



Yiğit Öner

H5210016

Yazılım Geliştirme Ortam ve Araçları

2022-2023 Final Raporu

Proje Tanımı ve Amacı

Projenin Amacı

Proje 14 Mayıs 2023 tarihinde yapılacak olan Türkiye Büyük Millet Meclisi üyelerinin seçilmesi (Milletvekili) için kullanılması öngörülen bir projedir . Milletvekili oylarının sisteme giriş yapılmasını sağlayan Maven projesidir.

Projenin istenilen amacı : 14 Mayıs 2023 tarihinde yapılacak olan Milletvekili seçiminin **Kayseri ilindeki** vatandaşların kullandığı oyları sayarak sisteme oylara giriş yapacak kişinin(Sandık başkanı) Mevcut oyları görüntüleme , Oy ekleme (birer birer) , Oy azaltma (birer birer) işlemlerini yerine getirmesini sağlaması amaçlanmıştır.

Projenin Tanımı

Proje'nin tanımı: Türkiye Cumhuriyeti Vatandaşlarının Milletvekili seçimlerinde oy verdikleri partinin belirlemiş olduğu Adayların Meclise girebilmesini mümkün kılan bu seçimin simülasyonudur. Bu yapı sayesinde oy damgası üzerinde geçerli olan pusulaların kaybolma , geçersiz hale gelmesi vs. gibi durumlara karşı önlem alınmış olunacaktır

Kullanım tasarımı şu şekilde olması planlanmıştır: Oy sandığı başındaki gerekli yetkili merci sisteme giriş yaparak oy sayım anında hangi partide kaç oy almış ise bunu sisteme kayıt ederek resmi hale getirmiş olacaktır.

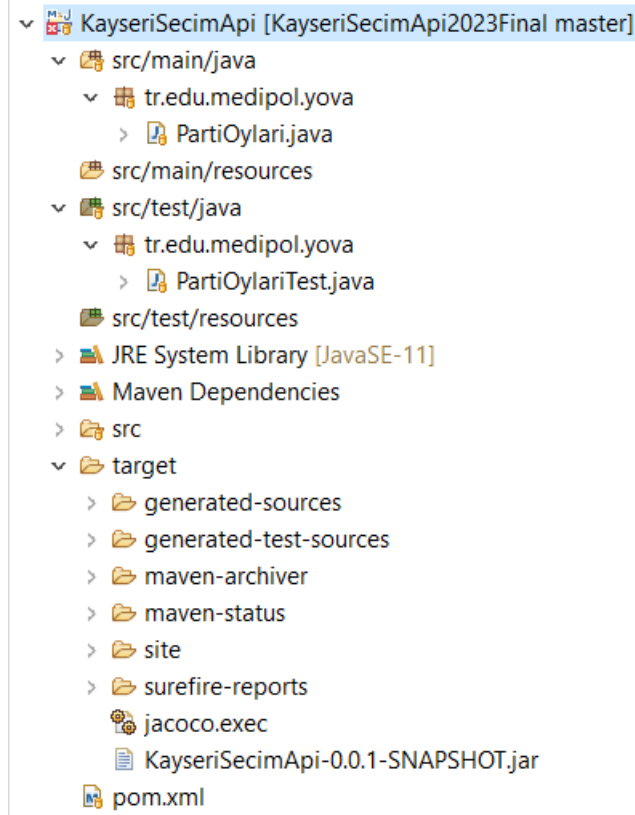
Kayseri İline ait geçmiş bilgiler

[2017 Türkiye anayasa değişikliği referandumu](#) ile ülkenin yönetim şekli başkanlık sistemi olarak belirlendi. 16 Mart 2018 tarihinde, 26 maddelik "Seçimlerin Temel Hükümleri ve Seçmen Kütükleri Hakkında Kanun ile Bazı Kanunlarda Değişiklik Yapılmasına Dair Kanun" yürürlüğe girdi. Kanunla, seçimlere katılma yeterliliği taşıyan siyasi partilerin ittifak yaparak milletvekili seçimine katılmalarına imkân tanınırken, siyasi partilerin seçimlerde başka bir siyasi partiyi destekleme kararı almalarını yasaklayan hüküm kaldırıldı.

Kasım 2017'de [Milliyetçi Hareket Partisi](#) Genel Başkanı [Devlet Bahçeli](#)'nin [Adalet ve Kalkınma Partisi](#) ile ittifak yapmayı önermesi sonucunda Şubat 2018'de [Cumhur İttifakı](#) kuruldu.

Projede Kodları ve Açıklamaları

1) Package Explorer



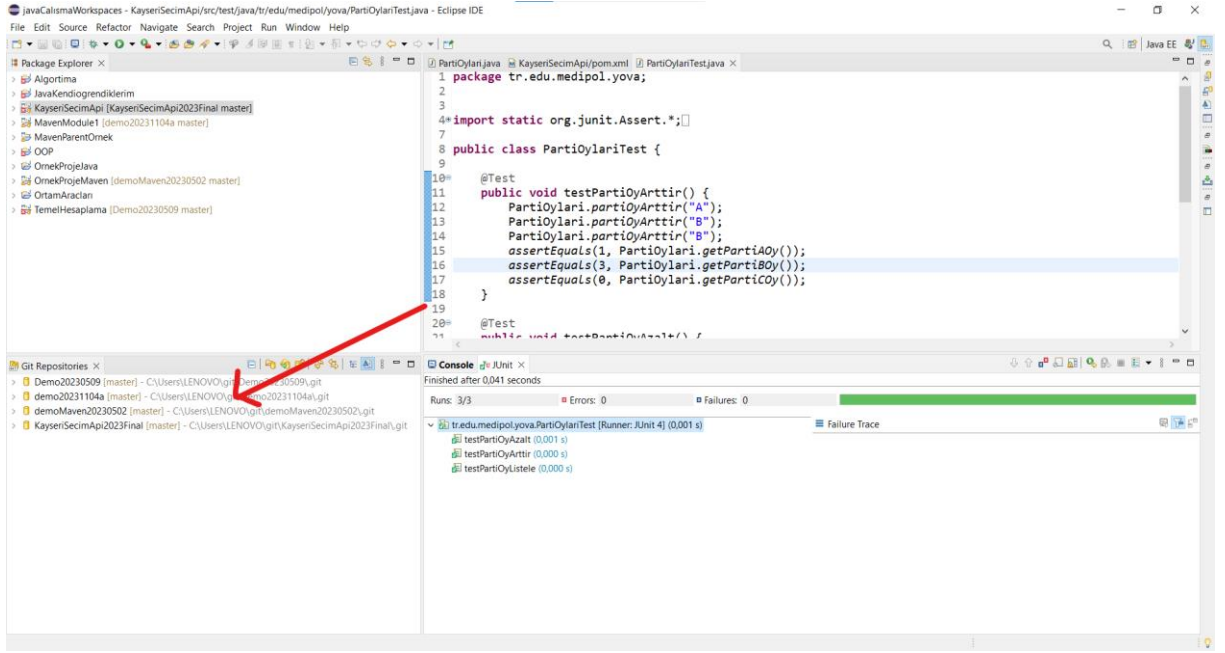
Açıklama:

Bu paket hiyerarşisi KayseriSecimApi Maven projesinin main yani ana sınıfların bulunduğu pakettir. tr.edu.medipol.yova Projenin main altında bulunan pakettir altında PartiOylari sınıfı bulunmaktadır. Bu sınıfta A B C partilerinin değer ataması , Oy arttırma (birer birer), Oy azaltma (birer birer) ve mevcut oyların gösterildiği metotlar bulunmaktadır .

Ayrıca partilerin mevcut oylarını çağırabilmek için bulunun get metotdu ile partilerin mevcut oylarının çağırma işlemi nesne yönelimli programlama için eklenmiştir. Test dizini altında bulunan PartiOylariTest sınıfı ise mevcut olan 3 partinin Oy arttırma (birer birer) , Oy azaltma (birer birer) , mevcut oyların gösterildiği metotların testlerinin bulunduğu sınıftır. Asıl amacı oluşturulan her sınıf içerisindeki metotların birim testinin gerçekleşerek doğru çıktı verip vermediğini doğrulamak ve analiz etmektir (test kapsama oranını kıyaslayarak).

Maven Dependencies eklentisi ise Projenin ihtiyaç duyduğu eklentiler sınıfının senkronizasyonunu sağlama aşamasında yardımcı olmaktadır alt dizinlerinde projeye ait ihtiyaç duyulan sınıflara ulaşılmasını sağlamaktadır.

2) GitHub Adresinin Eclipse içerisinde Eklenebilir



Açıklama:

GitHub Eclipse Paneli, Eclipse IDE'nin içine entegre edilmiş bir eklentidir. Bu panel, GitHub ile entegrasyon sağlar ve projelerinizi doğrudan Eclipse üzerinden GitHub ile senkronize etmemize olanak sağlamaktadır.

Kullanılan Repository : [KayseriSecimApi2023Final](#)

Eclipse GitHub Paneli'nin birkaç nedeni vardır:

Eclipse GitHub Paneli, geliştiricilerin GitHub üzerindeki projeleri kolayca yönetmesini, işbirliği yapmasını ve sürüm kontrolü sağlamasını amaçlayan kullanışlı bir eklentidir. Bu sayede, projelerinizi daha etkin bir şekilde geliştirebilir ve yönetebilirsiniz.

3) KayseriSecimApi/pom.xml dosyası

a) 1.0

```
PartiOylari.java KayseriSecimApi/pom.xml x PartiOylariTest.java
1<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/xsd/maven-4.0.0.xsd"
2  <modelVersion>4.0.0</modelVersion>
3
4  <groupId>tr.edu.medipol.yova</groupId>
5  <artifactId>KayseriSecimApi</artifactId>
6  <version>0.0.1-SNAPSHOT</version>
7
8  <properties>
9    <maven.compiler.source>11</maven.compiler.source>
10   <maven.compiler.target>11</maven.compiler.target>
11 </properties>
12
```

Açıklama:

Bu kod, Maven proje yönetim aracı için bir POM (Proje Nesne Modeli) dosyasının bir bölümünü temsil eder. POM dosyası, Maven tarafından kullanılan bir XML dosyası formatıdır ve bir projenin yapılandırma bilgilerini içerir.

- ``<modelVersion>4.0.0</modelVersion>``: POM dosyasının model sürümünü belirtir. Maven 4.0.0 sürümünü kullandığını gösterir.

- ``<groupId>tr.edu.medipol.yova</groupId>``: Projenin grubunun kimliğini belirtir. Grup kimliği, projeye ilişkili olan organizasyon, kuruluş veya kuruluş birimini tanımlar.

- ``<artifactId>KayseriSecimApi</artifactId>``: Projenin sanal bir varlık (artifact) kimliğini belirtir. Artifact kimliği, projenin adını veya ismini temsil eder. Bu örnekte, projenin adı "KayseriSecimApi" olarak belirtilmiştir.

- ``<version>0.0.1-SNAPSHOT</version>``: Projenin sürüm numarasını belirtir. Sürüm numarası, projenin belirli bir sürümünü tanımlar. Bu örnekte, "0.0.1-SNAPSHOT" sürümü kullanılmıştır. "SNAPSHOT" terimi, henüz tamamlanmamış, geliştirme aşamasında olan bir sürümü ifade eder.

- ``<properties>``: Proje özelliklerini tanımlayan bir bölümdür. Bu bölümde, Maven derleyicisi için kaynak ve hedef Java sürümlerini belirtmek için ``maven.compiler.source`` ve ``maven.compiler.target`` özellikleri tanımlanmıştır. Bu örnekte, Java 11 sürümü kullanıldığı belirtilmiştir.

b) 2.0

```
13<⌵    <dependencies>
14<⌵        <dependency>
15                <groupId>junit</groupId>
16                <artifactId>junit</artifactId>
17                <version>4.13.2</version>
18                <scope>test</scope>
19        </dependency>
20    </dependencies>
21
```

Açıklama :

Bu kod, bir Maven POM dosyasının ``<dependencies>`` bölümünü temsil eder. Bu bölüm, projenin bağımlılıklarını tanımlamak için kullanılır.

- ``<dependency>``: Bağımlılıkları tanımlayan bir etikettir. İçinde yer alan diğer etiketler, bir bağımlılığın detaylarını belirtir.

- ``<groupId>junit</groupId>``: Bağımlılığın grubunun kimliğini belirtir. Bu durumda, JUnit kütüphanesinin "junit" grubuna ait olduğunu gösterir.

- ``<artifactId>junit</artifactId>``: Bağımlılığın sanal bir varlık (artifact) kimliğini belirtir. Bu durumda, JUnit kütüphanesinin "junit" adını taşıdığını belirtir.

- ``<version>4.13.2</version>``: Bağımlılığın kullanılacak sürüm numarasını belirtir. Bu örnekte, JUnit kütüphanesinin 4.13.2 sürümü kullanılacağı belirtilmiştir.

- ``<scope>test</scope>``: Bağımlılığın kapsamını belirtir. Bu durumda, JUnit bağımlılığının yalnızca test sınıflarında kullanılacağını gösterir. Bu, JUnit'ın sadece testlerin derlenmesi ve çalıştırılması aşamasında gerekli olduğunu ifade eder.

c) 3.0

```
22<=    <build>
23<=        <plugins>
24<=            <plugin>
25<=                <groupId>org.jacoco</groupId>
26<=                <artifactId>jacoco-maven-plugin</artifactId>
27<=                <version>0.8.8</version>
28<=                <executions>
29<=                    <execution>
30<=                        <goals>
31<=                            <goal>prepare-agent</goal>
32<=                        </goals>
33<=                    </execution>
34<=                <execution>
35<=                    <id>report</id>
36<=                    <phase>test</phase>
37<=                    <goals>
38<=                        <goal>report</goal>
39<=                    </goals>
40<=                </execution>
41<=            </executions>
42<=        </plugin>
43<=    </plugins>
44<= </build>
45
46 </project>
```

Açıklama:

Bu kod parçası, Maven proje yapılandırmasının ``<build>`` bölümünü temsil eder. ``<build>`` bölümü, projenin derleme ve çalıştırma işlemleri için yapılandırma bilgilerini içerir.

Jacoco Maven eklentisinin projeye dahil edildiğini gösterir. Jacoco, test kapsamı kod kapsamı raporlaması sağlar ve testlerin kodunun ne kadarının test edildiğini ölçmek için kullanılır. Bu eklenti, ``prepare-agent`` hedefiyle testlerin çalıştırılması sırasında kod kapsamını izler ve ``report`` hedefiyle test sonuçlarını raporlar.

Bu şekilde, Jacoco Maven eklentisi projenin test sürecini geliştirmek ve **kod kapsamını ölçmek** için kullanılır.

- 4) PartiOylari.java Sınıfı kodları
a) 1.0

```
PartiOylari.java x KayseriSecimApi/pom.xml PartiOylariTest.java
1 package tr.edu.medipol.yova;
2
3 public class PartiOylari {
4
5     private static int partiAOy = 0;
6     private static int partiBOy = 0;
7     private static int partiCOy = 0;
8
9
10    public static void partiOyArttir(String partiAdi) {
11        if(partiAdi.equals("A")) {
12            partiAOy++;
13        }
14        else if(partiAdi.equals("B")) {
15            partiBOy++;
16        }
17        else if(partiAdi.equals("C")) {
18            partiCOy++;
19        }
20    }
21 }
22
```

Açıklama:

Bu kod parçası : 1'inci satırda hangi pakette olduğunu belirten koddur. 5-7'nci satırlar arasında bulunan kodlar mevcut partilerin oylarının atamasının yapıldığı yerdir.

10-21'inci satırlar arasında bulunun **partiOyArttir** metodu mevcut oyların arttırma işlevinin yerine getirildiği yerdir eğer String **partiAdi** doğru biçimde girilirse metot çalışır ve 11-18 arasındaki koşul döngülerine yönelir.

Eğer partiAdi A girilmiş ise partiAOy arttırılır ve değeri atanır (İfade ++).Eğer Doğru değil ise B koşulu olup olmadığı kontrol edilir ve partiBOy arttırılır ve değeri atanır.

Son olarak eğer girilen parti b de değilse C partisi olup olmadığı kontrol edilir ve son olarak bu koşula girilip partiCOy arttırılır ve değeri atanır.

b) 2.0

```
23 public static void partiOyAzalt(String partiAdi) {
24     if(partiAdi.equals("A")) {
25         if(partiAOy!=0 || partiAOy>0) {
26             partiAOy--;
27         }
28         else {
29             }
30     }
31 }
32 else if(partiAdi.equals("B")) {
33     if(partiBOy!=0 || partiBOy>0) {
34         partiBOy--;
35     }
36     else {
37         }
38 }
39 }
40 else if(partiAdi.equals("C")) {
41     if(partiCOy!=0 || partiCOy>0) {
42         partiCOy--;
43     }
44     else {
45         }
46 }
47 }
48 }
49 }
```

Açıklama:

Bu kod, bir Java sınıfının içinde bulunan `partiOyAzalt` adlı bir metodu temsil eder. Bu metod, belirli bir parti adına göre oyları azaltmak için kullanılır. Belirli bir parti adına göre oyları azaltmak için kullanılan bir metodu içermektedir. Hangi parti adının girildiğine bağlı olarak ilgili partiye ait oy sayısı azaltılır.

1. `partiOyAzalt` adında bir metod tanımlanmıştır. Bu metod, `String` tipinde bir `partiAdi` parametresi alır.

2. İlk `if` bloğunda, `partiAdi` değişkeni "A" olarak belirtilen partiyle eşit mi diye kontrol edilir.

3. Eğer `partiAdi` "A" ise, 21'inci satırdaki `if` bloğuna girilir ve `partiAOy` değişkeni 0'dan farklı veya 0'dan büyük mü diye kontrol edilir.

4. Eğer `partiAOy` 0'dan farklı veya 0'dan büyük ise, `partiAOy` değişkeni bir azaltılır (`partiAOy--` ifadesi).

5. Diğer partiler için benzer bir yapı kullanılır. 32'nci satırdaki `else if` bloğu "B" parti adı için kontrol yapar ve `partiBOy` değişkenini azaltır. Üçüncü `else if` 40'ınci satırdaki blok ise "C" parti adı için kontrol yapar ve `partiCOy` değişkenini azaltır.

c) 3.0

```
51 public static String partiOyListele() {  
52     return "Parti A: " + partiAOy + "\n" +  
53         "Parti B: " + partiBOy + "\n" +  
54         "Parti C: " + partiCOy + "\n";  
55 }  
56  
57 public static int getPartiAOy() {  
58     return partiAOy;  
59 }  
60  
61 public static int getPartiBOy() {  
62     return partiBOy;  
63 }  
64  
65 public static int getPartiCOy() {  
66     return partiCOy;  
67 }  
68
```

Açıklama:

Bu kod grubu, parti oylarını listelemek ve her bir partiye ait oy sayısını elde etmek için kullanılan metotları içerir. Parti oylarının listelenmesi ve her bir partiye ait oy sayılarının elde edilmesi gibi işlevleri gerçekleştirmek için kullanılır. `partiOyListele` metodu, oyları okunabilir bir formatta listelerek kullanıcıya sunar, `getPartiAOy`, `getPartiBOy` ve `getPartiCOy` metotları ise her bir partiye ait oy sayılarını diğer kod parçalarından elde etmek için kullanılabilir.

1. `partiOyListele` metodu:

- Bu metot, parti oylarını bir metin olarak döndürür.
- Dönen metinde "Parti A:", "Parti B:" ve "Parti C:" ifadeleriyle birlikte ilgili partiye ait oy sayıları yer alır.
- Metin, satır satır olacak şekilde oluşturulur ve her parti ve oy sayısı **arasında bir satır atlama** yapılır.

2. `getPartiAOy` metodu:

- Bu metot, Parti A'ya ait oy sayısını döndürür.
- `partiAOy` değişkeninin değerini döndürür.

3. `getPartiBOy` metodu:

- Bu metot, Parti B'ye ait oy sayısını döndürür.
- `partiBOy` değişkeninin değerini döndürür.

4. `getPartiCOy` metodu:

- Bu metot, Parti C'ye ait oy sayısını döndürür.
- `partiCOy` değişkeninin değerini döndürür.

5) PartiOylariTest.java Sınıfın kodları

```
1 package tr.edu.medipol.yova;
2
3
4 import static org.junit.Assert.*;
5
6 public class PartiOylariTest {
7
8     @Test
9     public void testPartiOyArttir() {
10         PartiOylari.partiOyArttir("A");
11         PartiOylari.partiOyArttir("B");
12         PartiOylari.partiOyArttir("B");
13         assertEquals(1, PartiOylari.getPartiAOy());
14         assertEquals(3, PartiOylari.getPartiBOy());
15         assertEquals(0, PartiOylari.getPartiCOy());
16     }
17
18     @Test
19     public void testPartiOyAzalt() {
20         PartiOylari.partiOyArttir("A");
21         PartiOylari.partiOyArttir("B");
22         PartiOylari.partiOyAzalt("A");
23         assertEquals(0, PartiOylari.getPartiAOy());
24         assertEquals(1, PartiOylari.getPartiBOy());
25         assertEquals(0, PartiOylari.getPartiCOy());
26     }
27
28     @Test
29     public void testPartiOyListele() {
30         String expected = "Parti A: 1\n" +
31             "Parti B: 3\n" +
32             "Parti C: 0\n";
33         assertEquals(expected, PartiOylari.partiOyListele());
34     }
35 }
36
37
38
39
40
```

Açıklama:

"PartiOylari" adlı bir sınıfın testlerini içeren bir JUnit test sınıfını temsil eder. Bu testler, "PartiOylari" sınıfındaki metotları test etmek ve beklenen sonuçlarla gerçek sonuçları karşılaştırmak için kullanılır. İşlevleri aşağıda açıklıyorum:

1. `testPartiOyArttir` metodu:

- Bu test metodu, "PartiOylari" sınıfının `partiOyArttir` metotunu test eder.
- "A" ve "B" partilerinin oy sayılarını artırır.
- `assertEquals` yöntemiyle beklenen sonuçlarla gerçek sonuçlar karşılaştırılır.

2. `testPartiOyAzalt` metodu:

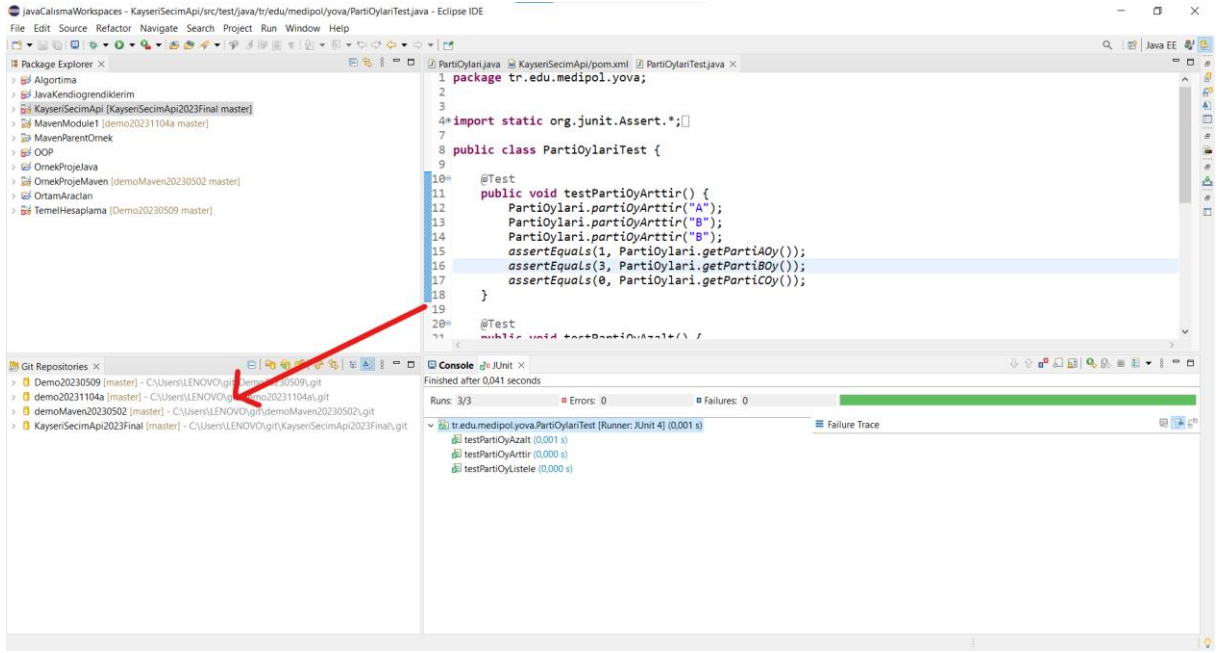
- Bu test metodu, "PartiOylari" sınıfının `partiOyAzalt` metotunu test eder.
- "A" ve "B" partilerinin oy sayılarını artırır, ardından "A" partisinin oy sayısını azaltır.
- `assertEquals` yöntemiyle beklenen sonuçlarla gerçek sonuçlar karşılaştırılır.

3. `testPartiOyListele` metodu:

- Bu test metodu, "PartiOylari" sınıfının `partiOyListele` metotunu test eder.
- `partiOyListele` metoduyla elde edilen sonucu bir örnek metinle karşılaştırır.
- `assertEquals` yöntemiyle beklenen sonuçlarla gerçek sonuçlar karşılaştırılır.

Bu test sınıfı, "PartiOylari" sınıfının doğru şekilde çalışıp çalışmadığını kontrol etmek için kullanılır. Her bir test metodu, belirli bir işlevin doğruluğunu doğrulamak için beklenen sonuçlarla gerçek sonuçları karşılaştırır. Eğer beklenen sonuçlarla gerçek sonuçlar uyuşmuyorsa, bir hata olabileceği anlaşılır ve sorunun kaynağı tespit edilerek düzeltilmesi sağlanır.

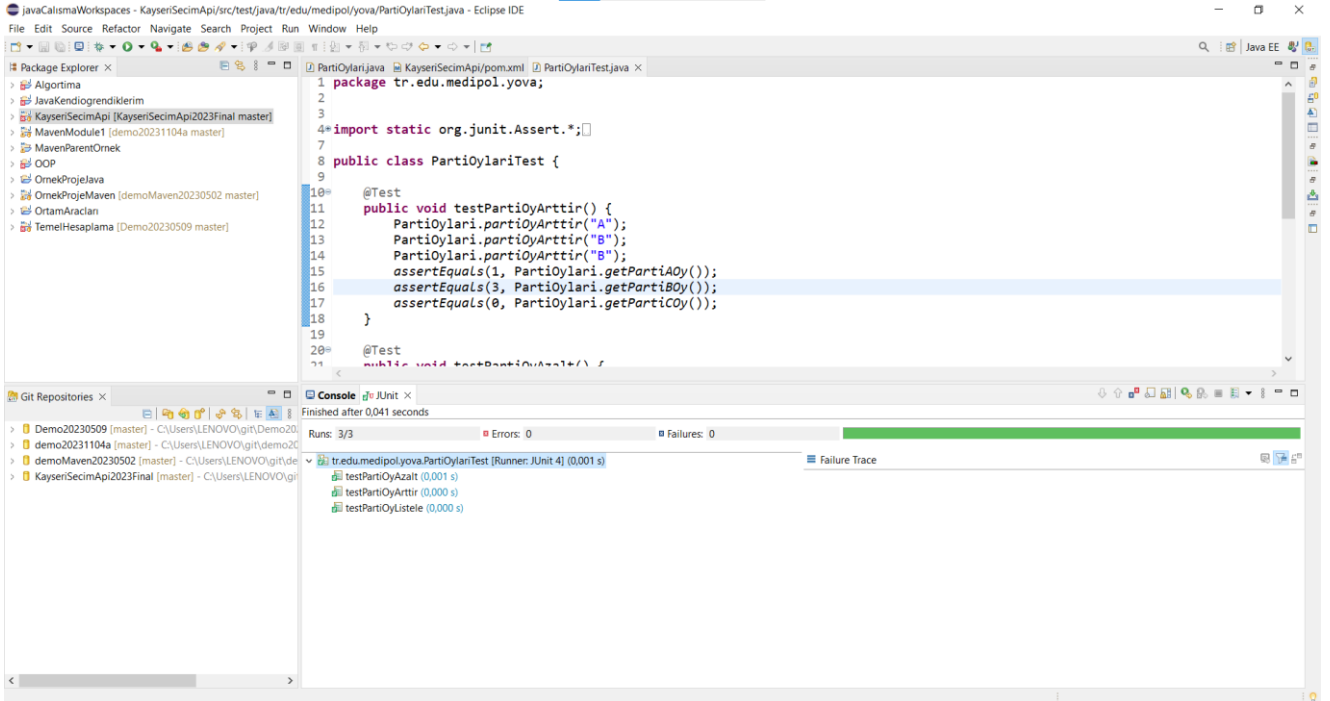
6) GitHub Adresi



Açıklama:

Bu içerik oluşturulan projenin eclipse üzerinden hangi repository'e gideceğini göstermektedir. Ayrıca Packed Explorer'da ilgili Proje'nin yanında hangi repository'e gönderildiğini göstermektedir.

7) JUnit Test Run as Sonucu



Açıklama :

Oluşturulan test sınıfı içerisindeki 3 temel metodun(Oy artırma (biri birer), Oy azaltma (biri birer),Mevcut oyları gösterme) testinin başarı ile gerçekleştiğinin gösteren içeriktir .

8) Maven Test Çıktı Sonucu

```
javaCalsmaWorkspaces - KayseriSecimApi/src/test/java/tr/edu/medipol/yova/PartiOylariTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer
> Algoritma
> JavaKendiogrendiklerim
> KayseriSecimApi [KayseriSecimApi2023Final master]
> MavenModule1 [demo20231104a master]
> MavenParentOrnek
> OOP
> OrnekProjeJava
> OrnekProjeMaven [demoMaven20230502 master]
> OrtamAraclar
> TemelHesaplama [Demo20230509 master]

PartiOylariTest.java
1 package tr.edu.medipol.yova;
2
3
4 import static org.junit.Assert.*;
5
6 public class PartiOylariTest {
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Console - JUnit
<terminated> C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (10 Haz 2023 17:13:55) [pid: 8336]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ KayseriSecimApi ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ KayseriSecimApi ---
[INFO] Surefire report directory: C:\Users\LENOVO\git\KayseriSecimApi2023Final\KayseriSecimApi\target\surefire-reports

-----
T E S T S
-----
Running tr.edu.medipol.yova.PartiOylariTest
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.144 sec

Results :
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0

[INFO] --- jacoco-maven-plugin:0.8.8:report (report) @ KayseriSecimApi ---
[INFO] Loading execution data file C:\Users\LENOVO\git\KayseriSecimApi2023Final\KayseriSecimApi\target\jacoco.exec
[INFO] Analyzed bundle 'KayseriSecimApi' with 1 classes
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.546 s
[INFO] Finished at: 2023-06-10T17:13:59+03:00
[INFO] -----
```

Açıklama :

Bu içerik Projeye sağ tıklayıp Maven>Maven Test işlemi sonucunda konsoldan gelen çıktıdır . 3 temel metodun(Oy arttırma (biri birer), Oy azaltma (biri birer),Mevcut oyları gösterme) testinin başarı ile gerçekleştiğinin gösteren içeriktir .

9) Paketteki Tüm testlerin kapsama oranının verisi

Element	Missed Instructions	Cov	Missed Branches	Cov	Missed Ctxy	Missed Lines	Missed Methods	Missed Classes
tr.edu.medipol.yova	38 of 87	%56	19 of 24	%20	12 20	9 25	1 8	0 1
Total	38 of 87	%56	19 of 24	%20	12 20	9 25	1 8	0 1

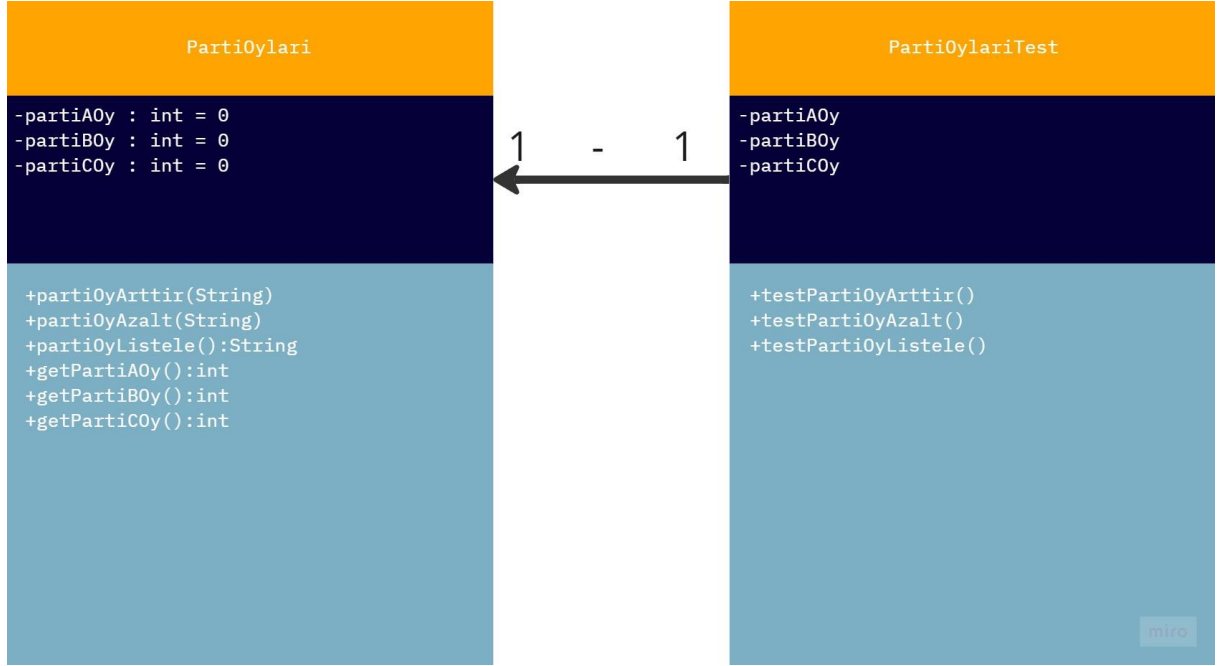
Element	Missed Instructions	Cov	Missed Branches	Cov	Missed Ctxy	Missed Lines	Missed Methods
partiOyAzalt(String)	9	%11	6	%10	0 10	0 1	0 1
partiOyArttir(String)	2	%4	2	%7	0 1	0 1	0 1
PartiOylari()	1	%1	1	%1	1 1	1 1	1 1
static(..)	0	%100	0	%100	0 3	0 1	0 1
partiOylListele()	0	%100	0	%100	0 1	0 1	0 1
getPartiAOyl()	0	%100	0	%100	0 1	0 1	0 1
getPartiBOyl()	0	%100	0	%100	0 1	0 1	0 1
getPartiCOyl()	0	%100	0	%100	0 1	0 1	0 1
Total	38 of 87	%56	19 of 24	%20	12 20	9 25	1 8

Açıklama :

Bu içerik 3 temel metodun(Oy arttırma (biri birer), Oy azaltma (biri birer),Mevcut oyları gösterme) testlerinin hangi metotlara sahip olduğunu ve bu metotların kapsama oranının yüzdesel oranını göstermektedir.

10) UML ve Use Case

a) UML



Açıklama:

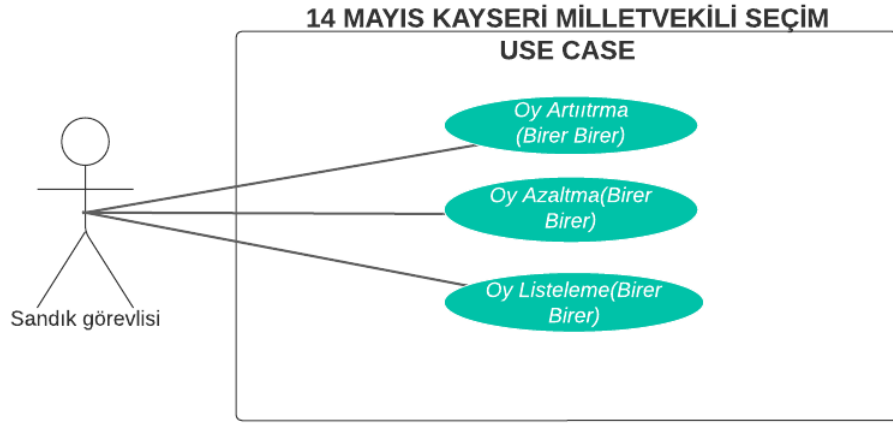
Bu PartiOylari sınıfının Uml Görüntüsü sol tarafta private partiAOy , partiBOy , partiCOy değişkenlerinin hangi veri tipine sahip olduğu ve atamasının yapıldığını gösteren verilere sahiptir.

Public partiOyArttir , partiOyAzalt , partiOyListele , getPartiAOy , getPartiBOy, getPartiCOy'un veri tipleri inputları ve dönüş tiplerini göstermektedir

Bu PartiOylari sınıfının Uml Görüntüsü sağ tarafta private partiAOy , partiBOy , partiCOy değişkenlerine sahip olduğunu göstermektedir . Bu değerler PartiOylari sınıfında atandığından burada değer atanmaz

Public testPartiOyArttir , testPartiOyAzalt , testPartiOyListele test sınıflarının geri dönüş tipine sahip olmayan metotlar olduğunu göstermektedir.

b) Use Case



Açıklama:

Bu içerik KayseriSecimApi maven projesindeki Oy Arttırma Oy azaltama ve listeleme validasyonlarının yetkilerinin kim tarafından kullanıldığını göstermektedir.

11) Git İşlemleri

MINGW64:/c:/Users/LENOVO/Desktop/H5210016_Yigit_Oner_FinalRapor

```
LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner_FinalRapor
$ git init
Initialized empty Git repository in C:/Users/LENOVO/Desktop/H5210016_Yigit_Oner_FinalRapor/.git/

LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner_FinalRapor (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        "yi\304\237it_\303\266ner_H5210016.docx"
```

İlk olarak git init ile git dosyası oluşturuldu ve statustan değişikliğe bakıldı

MINGW64:/c:/Users/LENOVO/Desktop/H5210016_Yigit_Oner_FinalRapor

```
LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner_FinalRapor (master)
$ git status
On branch master
nothing to commit, working tree clean

LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner_FinalRapor (master)
$ git remote add origin https://github.com/yigitoner28/KayseriSecimApi2023Final.git
```

Git remote komutu ile uzaktaki sunucu eşleştirilip eklendi

MINGW64:/c:/Users/LENOVO/Desktop/H5210016_Yigit_Oner_FinalRapor

```
LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner_FinalRapor (master)
$ git add .
$ git add .

LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner_FinalRapor (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   "yi\304\237it_\303\266ner_H5210016.docx"

LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner_FinalRapor (master)
$ git commit -m "commit1"
[master (root-commit) 1273001] commit1
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "yi\304\237it_\303\266ner_H5210016.docx"

LENOVO@DESKTOP-73K0559 MINGW64 ~/Desktop/H5210016_Yigit_Oner_FinalRapor (master)
$ git status
On branch master
```

Commit İşlemi gerçekleşti.

Github Adresi : <https://github.com/yigitoner28/KayseriSecimApi2023Final>