

بسمه تعالی

استاد: مهندس احمد زاده

تمرین: اول

نام و نام خواندگی: علی حاجی حسینی

درس: توسعه نرم افزار

ترم بهمن 1403



۱- نمودار object diagram یک نمودار UML است که نمونه‌های خاص از کلاس‌ها و روابط بین آن‌ها را در یک لحظه خاص از زمان نشان می‌دهد.

۲ - state diagram - sequence diagram - class diagram - communication diagram

۳- کاربرد: نمایش وضعیت سیستم - تحلیل و اشکال زدایی - مستندات تست و امکان‌سنجی

۴- مزایا: وضوح - درک آسان - کمک به اشکال زدایی - مستندات سنجی

۵- معایب: محدودیت زمان - پیچیدگی در سیستم‌های بزرگ - زمان‌بر

۶- ارتباط: نمایش به‌فراوان در فازهای خاص از روابط بین آن‌ها / برای نمایش رفتار سیستم با کمک نمودارهای دیگر

۱- نمودار Resource Diagram: یکی نمودار UML است که ترکیب تعاملات بین اشیاء و ادیت سیستم به صورت گام به گام در طول زمان نشان می دهد. این نمودار بر روی پیام های جابجایی شده و ترکیب زمان آن ها متمرکز است. در UML Diagram - communication Diagram - class Diagram - usecase Diagram - activity Diagram

۲- کاربرد: مدل سازی تعاملات - طراحی سیستم - تحلیل رفتار سیستم - مستندات - مزایا: واضح - تمرکز بر زمان - دقت اسان - کمک به اشکال زدایی

۳- عایب: پیچیدگی در سیستم های بزرگ - عدم ساختار ایستا - زمان طولانی برای ایجاد اشیاء؛ نمایش ساختار ایستای و ارتباط بین آن ها؛ مزایا: نمایش ترکیب پیام ها و تعاملات - بین اشیاء در طول زمان

۱- نمودار Communication Diagram: یک نمودار UML است

که تعاملات بین اشیاء را در یک سیستم نشان می‌دهد. این نمودار برای

ارتباطات و پیام‌ها با جابجایی بین اشیاء ترسیم می‌شود.

۲- sequence Diagram - class Diagram - use case Diagram.

activity Diagram

۳- کاربرد: مدل‌سازی تعاملات - طراحی سیستم - تحلیل رفتار سیستم -

مستندات

۴- مزایا: ترسیم ارتباطات - انعطاف‌پذیری - درک انسان‌محور

اشکال‌زدایی

۵- معایب: پیچیدگی در سیستم‌های بزرگ - عدم نمایش دقیق زمان -
زیاد بودن

۶- ایستا: نمایش ساختار اشیاء و ارتباط بین آن‌ها به صورت: نمودار پیام

و تعاملات بین اشیاء در طول زمان

۵- اینجا! نمایان ساختار حالت ها و روابط بین آن ها / یوا: نمایان تغییرات حالت ها و زمان ها / زمان در طول زمان

۱- نمودار interaction diagram: یک نمودار UML است که تعاملات بین اجزای سیستم نشان می دهد. هدف اصلی آن نمایش مجموعه تعاملات است برای اجزای سیستم.

communication diagram sequence diagram
interaction overview diagram timing diagram

۶- کاربرد: مهندسی تعاملات - طراحی سیستم - تحلیل رفتار سیستم - مهندسی

۴- انواع موضوع - اشیاء سیستم - درک انسان - مکان
۵- اشکال زمانی

۷- معایب: پیچیدگی در سیستم ها با بزرگ - عدم نمایش ساختار کلی - زمان بردن

۱- منبع ایستا؛ نمایش ساختار کلی فرایند فعالیت ما / پویا: نمایش
تغییرات متغیر و ترکیب اجزای فعالیت ما

۲- نمودار Activity Diagram: یک نمودار UML است که حالت‌های
متغیر یک شیء و اشکال بین آن‌ها را در طول حلقه زندگی آن نشان می‌دهد.
این نمودار برای مدل‌سازی و رفتار یک شیء در پاسخ به رویدادها یا کاربری
استفاده می‌شود.

۳- Activity Diagram - Sequence Diagram - class diagram
use case diagram

۴- کاربرد: مدل‌سازی رفتار شیء - حل مسائل سیستم‌ها یا پیچیده - مدل‌سازی

۵- مزایا: وضوح - تمرکز بر رفتار - پشتیبانی از سیستم‌ها یا پیچیده - درک آسان

۶- معایب: پیچیدگی در سیستم‌ها یا بزرگ - عدم نمایش جزئیات فنی - زمان‌بر بودن

۷- ایستا: نمایش ساختار حالت‌ها و روابط بین آن‌ها / پویا: نمایش تغییرات
و رفتار دینامیکی سیستم در پاسخ به رویداد

شنبه ۲۰ | Saturday 10

۱- نمودار UML و $Diagram$ یک نمودار UML است که تغییر حالت های یک
سیستم یا سیستم را در طول زمان نشان می دهد و این نمودار برای
زمان بندی و مدت زمان تغییرات حالت استفاده می شود و معمولاً برای
یک سیستم با درخت استفاده می شود.

۲- $Activity Diagram$ - $State Diagram$ - $Sequence Diagram$ -
 $Use Case Diagram$

۳- کاربرد: مدل سازی سیستم های با درخت، تحلیل زمان بندی -

همه اوقات سیستم های سخت افزار و نرم افزار - مستند سازی

۴- برای: موضوع - ترسیم زمان بندی - مناسب برای سیستم های با درخت

درخت زمان

۵- کاربرد: پیچیدگی در سیستم ها - ترسیم - علم مهندسی

سخت افزار (سیستم) - زمان بندی

۱- نمودار components diagram: یک نمودار UML است که ساختار سیستم را با نمایش کامپوننت‌ها و واسطه‌ها و وابستگی بین آن‌ها نشان می‌دهد.

۲- class Diagram - package diagram - Deployment Diagram - composite structure diagram

۳- کاربرد: طراحی سیستم مبتنی بر کامپوننت - مدل سازی واسطه‌ها - تحلیل وابستگی‌ها - کمک به اشکال زدایی

۴- مزایا: وضوح - تمرکز بر مائولاریتی - درک آسان - سبب اشکال زدایی

۵- معایب: پیچیدگی در سیستم‌های بزرگ - عدم نمایش رفتار پویا - زمان‌بر بودن

۶- ایستگاه: نمایش ساختار کامپوننت‌ها و وابستگی‌ها بین آن‌ها / پویا

نمایش رفتار پویای سیستم با کمک نمودارهای دیگر

use case Diagram: یک نوع نمودار UML است که نشان می‌دهد که کاربران چه کارهایی می‌توانند با یک سیستم انجام دهند. این نمودار به سادگی تعاملات بین کاربران و سیستم را نمایش می‌دهد و برای درک نیازهای سیستم و عملکرد آن مفید است.

- activities Diagram sequence Diagram class Diagram
state Diagram

۱- کاربرها: تحلیل نیازمندی ها - طراحی سیستم - مستقرسازی -

۲- مزایا: سادگی، ارتباط موثر - تمرکز بر کاربر

۳- معایب: عدم جزئیات - استاتیک بودن

۴- جنبه‌ها: نمایش تعاملات و جریان کار بین actors و سیستم (در نمودارهای دیگر مانند sequence Diagram)

۱- نمودار Composite Structure Diagram: یک نمودار UML است که نشان می‌دهد اجزای یک کلاس یا کامپوننت را چگونه می‌توان به یکدیگر متصل کرد. این نمودار در درون اجزای داخلی، ارتباط بین آن‌ها و نحوه تعامل آن‌ها مشخص می‌کند.

۲- Deployment Diagram, Component Diagram, Class Diagram, Object Diagram

۳- کاربرد: مهندسی نرم افزار، ساختار داخلی، طراحی سیستم‌های پیچیده، تحلیل وابستگی

۴- مزایا: موضوع - تمرکز بر جزئیات - درک آسان - کمک به امکان‌پذیری

۵- معایب: پیچیدگی در سیستم‌ها، بزرگ - عدم تعادل در تعاریف

۶- استناد: شاید ساختار داخلی و ارتباط بین اجزا / هویت / تعریف شده
تفصیل اجزای داخلی و نحوه تعامل آن‌ها

دوشنبه
Monday ۲۲ | 12

۶- نویسنده! شاید ساختار اسباب را ارتباط بین آن ها / برآید!
نمایشن نکات و پیام های بسیار و نکته بین اسباب در طول زمان

۱- نمودار (profile diagram) یک نمودار VML است که برای تغییر و
گسترش مفاهیم خاص در یک حوزه یا به مقصود خاص استفاده می شود

۲- component diagram, package diagram, class diagram
Deployment Diagram

۳- کاربرد و VML مفاهیم ساز - هر سازی (اینها خاص -
پیشگانی از بلکترج خاص - هسته ساز

۴- برای! اینها یک نوعی موضوع - پیشگانی از داستان مختلف - در
ایمان

۵- مایب! پیچیده - زمان بر بودن - نیاز به عضو

۶- ایست! شاید ساختار مفاهیم و نمایشن خاص / برآید! شاید
مقدار و نتایج خاص در دامن یا بلکترج

۱- نمودار ^{Diagram} activities یک نمودار UML است که برای نمایش جریان کار یا فرایندهای کسب و کار در سیستم استفاده می‌شود.

این نمودار در مراحل مختلف یک فرایند، تصمیم‌گیری‌ها و جریان کنترل بین فعالیت‌ها را نشان می‌دهد.

۲- Sequence Diagram - Use case Diagram
Flowchart - State Diagram

۳- کاربر (User)؛ مدل‌سازی فرایند کسب و کار - طراحی سیستم -

تحلیل و بهینه‌سازی ساختار - ساختار سازی

۴- کارایا؛ موضوع - فعالیت‌پذیری - درک انسان -

پیشگامان از موانع سازی

۵- پیچیدگی در فرایندهای بزرگ - عدم نمایش جزئیات فردی -

موسسان برپودن

۱- نمودار Deployment Diagram: یک نمودار UML است که توزیع استقرار نرم افزار را بر روی سخت افزار، نشان می دهد.

۲- Use case Diagram - Package Diagram - Class Diagram - Component Diagram

۳- کاربرد: مدل سازی استقرار - طراحی سیستم های توزیع شده - تحلیل سیستم سازی

۴- مزایا: تمرکز بر روی سیستم - موقعیت - درک انسان - کمک به یادگیری

۵- معایب: پیچیدگی در سیستم های بزرگ - عدم نمایش رفتار پویا - زمان بر بودن

۶- رستا: نمایش ساختار و ارتباط بین آن ها
پویا: نمایش رفتار پویای سیستم با کمک نمودارهای دیگر

15

Thursday

۲۵

۱- نمودار package diagram و یک نمودار UML است که ساختار سیستم را در سطح بالاتر نشان می‌دهد و وابستگی بین آن‌ها را نشان می‌دهد.

۲- class diagram . component diagram . use case diagram . deployment diagram

۳- کاربرد: سازمان درون سیستم - مداخلات ساختار سیستم - طراحی سیستم - مستندات

۴- چرا؟! و غرض - سازمان درون - درک انسان - کامپیوتر

۵- ~~نمایش~~ نمایش! عدم نمایش جزئیات - پیچیدگی در سیستم‌های بزرگ - زمان

۶- اینها! نمایش ساختار کلیه ما و وابستگی بین آن‌ها / بویا! نمایش و بویا سیستم با کمک نمودارهای دیگر

۱- نمودار class diagram: یک نمودار UML است که ساختار سیستم را با یکتای کلاس‌ها، ویژگی‌ها، عملیات‌ها، روابط بین آن‌ها نشان می‌دهد.

۲- component diagram - package diagram - object diagram - composite structure diagram

۳- کاربرد: طراحی سیستم، تحلیل نیازمندی‌ها، سکو سازی - پلاده سازی

۴- مزایا: وضوح - پستیابی از شی گرای - درک آسان - محکم
به اشکال برداری

۵- معایب: پیچیدگی در سیستم‌های بزرگ - عدم نمایش رفتار پویا
زمان بر بودن

۶- ایستای نمایش ساختار کلاس‌ها و روابط بین آن‌ها / پویا:

نمایش رفتار پویای سیستم با کمک نمودارهای دیگر

ASP! متوقف! Active server pages است و یک فناوری قدیمی تر
 برای ساخت وب است که توسط مایکروسافت ایجاد شده. این فناوری به توسعه
 برنامه‌نویسان اجازه می‌دهد تا صفحات وب پویا و تعاملی ایجاد کنند. از اسکرپت
 زبان روسی سرور استفاده می‌کنند و معمولاً با زبان‌های مانند VBScript یا
 JScript اسکریپت می‌نویسند.

dotNET! یک پلتفرم توسعه نرم افزار است که توسط مایکروسافت
 ایجاد شده و برای ساخت و ایجاد برنامه‌های کاربردی تحت ویندوز و وب،
 موبایل و سایر پلتفرم استفاده می‌شود. این پلتفرم از زبان‌های برنامه‌نویسی
 مختلف استفاده می‌کند و ابزار و کتابخانه قدرتمندی را در اختیار توسعه‌دهندگان
 قرار می‌دهد.

ASP.NET core! یک چوب‌مدار متن‌باز و چند پلتفرمی برای
 ساخت برنامه‌های وب و سرور تحت وب است که توسط مایکروسافت
 توسعه داده شده است.

20

سه شنبه

Tuesday

۳۰

۴- middleware! در برنامه نویسی، به ویژگی در میان اجزای وب مانند ASP.NET core، به قطعه‌ای که امر افزارگی نظام می‌دهد که در لایه‌های پایین درخواست و پاسخ قرار می‌گیرد و وظیفه‌ها را در زیر درخواست‌ها می‌رود و پاسخ‌های خروجی را برمی‌دارد.

۵- Web Routing: یک سیستم است که تعیین می‌کند چگونه URL درخواست شده توسط کاربر به منابع مناسب در برنامه وب نگاشت شوند.

۶- Data Annotation: یک ویژگی در میان اجزای وب، برنامه نویسی است که به توسعه دهندگان امکان می‌دهد تا قوانین اعتبار سنجی و داده‌ها را به هر درستی ساده و با استفاده از ویژگی‌ها به مدل‌ها داده اضافه کنند.

22

پنج شنبه

Thursday

۱

۱۱- ORM! یک تکنیک برنامه نویسی است که به کمک امکان
 می دهد با استفاده از اشیا و کلاس ها در برنامه نویسی با پایگاه
 داده ها رابطه ای کار کنید

۱۲- Entity framework! یک ~~چارچوب~~ چارچوب ORM است که توسط
 مایکروسافت برای پلتفرم دات نت توسعه داده شده است. این چارچوب
 به برنامه نویسان کمک می کند تا با استفاده از اشیا و کلاس ها
 در #C یا VB.NET با پایگاه داده ها رابطه ای کار کنند.

۱۳- Razor! یک موتور نمایش است که توسط مایکروسافت برای
 ساخت صفحات وب به زبان درجه اول چارچوب ها و پرتفله داده شده
 است

۱۴- Validation! به فرایند بررسی و اطمینان از صحت داده
 ها و ورودی کاربر یا سیستم گفته می شود.

۱- Db Context : یک کلاس اصلی در Entity Framework است

در نقش یک ارتباطی بین برنامه و پایگاه داده را ایفا می‌کند. این کلاس به شما امکان می‌دهد تا با پایگاه داده کار کنید بدون اینکه نیاز به نوشتن کوئری‌های SQL خاص داشته باشید

۲- connection string : یک رشته متنی است که شامل اطلاعات لازم برای اتصال به یک پایگاه داده یا منبع داده دیگر باشد. این رشته شامل جزئیاتی مانند نام سرور، نام پایگاه داده، اعتبارنامه‌ها و سایر تنظیمات مورد نیاز برای برقراری ارتباط است.

۳- Dependency Injection : یک الگوی طراحی در برنامه نویسی است که به شما کمک می‌کند وابستگی‌های یک کلاس را از خارج به آن کلاس تزریق کنید و به جای اینکه کلاس خودش آن‌ها را ایجاد کند.

۴- Controller : یکی از اجزای اصلی در الگوی MVC است که نقش مدیریت منطق برنامه و هماهنگی بین Model و View را بر عهده دارد

منبع:

<https://www.deepseek.com/>