

# Programlama Dillerin Presepleri

ÖDEV 1

ZIDAN OMAR HAMID | B181210568

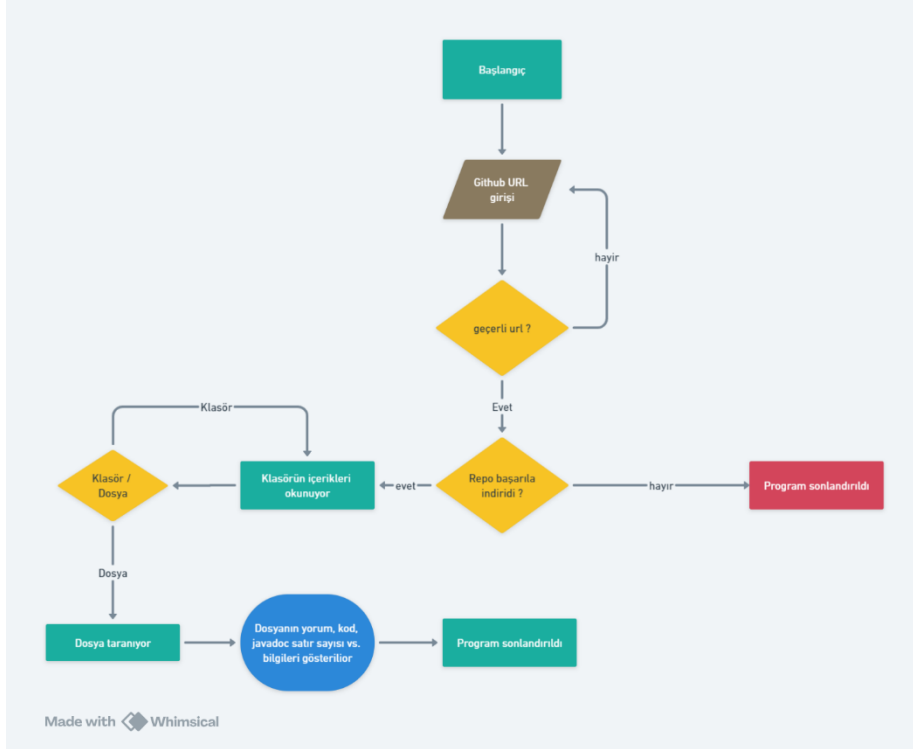
### 1. Ödevde istenilen

Bu ödevde, bir konsol üzerinden girilen bir git reposunu lokal bir klasöre klonlayan bir uygulama oluşturulacak. Klonlanan repodan JavaDoc satır sayısı, yorum satır sayısı, kod satır sayısı, toplam satır sayısı (boşluklar hariç), fonksiyon sayısı ve yorum oranı gibi değerlendirilecek.

### 2. Öğrendiklerim

Java diliyle bir proje geliştirmemiştım ve genel syntax dışında Java bilgim yoktu. Ancak bu ödev sayesinde Java syntax'ıyla tanıştım ve ayrıca Java'da başka kütüphaneleri nasıl kullanacağımı öğrendim.

### 3. Yaptıklarım



Program çalıştırdığında kullanıcıdan github repo url isteniyor. Kullanıcı github repo url'i girdikten sonra url geçerli bir github url olup olmadığını regex ile kontrol ediyor ve kullanıcı geçerli bir repo girene kadar program tekrarlanacak.

Geçerli repo url girdikten sonra repo `./public/(timestamp)` isim klasöründe `jgit` kütüphanesi ile indirilecek/clonelanacak. Klasör ismi timestamp olarak kullanmamı sebebi ise eğer daha önce aynı repo klonlanacak ise mevcut klasörü silmeme veya üstüne tekrar yazılmama gerek kalmayacak çünkü bazen mevcut klasörü silerken Administaror yetki hatası çıkar. Bunun için her repoya timestamp ile yeni klasör oluşturulacak.

Sonra `./public/ (timestamp)` klasörün içindeki dosyalar ve içindeki en derin klasörüne kadar tek tek dosyaları okuyacak.

```

private void analyzeFile(File file) {
    Object[] javadocCounterArr = new Object[2];
    javadocCounterArr[0] = 0;
    javadocCounterArr[1] = false;

    Object[] commentLineCounterArr = new Object[3];
    commentLineCounterArr[0] = 0;
    commentLineCounterArr[1] = false;
    commentLineCounterArr[2] = false;

    Object[] codeLineCounterArr = new Object[2];
    codeLineCounterArr[0] = 0;
    codeLineCounterArr[1] = false;

    int lineCounter = 0;

    Object[] functionCounterArr = new Object[2];
    functionCounterArr[0] = 0;
    functionCounterArr[1] = false;

    try (BufferedReader reader = new BufferedReader(new FileReader(file))) {
        String line;
        while ((line = reader.readLine()) != null) {
            line = line.trim();
            lineCounter++;
            countJavadocLine(javadocCounterArr, line);
            countCommentLine(commentLineCounterArr, line);
            countCodeLine(codeLineCounterArr, line);
            countFunction(functionCounterArr, methodPattern, line);
        }

        this.totalJavadocLine = (int) javadocCounterArr[0];
        this.totalCommentLine = (int) commentLineCounterArr[0];
        this.totalCodeLine = (int) codeLineCounterArr[0];
        this.totalLine = lineCounter;
        this.totalFunction = (int) functionCounterArr[0];
        this.commentDeviation = calculateCommentDeviation();
    } catch (IOException e) {
        System.err.println("Error reading file: " + e.getMessage());
    }
}

```

Programın performansı yükseltmek ve hafif bir şekilde çalışabilmesi için her buffer okunduğunda bütün dosya taratmak yerine satır satır inceleniyor. Tarama fonksiyonların parametreleri üzerinden girilen değişkenler sürekli aynı referans(pass by reference) ile çalışabilmesi için array olarak yolladım. Yeni satıra geçince islnMultilineComment, islnJavadoc, line counter gibi değişkenleri sıfırlanmayacaktır. Analizleri yapıldıktan sonra sonuçları ekrana yazdırdım.

#### 4. Eksik bıraktığım yerler

*/\* bu bir yorum \*/ return 1*

Yukarıdaki yorum gibi multiline commenti inline comment gibi davranıyorsa bir yorum sayılır mı diye özeldürümlerde yorumların toplam satır hesaplaması yanlış çıkabilir çünkü kesin bir Kural belirtilmemiş

#### 5. Zorladığım kısımlar

Genel olarak java syntax ile alıştıktan sonra pek zorluk yaşamadım sadece git klonlamada sürekli hata gösterdiği için jgit kütüphanesi kullandım ve dosya analiz fonksiyonları geliştirmeye odaklayabildim.