**BİTİRME PROJESİ**

**Konu : Kaynak kod versiyonlama sistemi : detaylı github örnekleri**

**Hazırlayan : Hüdayfa ÇİFTÇİ**

**Okul Numarası : 2015123038**

**Proje adımları :**

**1. GitHub’a ÜYE OLMA**

**2.REPOSITORY OLUŞTURMA**

**3.GitHub DESKTOP UYGULAMASINI İNDİRME VE KURMA**

**4.REPOSITORYE DOSYA COMMIT ETME**

**5.GitHub ECLIPSE İLE ENTEGRE ETME**

**6.JAVA KODU COMMIT ETME**

**7.JAVA KODU DEĞİŞTİRİLİP COMMIT ETME**

**8.JAVA KODU ESKİ VERSİYONUNA GERİ DÖNDÜRÜLÜP COMMIT ETME**

**9.CONTRIBUTOR EKLEME**

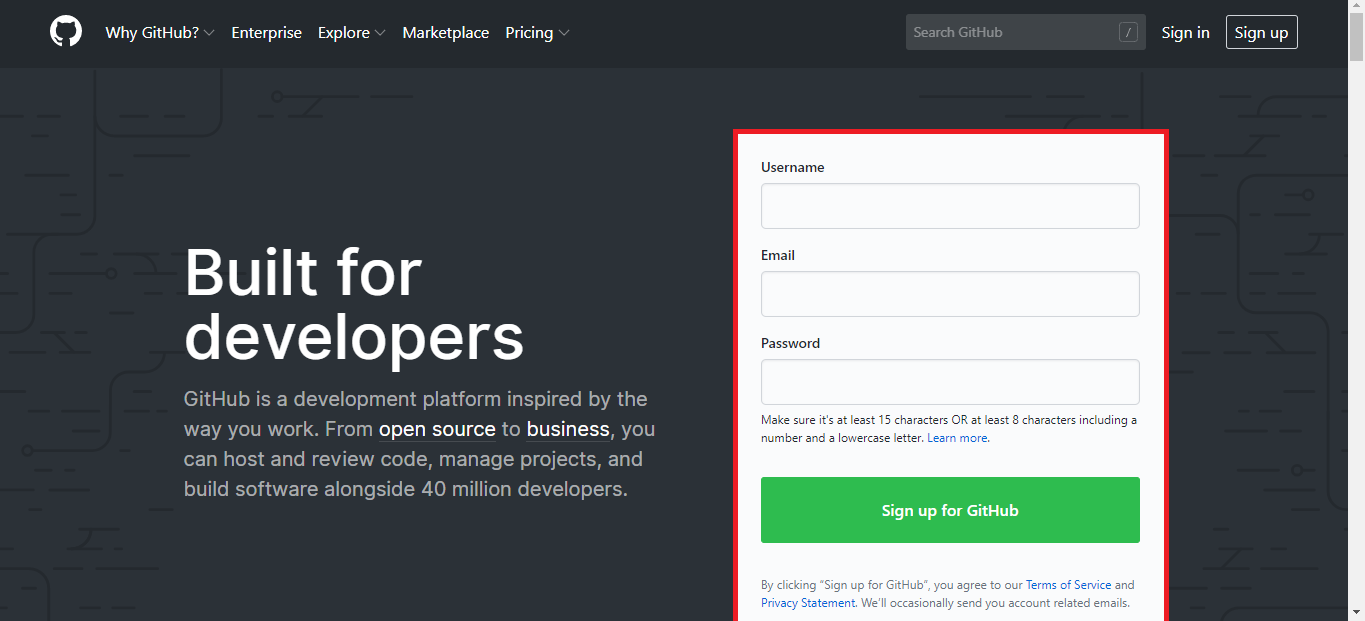
**10.FARKLI KULLANICIDAN AYNI REPOSITORYE COMMIT İŞLEMLERİ**

**11.CONFLICT İŞLEMLERİ**

**12.ISSUE İŞLEMLERİ**

**1. GitHub’a ÜYE OLMA**

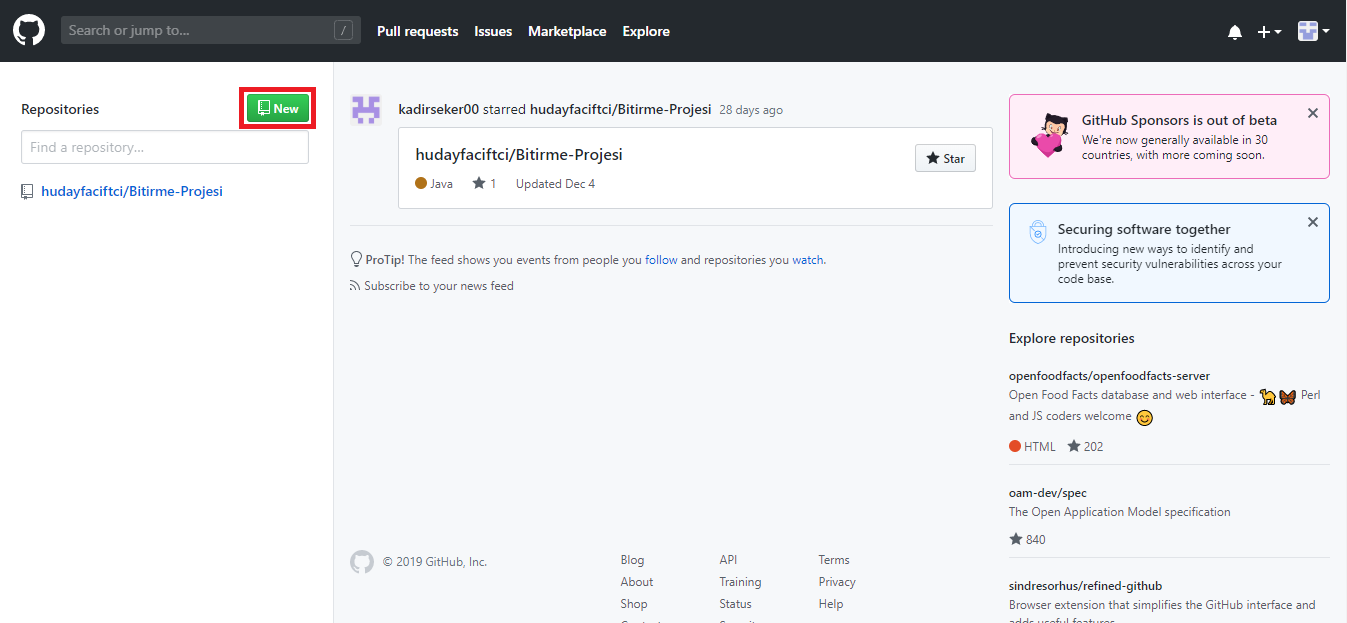
İlk önce tarayıcı üzerinden [www.github.com](http://www.github.com) adlı siteye girilir.



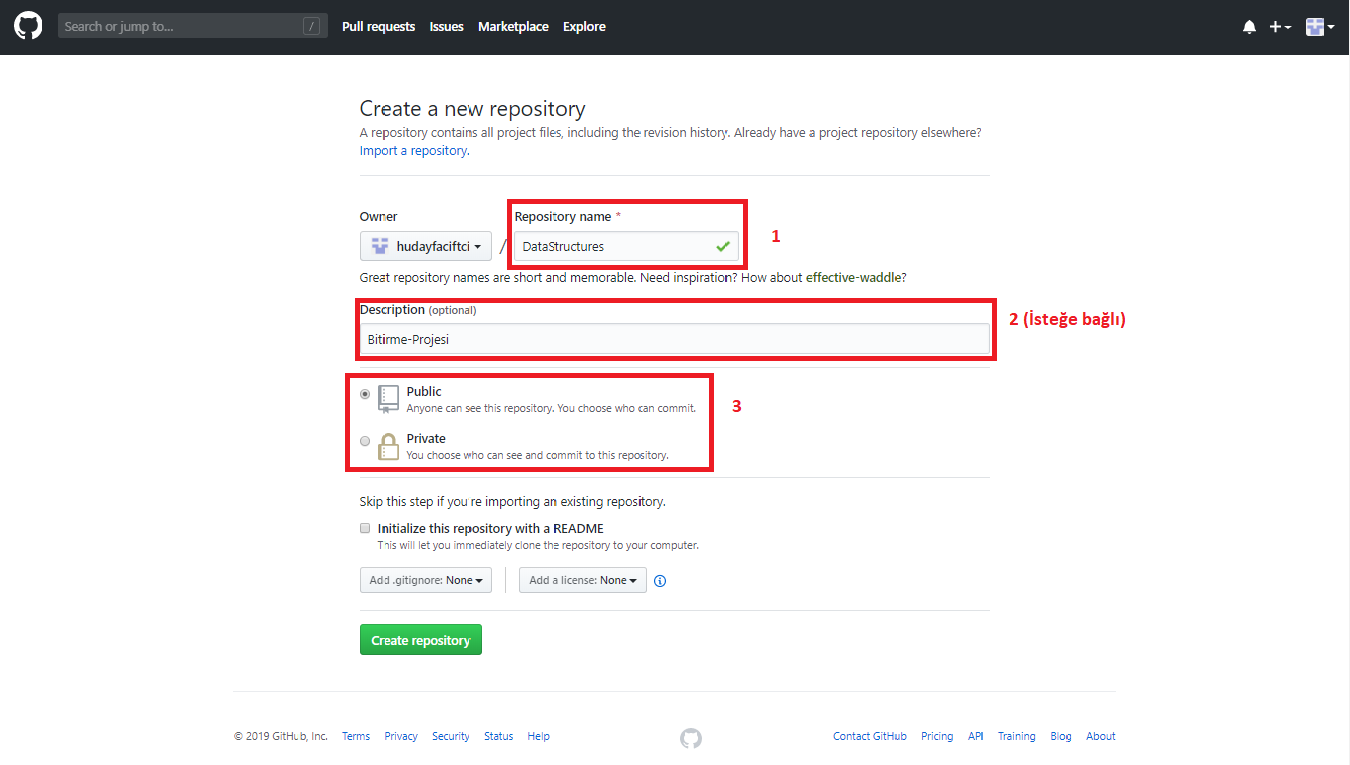
(Resim de kırmızı çerçeveli yer)Kayıt formunu doldurup Sign up for GitHub butonuna tıklayıp üye olunur.

**2.REPOSITORY OLUŞTURMA**

GitHub’a üye olduktan sonra aynı sayfa üzerinden giriş yapılır daha sonra açılan sayfada Repositories yazısının yanında bulunan “New” butonuna tıklanır.

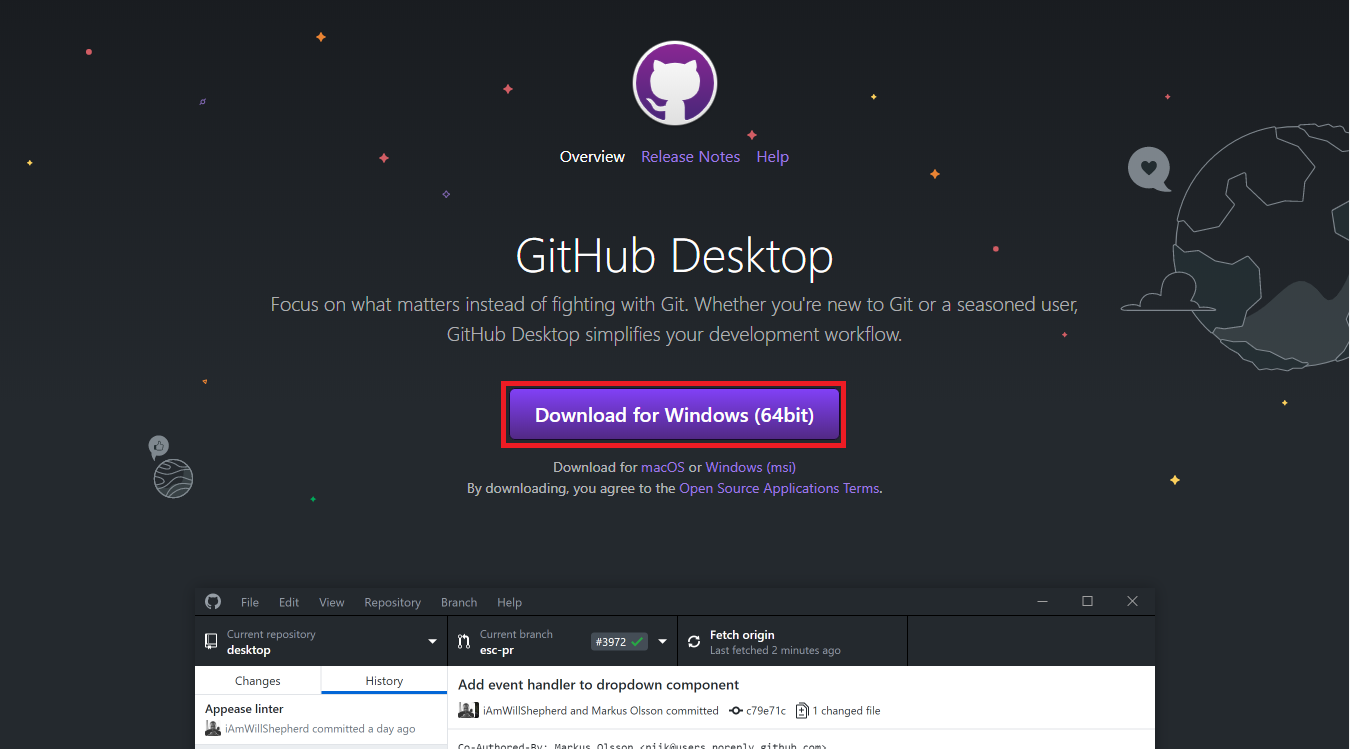


Daha sonra yeni gelen sayfada (altta bulunan resim) 1 numaralı yere repository adı yazılır , 2 numaralı yere repositorynin açıklaması yazılır (isteğe bağlı) , 3 numaralı yer de ise repositorynin herkese açık ya da gizli olup olmadığı seçilir (Public : Genel , Private : Özel), en altta bulunan Create repository butonuna tıklanır.



**3.GitHub DESKTOP UYGULAMASINI İNDİRME VE KURMA**

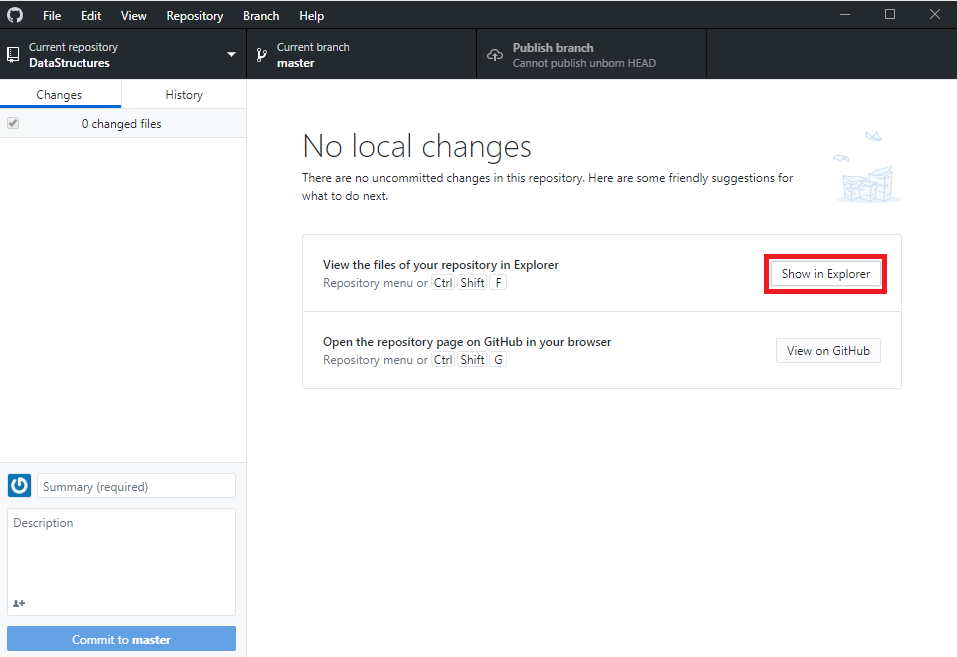
[www.desktop.github.com](http://www.desktop.github.com)adlı siteye girdikten sonra “Download for Windows (64 bit)” butonuna tıklayıp indirilir (eğer işletim sisteminiz farklı ise işletim sisteminize uygun olana tıklayıp indiriniz).



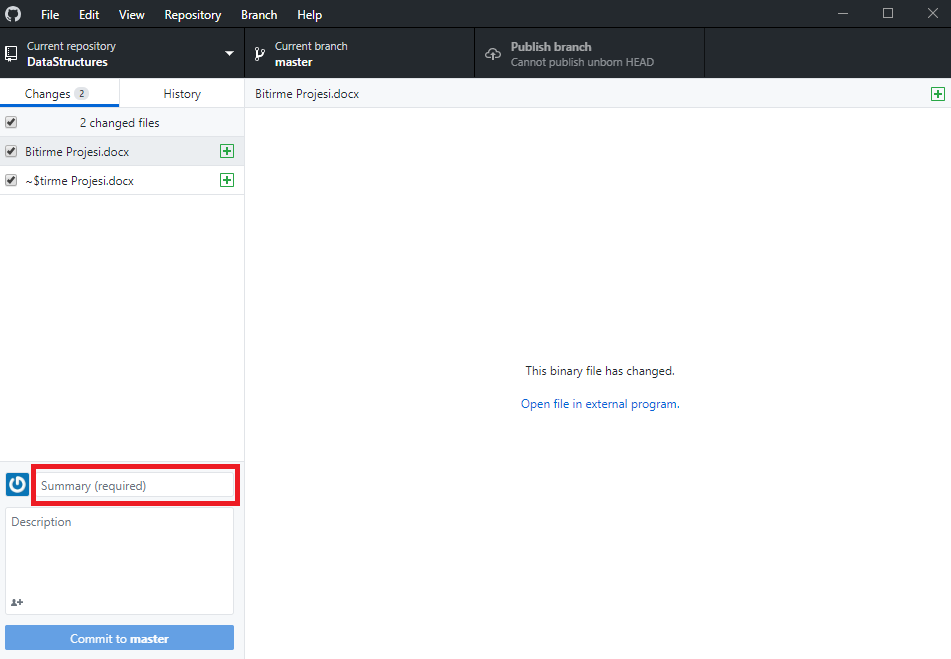
İndirilen dosyaya tıklayıp kurulur (kurulum sırasında herhangi bir işleme gerek duyulmaz). (Kurduktan sonra masaüstüne iconu gelecektir.)

**4.REPOSITORYE DOSYA COMMIT ETME**

İndirilen GitHub desktop uygulaması açılır , “Show in Explorer” butonuna tıklanır , ekrana gelen dosyanın içine repositorye commit edilmek istenen dosyalar atılır.

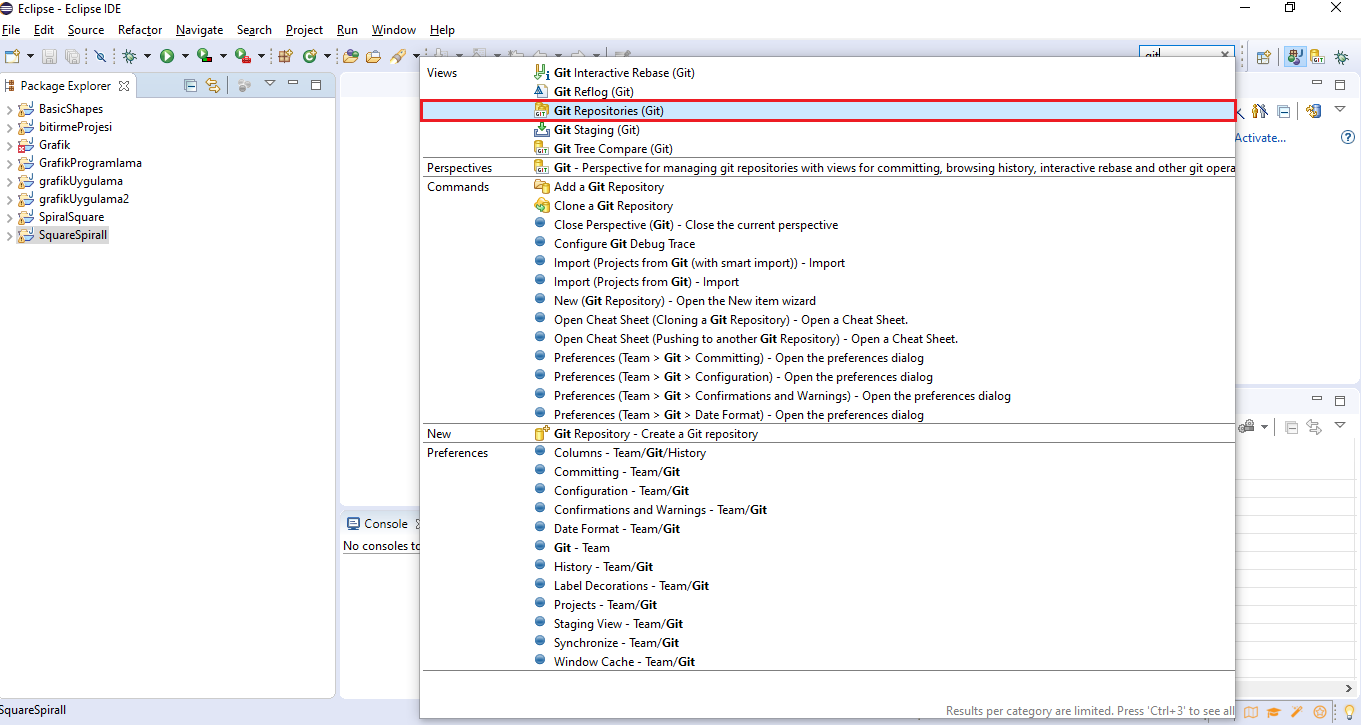


Dosyaları repositorynin klasörüne attıktan sonra uygulamada atılan dosyalar gözükecektir. Daha sonra altta bulunan Summary kısmına açıklama yazılıp “Commit to master” butonuna tıklanır, tıkladıktan sonra mavi yazılı “Publish branch” butonuna tıklayıp repository adresine upload edilir.

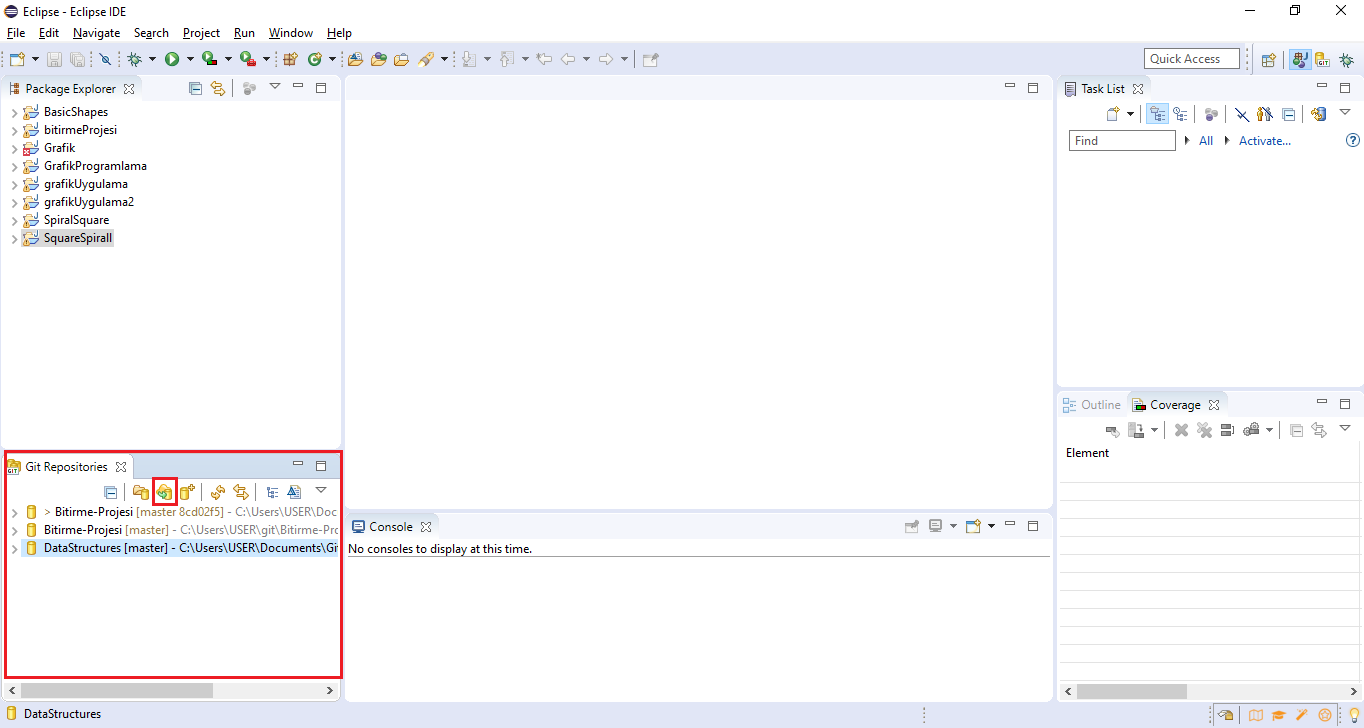


**5.GitHub ECLIPSE İLE ENTEGRE ETME**

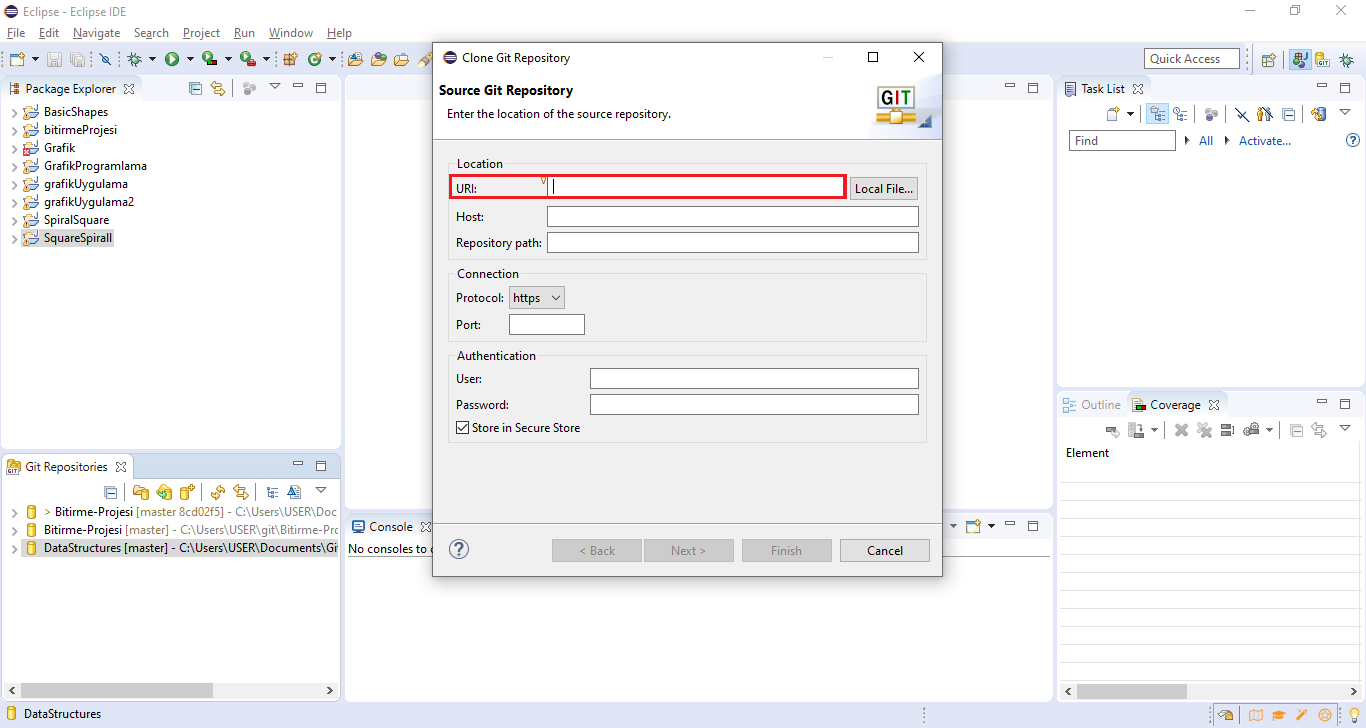
Eclipse yazılım programı açılır, sonrasında sağ üstteki arama yerine “Git” yazıp “Git Repositories (Git)” kısmına tıklanır.



Daha sonra sol kısımda açılan Git Repositories bölümünden Clone butonuna tıklanır (alt resimdeki kırmızı kutucuk).

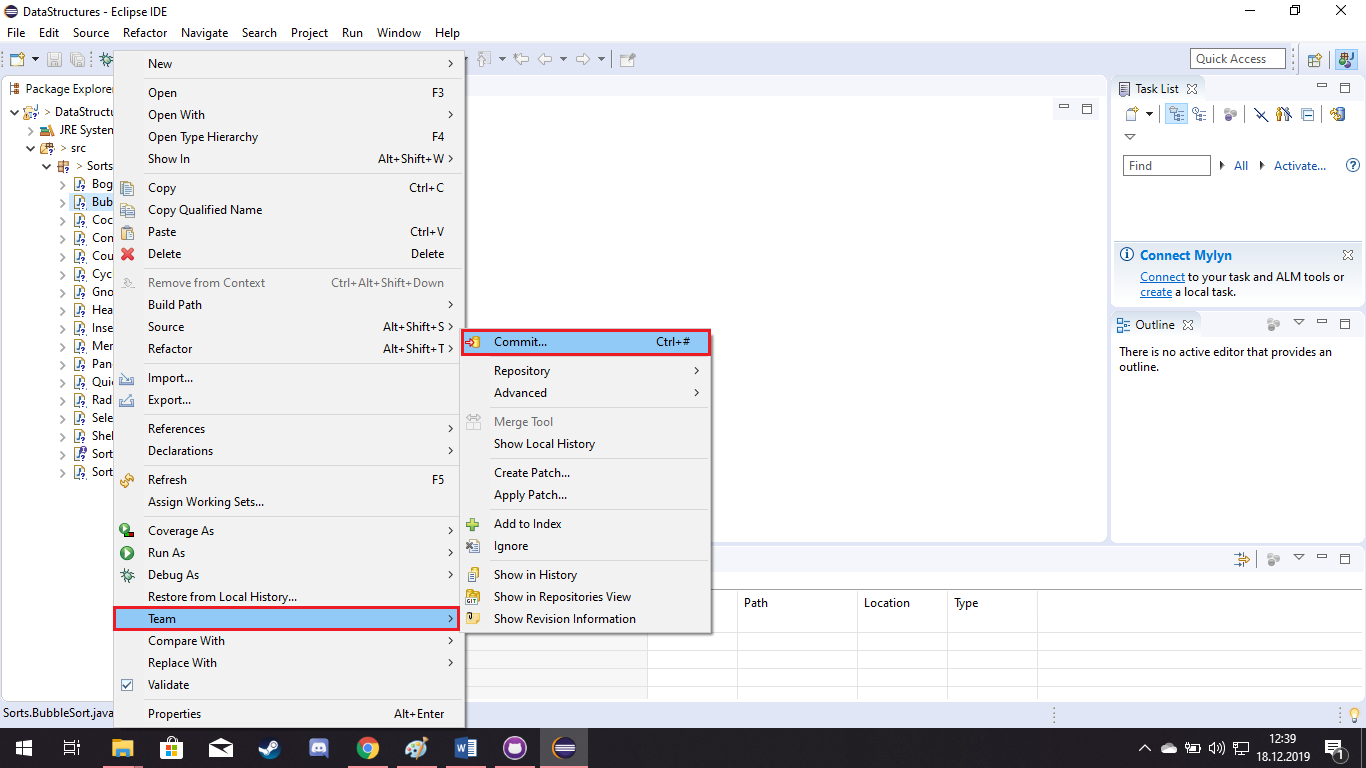


Açılan ekranda url yerine repositorynin linki kopyalanır ve altta bulunan “User” “Password” kısımlarına github kullanıcı adı ve şifresi girilir daha sonra “Finish” butonu gelene kadar hiçbir değişiklik yapmadan “Next” butonuna tıklayıp en son “Finish” butonuna tıklanır.

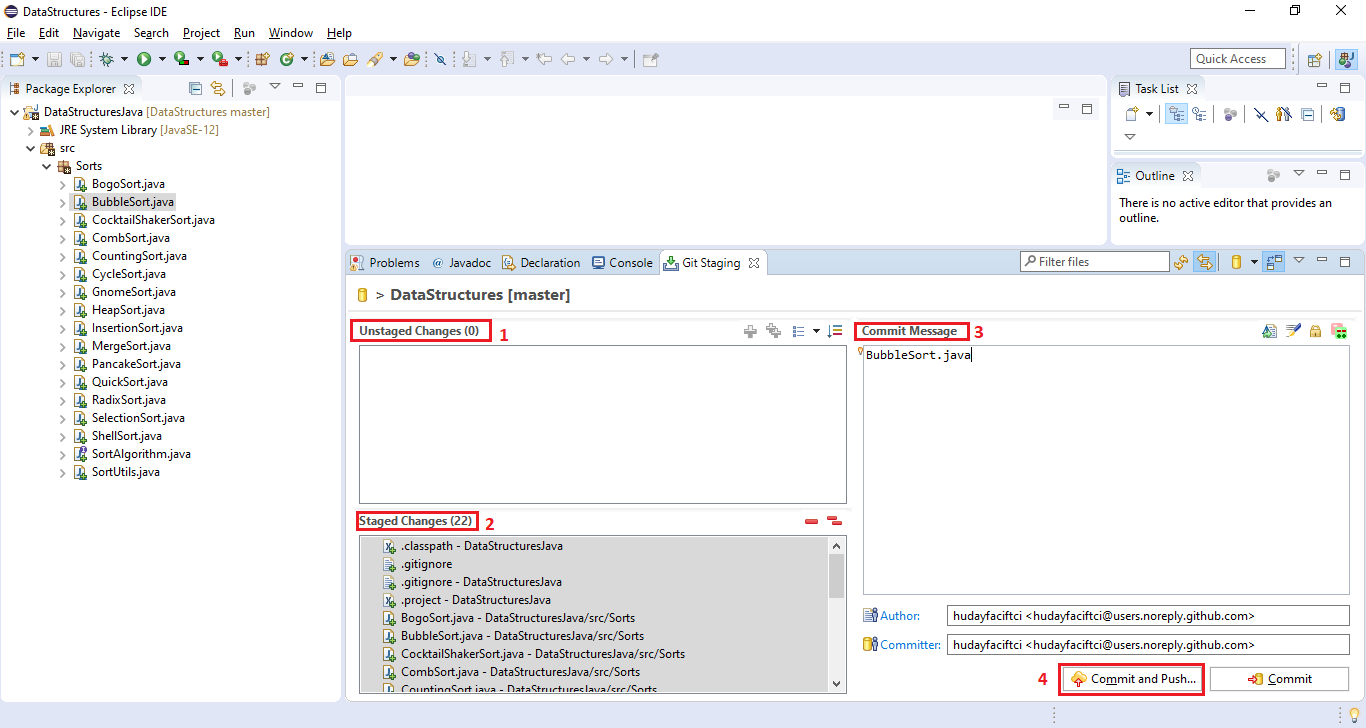


**6.JAVA KODU COMMIT ETME**

Java projesi repository klasörünün içine açılır daha sonra eclipsede kodun üstüne sağ tıklayıp “Team” butonuna gelinir sağ tarafta çıkan sekme de “Commit” butonuna tıklanır.

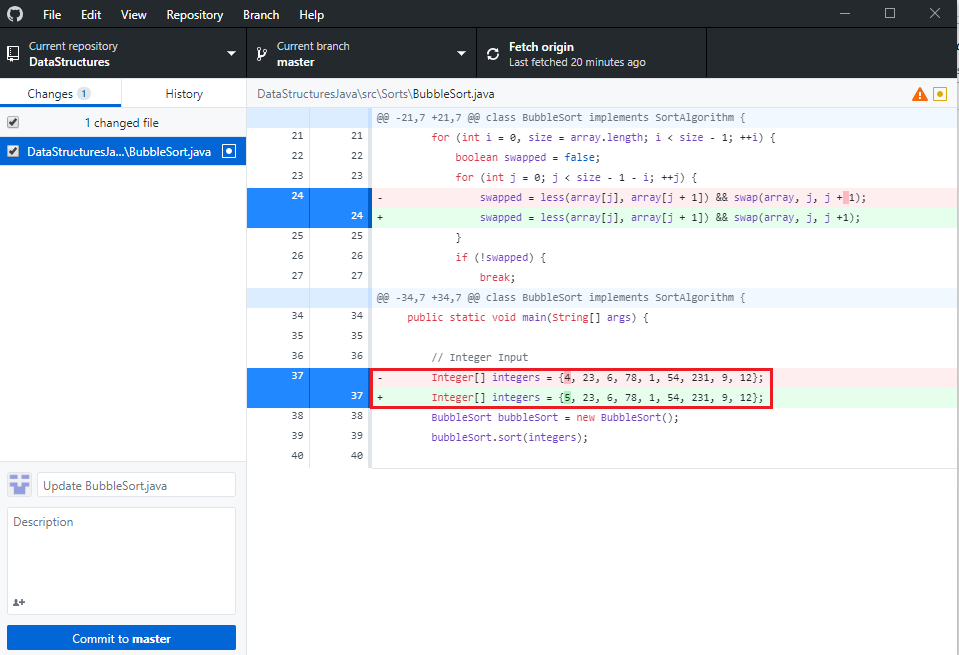


Daha sonra alt kısımda açılan sekme de “Unstaged Changes” kısmında bulunan tüm dosyalar seçilip altta bulunan “Staged Changes” kısmına sürüklenir sonrasında “Commit Mesage” yerine açıklama yazılıp “Commit and Push” butonuna tıklanır.

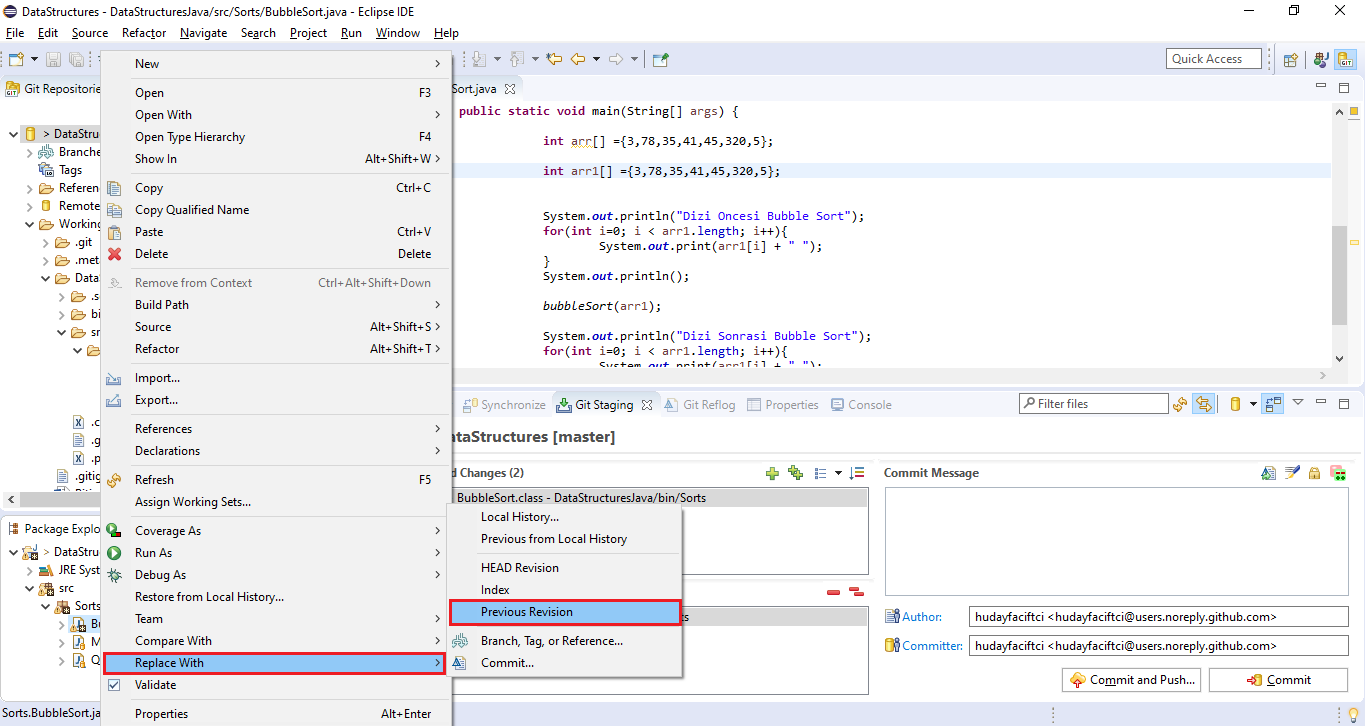


**7.JAVA KODU DEĞİŞTİRİLİP ENTEGRE ETME**

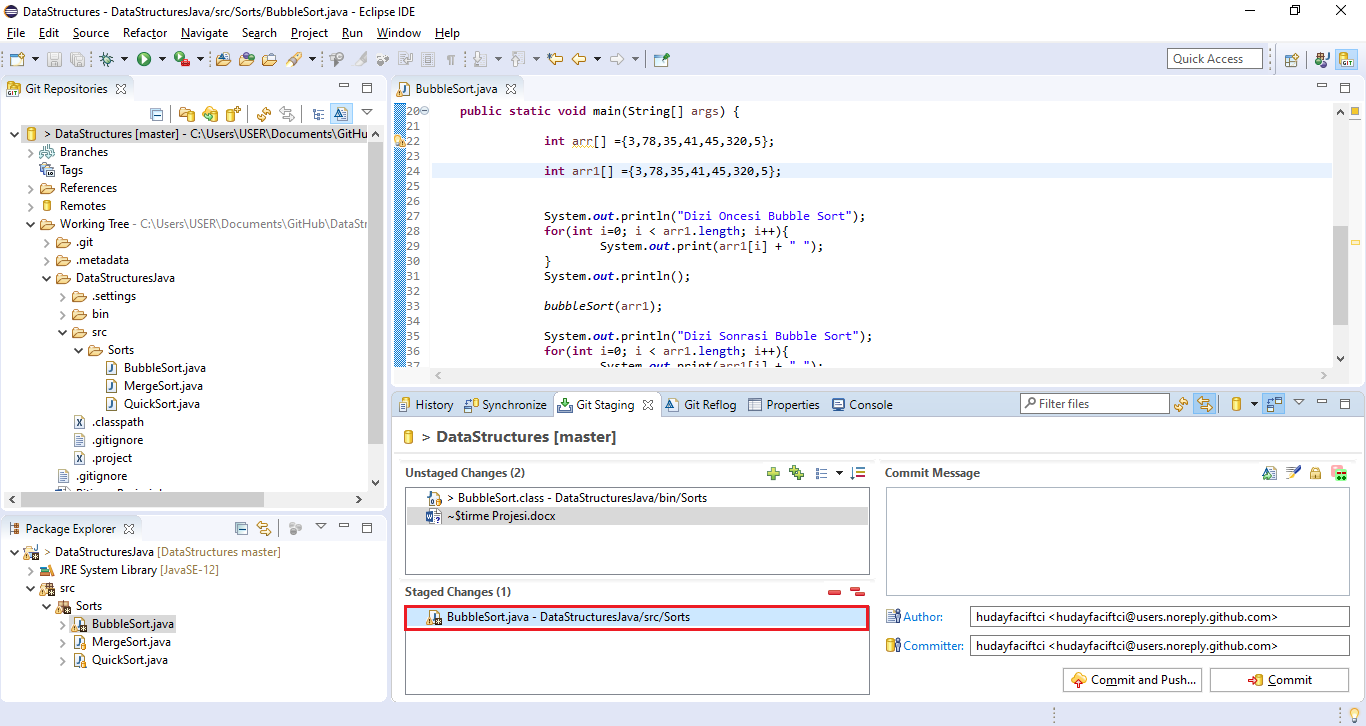
Repositorye commit edilen java kodu değiştirilip tekrar commit edilir, daha sonrasında github uygulamasında kod üzerinde yapılan değişiklikler gösterilip tekrar commit edilmesi istenilir.



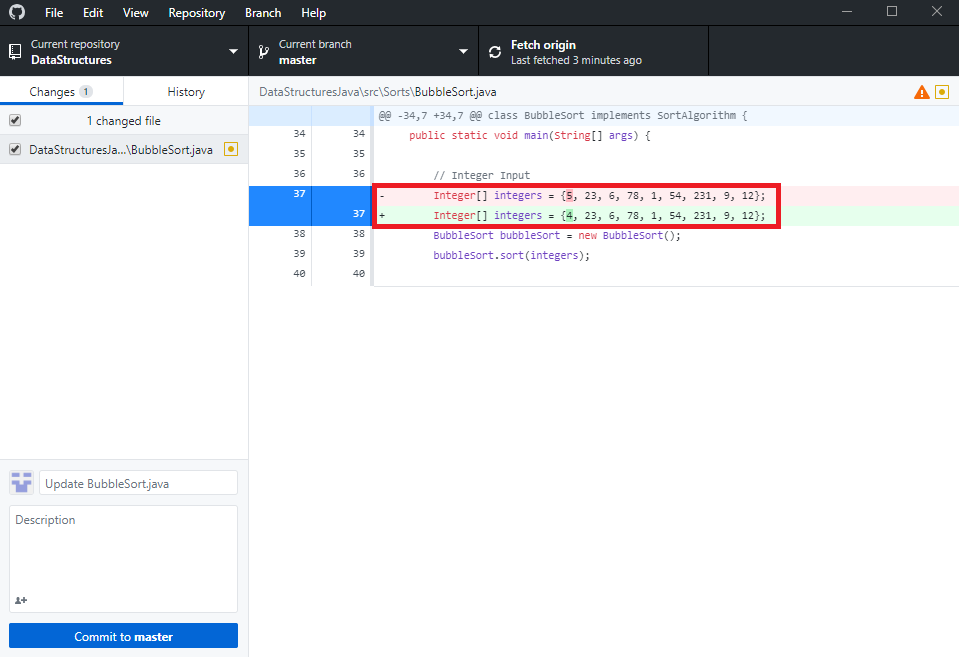
**8.JAVA KODU ESKİ VERSİYONUNA GERİ DÖNDÜRÜLÜP COMMIT ETME**

Değiştirilen kodu eski versiyonuna geri döndürünce github desktop uygulamasında önceki sürümüne döndürüldüğü gösterilir ve tekrar commit edilmesi istenilir. 

İlk önce package explorer sekmesinden değiştirilen kodun üzerine sağ tıklayıp “Replace With” butonuna gelip “Previous Revision” yazısına tıklanır ve bir önceki sürüme geri dönderilmiş olur.

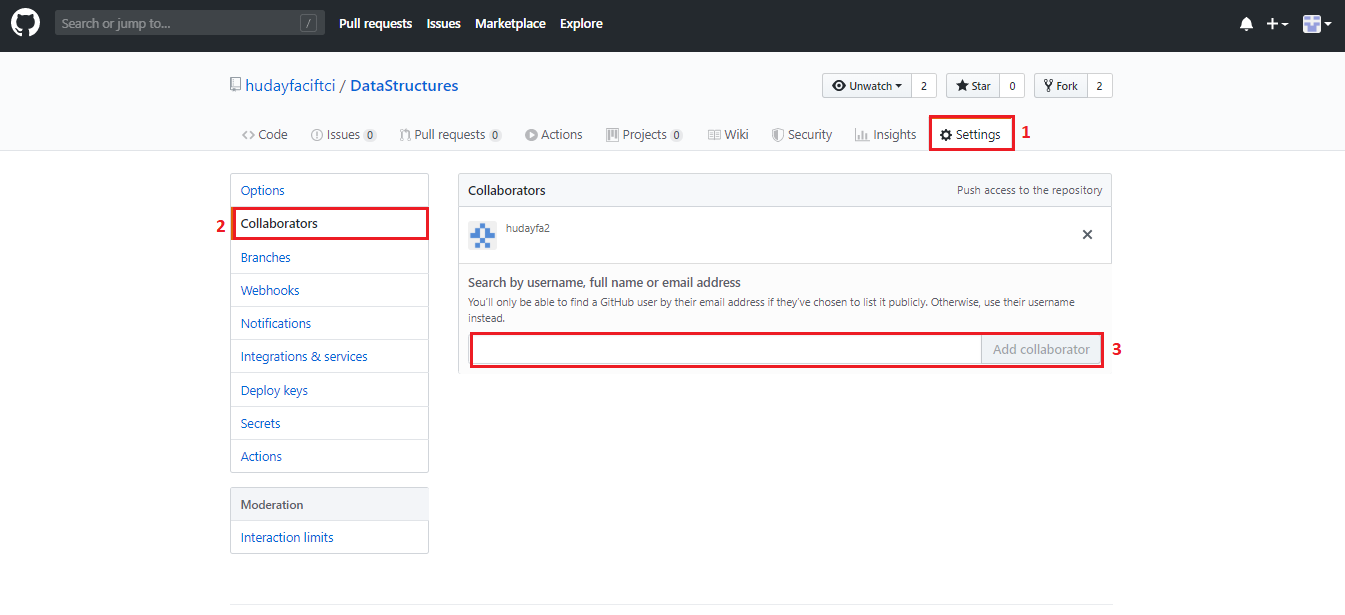


Eski sürüme geri döndürüldükten sonra “Git Staging” sekmesinde değiştirilen kod “Staged Changes” sekmesinin altında gözükür yani commit edilmesi istenilir. Aşağıdaki görselde de github desktop uygulamasında kodun önceki sürümüne göre farklı olduğu gösterilip commit edilmesi istenilir.



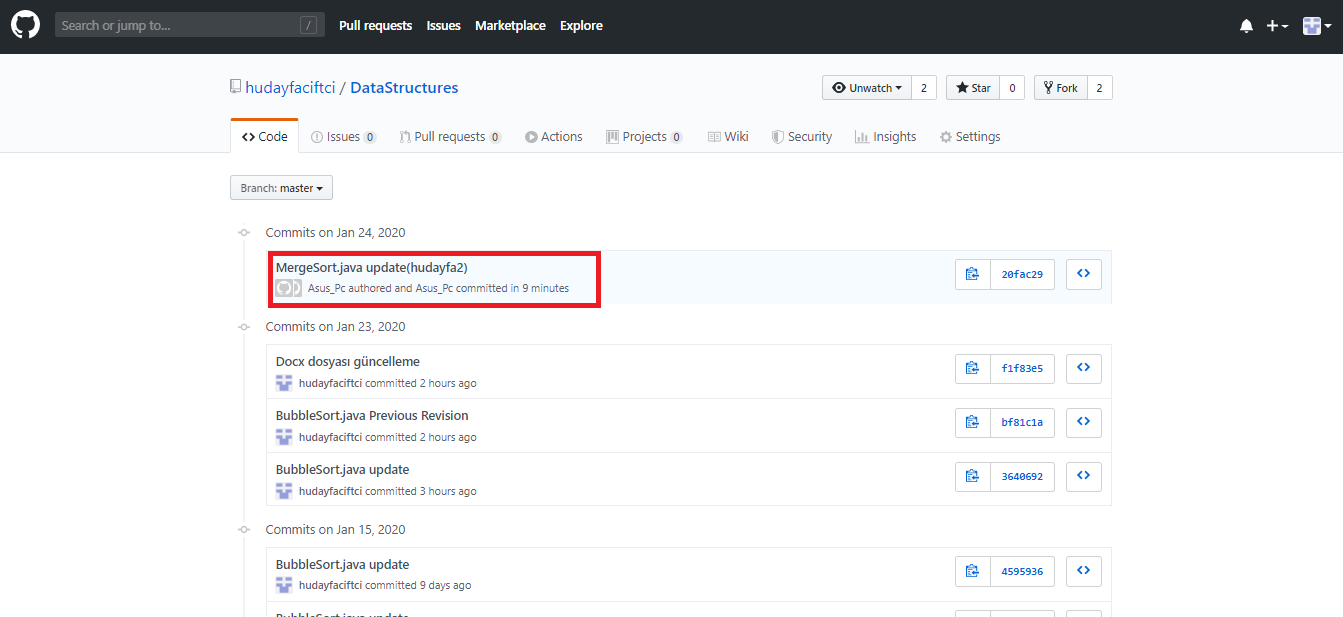
**9.CONTRIBUTOR EKLEME**

Contributor ekleme yani repositorye işbirlikçi ekleme işlemi için öncelikle repository sahibi repositorynin sayfasından “Settings” sekmesine tıklayıp sol kısımda bulunan “Collaborators” kısmına gelip oradaki boş kutucuğa eklemek istediği işbirlikçinin github username ini yazıp davet gönderir. Davet gönderildikten sonra gönderilen kişiye mail olarak iletilir ve kabul ettiği takdirde repositorye contributor olur.

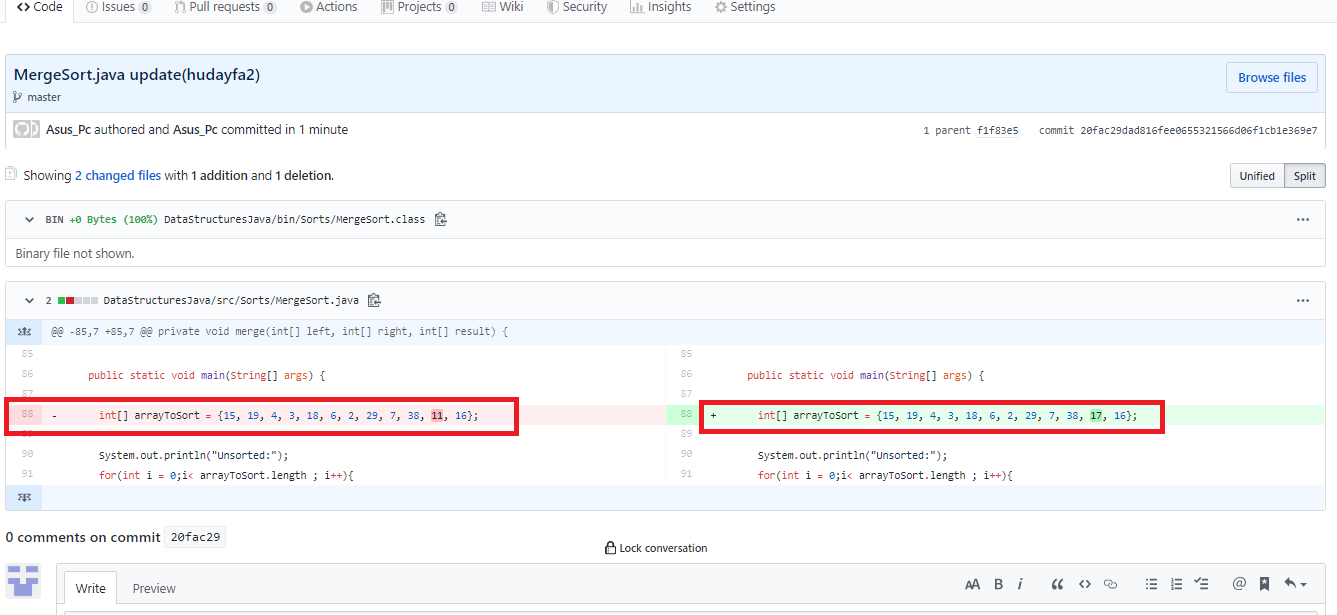


**10.FARKLI KULLANICIDAN AYNI REPOSITORYE COMMIT İŞLEMLERİ**

Master tarafından repositorye contributor olarak eklenen diğer kullanıcılar repositorye commit işlemleri yapabilir. Aşağıdaki resimde de görüldüğü üzere commit eden kullanıcının yazılım programında “Commiter” kısmına yazdığı isim görülür.

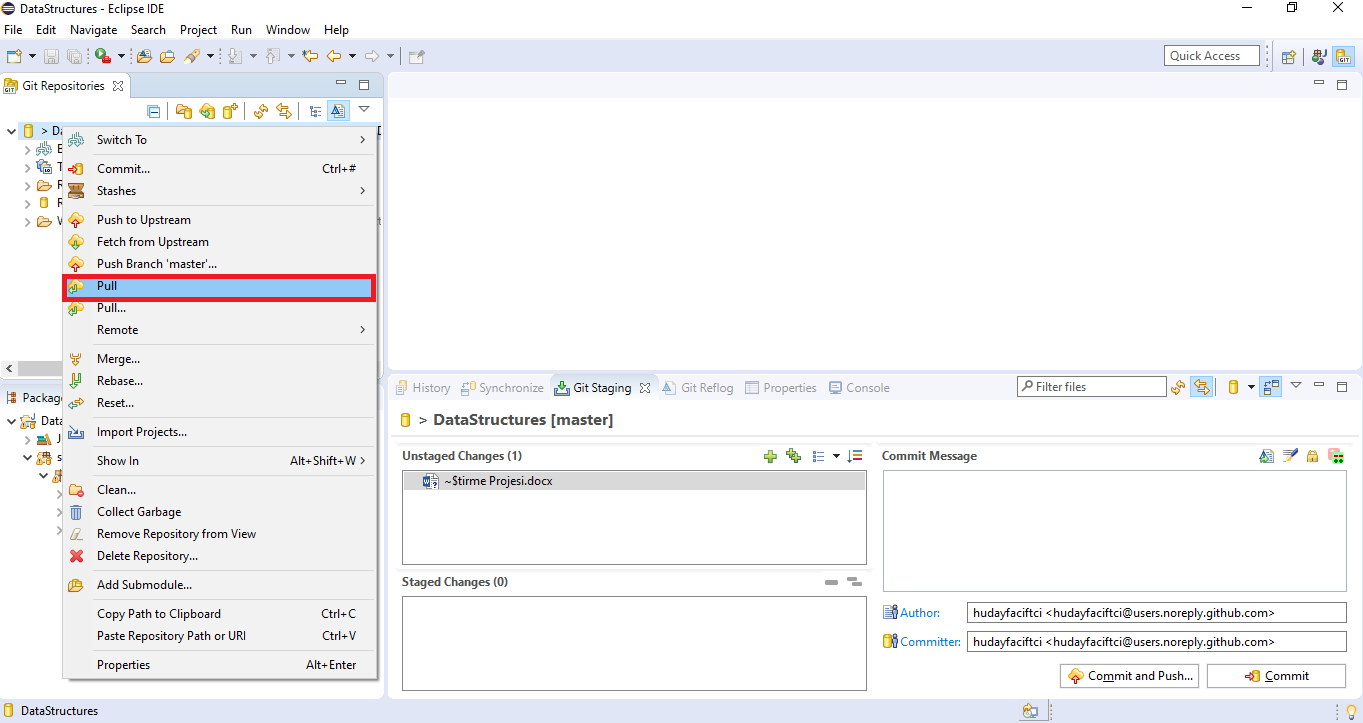


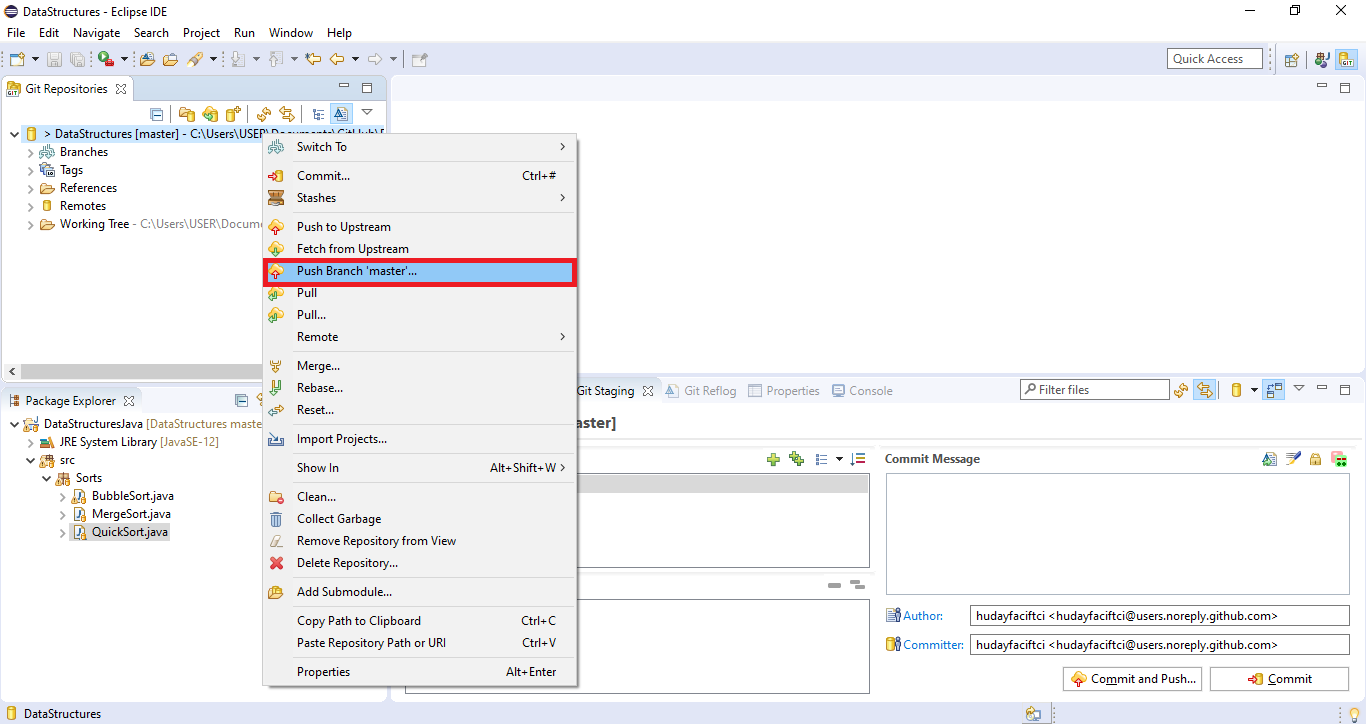
Kullanıcın yaptığı committen sonra commit içeriğine girilince ne silip ne eklediği gösterilir.

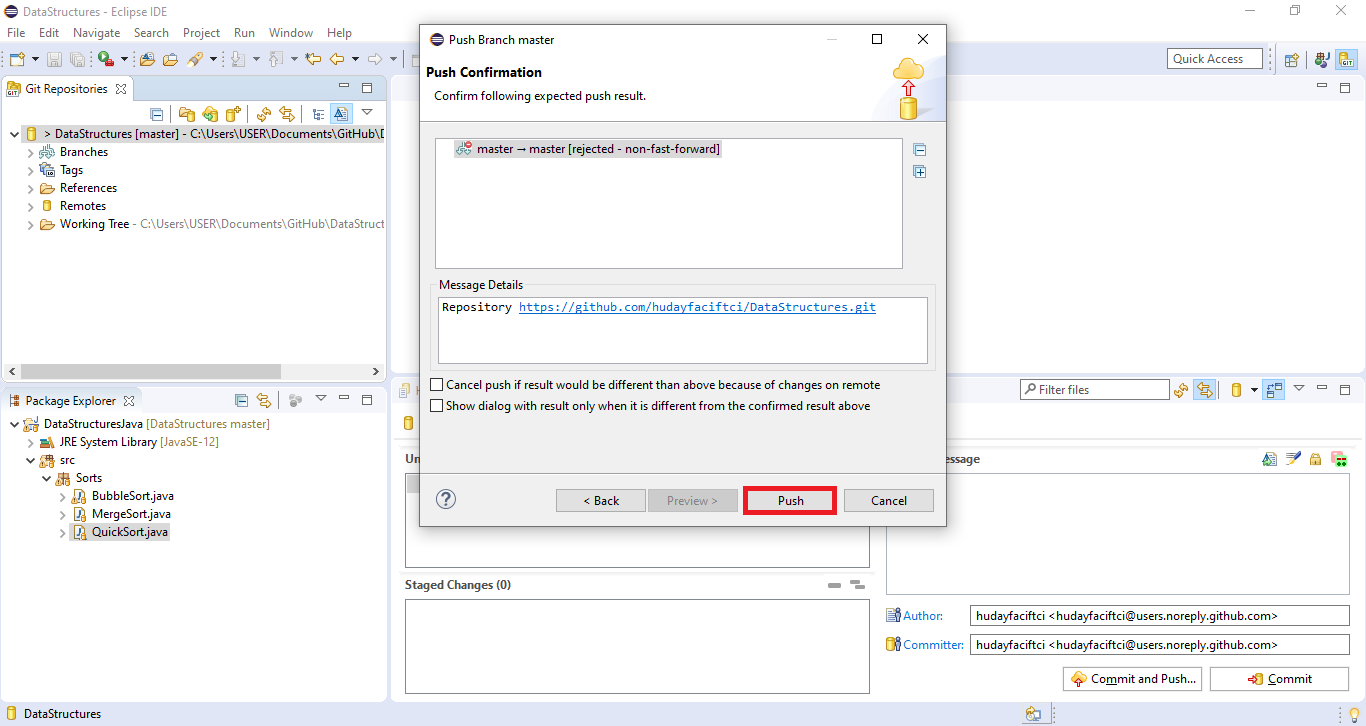
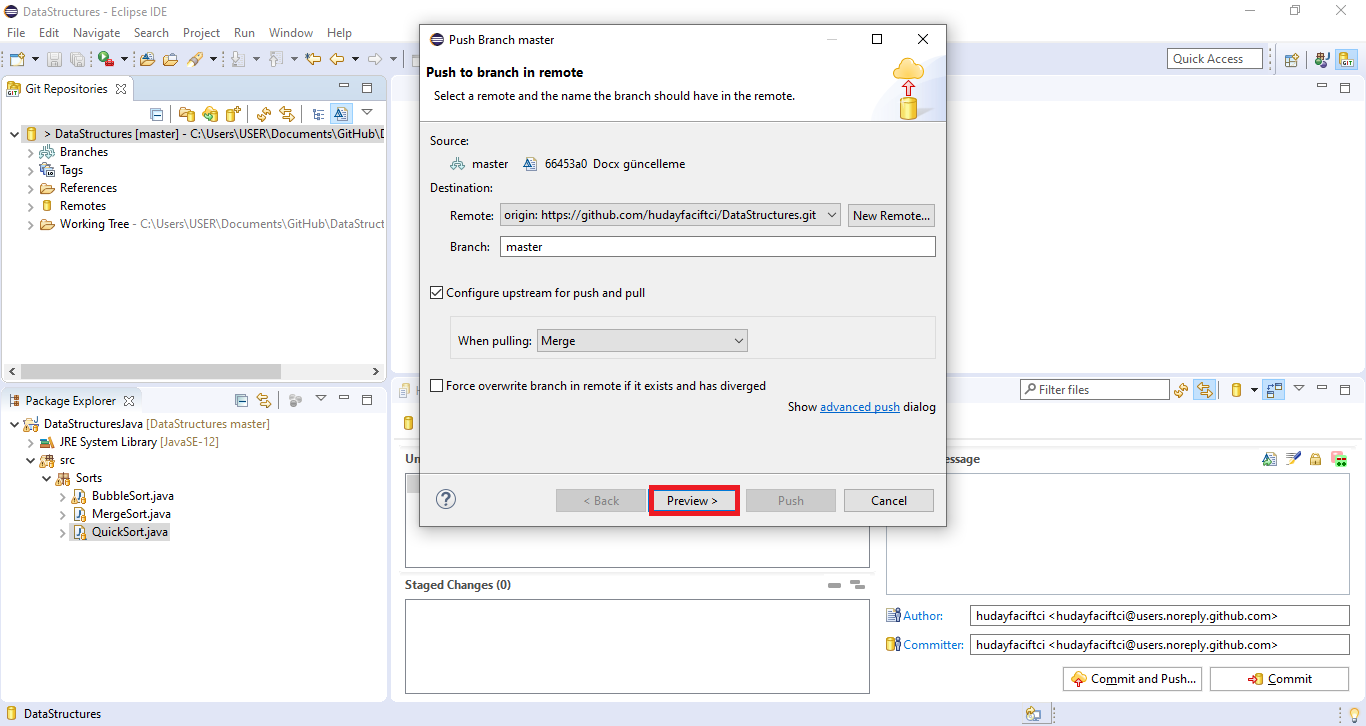


**11.CONFLICT İŞLEMLERİ**

Başka kullanıcı repositorye commit işlemi yaptıktan sonra aynı repoya tekrar commit etmek için pull yapmak gerekir aksi takdirde program commit yapılmasını reddeder. Eğer program pull yaptırmıyorsa önce merge etmek gerekir o da “Pull” butonunun 2 buton altındadır.

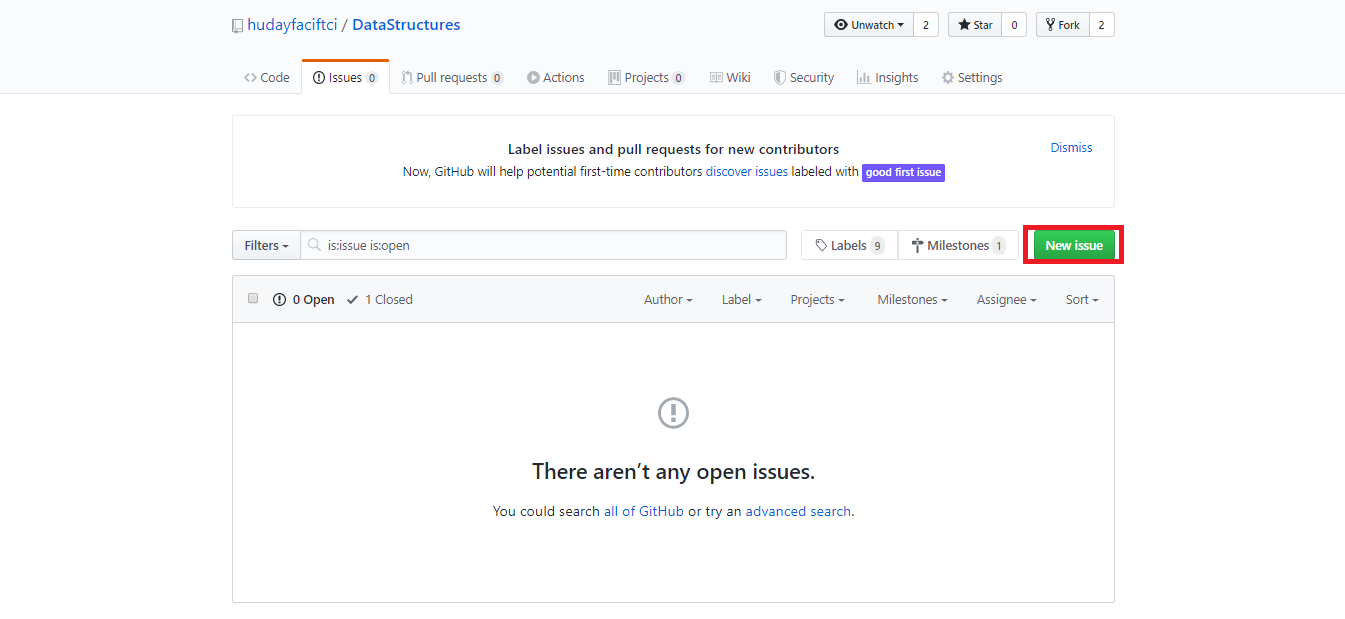


Kullanıcılar aynı kod üzerinde değiştirme yaptılarsa merge edince repositorydeki farklı satırlar koda eklenir bu da kodun hata vermesine yol açar. Bu yüzden kodlar değiştirilmeden önce branch edilip sonrasında merge edilip commit edildiğinde problem oluşmayacaktır.

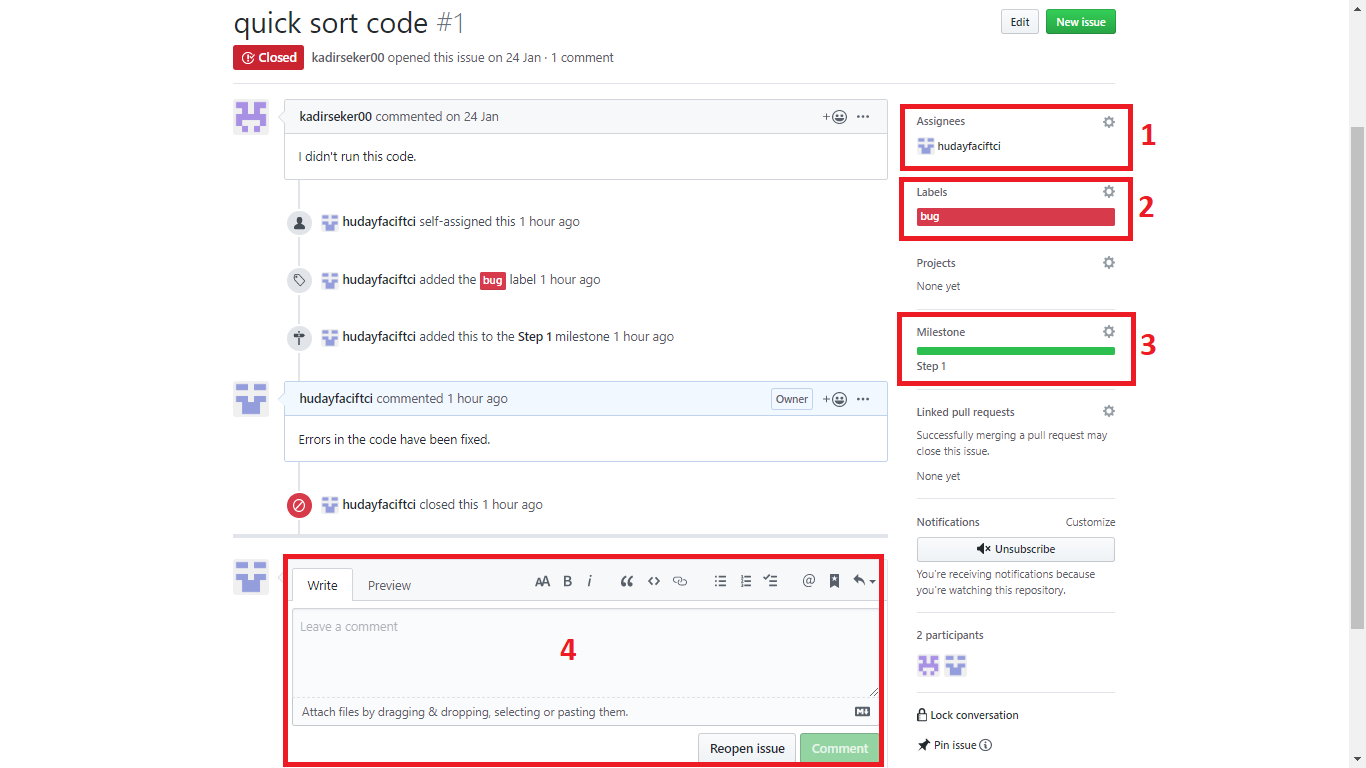


**12.ISSUE İŞLEMLERİ**

Repository adresindeki sekmelerde bulunan issue ya tıklanır. Orada bulunan “New Issue” butonuna tıklanarak yeni issue oluşturulur.



Sonrasında issue ya assignee, label ve milestone atanır.



Yukarıdaki resimde 1 numaralı gösterilen kısımdan “Assignee” atanır, 2 numaralı gösterilen kısımda problemin hangi Label(etiket) grubundan olduğu girilir, 3 numaralı gösterilen kısımda issue oluşturulan herhangi bir Milestone’a atanır ve son olarak 4 numaralı gösterilen kısıma issueya atanan Assignees’ler issue hakkında gelişmeleri yorum olarak belirtip problem çözüldükten sonra son yorum yapılıp “Closed Issue” butonuyla issue kapatılır. (4 numaralı kısımda “Reopen Issue” yazan yerde “Closed Issue” yazıyor ben issueyu kapattığım için öyle gösteriyor.)