



YAZILIM GELİŞTİRME ORTAM VE ARAÇLARI

FİNAL RAPORU

MUHAMMET TALHA ALADAĞ

H5210009

Cüzdan projesini gerçekleştirmek için Eclipse üzerinde Maven projesi oluşturdum,gerekli bağımlılıklar ve düzenlemelerden sonra quickstart paketi altında CuzdanApi sınıfı oluşturuldu.CuzdanApiTesti sınıfı para harcama bakiye sorgulama ve para yükleme işlemlerini sağlamaktadır

```
1<?xml version="1.0" encoding="UTF-8"?>
2<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4  <modelVersion>4.0.0</modelVersion>
5  <groupId>com.example</groupId>
6  <artifactId>cuzdan-projesi</artifactId>
7  <version>1.0-SNAPSHOT</version>
8  <properties>
9    <maven.compiler.source>11</maven.compiler.source>
10   <maven.compiler.target>11</maven.compiler.target>
11 </properties>
12 <dependencies>
13
14
15   <dependency>
16     <groupId>org.junit.jupiter</groupId>
17     <artifactId>junit-jupiter-api</artifactId>
18     <version>5.8.1</version>
19     <scope>test</scope>
20   </dependency>
21   <dependency>
22     <groupId>org.junit.jupiter</groupId>
23     <artifactId>junit-jupiter-engine</artifactId>
24     <version>5.8.1</version>
25     <scope>test</scope>
26   </dependency>
27   <!-- https://mvnrepository.com/artifact/junit/junit -->
28   <dependency>
29     <groupId>junit</groupId>
30     <artifactId>junit</artifactId>
31     <version>4.13.2</version>
32     <scope>test</scope>
33   </dependency>
34 </dependencies>
35 </project>
```

```

1 package quickstart;
2
3
4 public class Cuzdan {
5     private double bakiye;
6
7     public void paraYukle(double miktar) {
8         bakiye += miktar;
9     }
10
11     public void paraHarca(double miktar) {
12         if (miktar <= bakiye) {
13             bakiye -= miktar;
14         } else {
15             throw new IllegalArgumentException("Yetersiz bakiye!");
16         }
17     }
18
19     public double bakiyeSorgula() {
20         return bakiye;
21     }
22 }
23
24
25
26
27

```

paraYukle(double miktar): Bu metot, cüzdana belirli bir miktar para ekler. miktar parametresi olarak geçirilen miktar, cüzdana yüklenecek olan para miktarını belirtir. Bu miktar, cüzdanın içindeki bakiye değişkenine eklenir.

paraHarca(double miktar): Bu metot, cüzdandan belirli bir miktar para harcar. miktar parametresi olarak geçirilen miktar, harcanacak olan para miktarını belirtir. İlk olarak, miktar parametresi, cüzdanın içindeki bakiye değişkeni ile karşılaştırılır. Eğer miktar cüzdanın bakiyesinden küçük veya eşitse, miktar bakiyeden çıkarılır. Ancak, miktar cüzdanın bakiyesinden daha büyükse, IllegalArgumentException hatası fırlatılır (Yetersiz bakiye hatası).

bakiyeSorgula(): Bu metot, cüzdanın mevcut bakiyesini döndürür. Cüzdanın bakiyesi, bakiye değişkeni aracılığıyla tutulur ve metot bu değeri döndürür.

```

1 package quickstart;
2
3 public class CuzdanApi {
4     private Cuzdan cuzdan;
5
6     public CuzdanApi() {
7         cuzdan = new Cuzdan();
8     }
9
10    public void paraYukle(double miktar) {
11        cuzdan.paraYukle(miktar);
12    }
13
14    public void paraHarca(double miktar) {
15        cuzdan.paraHarca(miktar);
16    }
17
18    public double bakiyeSorgula() {
19        return cuzdan.bakiyeSorgula();
20    }
21 }
22

```

CuzdanApi(): Bu, CuzdanApi sınıfının yapıcı metodudur. Yapıcı, Cuzdan sınıfından bir cuzdan nesnesi oluşturur ve cuzdan değişkenine atar.

paraYukle(double miktar): Bu metod, belirtilen miktar kadar parayı cüzdana yükler. İçerisindeki cuzdan.paraYukle(miktar) ifadesi, Cuzdan sınıfında tanımlanmış olan paraYukle metodunu çağırır ve miktarı cüzdana yükler.

paraHarca(double miktar): Bu metod, belirtilen miktar kadar parayı cüzdandan harcar. İçerisindeki cuzdan.paraHarca(miktar) ifadesi, Cuzdan sınıfında tanımlanmış olan paraHarca metodunu çağırır ve miktarı cüzdandan düşer.

bakiyeSorgula(): Bu metod, cüzdandaki mevcut bakiyeyi döndürür. İçerisindeki cuzdan.bakiyeSorgula() ifadesi, Cuzdan sınıfında tanımlanmış olan bakiyeSorgula metodunu çağırır ve cüzdandaki bakiyeyi alır.

```

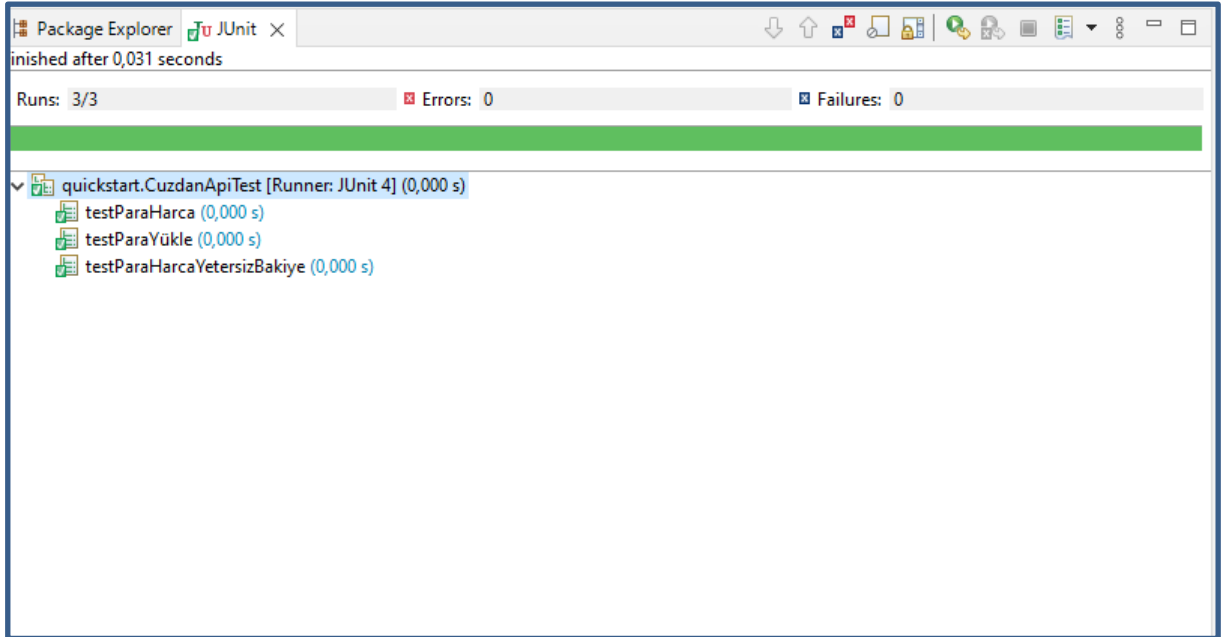
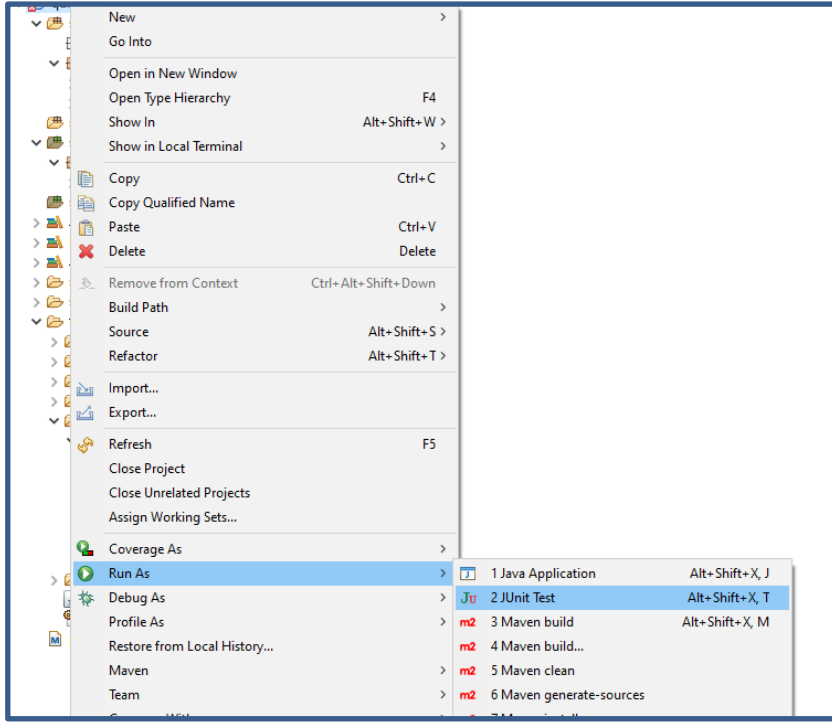
1 package quickstart;
2
3
4 import org.junit.Before;
5 import org.junit.Test;
6 import static org.junit.Assert.assertEquals;
7
8 public class CuzdanApiTest {
9
10     private CuzdanApi cüzdanApi;
11
12     @Before
13     public void setUp() {
14         cüzdanApi = new CuzdanApi();
15     }
16
17     @Test
18     public void testParaYükle() {
19         cüzdanApi.paraYukle(100);
20         assertEquals(100, cüzdanApi.bakiyeSorgula(), 0.001);
21     }
22
23     @Test
24     public void testParaHarca() {
25         cüzdanApi.paraYukle(100);
26         cüzdanApi.paraHarca(50);
27         assertEquals(50, cüzdanApi.bakiyeSorgula(), 0.001);
28     }
29
30     @Test(expected = IllegalArgumentException.class)
31     public void testParaHarcaYetersizBakiye() {
32         cüzdanApi.paraYukle(30);
33         cüzdanApi.paraHarca(50);
34     }
35 }

```

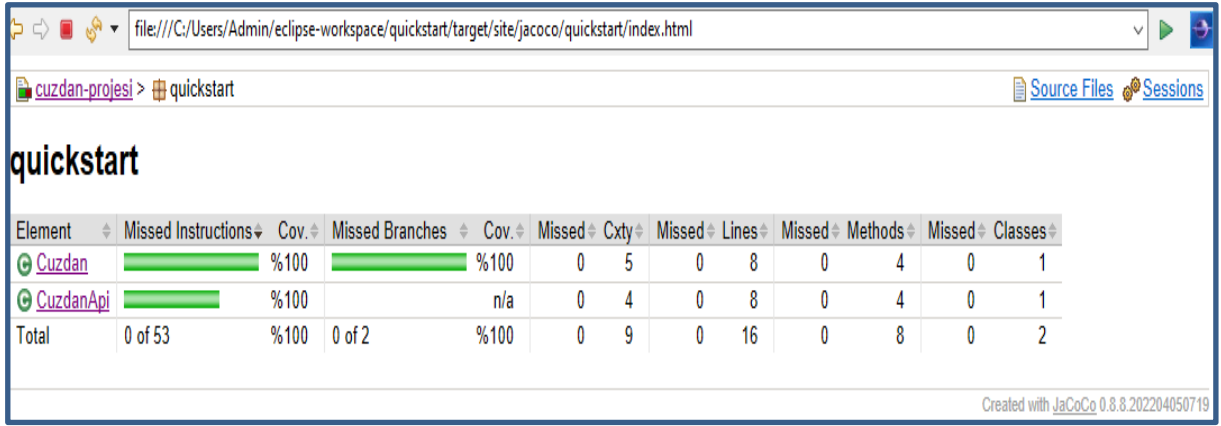
testParaYükle(): Bu test, paraYukle metoduyla cüzdana 100 birimlik bir miktar yüklendikten sonra, bakiyeSorgula metoduyla cüzdandaki bakiyenin 100 birim olup olmadığını kontrol eder. assertEquals yöntemi, beklenen değeri (100) ve gerçek değeri (cuzdanApi.bakiyeSorgula()) karşılaştırır ve eşit olup olmadıklarını doğrular.




testParaHarca(): Bu test, önce paraYukle metoduyla cüzdana 100 birimlik bir miktar yükler, ardından paraHarca metoduyla 50 birimlik bir miktar harcar. Son olarak, bakiyeSorgula metoduyla cüzdandaki bakiyenin 50 birim olup olmadığını kontrol eder. Yine assertEquals yöntemi kullanılarak beklenen değer (50) ve gerçek değer (cuzdanApi.bakiyeSorgula()) karşılaştırılır.

testParaHarcaYetersizBakiye(): Bu test, paraHarca metoduyla yetersiz bakiye durumunu test eder. Burada expected özelliği IllegalArgumentException.class olarak ayarlanmıştır, yani bu testin, paraHarca metodu çağrıldığında IllegalArgumentException fırlatması gerektiği beklenmektedir. Bu durum, cüzdanın bakiyesinin harcanmak istenen miktardan daha az olduğunda ortaya çıkar.



Ayrıca CuzdanApi sınıfının doğru çalıştığını ve beklenen davranışları sergilediğini doğrulamak için JUnit kullanıldı. Bu testlerin temel amacı, CuzdanApi'nin metotlarının beklenen sonuçları doğru şekilde üretip üretmediğini ve hatalı girişlere uygun tepkiler verip vermediğini kontrol edilmesidir. Testlerin sonuçları, CuzdanApi sınıfının beklenen davranışlarını sergileyerek işlevselliğini doğrulamayı amaçlamaktadır.



Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Cxty	Missed Lines	Missed Methods	Missed Classes
Cuzdan		%100		%100	0 5	0 8	0 4	0 1
CuzdanApi		%100	n/a	n/a	0 4	0 8	0 4	0 1
Total	0 of 53	%100	0 of 2	%100	0 9	0 16	0 8	0 2

Created with JaCoCo 0.8.8.202204050719


Bu aşamadan sonra ayrıca coverage metriği kullanılarak yazılım testlerinin ne kadarının kod tarafından kontrol edildiği ölçülmüştür. Bu metrik, test sürecinin etkinliğini değerlendirmek ve yazılımın hangi kısımlarının test edilmediğini belirlemek açısından önemlidir. Yukarıda belirtilen ekran görüntüsünde CuzdanApi sınıfı için gerçekleştirilen testlerin yüksek bir code coverage oranı elde ettiği gözlemlenmiştir.

Ayrıca proje geliştirme sürecinde test kapsamını değerlendirmek için Jacoco adlı aracı kullandık. Jacoco, testlerin kod kapsamını ölçmek ve raporlamak için kullanılan bir araçtır. Testlerin hangi kod satırlarını kapsadığını, hangi bölgelerin test edilmediğini ve test kapsamının yüzdesini göstermek için Jacoco'nun sağladığı raporlama özelliklerinden faydalandık, sonucu index.html olarak yazdırdıktan sonra gerekli ekran görüntüleri alındı.

Clone Git Repository

Source Git Repository

Enter the location of the source repository.



Location

URI:

Host:

Repository path:

Connection

Protocol:


Port:

Authentication

User:


Password:


☒ Store in Secure Store



Push Results: <https://github.com/mtaladag03/Cuzdan>

Pushed to <https://github.com/mtaladag03/Cuzdan>

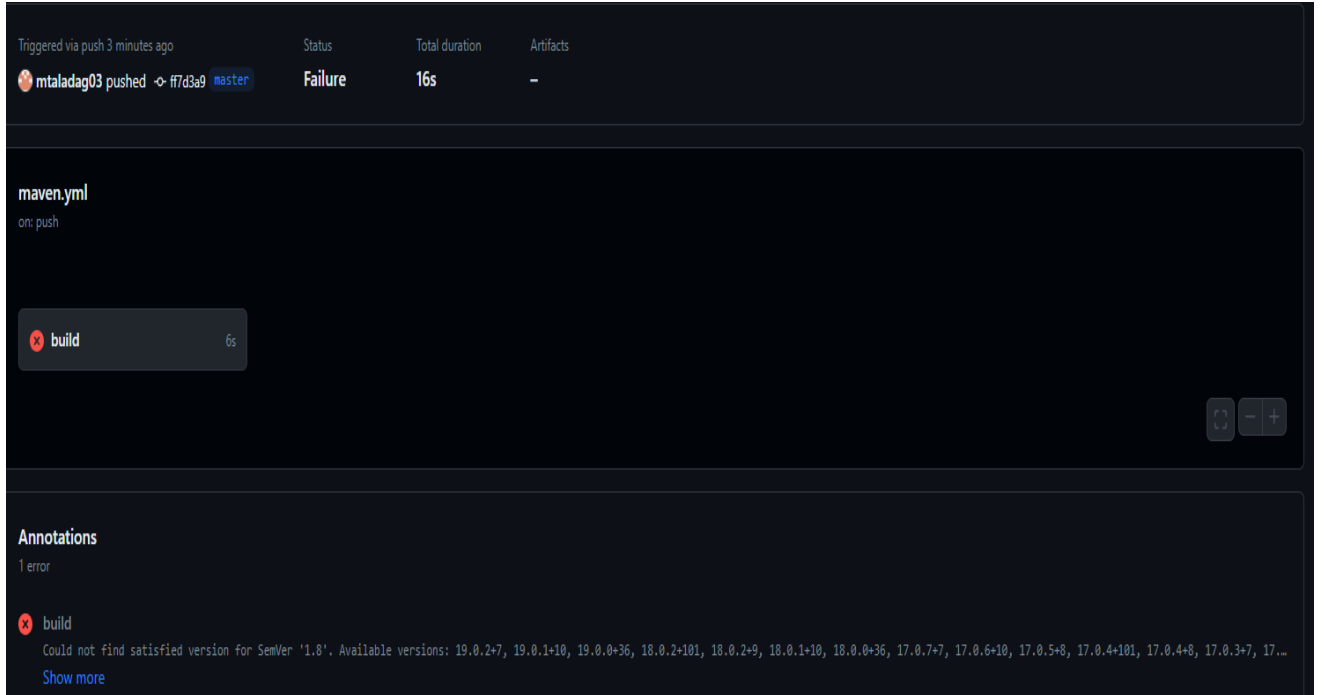
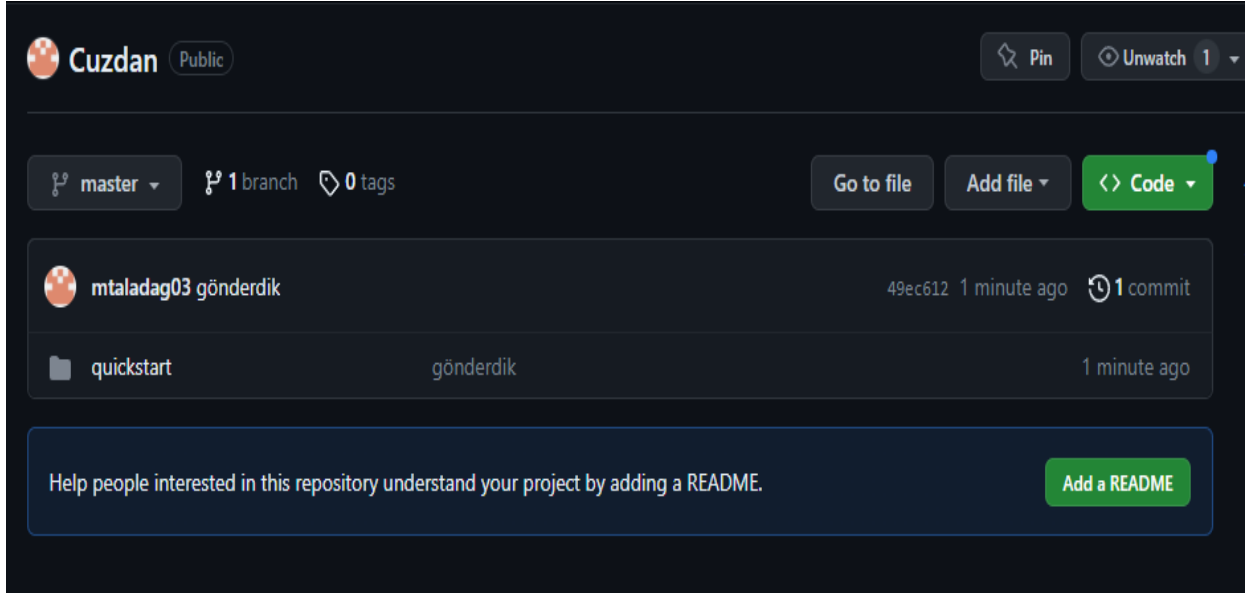


 master → master [new branch]

Message Details

Repository <https://github.com/mtaladag03/Cuzdan>

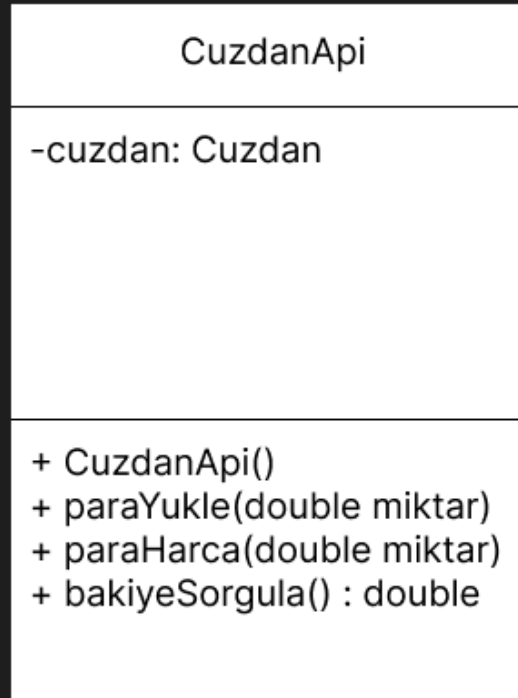
Resolving deltas: 0% (0/6)



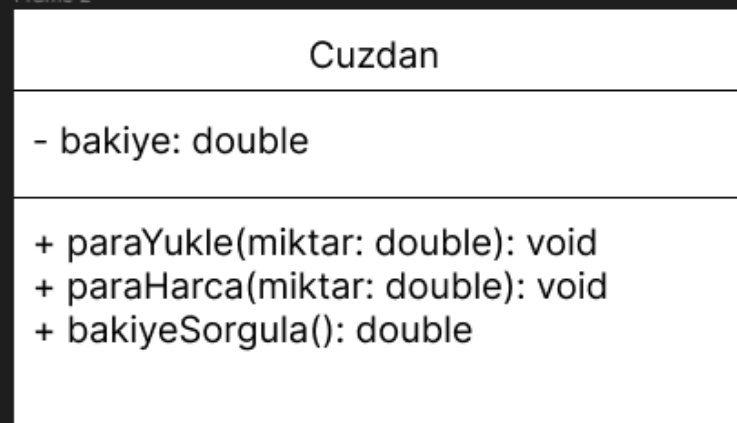
Actions kısmında böyle bir hata alıyorum saatlerce uğraştım fakat çözemedim yine de bunu sizinle paylaşmak istedim yeni repostoriesler açıp denememe kodları değiştirmeme rağmen bir türlü düzelmedi.

Uml sınıf diafgramı:

Frame 1



Frame 2



Use case diagramı.

