بسم الله الرحمن الرحیم

تمرینات درس نرم افزار   
UML بخش اول

**1 - تعریف UML**

زبان UML یا Unified Modeling Language یک زبان استاندارد برای طراحی و مستندسازی سیستم‌های نرم‌افزاری و حتی غیر نرم‌افزاری است. این زبان با استفاده از نمودارهای گرافیکی، ساختار و رفتار سیستم را مدل‌سازی می‌کند.

**2 - فهرست نمودارهای UML**

1. Use Case Diagram
2. Activity Diagram
3. State Diagram
4. Communication Diagram
5. Sequence Diagram
6. Timing Diagram
7. Interaction Overview Diagram
8. Profile Diagram
9. Composite Structure Diagram
10. Class Diagram
11. Package Diagram
12. Object Diagram
13. Component Diagram
14. Deployment Diagram

**3 - کاربرد UML**

UML ابزار بسیار مفیدی برای تحلیل، طراحی، مستندسازی و پیاده‌سازی نرم‌افزار است. با کمک آن می‌توان تعاملات سیستم را به‌صورت دیداری نمایش داد و ارتباط بین اجزای سیستم را درک کرد.

**4 - مزایای UML**

* درک بهتر سیستم قبل از کدنویسی
* تسهیل ارتباط بین اعضای تیم توسعه
* کاهش احتمال بروز خطا در مراحل پیاده‌سازی
* مستندسازی منسجم برای پروژه‌های آینده

**5 - معایب UML**

* یادگیری آن در ابتدا ممکن است دشوار باشد
* برای پروژه‌های کوچک ممکن است اضافه‌کاری به نظر برسد
* نمودارهای پیچیده ممکن است برای افراد غیر فنی گمراه‌کننده باشد

**6 - جنبه‌های ایستا و پویا**

* جنبه ایستا (ساختاری): شامل نمودارهایی مانند کلاس، کامپوننت و پکیج است که ساختار سیستم را نمایش می‌دهند
* جنبه پویا (رفتاری): شامل نمودارهایی مانند فعالیت، توالی و وضعیت است که رفتار سیستم در زمان اجرا را نشان می‌دهند

**7 - نمودارهای UML با تعریف و مثال**

1. Use Case Diagram  
   تعریف: تعامل کاربران با سیستم را نشان می‌دهد  
   مثال: کاربر می‌تواند «ورود» یا «ثبت‌نام» کند  
   توضیح: Actor ها مشخص می‌شوند و سناریوهای کاربردی نمایش داده می‌شوند
2. Activity Diagram  
   تعریف: فرآیندها و گردش کار را مدل‌سازی می‌کند  
   مثال: خرید آنلاین شامل مراحل ثبت سفارش، پرداخت، تأیید  
   توضیح: جریان فعالیت‌ها به صورت فلش نمایش داده می‌شود
3. State Diagram  
   تعریف: وضعیت‌های مختلف یک شی در طول زمان را نمایش می‌دهد  
   مثال: وضعیت یک سفارش: ثبت شده ← در حال ارسال ← تحویل شده  
   توضیح: رویدادها باعث تغییر وضعیت‌ها می‌شوند
4. Communication Diagram  
   تعریف: نحوه ارتباط بین اشیا را نمایش می‌دهد  
   مثال: کاربر ← کنترلر ← سرویس ← پایگاه داده  
   توضیح: تمرکز روی ساختار ارتباطی و تبادل پیام‌هاست
5. Sequence Diagram  
   تعریف: ترتیب زمانی پیام‌ها بین اشیا را نشان می‌دهد  
   مثال: کاربر وارد می‌شود ← اعتبارسنجی ← نمایش داشبورد  
   توضیح: محور عمودی زمان و افقی اشیاء هستند
6. Timing Diagram  
   تعریف: زمان‌بندی تغییر وضعیت‌ها در طول زمان  
   مثال: سنسور فعال ← ثبت داده ← غیرفعال  
   توضیح: برای سیستم‌های حساس به زمان کاربرد دارد
7. Interaction Overview Diagram  
   تعریف: ترکیب چند تعامل به صورت کلی  
   مثال: عملیات خرید شامل احراز هویت، انتخاب محصول، پرداخت  
   توضیح: نمای کلی از چند نمودار تعاملی
8. Profile Diagram  
   تعریف: توسعه سفارشی برای UML  
   مثال: تعریف stereotype جدید برای اشیای خاص  
   توضیح: برای ایجاد قواعد دامنه‌ای خاص استفاده می‌شود
9. Composite Structure Diagram  
   تعریف: ساختار داخلی کلاس‌ها یا مؤلفه‌ها  
   مثال: کلاس خودرو شامل موتور، فرمان، چرخ‌ها  
   توضیح: روابط بین اجزای داخلی کلاس‌ها مشخص می‌شود
10. Class Diagram  
    تعریف: ساختار کلاس‌ها، ویژگی‌ها و متدها  
    مثال: کلاس دانشجو با نام، کدملی، شماره دانشجویی  
    توضیح: نمایش روابط ارث‌بری، انجمنی، ترکیبی و …
11. Package Diagram  
    تعریف: گروه‌بندی کلاس‌ها در بسته‌ها  
    مثال: پکیج حسابداری شامل کلاس‌های کاربر، حساب، تراکنش  
    توضیح: سازمان‌دهی منطقی کلاس‌ها
12. Object Diagram  
    تعریف: نمایش اشیای نمونه از کلاس‌ها در یک لحظه  
    مثال: نمونه‌ای از کلاس ماشین با رنگ قرمز  
    توضیح: وضعیت فعلی سیستم را نشان می‌دهد
13. Component Diagram  
    تعریف: اجزای منطقی و فیزیکی سیستم  
    مثال: کامپوننت ورود کاربر، کامپوننت گزارش‌گیری  
    توضیح: برای معماری سطح بالا مناسب است
14. Deployment Diagram  
    تعریف: نحوه استقرار اجزای نرم‌افزار روی سخت‌افزار  
    مثال: نصب پایگاه داده روی سرور A و وب‌سایت روی سرور B  
    توضیح: شامل Node ها، اجزا، و ارتباطات بین آن‌هاست