



دانشگاه صنعتی شریف  
دانشکده‌ی مهندسی کامپیوتر

درس مهندسی نرم‌افزار

عنوان:

## خودکار سازی فرآیند تست‌ها در نرم‌افزار

نگارش:

مصطفی قدیمی

سید علی‌رضا فاطمی جهرمی

استاد راهنما:

دکتر مهران ریواده

فروردین ۱۳۹۹

سلام

## چکیده

همان‌طور که بیش‌تر آدم‌های داخل صنعت نرم‌افزار می‌دانند، تفاوت‌های مشخص و واضحی بین تست دستی و تست اتوماتیک نرم‌افزار وجود دارد. تست دستی نیاز به زمان و تلاش‌هایی دارد تا مطمئن شویم که منبع کد نرم‌افزار همان کارهایی را که موظف است، به‌درستی انجام می‌دهد. علاوه‌براین، تست‌کننده‌های دستی همه‌ی چیزهایی را که پیدا می‌کنند ضبط و ذخیره می‌کنند. این کار شامل چک کردن فایل logها، سرویس‌های خارجی و پایگاه‌داده‌ای از خطاها می‌باشد. تست اتوماتیک همان‌طور که از نامش پیداست، بدون دخالت انسانی صورت می‌گیرد. در مقایسه با تست دستی، چون تست اتوماتیک با استفاده از ابزارهای خودکارسازی انجام می‌گیرد، وقت کم‌تری نیاز است تا تست‌ها انجام بگیرد اما زمان بیش‌تری برای نگهداری افزایش پوشش تست‌ها نیاز است. فرآیند تست دستی می‌تواند بسیار وقت‌گیر و تکراری باشد. تست خودکار برای پروژه‌های بزرگ، به دلیل نیاز به تست کردن یک ناحیه به‌طور مکرر، بسیار مناسب است.

**کلیدواژه‌ها:** خودکارسازی، تست، نرم‌افزار

## فهرست مطالب

۵	۱ مفهوم تست خودکار
۵	۱-۱ خودکارسازی تست چیست؟
۵	۲-۱ ملاک‌های انتخاب ابزار
۶	۳-۱ انواع چارچوب‌ها
۶	۴-۱ تست کیس
۶	۵-۱ طراحی تست کیس
۸	۲ پیاده‌سازی تست خودکار
۸	۱-۲ جامعیت تست خودکار
۹	۲-۲ تصوره‌های غلط رایج مربوط به تست خودکار
۱۰	۳-۲ تشریح چارچوب‌های تست خودکار
۱۳	۳ تست اتوماتیک در گیت‌لب
۱۳	۱-۳ گیت‌لب چیست؟

## فصل ۱

# مفهوم تست خودکار

در این فصل نکات کلی و برخی مفاهیم در خودکارسازی فرآیند تست نرم افزار به اختصار توضیح داده می شود.

### ۱-۱ خودکارسازی تست چیست؟

مفهوم فرآیند خودکارسازی تست ها به استفاده از ابزارهای خاصی برای کنترل اجرای تست ها و مقایسه ی نتایج آن ها با نتایج مورد انتظار می گویند. ابزارهای تست نه تنها به ما کمک می کنند تا تست های رگرسیون را انجام دهیم، بلکه کمک می کند تا تولید داده ها، نصب محصول، تعامل با رابط گرافیکی و... خودکار شود.

### ۱-۲ ملاک های انتخاب ابزار

برای خودکارسازی هر برنامه ای، باید به موارد زیر توجه کنیم:

- قابلیت داده محور بودن
- عدم وابستگی به پلتفرم
- قابلیت اجراپذیری و شخصی سازی
- اعلان ایمیل

- قابلیت استفاده از Version Control

- قابلیت عیب‌یابی و logging

- ...

### ۳-۱ انواع چارچوب‌ها

به طور معمول، چهار چارچوب خودکارسازی تست وجود دارد که در حین خودکارسازی برنامه‌ها انتخاب می‌شوند:

- چارچوب خودکارسازی داده‌محور

- چارچوب خودکارسازی کلیدواژه‌محور

- چارچوب خودکارسازی ماژولار

- چارچوب خودکارسازی ترکیبی

### ۴-۱ تست کیس

تست کیس یک سندی است که دارای مجموعه‌ای از داده‌های تست، پیش‌شرط‌ها، نتایج مورد انتظار و پس‌شرط‌ها است که برای یک سناریو تست مشخص طراحی شده تا یک نیاز مشخص را تایید و اعتبارسنجی کند.

### ۵-۱ طراحی تست کیس

در قسمت زیر، تکنیک‌های متداول طراحی تست کیس در مهندسی نرم‌افزار آورده شده است.

۱. استخراج تست کیس‌ها به‌طور مستقیم از نیازمندی‌های تعریف شده یا تکنیک طراحی جعبه سیاه. این تکنیک شامل:

- تجزیه و تحلیل مرز مقادیر

- افراز به قسمت‌های برابر

- جدول تصمیم‌گیری تست

- نمودار انتقال حالت

- تست موارد استفاده

است.

۲. استخراج تست کیس‌ها به طور مستقیم از ساختار یک جزء یا سیستم.

- پوشش Statement

- پوشش Branch

- پوشش Path

- تست LCSAJ

۳. استخراج تست کیس‌ها براساس تجربه‌ی آزمون‌گر روی سیستم‌های مشابه یا شهود.

- خطا در حدس زدن

- تست اکتشافی

## فصل ۲

# پیاده‌سازی تست خودکار

در این فصل به نحوه‌ی پیاده‌سازی تست خودکار می‌پردازیم.

## ۱-۲ جامعیت تست خودکار

موقع ارزیابی یک راه‌حل تست، داشتن یک ابزار که مناسب نیازهای همه‌ی اعضای تیم فرآیند تست، بسیار ضروری است. این افراد عبارتند از:

- **آزمون‌گرهای دستی:** ضبط و اجرای دوباره یک کار حیاتی برای آزمون‌گرهای دستی است. مخصوصاً آن دسته که در تست خودکار تازه‌کار هستند. توانایی استفاده از اسکریپت‌های ضبط شده یکسان با داده‌های مختلف می‌تواند به صورت دستی حین شناسایی و رفع مشکلات از طریق چندین محیط انجام شود.
- **مهندسان خودکارسازی:** برای مهندسان خودکارسازی، پشتیبانی قدرتمند از زبان‌های اسکریپت‌نویسی، ادغام با سیستم‌های CI و توانایی مقیاس‌پذیر کردن تست‌ها می‌تواند مهم باشد.
- **توسعه‌دهندگان:** پیاده‌سازی تست‌ها در فرآیند توسعه نیازمند توانایی برای انجام دادن تست‌ها درون IDE‌های مختلف

نظیر Visual Studio Code



## ۲-۲. تصورات غلط رایج مربوط به تست خودکار

### ۱. خودکارسازی وقت آزاد بیش‌تری فراهم می‌کند.

این تصور که تست خودکار وقت آزاد بیش‌تری به ما می‌دهد هم درست است و هم غلط. در تست دستی، بیش‌تر زمان برای اکتشاف و تست کردن کارایی اختصاص می‌یابد که باید به طور دستی به دنبال خطاها بگردیم. یک‌بار که این کار انجام می‌شود، آزمون‌گر دستی باید مکرراً این کارها را اجرا کند. اما استفاده از تست خودکار باعث می‌شود که این زمان به شدت کاهش یابد. کار آزمون‌گرهای تست خودکار این است که زمان‌شان را صرف کد زدن تست‌ها و اعمال بهبود روی آن تست‌ها به صورت مکرر است. در اصل، زمان صرف‌شده برای کارهای روزمره و تکراری آزمون‌گر دستی برای آزمون‌گر خودکار صرف تمرکز روی موضوعات بزرگ‌تر و مهم‌تری که شامل نرم‌افزار در حال توسعه می‌شود، می‌باشد.

### ۲. هزینه‌ی تست خودکار بسیار بالا است.

در ابتدا، سرمایه‌گذاری روی تست اتوماتیک ممکن است هزینه‌بر به نظر برسد، به خصوص اگر صاحب یک شرکت کوچک باشیم. اما تحلیل‌ها نشان داده است که با گذشت زمان، تست خودکار هزینه‌ی خودش را می‌پردازد (هزینه‌ی اضافی ندارد). همان‌طور که قبلاً اشاره شد، تست اتوماتیک شما را آزادتر می‌کند تا بتوانید روی مسائل مهم‌تر مانند نیازهای مشتری، کارایی و بهبودها تمرکز کنید. هم‌چنین تست خودکار هزینه‌ها و نیاز به بازبینی برنامه را کاهش می‌دهد. علاوه‌براین هر بار که منبع کد تغییر پیدا می‌کند، تست‌های نرم‌افزار می‌تواند تکرار شود. تکرار این تست‌ها به صورت دستی هزینه‌بر است و زمان زیادی را مصرف می‌کند اما تست خودکار می‌تواند بدون هزینه‌ی اضافی تکرار شود.

### ۳. تست اتوماتیک بهتر از تست دستی است.

واقعیت این است که بهتر و بدتری وجود ندارد. هر کدام از راهکارها خوبی‌ها و بدی‌های خودشان را دارند. تست دستی توسط یک نفر که جلوی کامپیوتر نشسته است به کمک بررسی دقیق logها، تلاش برای دادن ورودی‌های مختلف، مقایسه‌ی نتیجه با رفتار مورد انتظار و ضبط نتایج انجام می‌شود. این در حالی است که تست خودکار معمولاً وقتی نسخه‌ی اولیه‌ی نرم‌افزار توسعه داده شده است، انجام می‌گیرد.

### ۴. تست خودکار از تعامل انسانی جلوگیری می‌کند.

یکدیگر از تصورات غلط درباره تست خودکار این است که تعامل انسانی را تضعیف می‌کند. به طور دقیق‌تر، تست‌های اتوماتیک سریع‌تر از توانایی انسان می‌تواند آزمایش‌ها را انجام دهد با این مزیت که بدون خطای انسانی است. بنابراین این تصور قابل درک است.

البته این موضوع جایگزین روابط چهره به چهره و ملاقاتی که برای قسمت‌های توسعه نرم‌افزار ضروری است، نمی‌باشد. در عوض، با ارائه‌ی یک کانال دیگر ارتباطی، این جنبه را بهبود می‌بخشد. برای مثال ایمیل جای‌گزین تلفن نشد و صرفاً یک ابزار اضافی برای برقراری ارتباط بود.

## ۳-۲ تشریح چارچوب‌های تست خودکار

### ۱. چارچوب خودکارسازی داده‌محور

- **تعریف** با استفاده از چارچوب داده‌محور، داده‌های تست را از منطق اسکریپت جدا می‌شوند، به این معنی که مهندسان تست می‌توانند داده‌ها را در یک منبع جدا ذخیره کنند. خیلی اوقات، مهندسان تست در شرایطی قرار می‌گیرند که باید چندین بار ویژگی‌ها یا عمل‌کردهای یک برنامه را با مجموعه‌های مختلف داده تست کنند. در این موارد، بسیار مهم است که داده‌های آزمون در خود اسکریپت hard-code نباشند، این اتفاقی است که با یک چارچوب تست خطی یا مبتنی بر ماژولار رخ می‌دهد.
- تنظیم چارچوب تست داده‌محور به مهندسان تست این امکان را می‌دهد تا پارامترهای ورودی/خروجی را آزمایش کند و اسکریپت‌ها را از یک منبع داده خارجی مانند CSV، Text Files، Excel Spreadsheets، جداول SQL یا مخازن ODBC ذخیره کند.
- اسکریپت‌های تست به منبع داده خارجی متصل شده و در صورت لزوم داده‌های لازم را می‌خوانند.
- **مزایا** تست‌ها را می‌توان با مجموعه داده‌های متعدد اجرا کرد.
- با تغییر داده‌ها می‌توان به سرعت چندین سناریو را آزمایش کرد و از این طریق تعداد اسکریپت‌های مورد نیاز را کاهش داد.
- از hard-code کردن داده‌ها می‌توان جلوگیری کرد بنابراین هرگونه تغییر در اسکریپت‌های آزمایشی روی داده‌های مورد استفاده تاثیر نمی‌گذارد و برعکس.
- با اجرای سریع‌تر تست‌های بیشتر، در وقت صرفه‌جویی می‌شود.
- **معایب**
- برای استفاده صحیح از این چارچوب، به یک مهندس تست باتجربه و دارای مهارت در زبان‌های مختلف برنامه‌نویسی نیاز دارید. آن‌ها باید منابع داده خارجی را شناسایی و قالب‌بندی کنند و کدی را بنویسند (ایجاد توابع) که آزمایش‌ها را به صورت یک‌پارچه به منابع داده خارجی متصل می‌کند.
- تنظیم یک چارچوب داده‌محور زمان زیادی را می‌طلبد.

## ۲. چارچوب خودکارسازی کلیدواژه‌محور

- **تعریف** در یک چارچوب کلیدواژه‌محور، هر عمل‌کرد برنامه مورد تست در یک جدول با یک سری دستورالعمل به ترتیب متوالی برای هر تست که باید اجرا شود، قرار داده شده است. با روشی مشابه چارچوب داده‌محور، داده‌های تست و منطق اسکریپت در یک چارچوب کلیدواژه‌محور از هم جدا می‌شوند، اما این رویکرد آن را یک قدم فراتر می‌برد.

با این رویکرد، کلمات کلیدی نیز در یک جدول داده خارجی ذخیره می‌شوند و باعث می‌شوند که آن‌ها از ابزار تست خودکار که برای اجرای تست‌ها استفاده می‌شود، مستقل شوند. کلمات کلیدی بخشی از یک اسکریپت است که نمایانگر اقدامات مختلفی است که برای تست GUI یک برنامه انجام می‌شود. این موارد می‌توانند به سادگی با "click"، یا "login" یا برچسب‌های پیچیده مانند "clicklink"، یا "verifylink" برچسب‌گذاری شوند.

در جدول، کلمات کلیدی به صورت مرحله به مرحله با یک آبجکت مرتبط یا بخشی از رابط کاربری که عمل در آن انجام می‌شود، ذخیره می‌شوند. برای این‌که این رویکرد به درستی کار کند، یک مخزن آبجکت مشترک برای یافتن اقدامات مرتبط با هر آبجکت لازم است.

جدول

- **مزایا** دانش برنامه‌نویسی حداقل مورد نیاز است.

یک کلمه کلیدی واحد را می‌توان در چندین اسکریپت تست استفاده کرد، بنابراین کد قابل استفاده مجدد است.

اسکریپت تست را می‌توان مستقل از نرم‌افزار تحت آزمون ساخته شده است.

- **معایب** هزینه اولیه تنظیم چارچوب زیاد است. وقت‌گیر و پیچیده است. کلمات کلیدی نیاز به تعریف دارند و باید مخازن/کتابخانه‌های آبجکت تنظیم شود.

شما به یک مهندس تست با مهارت‌های اتوماسیون بالا نیاز دارید.

هنگام مقیاس دادن به عملیات تست، کلمات کلیدی می‌توانند به دردت تبدیل شوند. شما نیاز به ادامه ساخت مخازن و جداول کلمات کلیدی دارید.

## ۳. چارچوب خودکارسازی ماژولار

- **تعریف** اجرای یک چارچوب ماژولار به مهندسين تست نیاز دارد تا برنامه مورد آزمایش را به واحدها، کارکردها یا بخش‌های جداگانه تقسیم کنند که هر یک از آن‌ها به صورت جداگانه تست می‌شوند. پس از شکستن برنامه

به ماژول‌های جداگانه، یک اسکریپت تست برای هر قسمت ایجاد می‌شود و سپس برای ساخت تست‌های بزرگ‌تر به صورت سلسله مراتبی ترکیب می‌شود. این مجموعه‌های تست بزرگ‌تر شروع به نمایش تست کیس‌ها می‌کنند.

یک استراتژی اساسی در استفاده از چارچوب ماژولار ایجاد یک لایه انتزاع است، به طوری که هرگونه تغییر در بخش‌های خاص بر ماژول اصلی تاثیر نمی‌گذارد.

- **مزایا** در صورت ایجاد هرگونه تغییر در برنامه، فقط باید ماژول و آن اسکریپت تست مرتبط با آن تغییر یابد، به این معنی که لازم نیست بقیه تست‌ها را تغییر دهید و می‌توانید آن‌ها را دست نخورده بگذارید. ایجاد موارد آزمایشی تلاش کم‌تری می‌کند زیرا می‌توان از اسکریپت‌های آزمون برای ماژول‌های مختلف استفاده مجدد کرد.

- **معایب** داده‌ها از آن‌جا که تست‌ها به طور جداگانه انجام می‌شوند به صورت hard-code در تست‌ها قرار دارند، بنابراین نمی‌توانید از چندین مجموعه داده استفاده کنید. دانش برنامه نویسی برای تنظیم این چارچوب نیاز است.

#### ۴. چارچوب خودکارسازی ترکیبی

- **تعریف** مانند اکثر فرایندهای تست امروز، چارچوب‌های تست خودکار شروع به یک پارچه‌سازی و هم‌پوشانی با یک‌دیگر می‌کنند. یک چارچوب ترکیبی، ترکیبی از هر یک از چارچوب‌های قبلی است که برای استفاده از مزایای برخی و کاهش نقاط ضعف برخی دیگر تنظیم شده است. هر برنامه متفاوت است، بنابراین باید فرایندهای تست آن‌ها نیز متفاوت باشد. با حرکت تیم‌های بیش‌تر به یک مدل چابک، تنظیم یک چارچوب انعطاف‌پذیر برای تست خودکار بسیار مهم است. یک چارچوب ترکیبی می‌تواند به آسانی با نیاز ما سازگار شود تا بهترین نتایج تست را به دست آوریم.

## فصل ۳

# تست اتوماتیک در گیت لب

در این فصل به پیاده‌سازی تست اتوماتیک در سرویس گیت لب می‌پردازیم.

### ۱-۳ گیت لب چیست؟