

بسمه تعالی

شماره دانشجویی: 02121153705039

نام و نام خانوادگی: علی حاجی حسینی

استاد: مهندس محمد احمدزاده

درس: مهندسی نرم افزار



## Configuration Script (اسکرپت پیکربندی):

Configuration Script (اسکرپت پیکربندی) به برنامه‌ای اطلاق می‌شود که برای تنظیم و پیکربندی سیستم‌ها، نرم‌افزارها، یا محیط‌های مختلف به کار می‌رود. این اسکرپت‌ها عموماً شامل مجموعه‌ای از دستورات و تنظیمات هستند که می‌توانند به طور خودکار برای راه‌اندازی و تنظیم خدمات، پایگاه‌های داده، سرورها یا هر نوع نرم‌افزاری که نیاز به پیکربندی دارد، اجرا شوند.

ویژگی‌های مهم اسکرپت‌های پیکربندی عبارتند از:

**اتوماسیون:** این اسکرپت‌ها می‌توانند فرآیندهای تکراری را خودکار کنند و زمان و تلاش را برای راه‌اندازی سیستم‌ها کاهش دهند.

**قابلیت تکرار:** با استفاده از این اسکرپت‌ها می‌توان سیستم‌ها را به صورت یکنواخت و قابل پیش‌بینی پیکربندی کرد.

**مدیریت تغییرات:** تغییرات در پیکربندی می‌تواند به راحتی توسط ویرایش اسکرپت‌ها پیگیری و اعمال شود.

اسکرپت‌های پیکربندی می‌توانند با زبان‌های مختلفی مانند Bash، Python، PowerShell و غیره نوشته شوند و در بسیاری از چارچوب‌ها و ابزارهای DevOps مورد استفاده قرار می‌گیرند.

## تست اتوماسیون (Test Automation):

تست اتوماسیون (Test Automation) به فرآیند استفاده از نرم افزار برای اجرای تست های نرم افزاری به طور خودکار اشاره دارد. این روش به جای انجام تست ها به صورت دستی، از ابزارها و اسکریپت ها برای اجرای تست ها و بررسی نتایج استفاده می کند. در زیر به چند نکته کلیدی در مورد تست اتوماسیون اشاره می کنم:

### 1. مزایای تست اتوماسیون:

**افزایش سرعت:** تست های اتوماسیون می توانند سریع تر از تست های دستی اجرا شوند و به توسعه دهندگان و تیم های QA این امکان را می دهند که به سرعت به بازخورد نیاز داشته باشند.

**دقت بیشتر:** خطاهای انسانی در تست های دستی کاهش می یابد، زیرا تست ها به صورت خودکار بدون خطای انسانی اجرا می شوند.

**تکرارپذیری:** تست های اتوماسیون می توانند چندین بار با ورودی های مختلف اجرا شوند، که این امر به بررسی سناریوهای مختلف کمک می کند.

**صرفه جویی در هزینه:** علی رغم هزینه های اولیه، در درازمدت، تست های اتوماسیون می توانند هزینه های کلی را کاهش دهند.

### 2. فرآیند تست اتوماسیون:

انتخاب ابزار مناسب: ابزارهای متنوعی برای اتوماسیون تست وجود دارد، از جمله Selenium، Appium، JUnit و TestComplete. انتخاب ابزار مناسب بستگی به نیاز پروژه و نوع تست دارد.

**نوشتن تست ها:** اسکریپت های تست باید نوشته شوند که شامل مواردی نظیر ورودی ها، انتظار خروجی و کنترل های مربوطه باشد.

**اجرای تست‌ها:** تست‌ها به صورت خودکار اجرا می‌شوند و خروجی‌ها جمع‌آوری می‌شوند.

**تحلیل نتایج:** نتایج تست باید تحلیل شوند و مشکلات شناسایی شده مستندسازی شوند.

### 3. چالش‌ها:

**هزینه‌های اولیه:** هزینه‌های راه‌اندازی و نگهداری سیستم‌های اتوماسیون ممکن است قابل توجه باشد.

**نگهداری:** تغییرات در کد یا ویژگی‌ها ممکن است به روزرسانی‌های مکرر تست‌های اتوماتیک را می‌طلبد.

**عدم توانایی در تست‌های UI پیچیده:** تست اتوماسیون ممکن است نتواند تمامی جنبه‌های رابط کاربری

را بررسی کند.

### 4. انواع تست‌های قابل اتوماسیون:

**تست‌های عملکردی:** تست‌های که درست کار کردن نرم‌افزار را بررسی می‌کنند.

**تست‌های غیرعملکردی:** مانند تست‌های بار و استقامت که به بررسی رفتار نرم‌افزار تحت فشار می‌پردازند.

**تست‌های رگرسیون:** برای اطمینان از اینکه تغییرات در کد تأثیر منفی بر روی عملکرد سیستم ندارند.

### 5. نتیجه‌گیری:

تست اتوماسیون می‌تواند به تیم‌های توسعه کمک کند تا نرم‌افزارهای خود را سریع‌تر و با دقت بیشتری منتشر

کنند. با این حال، نیاز به ارزیابی دقیق مزایا و چالش‌ها دارد تا مطمئن شویم که اتوماسیون به درستی

پیاده‌سازی می‌شود و واقعاً به نفع پروژه است.

## **:Build Automation**

Build Automation یا خودکارسازی ساخت به فرایند اجرا و مدیریت فرایند ساخت نرم افزار به صورت خودکار اشاره دارد. این فرآیند شامل مجموعه ای از ابزارها و تکنیک ها است که به توسعه دهندگان کمک می کند تا فرایند ساخت نرم افزار را سریع تر و قابل اعتمادتر کنند. در زیر به موضوعات مختلف مرتبط با خودکارسازی ساخت پرداخت شده است:

### **1. تعریف Build Automation:**

خودکارسازی ساخت به معنای این است که تمام مراحل ساخت نرم افزار، از کدگذاری گرفته تا آزمون و استقرار، به صورت خودکار انجام می شود. این کار معمولاً با استفاده از اسکریپت ها و ابزارهای ویژه ای انجام می گیرد.

### **2. مزایای Build Automation:**

**سرعت:** خودکارسازی فرایند ساخت، زمان لازم برای تولید نرم افزار را کاهش می دهد.

**دقت:** کاهش احتمال خطاهای انسانی تا حد زیادی.

**تکرارپذیری:** امکان ساخت نسخه های مختلف نرم افزار به راحتی.

**کارایی:** آزاد کردن زمان توسعه دهندگان برای انجام کارهای مهم تر.

**مدیریت نسخه:** اطمینان از اینکه هر بار که ساختی انجام می شود، کد صحیح و نسخه صحیح مورد استفاده قرار می گیرد.

### **3. فرآیندهای متداول در Build Automation:**

**تجزیه و تحلیل کد:** بررسی کد برای وجود اشکالات و نواقص.

**کامپایل:** تبدیل کد منبع به کد اجرایی.

**آزمون:** اجرای تست‌ها برای اطمینان از کارکرد صحیح نرم‌افزار.

**بسته‌بندی:** ایجاد بسته‌های قابل استقرار برای نرم‌افزار.

**استقرار:** نصب نرم‌افزار در محیط‌های مختلف.

#### 4. ابزارهای محبوب Build Automation:

**Maven:** یکی از محبوب‌ترین ابزارها برای پروژه‌های جاوا، که مدیریت وابستگی‌ها و ساخت را تسهیل می‌کند.

**Gradle:** ابزاری قدرتمند برای پروژه‌های جاوا و اندروید که از زبان Groovy برای تعریف ساخت استفاده می‌کند.

**Ant:** ابزار قدیمی‌تری برای خودکارسازی ساخت که بیشتر برای پروژه‌های جاوا استفاده می‌شود.

**Make:** ابزاری برای ساخت پروژه‌ها که معمولاً در سیستم‌های یونیکس استفاده می‌شود.

**CI/CD Tools:** مانند Jenkins، GitLab CI، CircleCI و Travis CI که به‌طور خاص برای Integration

و Continuous Deployment طراحی شده‌اند.

#### 5. تکنیک‌ها و الگوها:

**Continuous Integration (CI):** هر بار که یک تغییر در کد صورت می‌گیرد، فرآیند ساخت و آزمون

به‌صورت خودکار اجرا می‌شود.

**Continuous Deployment (CD):** نسخه‌های جدید نرم‌افزار به‌طور خودکار به محیط‌های مختلف مستقر

می‌شود.

**Semantic Versioning:** استفاده از سیستم نسخه‌بندی معنایی برای مدیریت نسخه‌های نرم‌افزار.

## 6. چالش‌ها:

**تنظیمات پیچیده:** پیکربندی و تنظیم ابزارهای خودکارسازی می‌تواند زمان‌بر و پیچیده باشد.

**تست و عیب‌یابی:** نیاز به تست‌های جامع و تأمین کیفیت پیش از استقرار در محیط‌های زنده.

## 7. خلاصه:

Build Automation به‌عنوان یک جزء ضروری در فرایند توسعه نرم‌افزار مطرح است. با استفاده از ابزارها و تکنیک‌های مناسب، سازمان‌ها می‌توانند کیفیت و سرعت ساخت نرم‌افزار را به‌طور قابل توجهی افزایش دهند. این پیشرفت‌ها به تیم‌های توسعه اجازه می‌دهد تا تمرکز بیشتری روی نوآوری و توسعه ویژگی‌های جدید داشته باشند.

## :Release automation

Release automation به فرایند خودکاری اشاره دارد که در آن مراحل مرتبط با انتشار یک نرم افزار یا سیستم به صورت خودکار انجام می شود. هدف این فرایند کاهش زمان و خطاهای مرتبط با انتشار نرم افزار، بهبود کیفیت و افزایش قابلیت اطمینان است.

این فرایند معمولاً شامل چندین مرحله مهم است، از جمله:

1. **ساخت (Build):** اتوماسیون مراحل ساخت نرم افزار، که شامل کامپایل کردن کد و ایجاد بسته های اجرایی است.
2. **تست:** اجرای تست های خودکار برای اطمینان از صحت عملکرد نرم افزار قبل از انتشار.
3. **استقرار (Deployment):** انتقال نرم افزار به محیط های مختلف (مانند توسعه، تست، و تولید) به صورت خودکار.
4. **پیکربندی (Configuration):** مدیریت و پیکربندی تنظیمات نرم افزار در محیط های مختلف.
5. **نظارت (Monitoring):** بررسی و پایش سلامت و عملکرد نرم افزار پس از انتشار.

Release automation به سازمان ها کمک می کند تا فرآیندهای انتشار خود را سریع تر، کارآمدتر و ایمن تر کنند و از این طریق بتوانند به نیازهای بازار و مشتریان بهتر پاسخ دهند.



## :Real-Time Software

نرم افزار زمان واقعی (Real-Time Software) به نرم افزاری اطلاق می شود که قابلیت پردازش و پاسخ به داده ها یا رویدادها را در زمان واقعی دارد. این به این معناست که نرم افزار باید بتواند به ورودی ها با تأخیر بسیار کم واکنش نشان دهد و این واکنش باید در زمان مشخصی انجام شود.

نرم افزارهای زمان واقعی معمولاً در سیستم هایی کاربرد دارند که در آن ها تأخیر در پردازش اطلاعات می تواند منجر به عملکرد نادرست یا خطرناک شود. از جمله این سیستم ها می توان به کنترل سیستم های صنعتی، هوافضا، پزشکی، خودروهای خودران و بازی های رایانه ای با نرخ فریم بالا اشاره کرد.

به طور کلی، نرم افزارهای زمان واقعی به دو دسته تقسیم می شوند:

1. سیستم های زمان واقعی سخت (Hard Real-Time Systems): در این سیستم ها، عدم رعایت زمان

واکنش می تواند عواقب جدی و خطرناک به همراه داشته باشد.

2. سیستم های زمان واقعی نرم (Soft Real-Time Systems): در این سیستم ها، تأخیر در پردازش

ممکن است قابل قبول باشد، اما در عین حال می تواند کیفیت عملکرد را کاهش دهد.

نرم افزارهای زمان واقعی به دلیل نیاز به پردازش سریع و دقیق، معمولاً نیاز به استفاده از زبان های برنامه نویسی و

ابزارهایی دارند که عملکرد بهینه و کنترل دقیق بر منابع سیستم را فراهم کنند.

## :Embedded software

Embedded software به نرم‌افزارهایی اطلاق می‌شود که به صورت خاص برای کنترل و مدیریت دستگاه‌های فیزیکی طراحی شده‌اند. این نرم‌افزارها معمولاً در سخت‌افزارهای خاصی پیاده‌سازی می‌شوند و به عملکرد و کارایی آنها بستگی دارند.

دو مورد از مهم‌ترین زبان‌های برنامه‌نویسی که برای توسعه این نوع نرم‌افزارها استفاده می‌شوند، C و Rust هستند:

1. **C:** زبان C یکی از زبان‌های اصلی برای برنامه‌نویسی سیستم و نرم‌افزارهای embedded است. این زبان به دلیل قابلیت کارایی بالا، سرعت اجرا و کنترل دقیق بر منابع سخت‌افزاری، در توسعه نرم‌افزارهای سیستم‌های تعبیه شده (Embedded Systems) به طور گسترده‌ای مورد استفاده قرار می‌گیرد. بسیاری از BIOSها و درایورها به زبان C نوشته شده‌اند.

2. **Rust:** Rust یک زبان برنامه‌نویسی مدرن است که به دلیل ایمنی حافظه و مدیریت مناسب منابع، به طور فزاینده‌ای در توسعه نرم‌افزارهای embedded مورد استفاده قرار می‌گیرد. Rust به شما امکان می‌دهد تا نرم‌افزارهایی با عملکرد بالا و همچنین ایمن بسازید، بدون آنکه با مشکلات رایج مانند نشت حافظه یا شرایط رقابتی مواجه شوید.

**BIOS (Basic Input/Output System)** یک نوع نرم‌افزار سیستم است که مسئول بوت کردن سیستم و مدیریت ارتباطات بین سیستم عامل و سخت‌افزارهای مختلف است. BIOS وظیفه اولیه خود را در شروع به کار سیستم انجام می‌دهد و معمولاً به زبان C نوشته می‌شود.

در نهایت، ترکیب این زبان‌ها و مفاهیم در توسعه نرم‌افزارهای embedded می‌تواند منجر به ساخت محصولات با کارایی بالا و ایمن شود.

## خط تولید نرم‌افزار (Software Product Line یا SPL):

خط تولید نرم‌افزار (Software Product Line یا SPL) یک رویکرد مهندسی نرم‌افزار است که بر اساس آن، مجموعه‌ای از محصولات نرم‌افزاری با ویژگی‌ها و قابلیت‌های مشابه، اما با تفاوت‌هایی مشخص برای پاسخ به نیازهای مختلف کاربران یا بازار، توسعه داده می‌شوند.

در این مدل، توسعه‌دهندگان ابتدا یک هسته (core) عمومی از ویژگی‌ها و قابلیت‌ها را طراحی می‌کنند که می‌تواند به عنوان پایه‌ای برای تولید محصولات مختلف مورد استفاده قرار گیرد. سپس این هسته با افزودن یا تغییر برخی ویژگی‌ها و قابلیت‌ها، به محصولات خاصی تبدیل می‌شود.

مزایای این رویکرد شامل کاهش زمان و هزینه توسعه، افزایش کیفیت نرم‌افزار و سهولت در نگهداری و به‌روزرسانی محصولات است. به همین دلیل، SPL به ویژه در صنایع بزرگ و پیچیده که نیاز به تولید محصولات متنوع دارند، بسیار مورد استفاده قرار می‌گیرد.

## تریدینگ اینترنتی (Internet Trading):

تریدینگ اینترنتی (Internet Trading) به فرآیند خرید و فروش انواع دارایی‌ها مانند سهام، فارکس، ارزهای دیجیتال و کالاها از طریق بسترهای آنلاین و اینترنتی اشاره دارد. این نوع تجارت به سرمایه‌گذاران این امکان را می‌دهد که به راحتی و در هر زمان و مکانی به بازارهای مالی دسترسی پیدا کنند.

تریدینگ اینترنتی معمولاً شامل موارد زیر است:

1. **پلتفرم‌های معاملاتی:** نرم‌افزارها یا وبسایت‌هایی که معامله‌گران می‌توانند از آن‌ها برای خرید و فروش دارایی‌ها استفاده کنند.
3. **تحلیل و تحقیق:** معامله‌گران معمولاً از تحلیل‌های فنی و بنیادی برای تصمیم‌گیری در مورد زمان و نحوه انجام معاملات استفاده می‌کنند.
4. **مدیریت ریسک:** استراتژی‌های مدیریت ریسک برای کاهش احتمال زیان و حفظ سرمایه اهمیت زیادی دارند.
5. **انواع معاملات:** شامل معاملات کوتاه‌مدت (Day Trading)، معاملات میان‌مدت (Swing Trading) و سایر استراتژی‌های معاملاتی.

تریدینگ اینترنتی به دلیل راحتی دسترسی و امکان انجام معاملات از هر نقطه‌ای برای بسیاری از سرمایه‌گذاران جذاب است، اما همچنین با ریسک‌ها و چالش‌های خاص خود نیز همراه است.

## پیش‌بینی Forecasting:

پیش‌بینی (Forecasting) به فرآیند تخمین یا پیش‌بینی وقایع آینده بر اساس داده‌ها و اطلاعات موجود اشاره دارد. این نوع تحلیل معمولاً در حوزه‌های مختلفی مانند اقتصاد، مدیریت، علوم اجتماعی، هواشناسی و ... استفاده می‌شود.

پیش‌بینی می‌تواند به دو دسته عمده تقسیم شود:

1. **پیش‌بینی کمی:** که بر اساس داده‌های عددی و آماری انجام می‌شود. این نوع پیش‌بینی از روش‌های ریاضی و مدل‌های آماری مانند رگرسیون، میانگین متحرک، و مدل‌های سری زمانی استفاده می‌کند.
  2. **پیش‌بینی کیفی:** که بر اساس نظر کارشناسان یا تحلیل‌های توصیفی انجام می‌شود. این نوع پیش‌بینی معمولاً در شرایطی که داده‌های کمی وجود ندارد یا در مورد مسائل نوظهور استفاده می‌شود.
- هدف اصلی پیش‌بینی، کمک به تصمیم‌گیری بهتر و برنامه‌ریزی برای آینده است. برای مثال، یک سازمان ممکن است از پیش‌بینی‌ها برای برآورد تقاضا، مدیریت موجودی یا برنامه‌ریزی تولید استفاده کند.

## تصمیم‌گیری (Decision Making)

تصمیم‌گیری (Decision Making) به فرایند انتخاب یک گزینه از میان چندین گزینه ممکن اطلاق می‌شود. این فرایند شامل شناسایی مشکل، جمع‌آوری اطلاعات، ارزیابی گزینه‌ها، و در نهایت انتخاب بهترین گزینه برای حل مشکل یا رسیدن به هدف است. تصمیم‌گیری می‌تواند به سطوح مختلفی از فردی تا گروهی انجام شود و تحت تأثیر عوامل مختلفی مانند تجربیات گذشته، دانش، بینش، و احساسات قرار دارد. این فرایند در جنبه‌های مختلف زندگی از جمله تجارت، سیاست، علوم اجتماعی و حتی تصمیمات روزمره مورد استفاده قرار می‌گیرد.

## معادلات غیر خطی (Non-linear Equations):

معادلات غیر خطی (Non-linear Equations) به معادلاتی اطلاق می شود که حداقل یکی از متغیرهای آنها به صورت غیر خطی در نظر گرفته شده است. به عبارت دیگر، در یک معادله غیر خطی، متغیرها نمی توانند تنها با استفاده از جمع و ضرب به یکدیگر مرتبط شوند. در این دسته از معادلات، روابط پیچیده تری وجود دارد که شامل توان های بالاتر از یک، جذرها، توابع مثلثاتی، نمایی و یا لگاریتمی می شود.

به عنوان مثال، معادله های زیر غیر خطی هستند:

1.  $x^2 + y^2 = 1$  (معادله دایره)

2.  $y = x^3 - 4x$  (یک معادله چندجمله ای)

3.  $\sin(x) + x^2 = 0$  (معادله ای با تابع مثلثاتی)

حل معادلات غیر خطی معمولاً پیچیده تر از معادلات خطی است و ممکن است نیاز به استفاده از روش های عددی یا الگوریتم های خاصی مانند روش نیوتن، روش رسم نمودار، یا روش های دیگر داشته باشد. در بسیاری از زمینه ها از جمله ریاضیات، فیزیک، مهندسی و اقتصاد، این نوع معادلات کاربردهای زیادی دارند.



## منطق (Logic)

منطق (Logic) به مطالعه اصول و قوانین استدلال و تفکر منطقی اشاره دارد. این علم به ما کمک می‌کند تا استدلال‌ها را به‌طور دقیق و منظم بررسی کنیم و از نادرستی‌ها و اشتباهات منطقی جلوگیری کنیم. منطق به دو دسته کلی تقسیم می‌شود: منطق صوری (formal logic) که به ساختار و قالب استدلال‌ها می‌پردازد، و منطق محتوایی (informal logic) که به بررسی محتوا و زمینه استدلال‌ها توجه دارد.

استدلال (Deduction) به فرآیند نتیجه‌گیری از پیش‌فرض‌ها یا قضایای موجود می‌باشد. در استدلال استنتاجی، اگر مقدمات صحیح باشند، نتیجه نیز باید صحیح باشد. به عبارت دیگر، در استدلال استنتاجی، نتیجه‌گیری از اطلاعات موجود و بر اساس قواعد منطقی انجام می‌شود. این نوع استدلال معمولاً در ریاضیات و فلسفه مورد استفاده قرار می‌گیرد.

به‌طور خلاصه، منطق و استدلال ابزارهایی هستند که به ما کمک می‌کنند تا تفکرات خود را ساماندهی کنیم و به نتایج دقیق و قابل اعتماد دست یابیم.