بالایی خواهد داشت
- ۲-تعداد زیر مسئله‌ها نباید زیاد یا کم باشد
- ۳-تعیین معیار شکستن یک مسئله مهم‌ترین عامل برای موفقیت استفاده از این ویژگی
- ۴-واحدها باید ویژگی‌های Building Blocks را داشته باشند یعنی:

1 Modularity

2 Units

3 Components

- استقلال^۱
- واسطه‌های خوش تعریف:^۲ یعنی رابط خوب تعریف و مستند شده باشد تا درک و نگهداری آن ساده باشد.

۱-۵-۳-۲-اصل مازولاریتی

قبل از بیان این اصل ابتدا به دو مفهوم زیر توجه کنید

-انسجام:^۳ عبارت است از درجه ارتباط عملکردهای عناصر داخلی یک مازول.

-وابستگی:^۴ عبارت است از درجه ارتباط مازولهای گوناگون با یکدیگر.

از سجام و وابستگی دو معیار اصلی در طراحی سیستم می‌باشد که اصل مهمی در مهندسی نرم‌افزار موسوم به اصل مازولاریتی^۵ بر اساس آن‌ها تعریف می‌شود. این اصل چنین است: کوپلاژ یا وابستگی کم و چسبندگی یا انسجام زیاد.^۶

برای رسیدن به چسبندگی یا انسجام زیاد باید بخش‌های غیر مرتبط از سیستم از هم جدا شوند و قطعات مربوط به سیستم باید باهم گروه‌بندی شوند. به عنوان مثال اجزای مرتبط باید در کنار هم قرار گیرند. مثلاً کدهای مربوط به هم را درون یک کلاس و کلاس‌های مرتبط به هم را درون یک فایل قرار داد و برنامه‌هایی که باهم ارتباط کمی دارند را در فایل‌های جدا قرار داد. ضمناً برای دستیابی به کوپلاژ یا وابستگی کم حتی‌الامکان باید از داده‌های public یا متغیرهای عمومی استفاده نکنید. با توجه به این اصل، می‌توان واحد بندی را به صورت زیر تعریف نمود:

سیستمی را واحد بندی شده گویند که به مجموعه‌ای از واحدهای منسجم و معنی‌دار که وابستگی بین آن‌ها حداقل است، تجزیه شده باشد.

1 Independent

2 Well-defined Interfaces

3 Cohesion

4 Coupling

5 principle of Modularity

6 Low coupling

7 High Cohesion

۱-۵-۳- نقش واحد بندی و تکنیک تجزیه ادر کنترل پیچیدگی

پیچیدگی سیستم‌های نرم‌افزاری روزبه روز در حال افزایش است و قدرت پردازش همزمان مغز انسان نیز محدود است. راه حل این مشکل استفاده از تکنیک تجزیه است. در واقع تکنیک کنترل و سلطه بر پیچیدگی همان اصل تفرقه بیندار و حکومت کن آست. برای استفاده از این اصل سیستم را به اجزای کوچک‌تر تقسیم می‌کنیم. سپس هر کدام از آن‌ها را به تنهایی حل می‌کنیم. به این ترتیب ذهن انسان برای درک هر سطح از سیستم مجبور نخواهد بود همه اجزاء سیستم را در نظر بگیرد بلکه با تعداد محدودی از اجزاء سرو کار خواهد داشت. بنابراین شکستن مسئله به اجزائی کوچک‌تر یکی از راه‌های کارا برای مقابله با پیچیدگی است.

۱-۵-۴- مزایای طراحی پیمانه‌ای کارآمد

- کاهش پیچیدگی
- تسهیل تغییرات
- امکان توسعه موازی مؤلفه‌های مختلف و درنتیجه تسهیل پیاده‌سازی

۱-۵-۴- سلسله‌مراتب^۲

سلسله‌مراتب عبارت است از مرتب ساختن تجریدها در سطوح مختلف. مطابق شکل ۴-۴ انواع سلسله‌مراتب عبارتند از:

- ۱- سلسله‌مراتب IS-A یا رابطه عام خاص یا رابطه وراثت (Inheritance). روابط وراثتی کلاس پدر، کلاس و کلاس فرزند در شکل ۵-۴ نمایش داده شده است.
 - ۲- سلسله‌مراتب PART-OF یا Has-A یا رابطه تجمعی.
- نکته: سطوح تجرید متفاوت در سطوح مختلف سلسله‌مراتب نمایان می‌گردد (شکل ۶-۴).

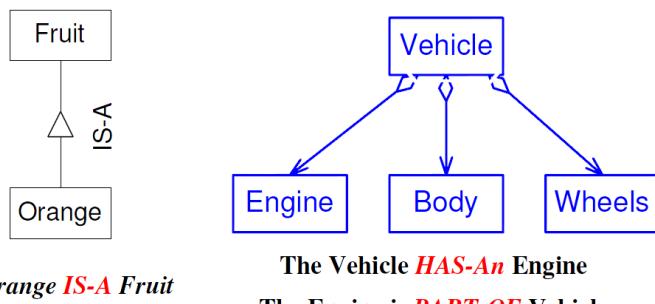
1 Decomposable

2 Divide and Rule

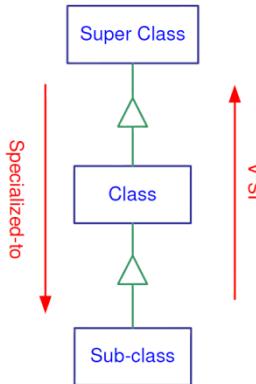
3 Hierarchy

۱-۴-۵-۱- نقش سلسله‌مراتب در کنترل پیچیدگی

- با سازمان‌دهی تجزیدها در سلسله‌مراتب PART-OF و IS-A درک ما نسبت به سیستم افزایش می‌یابد
- اهمیت سلسله‌مراتب PART-OF: روابط موجود بین اشیاء و فعل و انفعالاتی که رخداد را نمایان می‌سازد.
- اهمیت سلسله‌مراتب IS-A: افزونگی موجود در سیستم را مدیریت می‌نماید.
- استفاده از وراثت با محصورسازی تمام^۱ (محصورسازی تمام) تعارض دارد. زیرا مستلزم دسترسی مستقیم کلاس فرزند به بعضی از اعمال و داده‌های اختصاصی کلاس پدر است.

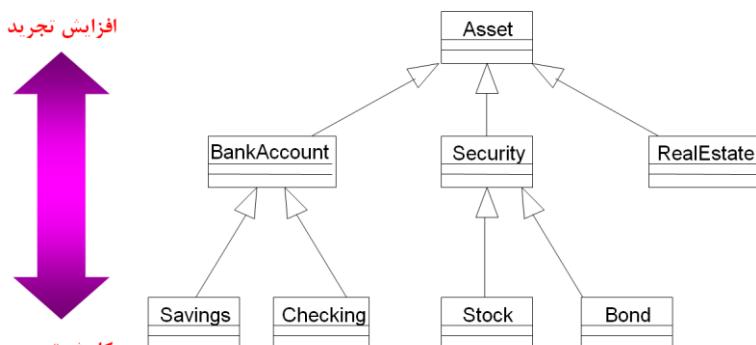


شکل ۴-۴: مثالی از سلسله‌مراتب IS-A و سلسله‌مراتب PART-OF



شکل ۴-۵: روابط وراثتی کلاس پدر، کلاس، و کلاس فرزند

1 entire encapsulation



شکل ۴-۶: سطح تجرید

۱-۶-مزایای مدل شیء

- ۱-هدف نهائی تکنولوژی OO انجام فرایند تولید نرم‌افزار به صورت مشابه فرایند تولید سخت‌افزار(فرایند استاندارد و سیستماتیک)
- ۲-قابلیت پشتیبانی از سیستم‌های توزیع شده(اشیاء یا مؤلفه‌ها روی سایت‌های گوناگون توزیع می‌شوند)
- ۳-ارائه مدل قوی‌تری که پتانسیل مدیریت پیچیدگی کاربردهای امروزی را دارد باشد
- ۴-کاهش هزینه تولید و نگهداری نرم‌افزار به وسیله در نظر گرفتن اشیاء به عنوان واحد مجتمع پذیر تفکیک‌نشدنی
- ۵-افزایش مقیاس‌پذیری و قابلیت توسعه سیستم‌ها به وسیله محصورسازی
- ۶-استفاده مجدد به وسیله تکنولوژی مؤلفه‌ها(.NET, COM, Java Beans) که بر مفاهیم مدل شیء مبتنی است

۱-۷-سؤالات تشریحی

- ۱-اصطلاحات زیر را توضیح دهید ؟
الف) تجرید ب) کپسوله سازی(محصورسازی) ج) وراثت
د) اصل ماژولاریتی
- ۲-نقش محصورسازی در کنترل پیچیدگی را توضیح دهید.

۱-۸-سؤالات چهارگزینه‌ای

- ۱- جدایی ساختار داخلی از نحوه استفاده از کلاس از مزایای کدامیک از اصول شیء‌گرایی است؟
- الف- وراثت ب- تجرید ج- محصورسازی د- چند شکلی
- ۲- تجرید
- الف- فرآیند متمرکز شدن روی ویژگی‌ها و رفتار اصلی یک پدیده و نادیده گرفتن ویژگی‌ها غیر مهم آن از یک زاویه دید مشخص است.
- ب- ابزار اصل کنترل پیچیدگی است
- ج- یکی از اصول شیء‌گرایی است
- د- همه موارد
- ۳- کدام گزینه تفاوت تجرید و محصورسازی را به درستی بیان می‌کند؟
- الف- تجرید یکی از اصول شیء‌گرایی است اما محصورسازی یک تکنیک در شیء‌گرایی است.
- ب- تجرید مکانیزم تعیین جزئیاتی است که باید پنهان شوند اما محصورسازی فرایند پنهان‌سازی جزئیات و کنترل دسترسی به آن خواهد بود.
- ج- محصورسازی مکانیزم تعیین جزئیاتی است که باید پنهان شوند اما تجرید فرایند پنهان‌سازی جزئیات و کنترل دسترسی به آن خواهد بود
- د- هیچ‌کدام
- ۴- کدام گزینه در مورد اصل مازولاریتی صحیح است؟
- الف- وابستگی کم و چسبندگی زیاد ب- انسجام و چسبندگی زیاد و وابستگی کم
- ج- کوپلائز زیاد و وابستگی کم د- هیچ‌کدام
- ۵- کدام گزینه غلط است
- الف- ابزار اصلی کنترل تجرید، نمودار کلاس است
- ب- با حرکت از بالای نمودار کلاس به سمت پایین، سطح تجرید کاهش می‌یابد.
- ج- سلسه‌مراتب کلاس‌ها را می‌توان بهوسیله روابط وراثت، تجمعی و انجمانی (همکاری) نمایش داد.
- د- رابطه تجمعی سلسه، مراتب ساختار شیء را نشان می‌دهد
- ۶- کدام گزینه صحیح است؟

- الف- به رابطه subclass و superclass از بالا به پایین، IS-A گفته می شود.
- ب- به رابطه subclass و superclass از بالا به پایین، Generalized-to گفته می شود.
- ج- به رابطه subclass و superclass از بالا به پایین، Specialized-to گفته می شود.
- د- به سلسله مراتب IS-A، رابطه وراثت نیز می گویند.

پاسخنامه:

د	ج	ب	الف	1
د	ج	ب	الف	2
د	ج	ب	الف	3
د	ج	ب	الف	4
د	ج	ب	الف	5
د	ج	ب	الف	6

