MERHABA!

ROBOT_DREAMS BİR PROGRAMLAMA, ANALİTİK VE VERİ BİLİMİ İÇİN ONLINE EĞİTİM PLATFORMUDUR.

✦ AŞAĞIDAKİLERİ ÖĞRETİYORUZ

KODLAMA
TEST
YAZILIM
ANALİZ

GELECEĞİN OKULU

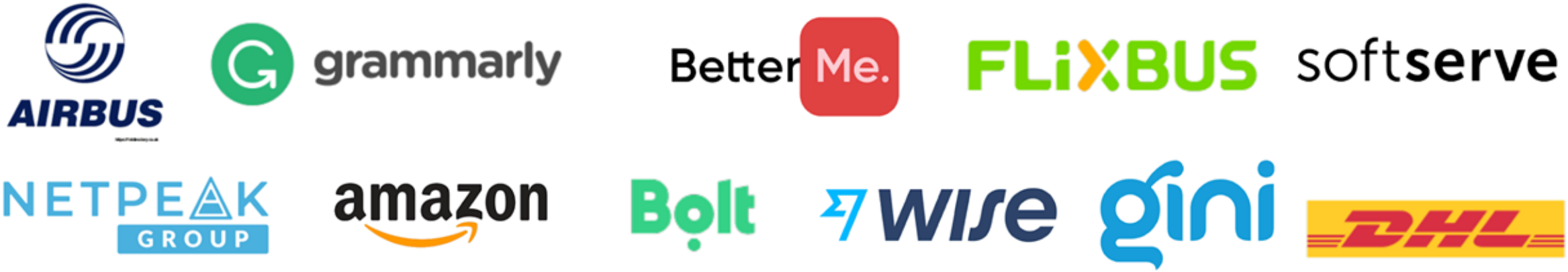
[HAKKIMIZDA]

robot_dreams online bir BT okuludur. Veri analitiği, programlama, test etme, Veri Bilimi ve Makine Öğrenimi gibi konularda kurslar bulunmaktadır.

Bilişim sektörünün en iyi uzmanları okulumuzda eğitim vermekte, ayrıca tüm mezunlarımız başarılı bir şekilde iş bulmaktadır.

>>>

Eğitmenlerimizin çalıştığı şirketlerden bazıları;



[Test Otomasyon Eğitimi]

Kurs hakkında

30 Kasım 2023 - 21 Mart 2024

ÇEVRİMİÇİ KURS | 29 DERS | 20 ÖDEV | 2 WORKSHOP | 1 GERİ BİLDİRİM OTURUMU

Test Otomasyon Eğitimi

Sektörel olarak test otomasyon eğitimi, günümüzün hızla gelişen teknoloji ortamında büyük bir önem taşımaktadır. Teknolojinin hızla ilerlemesi ve yazılım projelerinin karmaşıklığının artması, manuel test süreçlerinin yetersiz kalabileceği anlamına gelmektedir. Bu nedenle test otomasyonu, yazılım ürünlerinin daha hızlı, güvenilir ve tekrarlanabilir bir şekilde test edilmesini sağlayan bir gereklilik haline gelmiştir.

Test Otomasyonu eğitimi boyunca katılımcılar yazılım geliştirmenin temellerini, bir testin nasıl otomatize edileceğini öğrenecek ve testleri otomatize ederken ihtiyaç duyulacak tüm teknik donanımı teorinin yanında pratikte uygulayarak edinecekler.

Serkan Cura

Test Otomasyon Takım Lideri, Jotform

- Sistem Analisti/Tester olarak adımını attığı yazılım test alanının her alanında yaklaşık 11 yıldır görev almaktadır.
- Digitürk, Intertech, Getir gibi alanının öncü firmalarında Test Uzmanlığı, QA Mentör ve Test Otomasyon Mühendisi pozisyonlarında çalışmıştır ve şu anda Otomasyon Takım Liderliği yapmaktadır.
- C#, Java, JavaScript tabanlı Yazılım Test Frameworkleri ile çalışıp bunların detayları konusunda tecrübe edinmiştir.
- ISTQB CTFL sertifikasına sahiptir.
- Çevik Bir Dünyada TMMi®, Turkey Software Quality Report 2020-2021, Turkey Software Quality Report 2021-2022 yayınlarına katkı sağlamıştır.
- TestIstanbul Strateji Komitesi Üyeliği yapmıştır ve Live2Test konferansında panelist olarak yer almıştır.

[BU KURS KİMLER İÇİN?]

Yazılım Test Uzmanları

Otomasyon üzerine uzmanlaşmak isteyen, manuel test geliştirme süreçlerine hakim olanlar.

Yazılım Geliştiriciler

Test teorisi hakkında bilgi sahibi olan, test otomasyon mühendisi olarak çalışmak ya da dahil oldukları projelerinde otomasyondan faydalanmak isteyenler (test otomasyon sürecine dahil olmuş olamamaları gerekmektedir.)

Katılımcılar için ekstra gereklilikler

- Manuel olarak testleri gerçekleştirebilme, test teorisini anlama
- Web'in nasıl çalıştığını, URL'nin ne olduğunu anlama
- İlgili dokümantasyon yazabilme
- IT süreçlerini ve iş akışının nasıl yapılandırıldığını anlama
- Temel SQL bilgisine sahip olma

[KURS PROGRAMI] modül_01 OTOMASYON İÇİN GEREKLİ ARAÇLAR

Ders 1
- **Giriş**
- Test otomasyonunun amacı (otomasyon testlerinin manuel testlere karşı avantajları ve dezavantajları)
- Test Otomasyon türleri (Unit, Contract, API, Kullanıcı Arayüzü, performans)
- Test piramidi ve otomasyon bağlamında kullanımı
- Pratikte en sık karşılaşılan işletim sistemleri nelerdir

Ders 2
- **Git**
- Sürüm Kontrol Sisteminin (VCS) amacı
- Commit nedir
- Git deposunun yapısı (ana/dalga getir, uzak/yerel sürüm)
- Pull request nedir
- Çakışma (conflict) nedir ve nasıl çözülür

Ders 3
- **Docker**
- Docker'ın avantajları
- Docker imajı nedir
- Docker kayıt defteri (registry) nedir
- Docker Desktop'ın kurulumu
- Testçinin iş akışında kullanımı
- Veritabanını Docker'da başlatmak

Ders 4
- **Veri Depolama ve İletişim Formatları**
- XML
- JSON
- YAML
- TOML
- Modül sonu geri bildirimlerinin yapılması

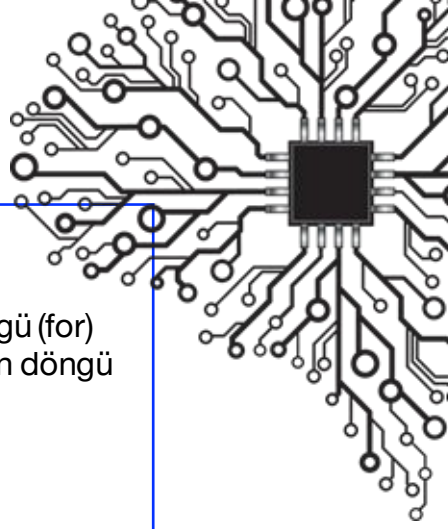
Dersten sonra kazanımlar;
- Test otomasyonunun gerekliliğini anlıyor ve savunabilirim.
- Hangi işlemlerin otomasyonla yapılması gerektiğini ve hangilerinin kaynak israfı olduğunu biliyorum.
- Hangi test türlerine en fazla kaynak harcaması gerektiğini anlıyorum.

Dersten sonra kazanımlar;
- Sürüm kontrol sisteminin nasıl çalıştığını anlıyorum.
- Git aracılığıyla diğer insanlarla işbirliği yapabiliyorum.

Dersten sonra kazanımlar;
- Containerization kavramını anlıyorum.
- Halka açık ve özel Docker imajlarının nerede saklandığını biliyorum.
- Docker'da uygulamaları başlatmayı biliyorum.

Dersten sonra kazanımlar;
- Temel veri temsili formatlarını okuyup yazabiliyorum.
- Hangi durumda hangi formatın daha iyi olduğunu biliyorum.

[KURS PROGRAMI] modül_02 JAVA



Ders 5

- **Hello World**
- JRE/JDK nedir, JDK'nın kurulumu
- Entegre Geliştirme Ortamı (IDE) nedir, neden gereklidir, kurulumu
- Java'da her şey nesnedir
- Temel veri türleri
- public static void main() {...} metodu
- System.out.println()
- String veri türü
- Dize biçimlendirme
- Nesnelerin alanları ve metotları

Ders 6

- **Operatörler ve Koşullar**
- Doğrudan atama "="
- Aritmetik operatörler
- Kısa atama "+=" "-=" "*=" "/=" "%="
- Artırma/azaltma "++" "--"
- Karşılaştırma "==" "!=" ">" "<" ">=" "<="
- Mantıksal operatörler "&&" "||" "!" “^”
- Basit if koşulu ve operatörlerle kullanımı
- Boolean tipinde basit if koşulu
- if-else, if-else if - else dallanma yapısı
- Üçlü operatör (ternary operator)
- Çoklu dallanma switch yapısı

Ders 7

- **Döngüler**
- Belirli sayıda yinelemeli döngü (for)
- Elemanları sıralı olarak gezen döngü (for each)
- Koşullu döngü (while)
- Sonlu döngü (do while)
- Sonsuz döngü
- Döngüyü "return" ile sonlandırma
- Döngüyü "break" ile sonlandırma
- Geçişi bir sonraki yinelemeye "continue" ile sağlama

Dersten sonra kazanımlar;

- Geliştirme ortamı kurulumu yapabiliyorum.
- Java'da nesne modelinin nasıl çalıştığını anlıyorum.
- Temel veri türlerinin ne olduğunu biliyorum.
- Basit bir uygulama oluşturup derleyebilir ve çalıştırabilirim.

Dersten sonra kazanımlar;

- Temel atama ve değiştirme operatörlerini kullanabiliyorum.
- Değişkenlerin nasıl karşılaştırılacağını biliyorum.
- Mantıksal operatörleri kullanarak karmaşık ifadeler oluşturabiliyorum.
- Koşullar temelinde algoritma oluşturabiliyorum.
- Bütün dallanma yapılarının avantajlarını ve dezavantajlarını biliyorum.
- Okunabilir ve verimli koşullar yazabiliyorum.

Dersten sonra kazanımlar;

- Yinelemelere dayalı algoritmalar oluşturabiliyorum.
- Verileri kısmen döngülerle işleyebiliyorum.
- Döngüler içinde mantıksal yapılar kurabiliyorum.
- Veri dizilerini işleyebiliyorum.

[KURS PROGRAMI] modül_02 JAVA

Ders 8

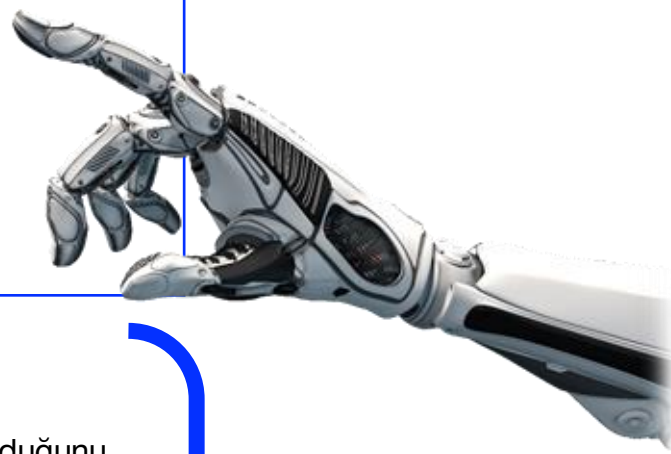
- **Otomasyon için Sıkça Kullanılan Veri Yapıları**
- Tek ve çok boyutlu diziler
- Kolleksiyonlar (ArrayList, LinkedList, HashSet)
- HashMap

Ders 9

- **Nesne Alanları ve Metotları**
- Alan türünü deklare etme, değer atama ve atamama
- Erişim belirleyicileri (public, private, protected)
- Sabitler (constants)
- Statik alanlar
- Metot adlandırma
- Metot imzası
- Dönüş tipi
- Yapıcı metotlar (constructor)
- Metotların aşırı yüklenmesi (overloading)
- Erişim belirleyicileri
- Statik metotlar
- Nesneyi bir değer olarak bir metoda aktarma

Ders 10

- **ENUM Numaralandırılmış Türü**
- ENUM türü nedir
- Sabitlerden farklılıklar
- ENUM ve switch yapısı
- ENUM sınıf yapısı
- ENUM dönüştürme



Dersten sonra kazanımlar;

- Java'da veri depolamak için hangi yapıları kullanmam gerektiğini biliyorum.
- Hangi durumda hangi veri yapısını seçmem gerektiğini biliyorum.
- Veriyi ekleme, çıkarma ve değiştirme becerilerine sahibim.

Dersten sonra kazanımlar;

- Java'da tam teşekküllü sınıflar oluşturabiliyorum.
- Erişim belirleyicilerinin verilere nasıl etki ettiğini biliyorum.
- Sabitleri nasıl tanımlayacağımı biliyorum.
- Belirli görevleri yerine getiren tam teşekküllü sınıflar oluşturabiliyorum.
- Prosedürel programlamadan nesne tabanlı programlamaya geçmeye hazırım.

Dersten sonra kazanımlar;

- ENUM türünün ne olduğunu ve hangi durumlar için kullanılması gerektiğini biliyorum.
- ENUM'ları kullanarak mantıksal dallanma yapısı oluşturabiliyorum.
- ENUM sınıf yapısını anlayabiliyorum.

[KURS PROGRAMI] modül_02 JAVA

Ders 11

- **Kapsülleme, Soyutlama ve Arayüzler, Miras ve Çok Biçimlilik**
- Erişim belirleyicilerin gözden geçirilmesi
- Gizli alanlara/metotlara erişim
- "this" anahtar kelimesi
- Soyut sınıflar
- Soyut metotlar
- Arayüzler
- Sınıf mirası
- Soyut sınıf mirası
- Arayüz uygulamaları
- Miras alınamayan sınıflar

Dersten sonra kazanımlar;

- Object-Oriented Programming (OOP) modeline uygun sınıflar oluşturabilirim.
- Mantıklı ve güvenli algoritmalar oluşturabilirim.
- Soyut sınıfların ve arayüzlerin neden var olduğunu anlıyorum.
- Sınıf mirasını ve arayüz uygulamalarını anlayabiliyorum.
- Daha az kodla çok katmanlı programlar oluşturabilirim.
- Sınıf mirasını ve arayüz uygulamalarını anlayabiliyorum.
- Kısmi yeniden kullanımı gerçekleştirebilen kod yazabilirim.
- Miras alınamayan sınıfları nasıl oluşturacağımı biliyorum.

Ders 12

- **Hata İşleme**
- Hata türleri
- Hata mirası
- Hata işleme
- Çok katmanlı hata işleme
- Hata ile iletişim aracı olarak hata
- try-with-resource yaklaşımı

Dersten sonra kazanımlar;

- Hataları düzgün bir şekilde işleyebilirim.
- Hataları geri bildirim aracı olarak kullanabilirim.

Ders 13

- **Dosya İşlemleri**
- Dosya sistemine erişim
- Dosya okuma ve yazma
- Kapanabilir kaynaklar
- Okuma ve yazma tamponlama

Dersten sonra kazanımlar;

- Java koduyla dosya sistemini etkili bir şekilde kullanabilirim.
- Dosya okuma ve yazma işlemlerini gerçekleştirebilirim.
- Kapanabilir kaynakları nasıl işleyeceğimi biliyorum.

Ders 14

- **Annotasyonlar, Generic Tipler ve Akışlar (Streams)**
- Annotasyonların genel bakışı
- Generic tiplerin genel bakışı
- Akışlar (Streams) ve lambda ifadelerinin genel bakışı. Fonksiyonel arayüzler

Dersten sonra kazanımlar;

- Annotasyonların ne olduğunu ve kod içinde nasıl kullanıldığını biliyorum.
- Generic tiplerin ne olduğunu ve kod içinde nasıl kullanıldığını biliyorum.
- Generic bir yöntem ve sınıf oluşturabilir ve kullanabilirim.
- Akışları ve lambda ifadelerini kullanarak koleksiyonlarla çalışabilirim.

[kurs hakkında] [eğitmen] [kimler için] [program] [araçlar] [sonuçlar] [format] [iletişim]

[KURS PROGRAMI] modül_02 JAVA

Ders 15

- **Java Uygulaması Tasarımı**
- Kötü kodun işaretleri. Code smells
- Kompozisyon ve agregasyon
- Tasarım Desenleri: Singleton ve Builder

Ders 16

- **Maven**
- Maven nedir
- pom.xml dosyasının yapısı
- Maven Merkezi Deposu
- Özel depolar
- Maven yürütme aşamaları
- Eklentiler (plugins)
- Proje yapısı

Ders 17

- **Java ile Veritabanı Çalışma (JDBC ve JDBI) - PostgreSQL Örneği**
- Veritabanına bağlanma
- Basit sorguları çalıştırma
- Statement kullanarak sorguları çalıştırma
- SQL Enjeksiyonu
- PreparedStatement kullanarak sorguları çalıştırma
- JDBI'nın genel bakışı
- Veritabanı işlemleri için mimari. Sorgu sonuçlarını nesnelerle işleme alma

Ders 18

- **JUnit/TestNG Test Çerçeveleri ve Veri Odaklı Test (Data Driven Testing)**
- JUnit'un genel bakışı
- TestNG nedir
- TestNG Annotasyonları ([Before/After] suite, groups, class, test, method)
- TestNG XML dosyaları
- Yapılandırma dosyalarıyla çalışma
- TestNG parametreli testler
- Testler için veri sağlayıcıları
- Modül sonu geri bildirimlerinin yapılması

Dersten sonra kazanımlar;

- Kötü kodun işaretlerini anlayabilirim.
- Kompozisyon ve agregasyonun ne olduğunu biliyorum.
- Singleton ve Builder tasarım desenlerini uygulayabilirim.

Dersten sonra kazanımlar;

- Maven kullanarak bağımlılıkları yönetmeyi biliyorum.
- Maven'i kullanarak proje derlemesini yapılandırabilirim.
- Maven projelerinin standart yapısını biliyorum.

Dersten sonra kazanımlar;

- Java ile temel veritabanı işlemlerini gerçekleştirebilirim.
- PreparedStatement kullanarak parametreli sorguları oluşturabilirim.
- JDBI kütüphanesiyle kod yapısını anlayabilirim.
- CRUD işlemlerini JDBI kütüphanesiyle gerçekleştirebilirim.

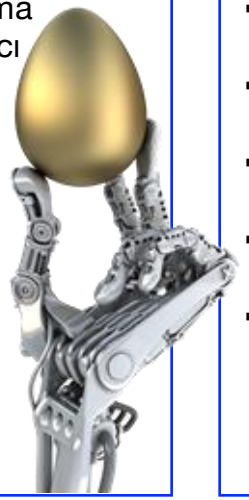
Dersten sonra kazanımlar;

- Testlerin kodla nasıl etkileşime girdiğini anlıyorum.
- Çalışmaya hazır testleri yazabiliyorum.
- Test ortamını nasıl hazırlayacağımı biliyorum.
- Testleri yürütmek için gruplar oluşturabilirim.
- Yapılandırma dosyalarıyla çalışabilirim.
- Aynı kodu farklı veri setleriyle çalışan testler olarak yazabilirim.
- Anlaşılır bir test yapılandırması oluşturabilirim.

[KURS PROGRAMI] modül_03 WEB ARAYÜZÜ TESTİ

Ders 19

- **Selenium WebDriver**
- Farklı tarayıcıları başlatma
- Chrome headless tarayıcı
- Elemanları bulma
- Elemanlarla etkileşim



Dersten sonra kazanımlar;

- Testlerin yürütülmesi için farklı tarayıcıları başlatabilirim.
- Sayfa elemanlarını bulma temellerini anlıyorum.
- Sayfa elemanlarıyla nasıl etkileşimde bulunacağımı biliyorum.

Ders 20

- **CSS Seçicileri (Selectors)**
- Özniteliklere (id ve class) göre eleman bulma
- Birden fazla özniteliğe göre eleman bulma
- Yerleşim yapısına göre eleman bulma
- Çoklu iç içe elemanlardan seçme
- Kısmi eşleme ile eleman bulma (^=, $=, *=,)

Dersten sonra kazanımlar;

- CSS seçicileriyle özniteliklere göre eleman bulma yeteneğine sahibim.
- Çok sayıda iç içe geçmiş eleman arasından seçim yapabilirim.
- Kısmi eşleme ile elemanları bulma yeteneğine sahibim.

Ders 21

- **XPath Lokatörleri**
- XPath nedir
- Mutlak ve göreli yol
- XPath bulma ve doğrulama için ChroPath
- Temel XPath
- Eleman tipini belirtmeden eleman bulma
- Parçalı eşlemeyle eleman bulma (contains)
- Başlangıçla eleman bulma (starts-with)
- Eleman metnini temel alarak eleman bulma
- XPath kullanarak dinamik elemanları bulma yöntemleri

Dersten sonra kazanımlar;

- Farklı özelliklere göre elemanları bulmak için XPath seçicileri oluşturabilirim.
- Dokümanın dinamik yapısında elemanları bulma yeteneğine sahibim.
- Google Chrome eklentisi kullanarak oluşturduğum XPath seçicilerini kontrol edebilirim.

Ders 22

- **Selenide**
- Selenide'da WebDriver yapılandırması
- Elemanlar ve eleman koleksiyonlarıyla çalışma
- Varsayılan bekleme ve bekleme işlemleri
- Selenide'da assertion (doğrulama) işlemleri
- AssertJ ile FluentAssertion kullanımı ve assertion işlemleri
- Page Object tasarım deseni kullanarak test yapısı

Dersten sonra kazanımlar;

- Selenide ile tarayıcıyı başlatabilirim.
- Selenide'da elemanlarla etkileşime geçebilirim ve doğrulama yapabilirim.
- Selenide kullanarak Web UI testleri yazabilirim.
- Testlerin ve sayfa nesnelerinin tasarım deseni olan Page Object yapısını biliyorum.

[KURS PROGRAMI] modül_04 API TESTİ

Ders 23

- **Cucumber**
- Yazılım geliştirmede BDD yaklaşımı
- BDD testlerinin yapısı ve Gherkin dili
- Cucumber'da veri geçişinin sorunu ve çözümü
- Parametreli ve veri tabanlı testler Gherkin dili ile
- Testleri Idea ve Maven ile nasıl çalıştıracağınız
- Cucumber ile paralel test çalıştırma ayarları
- Modül sonu geri bildirimlerinin yapılması

Dersten sonra kazanımlar;

- Cucumber ile yazılan testlerin yapısını biliyorum.
- Basit, parametreli ve veri tabanlı testler yazabilirim.
- Testleri Idea ve Maven aracılığıyla çalıştırabilirim.
- Cucumber ile paralel test çalıştırma ayarlarını yapabilirim.

Ders 24

- **HTTP + Postman**
- Restful yaklaşım (+ GraphQL hakkında kısa bir bilgi)
- HTTP yöntemleri (GET, POST, PUT, DELETE, HEAD, OPTIONS)
- Kaynak Endpoint
- Postman
- SWAGGER. Örnek olarak https://petstore.swagger.io/

Dersten sonra kazanımlar;

- HTTP'nin nasıl çalıştığını biliyorum.
- Hangi durumlarda hangi yöntemin kullanılacağını anlıyorum.
- Kaynak tabanlı yaklaşımın ne anlama geldiğini kavradım.
- Postman aracılığıyla HTTP endpoint'leri ile nasıl çalışacağımı öğrendim.
- Basit API testlerini Postman aracılığıyla yazabilirim.

Ders 25

- **REST Assured**
- Basit Java'da HTTP isteği ve REST Assured arasındaki farklar
- GWT modeli (Given-When-Then modeli)
- Sorgu parametreleri
- Yanıtın çıkarılması
- Yanıtın parçalarıyla çalışma
- Doğrulama (Validation)
- POST isteği

Dersten sonra kazanımlar;

- REST Assured ile ardışık HTTP istekleri gerçekleştirebilirim.
- Dönen verilerin doğruluğunu doğrulama yeteneğine sahibim.
- Given-When-Then modelinin ne anlama geldiğini anlıyorum.

Ders 26

- **Awaitility**
- Thread.sleep()'in sorunları
- Minimum ve maksimum bekleme süresi
- İstek gecikmesi (Delay)
- İstek aralığı (Interval)
- Dinamik istek aralıkları
- Hataların işlenişi
- Özel alanlara erişim
- Modül sonu geri bildirimlerinin yapılması

Dersten sonra kazanımlar;

- Dinamik olarak değişen durumları ve verileri içeren sistemlerde çalışabilen API'ler yazabilirim.
- Sistemdeki gereksiz yüklemeyi önlemek amacıyla doğru gecikme ve bekleme süresini ayarlayabilirim.

[KURS PROGRAMI]

modül_05 TEST RAPORLAMA

Ders 27

- **Test Raporlama ve Gelişmiş Test Raporlama**
- Log4j günlükleme
- Günlükleme seviyeleri
- log4j.properties dosyasıyla yapılandırma
- PatternLayout sınıfı ile log formatlama
- Gelişmiş test raporu ReportNG
- Allure genel bakış
- Allure Annotasyonları
- TestNG ve Maven projeleri için Allure yapılandırması
- Rapora dosya eklemek
- Modül sonu geri bildirimlerinin yapılması

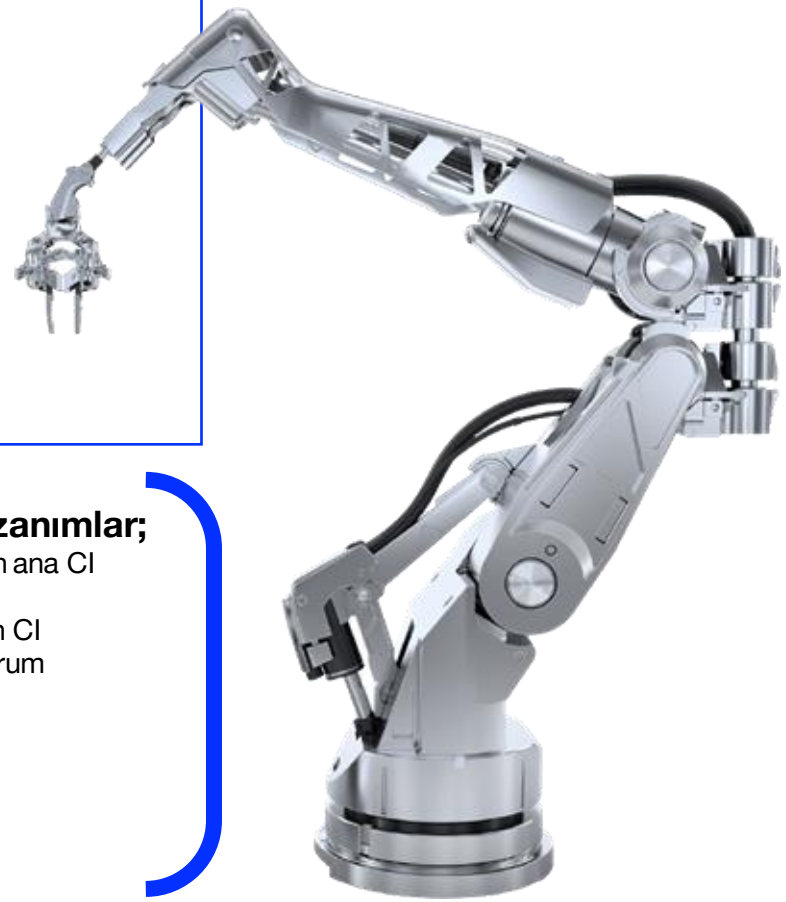
Dersten sonra kazanımlar;

- Test sonuçlarını düzgün bir şekilde görüntüleyebilirim.
- Hataların nerede meydana geldiğini anlayabilirim.
- Herkesin anlayabileceği bir test raporu yapısını nasıl yapılandıracağımı biliyorum.
- Testlere açıklama ve test durum kodu gibi meta bilgileri ekleyebilirim.
- Allure'u yapılandırabilirim.
- Raporlara dosya ekleyebilirim.

modül_06 Continuous Integration

Ders 24

- **CI Sistemlerine genel bakış**
- GitHub workflows
- GitLab pipelines
- CircleCI
- Jenkins



Dersten sonra kazanımlar;

- Uygulamada kullanılan ana CI sistemlerini biliyorum
- SaaS CI ile barındırılan CI arasındaki farkı anlıyorum

[KURS PROGRAMI] modül_07 PROJE ÇALIŞMASI

Ders 29

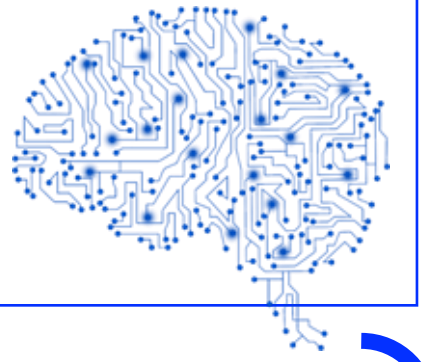
- **Proje Başlangıcı**
- Proje gereksinimlerinin anlaşılması
- Proje tasarımı ve yapılandırması
- Proje kapsamının belirlenmesi
- İş akışının oluşturulması
- Projenin geliştirilmesi ve kodlaması
- Projenin test edilmesi
- Projenin sonlandırılması ve teslim edilmesi
- Proje deneyimlerinin paylaşılması
- Modül sonu geri bildirimlerinin yapılması

Dersten sonra kazanımlar;

- Testlerin yürütülmesi için farklı (Ben ...):
- Bir projenin gereksinimlerini anlayabilirim.
- Proje tasarımı ve yapısını oluşturabilirim.
- Proje kapsamını belirleyebilirim ve iş akışını oluşturabilirim.
- Bir projeyi geliştirebilir, kodlayabilir ve test edebilirim.
- Proje sürecini yönetebilirim ve tamamlayabilirim.
- Proje deneyimlerimi ve sonuçlarını paylaşabilirim.

Ders 30

- **Geri Bildirim Oturumu**
- Kurs hakkında genel geri bildirimler.
- Gelecekte kariyerlerinde ve iş mülakatlarında karşılaşacakları zorluklar



Dersten sonra kazanımlar;

- Kurs esnasında yaşadığım deneyimleri paylaşırım.
- Gelecekte yaşayacağı sorunları daha net görürüm.

Workshop 1

- **Jenkins Yapılandırması**
- Kurulum
- Bildirime dayalı yapılandırma yoluyla bir pipeline oluşturma

Kazanımlar;

- Jenkins görevlerinin bildirimsel bir yaklaşımla nasıl tanımlandığını biliyorum
- Jenkins'i basit görevleri çalıştıracak şekilde yapılandırabilirim.

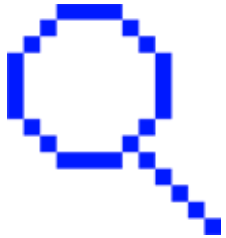
Workshop 2

- **Jenkins ile Testlerin çalıştırılması**
- Git deposuna erişim
- Görev yürütme artefactleri (test raporu)

Kazanımlar;

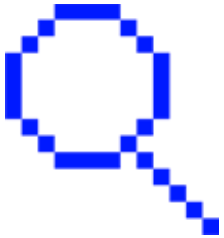
- Jenkins'i Git deposundaki kodla çalışacak şekilde nasıl yapılandıracağımı biliyorum
- Artefect'lerin ne olduğunu, onlara nasıl erişilebileceğini ve bunlara neden ihtiyaç duyulduğunu biliyorum

[ÖDEV/PROJELER]



- **#01** Kendinize ait herkese açık bir Repository Oluşturun.
- **#02** Docker üzerinden çalışan ve bir Database Yönetim Uygulaması aracılığı ile erişilebilen bir image oluşturun.
- **#03** XML, YAML ve JSON formatlarında örnek personel nesnesi oluşturun.
- **#04** Bir IK uygulaması için kullanılabilecek şirket departmanı için bir maaş hesaplayan kodu yazın.
- **#05** Bir dizi metni dolaşan bir algoritma oluşturun.
- **#06** Güneş sistemindeki tüm gezegenleri listeleyen bir Gezegen sınıfı tasarlayın.
- **#07** Bir Departman isminde sınıf oluşturup takım bileşenlerini yazılımsal olarak ifade edin
- **#08** Verilen bir yöntemin, giriş parametrelerine ve dış faktörlere bağlı olarak belirli hata türlerini döndürebilen bir yöntem olduğu bir senaryo oluşturun.
- **#09** Generic bir alan içeren bir sınıf ve bu alanı yazdıran bir metod yazın.
- **#10** Basit bir Singleton ve Builder tasarım deseni gerçekleştiren bir örnek yazın.
- **#11** Docker'da yerel bir veritabanına test metodundan bağlanın, basit sorguları PreparedStatement kullanarak çalıştırın.
- **#12** Test metoduyla uzaktaki bir veritabanına bağlanın ve CRUD işlemlerini JDBI kütüphanesi kullanarak gerçekleştirin.

[ÖDEV/PROJELER]



- **#13**
  Daha önce yazdığınız muhasebe maaş/prim hesaplama sınıfı için bir test yazın. Testi yapılandırmak için yapılandırma dosyası mekanizmasını ekleyin. https://demoqa.com/elements adresine gidip "Buttons" seçeneğini tıklayın, ardından "Click Me" düğmesini tıklayın ve görünen mesajı okuyun.

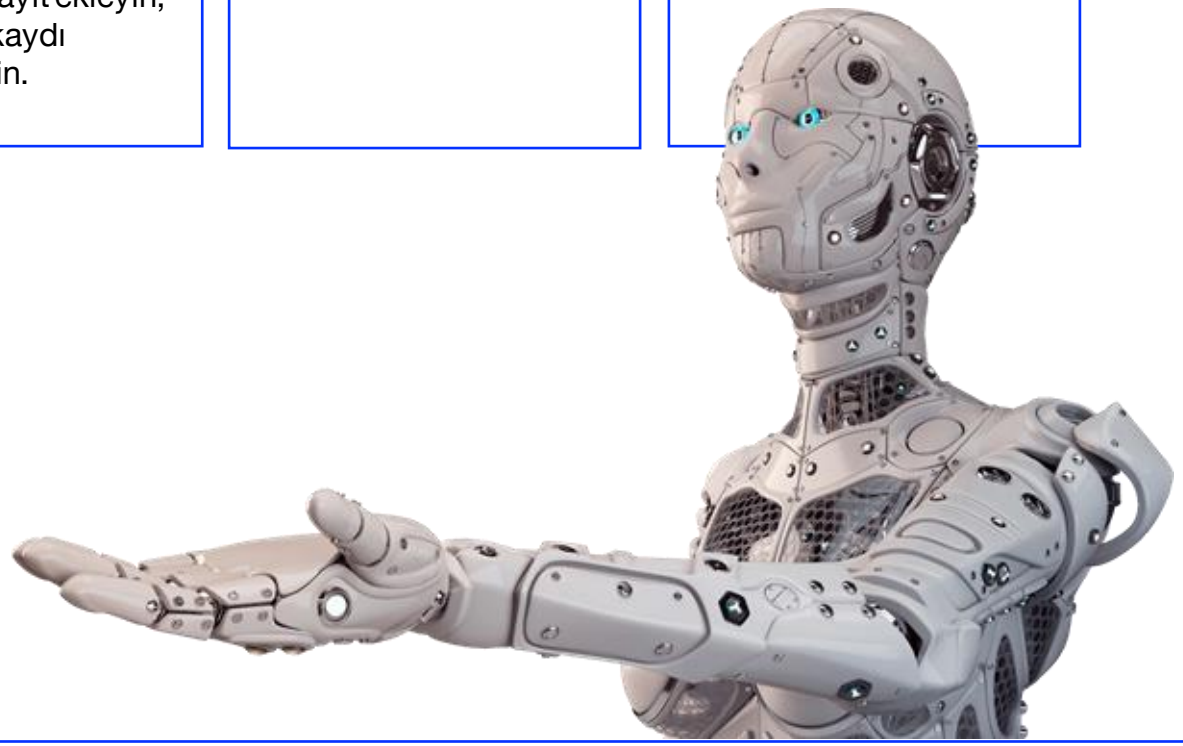
- **#14**
  https://demoqa.com/webtables adresine gidin, "ADD" düğmesini tıklayarak yeni bir kayıt ekleyin, eklenen kaydı düzenleyin.

- **#15**
  Daha önce yazılan testleri Selenide kullanarak gerçekleştirin

- **#16**
  Daha önce yazılan testleri Cucumber ile gerçekleştirin.

- **#17**
  Bir OpenSource API seçin ve bu API için testler yazın

- **#18**
  Yazdığımız testlere Allure Report ekleyin.



[ARAÇLAR]

[KURSTAN SONRA...]

- Farklı test otomasyon araçları ve framework'lerini anlama ve kullanma yeteneğinizi geliştirecek, bu araçları, test senaryolarını otomatikleştirmek ve test süreçlerini daha etkili hale getirmek için kullanacaksınız.
- Test otomasyon eğitimi ile, temel yazılım geliştirme prensiplerini ve pratiklerini öğreneceksiniz. Bu sayede test senaryolarını daha iyi anlayacak, daha sürdürülebilir ve esnek otomasyon projeleri oluşturacaksınız..
- Konteynerleştirme (Docker) ve sürekli entegrasyon / sürekli dağıtım (CI/CD) süreçlerini anlamanıza yardımcı olacak. Bu da otomasyon projelerinizi daha hızlı ve düzenli bir şekilde teslim etmenizi sağlayacak..
- Etkili test stratejileri oluşturma ve test senaryolarını planlama yeteneğinizi arttıracak. Otomasyon projelerinizin daha kapsamlı ve hedefe yönelik olmasını sağlayacaksınız.
- Farklı test seviyelerini (birim testleri, entegrasyon testleri, kabul testleri vb.) ve bunlara uygun otomasyon yaklaşımlarını anlayıp, doğru yaklaşımı seçip uygulayacaksınız..
- Kullanıcı arayüzü (UI) ve API testlerini otomatikleştirme yeteneğinizi geliştireceksiniz. Bu sayede hem web arayüzlerini hem de arka plandaki işlevleri test edebileceksiniz.
- Paralel test çalıştırma konseptini anlayacak ve otomasyon projelerinizi daha hızlı hale getirebilmek için aynı anda birden fazla testi çalıştırma yeteneği kazanacaksınız.
- Test otomasyon projelerinin hatalarını tespit etme ve sorun giderme yeteneğinizi geliştirecek, daha güvenilir ve kaliteli otomasyon projeleri oluşturacaksınız..

[KURS FORMATI]

Google Classroom, eğitimin gerçekleştiği kişisel hesabınızdır.
Kurs ile ilgili tüm bilgilere bu platform üzerinde sahip olacaksınız.

01 Kişisel hesabınızda çalışmanızı gönderebilir ve eğitmenden geri bildirim alabilirsiniz.

02 R_D Destek kapalı Discord grubu: Danışmak istediğiniz konu varsa bize her zaman ulaşabilirsiniz.

03 Kurstaki ders günleri ve ödevlerin son teslim tarihleri kaçırmamanız için bir Google takvimi ile senkronize edilebilecek bir program vardır.

04 Tüm dersleri kayıt altında olacak, ders kaçırdıysanız, dersin kaydı, dersin ertesi günü kişisel hesabınızda mevcut olacak.

05 Kursun sonunda, alınan puanlara bağlı olarak bir sertifika veya katılım sertifika alabileceksiniz.

06 Kurs ve tüm materyaller 1 yıl boyunca kişisel hesabınızda kalır. Daha sonra geri dönebilir ve bilginizi tazeleyebilirsiniz.

[İLETİŞİM]

>>> e-mail adreslerimiz

info@robotdreams.com.tr

support@robotdreams.com.tr

>>> sosyal medya

facebook

instagram

linkedin

r_d.com.tr

siz: tamam sırada ne var?

robot_dreams: yanıt bulundu ➡ bir kursa kaydolun