1. UML چیه؟

UML یه زبان تصویریه که کمک می‌کنه قبل از اینکه برنامه‌نویسی رو شروع کنیم، شکل و رفتار برنامه‌مون رو با نمودار نشون بدیم. مثلاً مثل کشیدن نقشه‌ی خونه قبل از ساختن اون.

---

2. نمودارهای UML (لیست ساده‌شده)

UML ۱۴ تا نمودار اصلی داره. بعضی‌هاش برای نشون دادن ساختار برنامه هستن (ثابت و ایستا)، بعضی‌هاش برای رفتار و عملکرد برنامه (پویا و متغیر):

نمودارهای ساختاری (ایستا):

1. نمودار کلاس (Class Diagram)

2. نمودار اشیا (Object Diagram)

3. نمودار بسته‌ها (Package Diagram)

4. نمودار اجزا (Component Diagram)

5. نمودار استقرار (Deployment Diagram)

6. نمودار ساختار ترکیبی (Composite Structure Diagram)

7. نمودار پروفایل (Profile Diagram)

نمودارهای رفتاری (پویا):

1. نمودار موارد استفاده (Use Case Diagram)

2. نمودار فعالیت (Activity Diagram)

3. نمودار حالت (State Diagram)

4. نمودار توالی (Sequence Diagram)

5. نمودار ارتباط (Communication Diagram)

6. نمودار زمان‌بندی (Timing Diagram)

7. نمودار نمای کلی تعامل (Interaction Overview Diagram

3. کاربرد UML

طراحی و برنامه‌نویسی بهتر

درک ساده‌تر سیستم توسط اعضای تیم

جلوگیری از اشتباه در پروژه

مستندسازی برای آینده

---

4. مزایای UML

باعث نظم و برنامه‌ریزی میشه

فهم سیستم رو آسون می‌کنه

بین اعضای تیم زبان مشترک ایجاد می‌کنه

قبل از نوشتن کد، ایرادها پیدا می‌شن

---

5. معایب UML

یادگیری همه‌ی نمودارها زمان‌بره

برای پروژه‌های کوچیک ممکنه زیادی باشه

گاهی طراحی زیاد باعث دیر شدن شروع کار میشه

---

6. ایستا و پویا یعنی چی؟

ایستا (ساختاری): نشون می‌ده برنامه از چی ساخته شده (مثل کلاس‌ها و اشیا)

پویا (رفتاری): نشون می‌ده برنامه چطوری رفتار می‌کنه (مثل تعامل کاربر با سیستم)

---

7. نمودارها همراه با تعریف ساده و مثال

1. Use Case Diagram (نمودار مورد استفاده)

چی نشون می‌ده؟ کارهایی که کاربر می‌تونه با سیستم انجام بده.

مثال:

سیستم کتابخانه:

کاربر می‌تونه «کتاب جستجو کنه»، «کتاب قرض بگیره»، «عضو بشه».

---

2. Activity Diagram (نمودار فعالیت)

چی نشون می‌ده؟ مراحل انجام یه کار مثل یک فلوچارت.

مثال:

ثبت‌نام کاربر:

وارد کردن اطلاعات → بررسی اطلاعات → تأیید ثبت‌نام

---

3. State Diagram (نمودار حالت)

چی نشون می‌ده؟ وضعیت‌های مختلف یک شی و تغییراتش.

مثال:

چراغ راهنمایی:

قرمز → سبز → زرد → قرمز

---

4. Communication Diagram (نمودار ارتباط)

چی نشون می‌ده؟ چطوری اشیا با هم حرف می‌زنن و پیام ردوبدل می‌کنن.

مثال:

کاربر → فرم ورود → سرور → دیتابیس

---

5. Sequence Diagram (نمودار توالی)

چی نشون می‌ده؟ ترتیب پیام‌ها بین اشیا به ترتیب زمانی.

مثال:

کاربر → ورود به سیستم → بررسی رمز → نمایش پیام ورود موفق

---

6. Timing Diagram (نمودار زمان‌بندی)

چی نشون می‌ده؟ وضعیت اشیا در طول زمان.

مثال:

سنسور دما:

دما افزایش پیدا می‌کنه → هشدار روشن میشه → فن روشن میشه

---

7. Interaction Overview Diagram (نمودار نمای کلی تعامل)

چی نشون می‌ده؟ خلاصه‌ای از چند تعامل مختلف در یک فلوچارت.

مثال:

فرآیند خرید آنلاین شامل:

انتخاب کالا

پرداخت

دریافت پیام تأیید

---

8. Profile Diagram (نمودار پروفایل)

چی نشون می‌ده؟ سفارشی‌سازی UML برای پروژه خاص.

مثال:

کلاسی که فقط برای موبایل طراحی شده رو با برچسب خاص نشون می‌دیم.

---

9. Composite Structure Diagram (ساختار ترکیبی)

چی نشون می‌ده؟ اجزای داخلی یک کلاس و نحوه ارتباطشون.

مثال:

کلاس «ماشین» شامل:

موتور

چرخ

فرمان

---

10. Class Diagram (نمودار کلاس)

چی نشون می‌ده؟ کلاس‌ها، ویژگی‌ها و روابط بینشون.

مثال:

کلاس «دانشجو» با ویژگی‌های: نام، شماره دانشجویی، درس‌ها

---

11. Package Diagram (نمودار بسته‌ها)

چی نشون می‌ده؟ گروه‌بندی کلاس‌ها یا بخش‌های برنامه.

مثال:

بسته‌ها: «کاربران»، «محصولات»، «پرداخت»

---

12. Object Diagram (نمودار اشیا)

چی نشون می‌ده؟ نمونه‌های واقعی از کلاس‌ها در یک لحظه.

مثال:

شی‌ای از کلاس دانشجو: نام = علی، شماره = 1234

---

13. Component Diagram (نمودار اجزا)

چی نشون می‌ده؟ اجزای اصلی نرم‌افزار و وابستگیشون.

مثال:

کامپوننت‌های سیستم بانکی: «UI»، «مدیریت حساب»، «پرداخت»

---

14. Deployment Diagram (نمودار استقرار)

چی نشون می‌ده؟ روی چه سخت‌افزاری و کجا برنامه اجرا می‌شه.

مثال:

سرور بانک:

نرم‌افزار پرداخت نصب شده روی سرور مرکزی

پایگاه‌داده روی سرور دوم

1. ASP

Active Server Pages (ASP) یک فناوری مایکروسافت برای ساخت صفحات وب داینامیک (پویا) است. اولین نسخه‌ها در زمان‌های قدیم‌تر با زبان VBScript بودند.

مثال: وقتی کاربر فرمی رو پر می‌کنه، ASP می‌تونه اطلاعات رو پردازش کنه و به پایگاه داده بفرسته.

---

2. .NET (دات‌نت)

.NET یک پلتفرم برنامه‌نویسی از مایکروسافته که برای ساخت نرم‌افزارهای مختلف (وب، دسکتاپ، موبایل، بازی) استفاده میشه.

مثال: با .NET می‌تونی یه اپلیکیشن فروشگاهی برای موبایل یا وب بنویسی.

---

3. ASP.NET Core

نسل جدیدی از ASP.NET هست که سریع‌تر، سبک‌تر و کراس‌پلتفرم (روی ویندوز، لینوکس و مک اجرا میشه) است. برای ساخت اپلیکیشن‌های مدرن وب، API و بلادرنگ استفاده میشه.

مثال: ساخت سایت خبری که روی لینوکس یا ویندوز اجرا میشه.

4. Middleware

Middlewareها بخش‌هایی از برنامه هستن که بین درخواست کاربر و پاسخ سرور قرار دارن و می‌تونن چیزی رو بررسی یا تغییر بدن.

مثال: Middleware برای بررسی اینکه کاربر لاگین کرده یا نه.

5. Web Routing

روتینگ یعنی تعیین اینکه وقتی کاربر به یه آدرس خاص میره، چه چیزی بهش نشون داده بشه.

مثال:

آدرس /products بره به کنترلر Products و تابع List()

---

6. Data Annotation

ویژگی‌هایی هستن که بالای کلاس یا فیلدها نوشته می‌شن تا ویژگی‌هایی مثل الزامی بودن، طول متن، فرمت ایمیل و... رو مشخص کنن.

مثال:

[Required]

[ Max Length (50)]

public string Name {get; set;}

7. Db Context

کلاسیه که ارتباط بین برنامه و پایگاه داده رو مدیریت می‌کنه. درواقع، نقش دروازه بین کد شما و دیتابیس رو داره.

مثال:

public class App Db Context : Db Context {

public Db Set<Product> Product { get ; set; }

}

---

8. Connection String

رشته‌ای از اطلاعات که به برنامه می‌گه به کدوم پایگاه داده وصل بشه و با چه تنظیماتی.

مثال:

"Connection Strings": {

"Default": "Server=.;Database=Shop Db ;Trusted \_Connection=True;"

}

--

9. Dependency Injection (DI)

روشی برای فرستادن وابستگی‌ها (مثل سرویس‌ها یا کلاس‌ها) به داخل کلاس‌های دیگه، به‌جای اینکه خودشون بسازنشون.

مثال:

یک کنترلر نیازی به ساخت سرویس نداره، سرویس بهش "تزریق" میشه.

---

10. Controller

کنترلر بخشی از برنامه است که درخواست‌های کاربر رو می‌گیره، بررسی می‌کنه، و پاسخ مناسب برمی‌گردونه (معمولاً به View یا API).

مثال:

public class controller: Controller {

public Result index () => view();

}

---

11. ORM (Object Relational Mapping)

فناوری‌ای که اجازه می‌ده با پایگاه داده مثل کار با اشیاء (کلاس‌ها) برخورد کنیم، نه با کوئری‌های SQL مستقیم.

---

12. Entity Framework (EF Core)

محبوب‌ترین ORM مایکروسافته که به کمکش می‌تونیم بدون نوشتن SQL، با پایگاه داده کار کنیم.

مثال:

Db Context product s . add(new Product { Name = "Laptop" });

Db Context. Save Changes();

---

13. Razor

Razor یک زبان ترکیبی HTML + C# برای نوشتن Viewها (صفحه‌های وب) در ASP.NET Core هست.

مثال:

<p>سلام @Model.name!</p>

---

14. Validation (اعتبارسنجی)

بررسی اینکه اطلاعات واردشده توسط کاربر درست هستن یا نه (مثلاً خالی نباشه، عدد باشه و...)

مثال:

[Required(error message= "وارد کردن نام الزامی است").