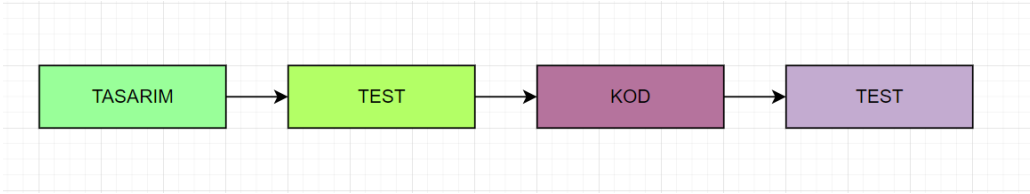


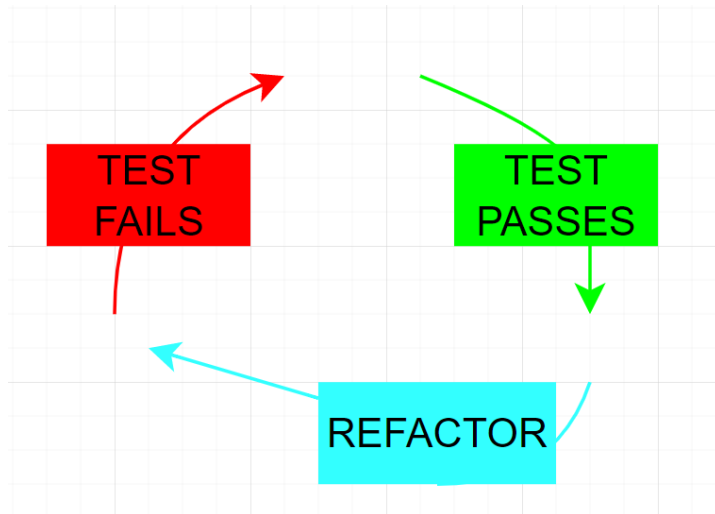
TDD (Test Driven Development)

TDD, **test odaklı geliştirme** anlamına gelir. Yazılacak kodun, neler yapacağını ve bunların doğruluğunu belirlemek amacıyla test senaryolarının geliştirilmesini sağlayan bir yaklaşımdır. **Test-first (önce test)** ilkesine bağlı olarak hareket eder. Bu yaklaşım ile birlikte önce düşünerek test oluşturulur sonrasında geliştirme yapılır. Yani önce geliştirme yapıp, sonrasında bunu test etmek TDD yaklaşımı olamaz. Geleneksel olarak geliştirilen yazılım projelerinden farkı burada başlar.

TDD yaklaşımına göre, ilk olarak **tasarım** kısmından başlanır ve sonrasında **test** aşamasına geçilir. Tasarım kısmında ortaya çıkan metotlarımızın, operasyonlarımızın vs. gibi bize nelerin gerek olduğunu tasarladığımız için test yazmak daha kolay olacaktır. Böylece kod yazmadan önce testimizi gerçekleştirmiş oluruz. Kod yazımı ile birlikte projemizi geliştirdikten sonra tekrardan test yaparız. Çıkan hatalara karşı yeniden düzenleme (refactoring) yaparak yazılan kodu iyileştiririz.



TDD yaklaşımının, yaşam döngüsü vardır. Bu yaşam döngüsü ilk olarak testin yazılması ile başlar. Test yazıldıktan sonra hatayla karşılaşmamız gerekir. Hata ile karşılaşılıyorsa yazdığımız testte bir sorun olduğu anlamına gelir. Sonrasında test başarılı hale getirilir. Tüm testlerin istediğimiz çıktıyla uyumlu yani başarılı olması sağlandıktan sonra yeniden düzenleme adımına geçeriz. Kod içerisinde bulunan iyileştirmelere vs. gibi noktalara bakarız.



TDD yaklaşımının bize sağlayacağı avantajlar vardır. Bunlardan en önemlisi geliştirme yapılırken ihtiyaç olmayan sınıf, nesne gibi proje tarafında ekstra maliyet getirecek şeylerin projeye eklenmemesini sağlayarak bir kazanç getirir. Gerekli olan sınıf, nesne ve metotlar olacağı için geliştirme daha hızlı olacaktır.

Test kapsamı daha geniş olacağından bulunacak olan hataların neden kaynaklandığı kolay bir şekilde ortaya çıkar ve geliştiricinin daha rahat görmesini sağlar. Yapılan testler, geliştirilecek olan projenin çalışma mantığını ortaya çıkarttığı için projenin amacı daha iyi anlaşılır. Tabi TDD yaklaşımını her projede kullanmak doğru olmayabilir. Bunu belirlemenin yolu geliştiricinin tecrübesi ve geçmiş projelerinde çıkan sorunlardan edindiği bilgiler ışığında olabilir.