Gelişen ve değişen teknolojiler ile birlikte, teknoloji de kendi içinde gelişti değişti ve alanlara ayrıldı. Özellikleri itibari ile farklı şekilde oluşan bu sistemler içinde insanların da görsel beklentileri oldu. Elbette ki bir web sitesinde aradığınızı bulmanızın yanı sıra, görselliğinde ön plana çıktığını anladık. Bu da yazılımı kendi içinde farklılaştırdı. Değişen ve özel bir şekilde hayatımıza giren bilişimin en önemli alanları tamamen bağımsız eğitimler ile gerçekleşmeye başladı. Bunlar içinde alabileceğiniz **yazılım prensipleri ve tasarım desenleri** eğitimi, oldukça farklı şekillerde yararlanabileceğiniz bir eğitim şeklidir. Sizler de özgün ve farklı tasarımları en etkin şekilde yaratabilmek adına, kesin olarak bu tarz eğitimlerden faydalanabilirsiniz. Alternatif olarak öğrenilmesi sizlere fayda sağlayacak olan bu prensipler ile sizler de en farklı yolları keşfedecek, denetimlerinizi bu şekilde yapabileceksiniz. Kodları doğru ve başarılı şekilde kullanabildiğiniz sürece, kontrolün sizde olduğunu göreceksiniz.
Yazılımlar belirli bir mantık içinde ilerlemek zorundadır. Bu şekilde yapılması gerekenler ve asla yapılmaması gerekenler diye yazılım prensiplerini ayırabilirsiniz. Ancak şunu belirtmek gerekir ki, bazı metotları kavradığınızda artık yazılımların çok daha hızlı bir şekilde ilerlediğini göreceksiniz. Başarılı bir şekilde alacağınız **yazılım prensipleri ve tasarım desenleri** eğitimi, sizlere zaman kaybetmeden kusursuz kodlamalar yapmanızı sağlayacaktır. Bu şekilde de geniş bir kapsamda işlemleri yapabileceksiniz. Edilgen bir yapıda olan kodlama sistemleri, hakimiyetin yoğun olması durumunda sizlere çok farklı kapıları açacaktır. Ancak her şeyde olduğu gibi koda yazmanın da bazı prensipleri ve yapılması gereken bir iş akışı olduğu kabul edilmelidir. Bu biçimden baktığınızda sizler de en özel sonuçları en hızlı şekilde elde edebilirsiniz. Farklı bir tarz ile sizlerin de rahatlıkla öğreneceğiniz bu işlemler, ileride inanılmaz faydalar alacağınız bir işlem ağına da hakim olmanızı sağlayacaktır.
Tasarımların göze farklı şekilde görünmesini sağlayan desenleri özellikle nesnelere yönelik yapılan yazılımların daha çok kapsamına girmektedir. Desin Pattern olarak karşımıza çıkan bu uygulama, dilimizde ise tasarım şablonları, desenleri ya da kalıpları olarak çevrilebilir. Kullanıcılara harika bir kolaylık sağlayan bu sistemler sayesinde sizler de çok farklı sonuçları elde edebilirsiniz. Karmaşık yapıdan uzak olan bu işlemler sayesinde, en farklı işlemleri yapmanız da mümkün olacaktır. **Yazılım prensipleri ve tasarım desenleri** eğitimi kullanıcıların harika bir şekilde yazılımlar üzerinde çalışmalarını da kolaylaştıracaktır. Farklı tasarım bozukluklarının tespit edilmesi ve bunların düzenlenmesi gibi ileri boyutlar da işlem yapabilme yetisine bu eğitimler ile ulaşabilirsiniz. SRP, OCP, LSP, DIP ve ISP olarak adlandırılan bu SOLID prensipler ile daha alışık bir şekilde işlem yapmanız da mümkün olacaktır. Tüm bu işlemler ile daha farklı ve özel bir ilerleme yapabilir, programcılık bilgisini ileri düzeye getirebilirsiniz. Sizler de uygulamalar ve eğitimlerimiz hakkında daha detaylı bilgileri bizlerden alabilirsiniz.

Eğitim Süresi: 5 gün

Amaç: Bir yazılımda değişim kaçınılmazdır. Kurumsal beklentiler, mevzuat değişiklikleri yazılımda değişimi gerektirir. Yazılımın kalitesi ise değişime verdiği direnç ile ters orantılıdır. Tasarım desenleri ile gerçek hayat uygulamaları üzerinden, değişime direnç göstermeyen yazılımların nasıl yapılacağı aktarılır. İyi bir C# bilgisi gerektirir.

İçerik

The Single Responsibility Principle
The Open / Closed Principle
The Liskov Substitution Principle
The Interface Segregation Principle
The Dependency Inversion Principle
The Don’t Repeat Yourself Principle,
Introduction to Design Patterns
Adapter Pattern
Bridge Pattern
The Builder Pattern
Chain of Responsibility
The Command Pattern
Composite Pattern

Decorator Design Pattern
Event Aggregator
Facade Pattern
Factory Patterns
Flyweight
Interpreter Pattern
Iterator Pattern
Lazy Load Pattern
Mediator Pattern
Memento
Model View Presenter (MVP) Pattern
Model View ViewModel (MVVM) Pattern
Null Object Pattern
Observer Pattern
The Prototype Pattern
Proxy Pattern
Repository
Singleton
Service Locator Pattern
State Pattern
Strategy Pattern
Template Method
Unit of Work
Visitor Pattern