

dosya. cukes runner dosyası, sahip olduğu etiketlerle eşleşen özellikleri çalıştırır.

GIT dallanma

git dallanmasını şimdi nasıl yapıyorsunuz?

Şu anda GRUP PROJELERİNDE NASIL YAPILIR:

Her ekip üyesi için ana şube ve ayrı şubeler vardır. birisi işi bitirdiğinde, kendi şubelerini inceledikten sonra ustalaşmak için birleştirilir.

NASIL YAPTIK?

projemde kişi için ustalık, geliştirme ve şubemiz vardı. bu nedenle, 2 otomasyon test cihazımız varsa,

usta

geliştirmek

test cihazı1

test cihazı2

her test kullanıcısı kendi şubelerine giriş yapar. incelendikten sonra şube geliştirmek için birleştirilir. ustayı birleştiriyoruz ve sadece bir sprint geliştirir.

🔗 görüşmede dallanma hakkında sorduklarında, otomasyon projesi dallanma stratejiniz hakkında konuşun.

Projenizde, kodunuz uygulama kod deposundan ayrı bir depodur. Otomasyon çerçevesi daha küçük bir kod tabanına sahiptir ve daha az insan dahil oldu. Böylece, daha az karmaşık dallanma politikasına sahip olabiliriz.

JAVA Mülakat Teknik Soruları

Ters Dize - Döngü İçin

```
public static String reverseString (String str) {  
    Dize ters = "";  
    for (int i = str.length () - 1; i >= 0; i--) {  
        ters += str.charAt (i);  
    }  
    tersine dönüş;  
}
```

EN İYİ YOL:

```
Dize adı = "Alper Aslan";  
String reversed = new StringBuilder (isim) .reverse (). ToString ();  
System.out.println (tersine çevrilmiş);
```

Ters Dize - StringBuilder

```
public static void main (String [] args) {  
    String str = "Merhaba Dünya";  
    StringBuilder sb = new StringBuilder (str);  
  
    Sistem. out .println (sb.reverse ()); }
```

Ters Dize - Karakter Dizisi

```
public void printReverse (karakter [] harfler, int boyut) {  
    for (int i = letters.length-1; i >= 0; i -) {  
        System.out.print ([i] harfleri);  
    }  
}
```

Asal sayı

```
public static boolean checkPrime (int n) {  
    eğer (n <= 1) {
```

```

        yanlış dönüş;
    }
    for (int i = 2; i < Math.sqrt(n); i++) {
        eğer (n % i == 0) {
            yanlış dönüş;
        }
    }
}
doğruya dön;
}

```

136

Sayfa 137

Palindrom

```

public static boolean isPalindrome (String str) {
    eğer (str == null)
        yanlış dönüş;
    StringBuilder sb = new StringBuilder (str);
    return sb.reverse (). toString (). equals (str); }

```

Palindrome - charAt () ile

```

public static boolean isPalindrome (String s) {
    int head = 0;
    int kuyruk = s.length () - 1;
    while (baş < kuyruk) {
        if (s.charAt (head) != s.charAt (kuyruk)) {
            yanlış dönüş;
        }
        kafa ++;
        kuyruk--;
    }
}
doğruya dön;
}

```

Ters Dize - ArrayList - Itirator

```

public static void main (String [] args) {
    String input = "Geeks For Geeks";
    List <Character> arrList = new ArrayList <> ();

    for (char c: input.toCharArray ()) {
        arrList.add (c);
    }
    Koleksiyonlar. ters (arrList);

    Nesne [] arr2 = arrList.toArray ();

    for (int i = 0; i < arr2.length; i++) {
        ters += dizi2 [i];
    }
    Sistem. out .println (ters);
}

```

DİĞER YOL

```

Dize ters = arrList.stream ()
    .map (String :: valueOf) // Akış <String>
    .collect (Collectors.joining ());

```

137

Sayfa 138

Faktöriyel

```
int sayı = 10;
int factorialSum = 1;

for (int i = 1; i <= sayı; i++) {
    factorialSum = factorialSum * i;
}
Sistem.out.println (" " + sayı + "faktöriyel" + faktör toplamı "dır); }
```

Basamakların Toplamı

```
int numarası = 1346;
int toplam = 0;

while (sayı> 0) {
    toplam += sayı% 10;
    sayı = sayı / 10;
}
Sistem.out.println (toplam); }
```

Armstrong

```
int toplam = 0;
int rakam;
int temp;
int sayı = 370;
temp = sayı;
while (temp> 0) {
    basamak = sıcaklık% 10;
    toplam = toplam + (basamak * basamak * basamak);
    sıcaklık = sıcaklık / 10;
}
eğer (sayı == toplam)
    System.out.println (sayı + "sağlam bir sayıdır");
} Başka
```

Merdiven

```
for (int x = 1; x <= 5; x++) {
    for (int y = 1; y <= x; y++) {
        System.out.print (y + "");
    }
    System.out.println ();
}
```

138

Sayfa 139

Fibonacci

```
int a = 0;
int b = 1;
System.out.print (a + "" + b + "");

for (int i = 2; i <= 10; i++) {
    int c = a + b;
    a = b;
    b = c;
    System.out.print (c + "+");
}
```

Geçici Bir Değişken Kullanmadan İki Numarayı Değiştirin

```
// tek satırlı yöntemler
a = a ^ b ^ (b = a); //Yöntem 1
```

```
b = (a + b) - (a = b); // yöntem 2
a += b - (b = a); // yöntem 3
```

```
// geçici değişken
int temp = a;
a = b;
b = sıcaklık;
```

Dizileri Kullanan Dizideki En Büyük Sayı

```
int [] dizi = {5, 6, 76, 31, 43, 1};
Arrays.sort (dizi);
System.out.println (arr [arr.length-1]);
```

Koleksiyonları Kullanan Dizideki En Büyük Sayı

```
public static int returnLargest (Tamsayı [] b, int toplam) {
    List <Integer> list1 = Diziler. asList (b);
    Koleksiyonlar. sort (list1);
    int en büyük = list1.get (toplam-1);
    en büyük dönüş; }

public static void main (String args []) {
    Tamsayı x [] = {4,3,2,12,54,34,88};
    Sistem. out .println ( returnLargest (x, 7));
```

- 139 -

Sayfa 140

Koleksiyonları Kullanan Dizideki En Büyük Sayı

```
public static int getLargest (int [] a, int toplam) {
    int temp;
    for (int i = 0; i <toplam; i ++) {
        for (int j = i + 1; j <toplam; j ++) {
            eğer (a [i] > a [j]) {
                temp = a [i];
                a [i] = a [j];
                a [j] = sıcaklık; } } }
    bir [toplam-1] döndürür; }
```

Ters Sayı

```
public static int reverse (int sayı) {
    int ters = 0;
    int kalan = 0;
    yapmak{
        kalan = sayı% 10;
        ters = ters * 10 + kalan;
        sayı = sayı / 10;
    } while (sayı > 0);
```

tersine dönüş; }

Bir Dizideki İlk İki Maksimum Sayıyı Bul

```
public void GetTwoMaxValues (int [] numaralar) {

    int maxOne = 0;
    int maxTwo = 0;

    Arrays.sort (nums);
    System.out.println ("Max1 -" + (nums [nums.length-1]));
    System.out.println ("Max2 -" + (nums [nums.length-2]));
```

}

Bölme veya Mod İşleci kullanmadan bölme

```

public static String division (int dividant, int divisor) {
    int quotient = 0;
    int kalan = 0;
    while (bölünen >= bölün) {
        bölünen = bölünen - bölün;
        bölüm ++;
        kalan = bölünen;
    }
    return "bölüm =" + bölüm + "kalan =" + kalan;
}

```

- 140 -

Sayfa 141**İkili Arama (Doğrusaldan Daha Hızlı) -**

Sıralanmış bir dizi içindeki bir hedef değerin konumunu bulur. İkili arama, dizinin orta elemanına hedef değer. Eşleşiyorsa, değilse true döndür ulaşana kadar tekrar bölüyorsunuz.

```

public static void binarySearch (int arr [], int first, int last, int key) {
    int orta = (ilk + son) / 2;
    while (ilk <= son) {
        eğer (arr [mid] <key) {
            ilk = orta + 1;
        } else if (arr [mid] == key) {
            System.out.println ("Öğe dizinde bulunur:"
                                + orta);
            kırmak;
        } Başka {
            son = orta - 1;
        }
        orta = (ilk + son) / 2;
    }
    eğer (ilk > son) {
        System.out.println ("Öğe bulunamadı!");
    }
}

public static void main (String args []) {
    int arr [] = {10,20,30,40,50};
    int anahtar = 30;
    int son = dizi. uzunluk-1;
    binarySearch (arr, 0, last, key);
}

```

Kabarcık Sırala

Mevcut eleman bir sonraki eleman ile karşılaştırılır. Geçerli öğe daha büyükse sonraki öğe değiştirilir.

```

public static void bubbleSort (int [] arr) {
    int n = dizi uzunluk;
    int temp = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 1; j < (n-i); j++) {
            eğer (dizi [j-1] > dizi [j]) {
                temp = dizi [j-1]; // öğeleri değiştir
                dizi [j-1] = dizi [j];
                arr [j] = sıcaklık;
            }
        }
    }
}

public static void main (String [] args) {
    int arr [] = {3,60,35,2,45,320,5};
    bubbleSort (arr); // bubble sort kullanarak dizi öğelerini sıralama
    System.out.println ("Kabarcık Sıralamasından Sonra Dizi");
    for (int i = 0; i < arr.length; i++) {
        System.out.print (arr [i] + " ");
    }
}

```

141

Sayfa 142

Bir Dizide Yinelenenleri Filtreleme

```

public static void main (String [] args) {
    ArrayList <String> list = new ArrayList <String> ();

    // 0-9 arası bir sayı listesi oluşturun.
    for (int i = 0; i <10; i++) {
        list.add (String.valueOf (i));
    }

    // 0-5 arasında yeni bir sayı kümesi ekleyin.
    for (int i = 0; i <5; i++) {
        list.add (String.valueOf (i));
    }

    System.out.println ("Giriş listesi:" + liste);
    System.out.println ("\nFiltre edilmiş kopyalar:" + processList (liste));
}

public static Set <String> processList (List <String> listContainingDuplicates) {

    final Set <String> resultSet = new HashSet <String> ();
    final Set <String> tempSet = new HashSet <String> ();

    for (String yourInt: listContainingDuplicates) {
        eğer (! tempSet.add (yourInt)) {
            resultSet.add (yourInt);
        }
    }
    return resultSet;
}

```

En Büyük Palindrom

```

public static String en büyükPalindrome (String str) {
    str = str.toLowerCase ();
    String majorPalindrome = "";
    Dize [] arr = str.split ("");

    for (String each: arr) {
        Dize ters = "";
        for (int i = each.length () - 1; i >= 0; i--) {
            ters += each.charAt (i);
        }

        eğer (each.equals (ters) && each.length () >
            en büyükPalindrome = her biri;
            mostPalindrome.length ()) {
        }
    }
    en büyükPalindrome'a dön; }

```

142

Sayfa 143**Bir Diziden Ekstra Boşlukları Kaldırma**

```

public class removeExtraSpaces {
    public static void main (String args []) {

        String input = "Fazla boşlukları kaldırmayı deneyin.";
        StringTokenizer substr = new StringTokenizer (input, "");
        StringBuffer sb = new StringBuffer ();

        while (substr.hasMoreElements ()) {
            sb.append (substr.nextElement (). append (""); }

        System.out.println ("Gerçek dize:" + girdi);
        System.out.println ("İşlenen dize:" + sb.toString (). Trim ()); }

```

DİĞER YOL

```
String input = "Fazla boşlukları kaldırmayı deneyin.";
String inputNew = input;

while (inputNew.contains(" ")) {
    inputNew = inputNew.replace(" ", "");
};

System.out.println (giriş);
System.out.println (inputNew);
```

Ünlülerin ve Ünsüzlerin sayısını sayın

```
public static String numaraları (String word) {
    int countVowels = 0;
    int countConsonants = 0;

    <Karakter> ünlüleri listele = Diziler. asList ('a', 'e', 'i', 'o', 'u');
    <Karakter> ünsüzleri Listele = new ArrayList <> ();

    for (char i = 'a'; i <= 'z'; i++) {
        eğer (! vowels.contains (i)) {
            ünsüzler.add (i); }
    }
    kelime = word.toLowerCase ();

    for (int i = 0; i <word.length (); i++) {
        if (vowels.contains (word.charAt (i))) {
            countVowels ++;
        }Başka {
            countConsonants ++; }
    }
    return "ünlülerin sayısı" + countVowels + "ve ünsüzlerin sayısı" +
countConsonants; }
```

- 143 -

Sayfa 144

Dinamik - Çalışma Zamanı Polimorfizmi

```
class Animal {
    void eat () {
        System.out.println ("yeme"); }
}
class Dog, Animal {
    void eat () {
        System.out.println ("meyve yeme");}
}
class BabyDog, Dog {
    void eat () {
        System.out.println ("içme sütü");
    }
}
public static void main (String args []) {
    Hayvan a1, a2, a3;
    a1 = yeni Hayvan ();
    a2 = yeni Köpek ();
    a3 = yeni BabyDog ();
    a1.eat (); // "yemek" yazdırır
    a2.eat (); // "yeme meyveleri" yazdırır
    a3.eat (); } // "içme sütü" yazdırır
```

Çok Düzeyli Kalıtım ile Dinamik Polimorfizm

```
class Animal {
    void eat () {
        System.out.println ("hayvan yiyor ...");
    }
}
class Dog, Animal {
    void eat () {
        System.out.println ("köpek yiyor ..."); }
}
class BabyDog1, Dog'u genişletir {
```

```
public static void main (String args []) {  
    Hayvan a = yeni BabyDog1 ();  
    a.cat ();  
}
```

Xpath - Kardeşin İzinde

```
Aşağıdaki-sibling :: siblingName [1];  
Parent :: parentTag  
// etiket [ile başlar (@ id, 'mesaj')]
```

- 144 -

Sayfa 145

Zaman Polimorfizmini Derleyin

```
class Overload {  
    void demo (int a) {  
        System.out.println ("a:" + a); }  
  
    void demo (int a, int b) {  
        System.out.println ("a ve b:" + a + ", " + b);  
    }  
    çift demo (çift a) {  
        System.out.println ("double a:" + a);  
        dönüş a * a; }  
}  
  
class MethodOverloading {  
    public static void main (String args []) {  
        Aşırı Yük Nesnesi = yeni Aşırı Yük ();  
        çift sonuç;  
        Obj. Demo (10);  
        Obj. Demo (10, 20);  
        sonuç = Obj .demo (5.5);  
        System.out.println ("O / P:" + sonuç); } }
```

1-100 arası tüm çift sayılar (i% 2 == 0)

Hepsi tek sayı (i% 2! = 1)

Veri üyeleri tarafından çalışma zamanı polimorfizmi elde edilemez.

String'i Double'a dönüştürmek için

```
Double doubleString = Double.parseDouble (toBeDouble);  
Double doubleStr = Double.valueOf (toBeDouble);
```

```
Double'yi String'e dönüştürmek için  
String strDouble = String.valueOf (toBeString);  
String stringDouble = toBeString.toString ();
```

```
public static char returnFirstChar (String str) {  
  
    dönüş str.charAt (0);  
}  
  
public static char returnLastChar (String str) {  
    dönüş str.charAt ((str.length () - 1));  
}
```


Sayfa 146

Uyarılar

```
Uyarı uyarısı = driver.switchTo.alert ();
alert.accept (); alert.dismiss (); alert.sendKeys () alert.getText ()
```

Örtülü Bekleme

```
driver.manage.timeouts.implicitlyWait (5 saniye)
```

Açık bekleme

```
WebDriverWait wait = new WebDriverWait (sürücü, 30);
WebElement öğesi = wait.until (
ExpectedConditions.elementToBeClickable (By.id (öge)));
visibilityOf (), alertIsPresent ()
```

Excel - Apache POI

```
public void readExcel () throws IOException {
    Dize Dosyası Path = "D: \\ sample.doc.xls";
    FileInputStream fis = new FileInputStream (FilePath);
    Çalışma kitabı wb = WorkbookFactory.create (fis);

    Sayfa sh = wb.getSheet ("Sayfa1"); // getSheetAt (1)
    // Sayfadaki mevcut satır sayısını almak için
    int totalNoOfRows = sh.getRows ();
    // Sayfada bulunan sütun sayısını almak için
    int totalNoOfCols = sh.getColumns ();
    for ( int satır = 0; satır < totalNoOfRows; satır ++ ) {
        for ( int col = 0; col < totalNoOfCols; col ++ ) {
            System.out.print (sh.getCell (sütun, satır) .getContents () + "\n t");
            System.out.println ();
        }
    }

    Row row = worksheet.getRow (0);
    Hücre hücre = row.getCell (0);
    System.out.println (cell.toString ());

    Hücre hücre = row.getCell (0);
    cell.setCellValue ("kahve");
    FileOutputStream outputStream = new FileOutputStream (filePath);
    workbook.write (outStream);
}
```

Sayfa 147

JDBC - Veritabanı Bağlantısı

```
Bağlantı bağlantısı = DriverManager.getConnection (Url, Kullanıcı, Geçiş);
İfade ifadesi = connection.createStatement
(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
ResultSet resultSet = statement.executeQuery ("ülkelerden * seçin");
sonuçSet.last (); // sonuç kümesinde kaç kayıt bulun
int rowCount = resultSet.getRow ();
System.out.println ("Satır sayısı:" + rowCount);

resultSet.first ();
while (resultSet.next ()) {
    System.out.println (resultSet.getString (1) + "-" + resultSet.getString ("ülke_adı") + "-
```

```
" + resultSet.getInt (" region_id "));}
```

Sınıf Seçin

```
ArabalarListesi seçin = yeni Seç (el);  
carList.getFirstSelectedOption ();  
assertequals ("bir metin", carList.getFirstSelectedOption (). getText ())
```

İframe

```
Driver.switchTo.frame (webElement veya Dize veya dizin olarak kimlik)
```

Pencere tutacağı

```
getWindowHandles ()  
for (Dize tutamacı: driver.get.getWindowHandles ())  
driver.switchTo.window ("handle") :: bir dizge alır  
Eğer (driver.getTitle (). Equals (beklenen başlık)  
kırmak;
```

Eylem Sınıfı

```
WebElement el = driver.findElement  
Eylemler eylemler = yeni Eylemler (sürücü) .perform  
actions.doubleClick (el) .perform
```

Dosya yükleme

```
Genel void fileUpload (Dize yolu) {  
WebElement karşıya yükleme = sürücü.findElement;  
Upload.sendKeys (yol)
```

147

Sayfa 148

SQL - İç Birleştirme

```
SEÇ e.FIRST_NAME, e.SALARY, j.JOB_TITLE  
DAN ÇALIŞANLARIN e INNER JOIN İŞLERİ j AÇIK e.JOB_ID = j.JOB_ID;
```

İstenen Yetenekler - Sertifika Verilmesi

```
DesiredCapabilities yeteneği = DesiredCapabilities.chrome ();  
// SSL sertifikasını kabul etmek için  
capability.setCapability (CapabilityType.ACCEPT_SSL_CERTS, true);  
System.setProperty ("webdriver.chrome.driver", "E: /chromedriver.exe");  
// Google Chrome örneği oluşturun ve maksimize edin  
sürücü = yeni ChromeDriver (yetenek);
```

5 Ekran inci tablodan Row

```
(EMPLOYEE_ID, FIRST_NAME, ROWNUM OLARAK ROWNUM SEÇİN.  
ÇALIŞANLARDAN) RN = 5 NEREDE;
```