



YAZILIM GELİŞTİRME ORTAM VE ARAÇLARI

**Üveys Selvi
HR210019**

Proje Özetim

Projenin amacı, oy verme sistemi üzerinde işlemler gerçekleştiren bir Java uygulaması geliştirmektir. Maven kullanarak oluşturulan proje, aşağıdaki işlemleri gerçekleştirecek metotları içermelidir:

`partiOyArttir(String partiAdi)`: Parametre olarak verilen partiye ait oyu bir artırır. Eğer parti için kayıt yoksa, yeni bir kayıt oluşturulur ve ardından oy artırılır.

`partiOyAzalt(String partiAdi)`: Parametre olarak verilen partiye ait oyu bir azaltır. Eğer parti için kayıt yoksa herhangi bir işlem yapılmaz veya hata verilir.

`partiOyListele()`: Tüm partilerin ve karşılık gelen oyların listesini döner.

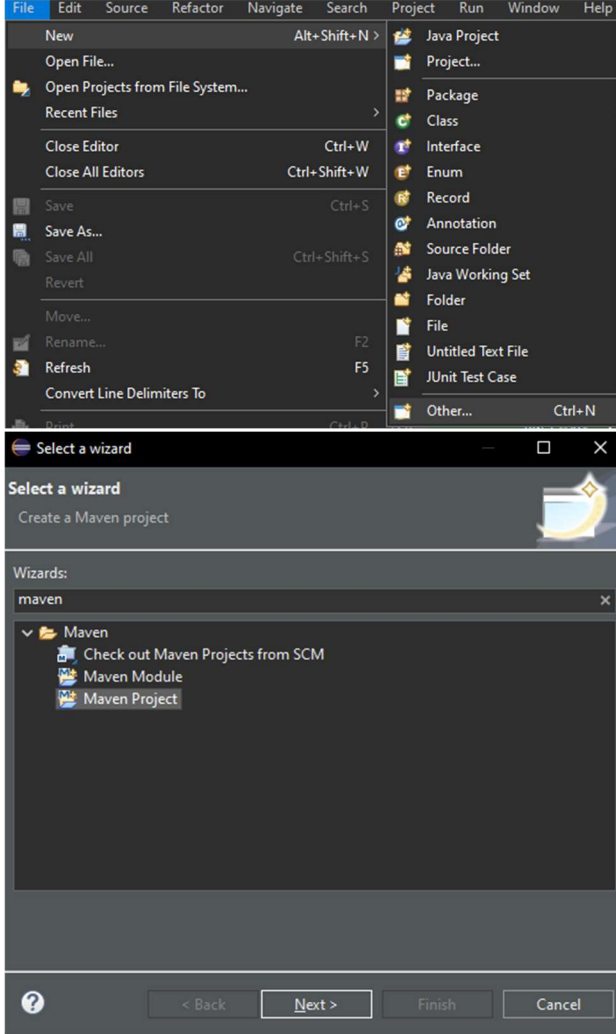
Partiler için sabit bir dizi kullanılabilir veya isteğe bağlı olarak nesneye yönelik programlama ilkelerine uygun olarak Parti adlı bir sınıf tanımlanabilir.

Proje aynı zamanda birim testlerini de içermelidir. Tüm metotlar için JUnit kullanarak birim testleri yazılmalı ve test kapsama oranı hesaplanarak rapora eklenmelidir.

Proje ismi, vize projesinde belirtilen il bilgisiyle birlikte tanımlanmalıdır. Örneğin, "TokatSecimApi" gibi bir isimlendirme kullanılabilir.

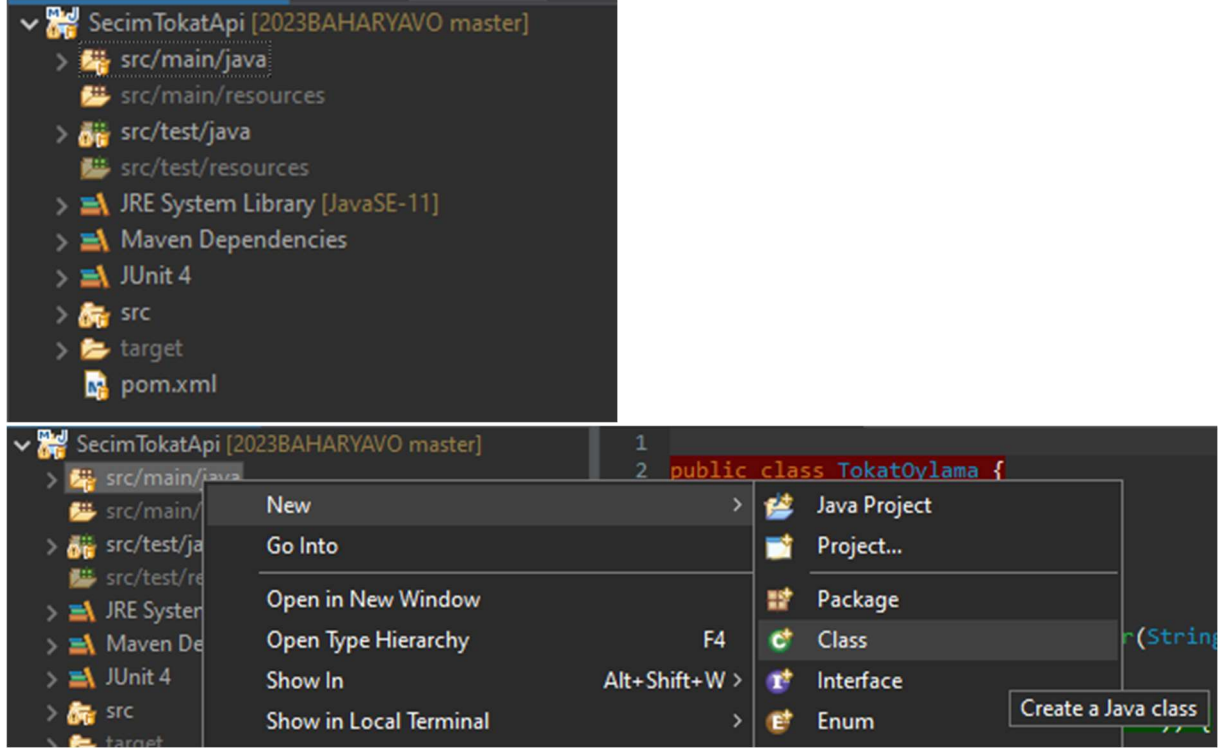
Proje Aşamaları

Eclipse IDE mizi açıp yeni SecimTokatApi adında bir maven projesi oluşturuyoruz



Uygun bir Group ID ve Artifact ID yazıyoruz ardından projemizi oluşturuyoruz.

Projemizi oluřturduktan sonra src-main-java altına yeni bir java class oluřturuyoruz



TokatOylama adında java class ıımızı açtıktan sonra proje ödevi için verilen bilgilere uyarak gerekli metotları yazıyoruz

```
public class TokatOylama {

    public static int partiAOy = 0;
    public static int partiBOy = 0;
    public static int partiCOy = 0;

    public static void partiOyArttir(String partiAdi) {
        if (partiAdi.equals("A")) {
            partiAOy++;
        } else if (partiAdi.equals("B")) {
            partiBOy++;
        } else if (partiAdi.equals("C")) {
            partiCOy++;
        }
    }

    public static void partiOyAzalt(String partiAdi) {
        if (partiAdi.equals("A")) {
            partiAOy--;
        } else if (partiAdi.equals("B")) {
            partiBOy--;
        } else if (partiAdi.equals("C")) {
            partiCOy--;
        }
    }

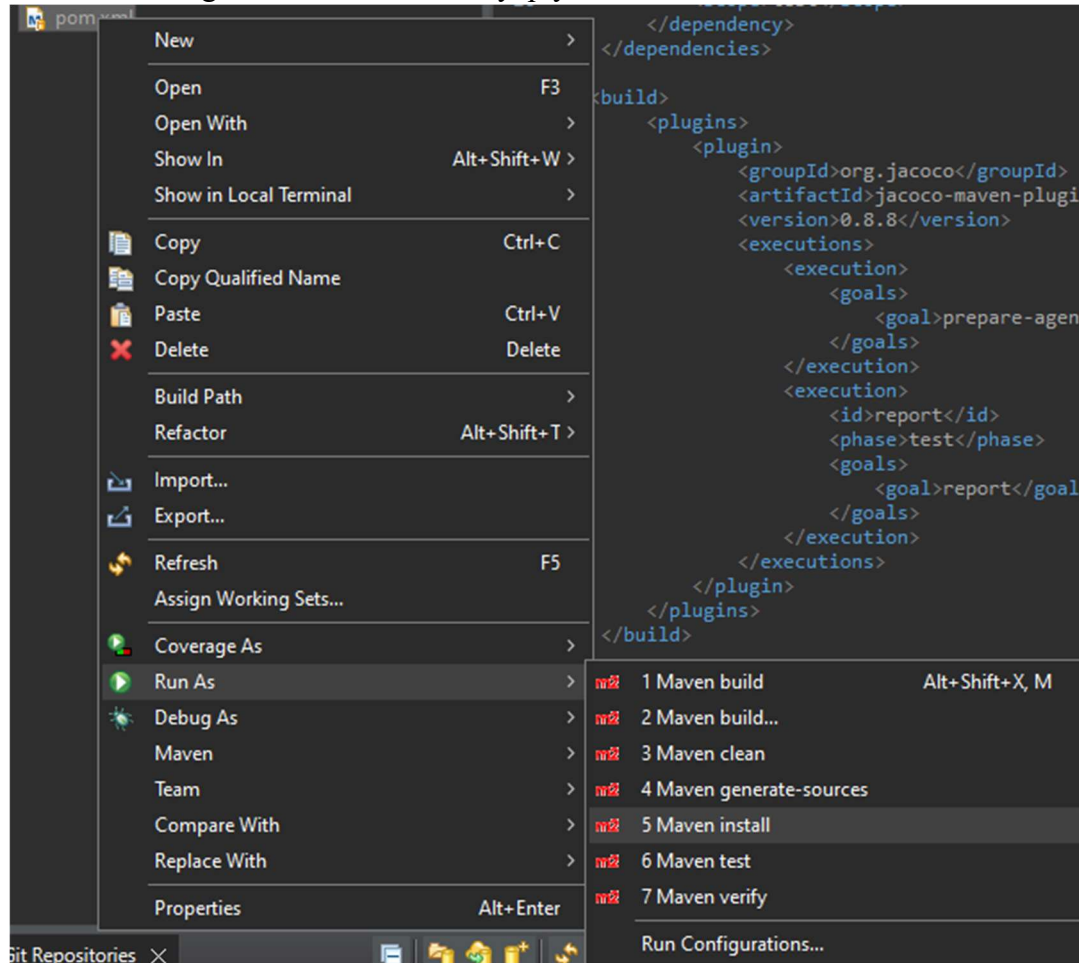
    //test
    public static String partiOyListele() {
        String yeni = "A partisi oy sayısı : " + TokatOylama.partiAOy + "B partisi oy sayısı : " + TokatOylama.partiBOy + "C partisi oy sayısı : " + TokatOylama.partiCOy;

        String[] dizi = new String[3];
        dizi[0] = "A partisi oy sayısı : " + TokatOylama.partiAOy ;
        dizi[1] = "B partisi oy sayısı : " + TokatOylama.partiBOy ;
        dizi[2] = "C partisi oy sayısı : " + TokatOylama.partiCOy ;
        return yeni;
    }
}
```

Java classımızı yazdıktan sonra pom.xml kısmına geçip dependencyleri ve pluginleri ekliyoruz

```
https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/
<modelVersion>4.0.0</modelVersion>
<groupId>tr.edu.medipol.tokat</groupId>
<artifactId>SecimTokatApi</artifactId>
<version>0.0.1-SNAPSHOT</version>
<properties>
<maven.compiler.source>11</maven.compiler.source>
<maven.compiler.target>11</maven.compiler.target>
</properties>
<dependencies>
<!-- https://mvnrepository.com/artifact/junit/junit -->
<dependency>
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>4.13.2</version>
<scope>test</scope>
</dependency>
</dependencies>
<build>
<plugins>
<plugin>
<groupId>org.jacoco</groupId>
<artifactId>jacoco-maven-plugin</artifactId>
<version>0.8.8</version>
<executions>
<execution>
<goals>
<goal>prepare-agent</goal>
</goals>
</execution>
<execution>
<id>report</id>
<phase>test</phase>
<goals>
<goal>report</goal>
</goals>
</execution>
</executions>
</plugin>
</plugins>
</build>
</project>
```

Ve ardından sağ tık run maven install yapıyoruz.



Ve build sonucunu bekliyoruz

```
[INFO] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build i
[INFO] Copying 0 resource
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ SecimTokatApi ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is plat
[INFO] Compiling 1 source file to C:\Users\selvi\git\2023BAHARYAVO\SecimTokatApi\target\cla
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ SecimTokatApi
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build i
[INFO] Copying 0 resource
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ SecimTokatApi ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is plat
[INFO] Compiling 1 source file to C:\Users\selvi\git\2023BAHARYAVO\SecimTokatApi\target\tes
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ SecimTokatApi ---
[INFO] Surefire report directory: C:\Users\selvi\git\2023BAHARYAVO\SecimTokatApi\target\sur

-----
T E S T S
-----
Running TokatOylamaTest
Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.113 sec

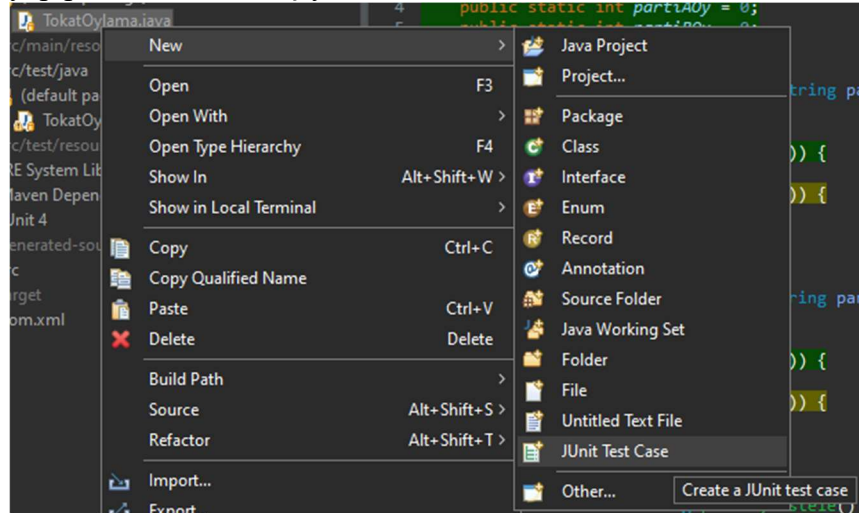
Results :

Tests run: 7, Failures: 0, Errors: 0, Skipped: 0

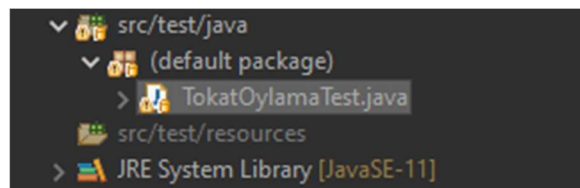
[INFO] --- jacoco-maven-plugin:0.8.8:report (report) @ SecimTokatApi ---
[INFO] Loading execution data file C:\Users\selvi\git\2023BAHARYAVO\SecimTokatApi\target\ja
[INFO] Analyzed bundle 'SecimTokatApi' with 1 classes
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ SecimTokatApi ---
[INFO] Building jar: C:\Users\selvi\git\2023BAHARYAVO\SecimTokatApi\target\SecimTokatApi-0.
[INFO] --- maven-install-plugin:2.4:install (default-install) @ SecimTokatApi ---
[INFO] Installing C:\Users\selvi\git\2023BAHARYAVO\SecimTokatApi\target\SecimTokatApi-0.0.1
[INFO] Installing C:\Users\selvi\git\2023BAHARYAVO\SecimTokatApi\pom.xml to C:\Users\selvi\
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.479 s
```

BUILD SUCCESS yazısını gördükten sonra işlemin başarılı olduğunu görüp devam edebiliriz.

Bu aşamadan sonra Junit testlerini gerçekleştirmek için açtığımız Java Classın üstüne sağ tık yapıp Junit test case açıyoruz.



Junit test classımızı oluşturduktan sonra src-test-java altında oluşuyor



Testlerimizi bu kısımda yazmaya başlıyoruz

```
import static org.junit.Assert.*;

public class TokatOylamaTest {

    @Test
    public void testAPartiOyArttir() {
        TokatOylama.partiOyArttir("A");

        int gercekSonuc = 1;

        assertEquals(gercekSonuc, TokatOylama.partiAOy);
    }

    @Test
    public void testBPartiOyArttir() {
        TokatOylama.partiOyArttir("B");

        int gercekSonuc = 1;

        assertEquals(gercekSonuc, TokatOylama.partiBOy);
    }

    @Test
    public void testCPartiOyArttir() {
        TokatOylama.partiOyArttir("C");

        int gercekSonuc = 1;

        assertEquals(gercekSonuc, TokatOylama.partiCOy);
    }

    @Test
    public void testAPartiOyAzalt() {
        TokatOylama.partiOyAzalt("A");
        int gercekSonuc = 0;
        assertEquals(gercekSonuc, TokatOylama.partiAOy);
    }

    @Test
    public void testBPartiOyAzalt() {
        TokatOylama.partiOyAzalt("B");
        int gercekSonuc = 0;
        assertEquals(gercekSonuc, TokatOylama.partiBOy);
    }
}
```

```
        TokatOylama.partiOyAzalt("B");
        int gercekSonuc = 0;
        assertEquals(gercekSonuc, TokatOylama.partiBOy);
    }

    @Test
    public void testCPartiOyAzalt() {
        TokatOylama.partiOyAzalt("C");
        int gercekSonuc = 0;
        assertEquals(gercekSonuc, TokatOylama.partiCOy);
    }

    @Test
    public void testOylarListele() {
        String yeni = "A partisi oy sayısı : " + TokatOylama.partiAOy + "B partisi oy sayısı : " + TokatOylama.partiBOy + "C partisi oy sayısı : " + TokatOylama.partiCOy;

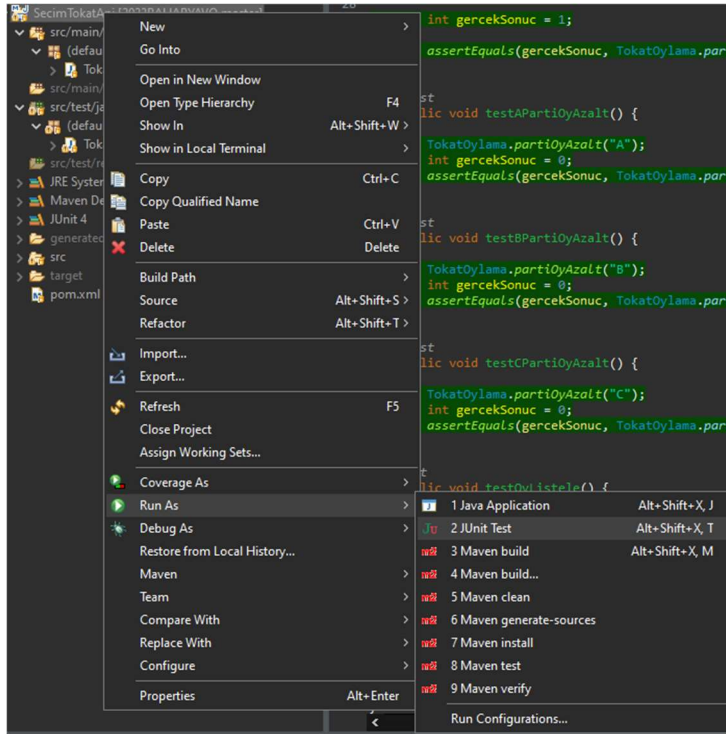
        String[] dizi = new String[3];
        dizi[0] = "A partisi oy sayısı : " + 0;
        dizi[1] = "B partisi oy sayısı : " + 0;
        dizi[2] = "C partisi oy sayısı : " + 0;

        String[] dizi1 = new String[3];
        String data = TokatOylama.partiOylarListele();

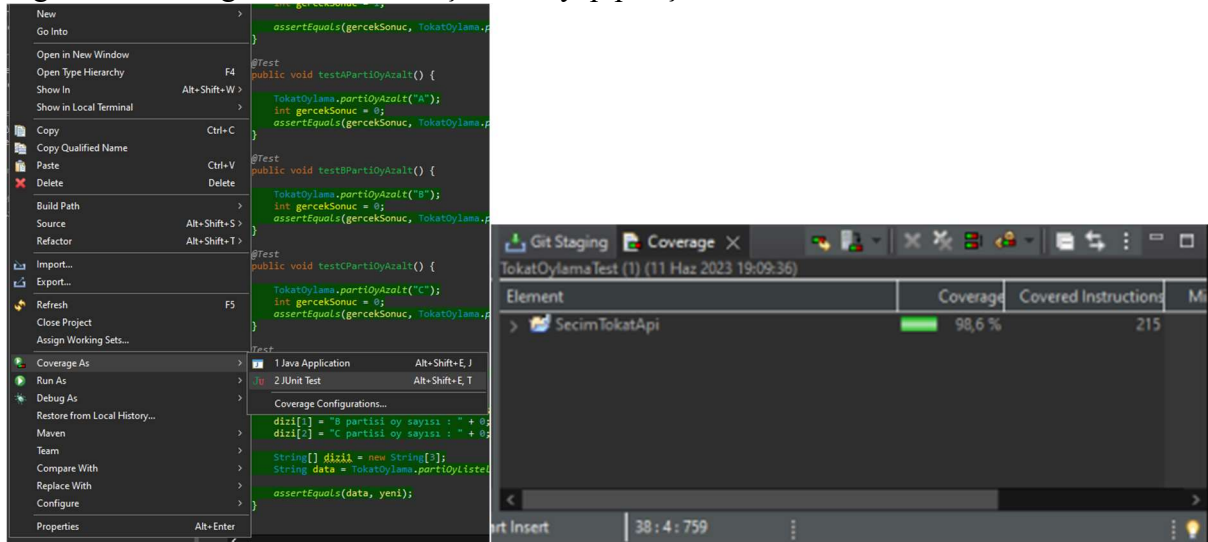
        assertEquals(data, yeni);
    }
}
```

Arttırma, Azaltma ve Listeleme testlerimizi burada yapıyoruz. Kodumuz bittikten sonra Junit testlerine başlayabiliriz.

Junit testlerine başlamak için: Sağ tık -> Run as -> Junit Test yapıyoruz.

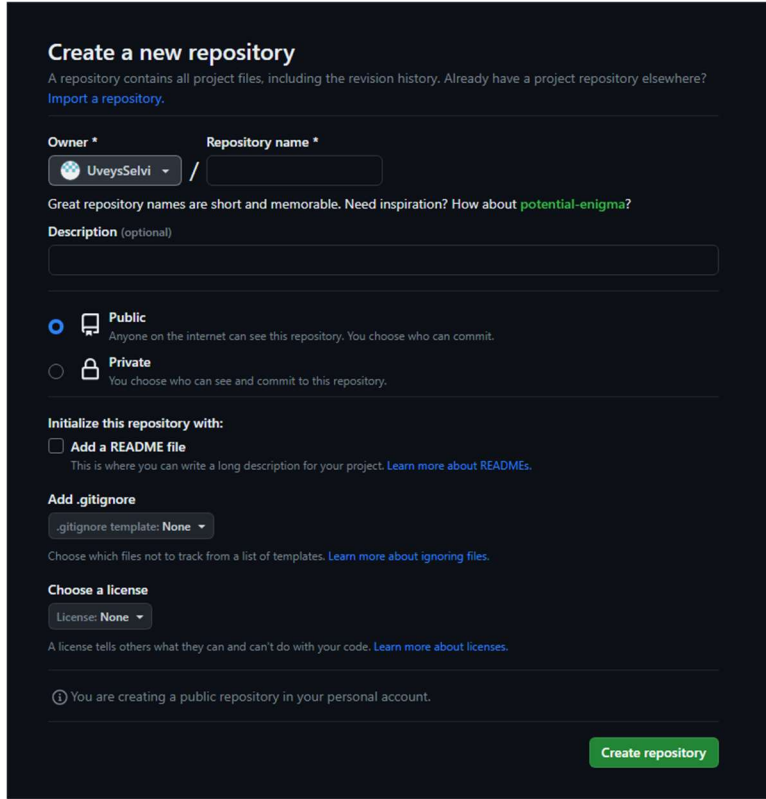


Testlerimizi başarılı bir şekilde tamamlayıp aldıktan sonra coverage testi yapabiliriz bu testide sağ tık -> coverage as -> Junit test şeklinde yapıp % şeklinde test sonucunu alabiliriz.



Bu aşamaları atladıktan sonra GitHub kısmına geçebiliriz.

Github.com u açıp yeni bir repository oluşturuyoruz



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * UveysSelvi / Repository name *

Great repository names are short and memorable. Need inspiration? How about [potential-enigma?](#)

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore
.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

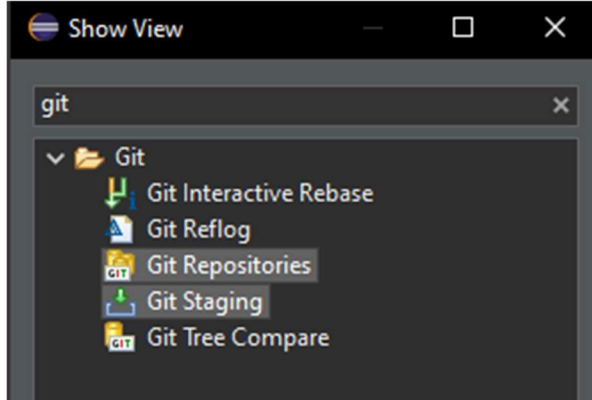
Choose a license
License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

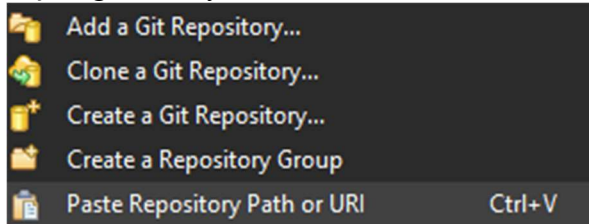
① You are creating a public repository in your personal account.

[Create repository](#)

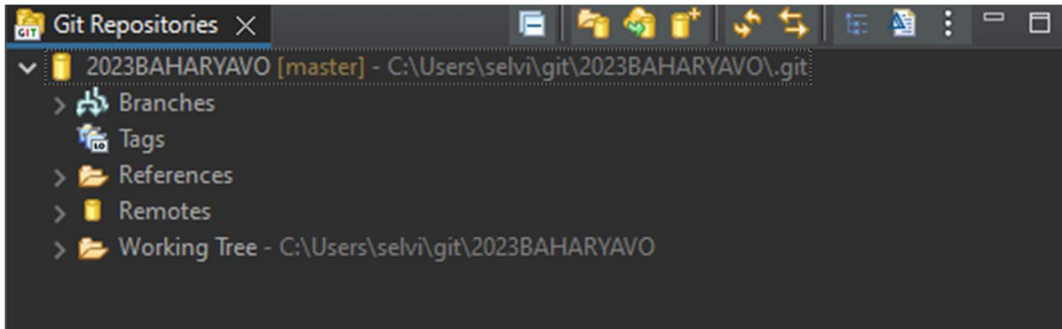
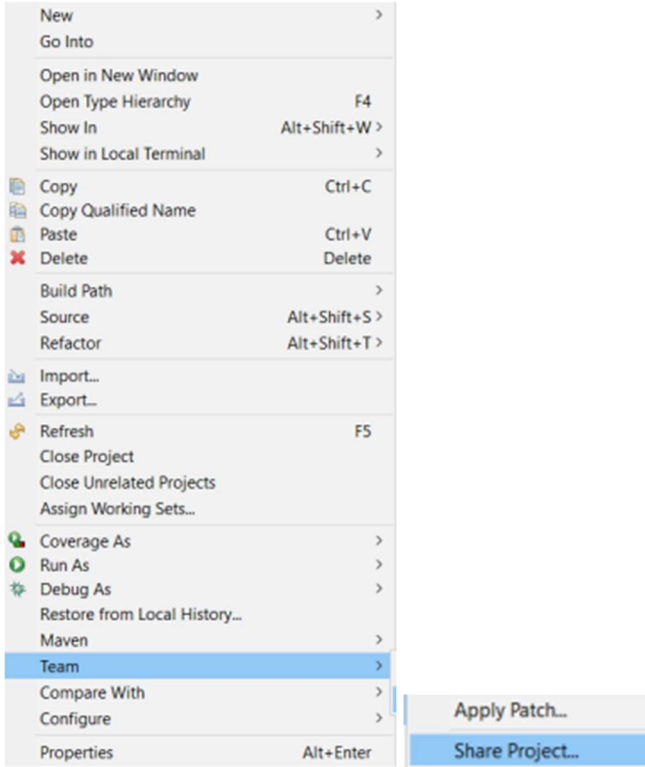
oluşturduktan sonra repository adresimizi kopyalayıp Eclipse te show view kısmından iki adet git bölümü ekliyoruz bunlar ;



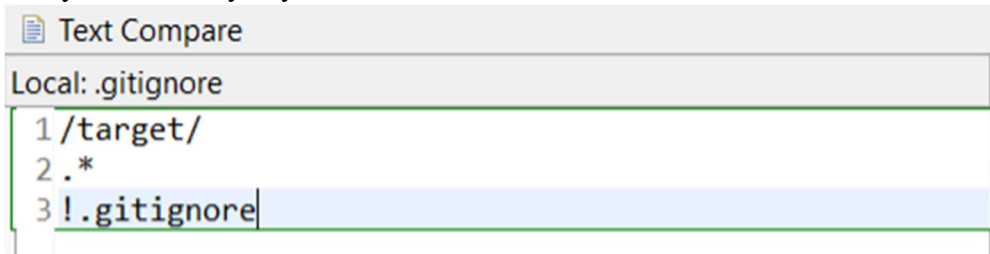
Daha sonra git repositories bölümünde sağ tık yapıp repository adresimizi yapıştırma seçeneğine basıyoruz.



Ve sağ tık yapıp team bölümünden paylaşıyoruz ardından github Kullanıcı adı- Şifremizi girerek repositorymizi oluşturuyoruz



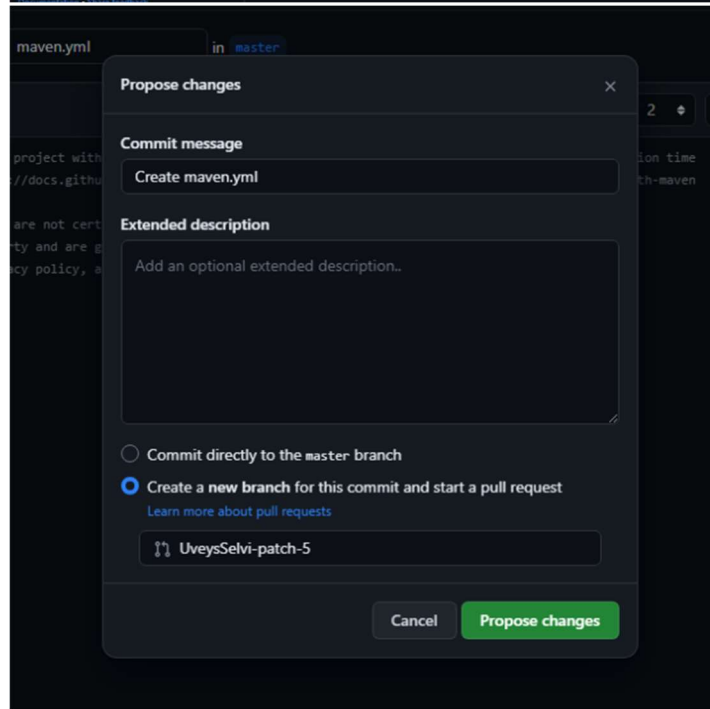
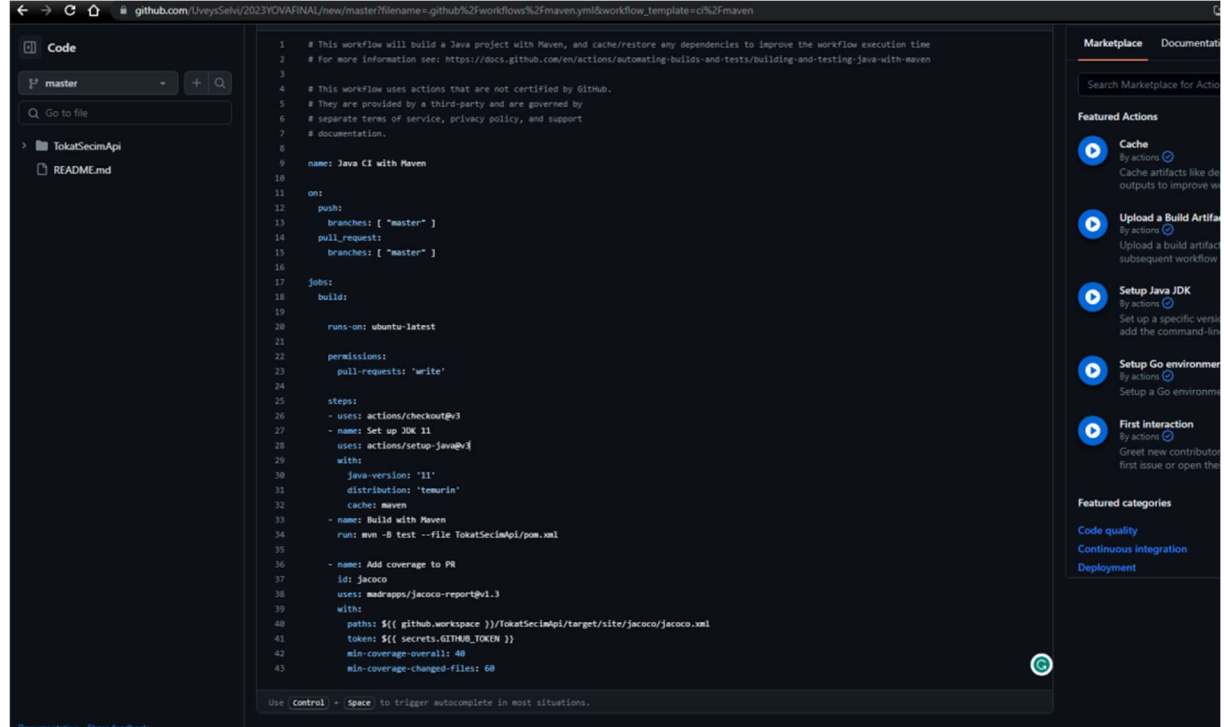
Daha sonra Git Staging bölümündeki gereksiz dosyaları kaldırmak için .gitignore dosyasına tıklayarak 3 satır yazıyoruz bunlar;



daha sonradan kalan dosyaları staged changes tarafına geçirip commit yapıyoruz ve dosyalar githuba geçmiş oluyor



ardından projemize workflow ekliyoruz ve açılan maven.yml dosyasını gerekli şekilde dolduruyoruz ve sonraki aşamaları yapıyoruz bu aşamalar şu şekildedir ;



Add more commits by pushing to the **ovejservice/patch-5** branch on **ovejservice/2025TOVA/initial**.

Add more commits by pushing to the **UveysSelvi-patch-1** branch on **UveysSelvi/2023BAHARYAVO**

Bu işlemlerin ardından Readme dosyasını gönderebiliriz Readme dosyasının aşamalarında sırası ile şu şekildedir;

The image shows a GitHub pull request interface. At the top, the repository is 'UveysSelvi / 2023BAHARYAVO'. The pull request is titled 'Create README.md #2'. The 'UveysSelvi' user wants to merge 1 commit into 'master' from 'UveysSelvi-patch-2'. The pull request has 0 conversations, 1 commit, 0 checks, and 1 file changed. The 'UveysSelvi' user commented 'No description provided.' The 'github-actions' bot commented 'There is no coverage information present for the Files changed' and showed a 'Total Project Coverage' of 96.63%. The pull request has a 'Verified' status and a 'Merge pull request' button. The bottom section shows the pull request successfully merged and closed, with a 'Delete branch' button.

UveysSelvi / 2023BAHARYAVO Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

2023BAHARYAVO / README.md in master

Edit Preview

1 # 2023BAHARYAVO
2 TEST

Cancel changes Commit changes

Spaces 2 No wrap

Create README.md #2

Open UveysSelvi wants to merge 1 commit into master from UveysSelvi-patch-2

Conversation 0 Commits 1 Checks 0 Files changed 1

UveysSelvi commented now

No description provided.

Create README.md Verified ✓ cba12e0

github-actions bot commented now

There is no coverage information present for the Files changed

Total Project Coverage 96.63%

Add more commits by pushing to the UveysSelvi-patch-2 branch on UveysSelvi/2023BAHARYAVO.

Require approval from specific reviewers before merging
Branch protection rules ensure specific people approve pull requests before they're merged. Add rule

All checks have passed
1 successful check Show all checks

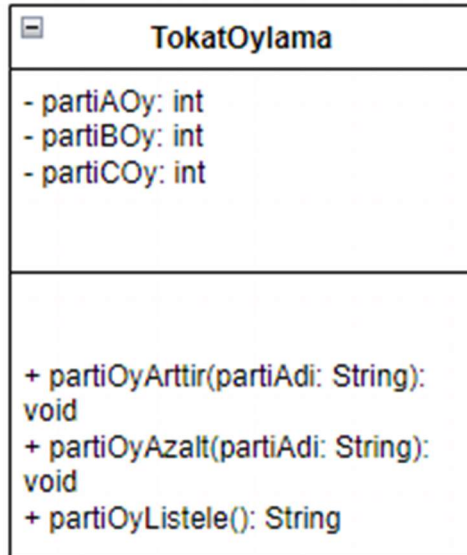
This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Pull request successfully merged and closed
You're all set—the UveysSelvi-patch-2 branch can be safely deleted. Delete branch

Ekran Görüntüleri projeyi bitirdikten sonra alındığı için bazı aşamaları tam gösterememiş olabilirim bunun için oluşan eksikliklerden özür diliyorum.

UML Diyagramlar



Use Case Diagramı

