

acetinkaya / VeriYapilari-ders6

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

Settings

MIT license

3 stars

0 forks

1 watching

Branches

Activity

Tags

Public repository

1 Branch

0 Tags

Go to file

t

Go to file

+

Add file

<> Code

acetinkaya Update README.md a3a1179 · 3 months ago

D1.png	Add files via upload	3 months ago
D10.png	Add files via upload	3 months ago
D11.png	Add files via upload	3 months ago
D12.png	Add files via upload	3 months ago
D13.png	Add files via upload	3 months ago
D14.png	Add files via upload	3 months ago
D15.png	Add files via upload	3 months ago
D16.png	Add files via upload	3 months ago
D17.png	Add files via upload	3 months ago
D18.png	Add files via upload	3 months ago
D2.png	Add files via upload	3 months ago
D3.png	Add files via upload	3 months ago
D4.png	Add files via upload	3 months ago
D5.png	Add files via upload	3 months ago
D6.png	Add files via upload	3 months ago
D7.png	Add files via upload	3 months ago
D8.png	Add files via upload	3 months ago
D9.png	Add files via upload	3 months ago
LICENSE	Initial commit	3 months ago
Programlama-9.png	Add files via upload	3 months ago
README.md	Update README.md	3 months ago

https://github.com/acetinkaya/VeriYapilari-ders6

1/13

Python - Veri Yapıları ve Algoritmalar

Ders 6. Akış Diyagramları



Bu ders içerisinde;

- 6.1. Akış Diyagramı / Şeması Nedir?
- 6.2. Akış Diyagramlarının Faydaları
- 6.3. Akış Diyagramında Kullanılan Çizimler
- 6.4. Akış Diyagramı Oluşturma Adımları
- 6.5. Akış Diyagramı Çizim Araçları
 - 6.5.1. Draw.io
 - 6.5.2. Diagrams.net
 - 6.5.3. Microsoft Visio
 - 6.5.4. Lucidchart
- 6.6. Akış Diyagramı Çizim Uygulamasının Gerçekleştirilmesi

6.1. Akış Diyagramı Nedir?

Akış diyagramı, bir algoritma veya sürecin adım adım görsel olarak ifade edilmesidir. Akış şemaları; başla ve bitir, geometrik çizimler ve oklardan oluşmaktadır.

6.2. Akış Diyagramlarının Faydaları

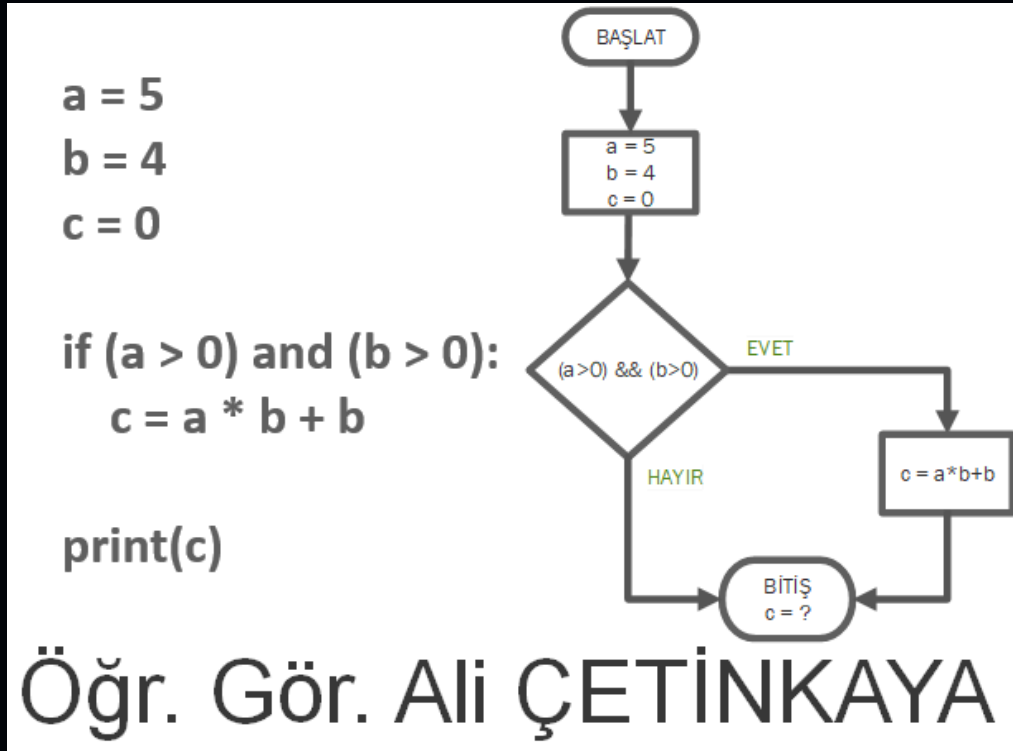
1. Süreçlerin daha iyi anlaşılmasını sağlar.
2. Hataların erken tespit edilmesine yardımcı olur.
3. Ekip içinde iletişimi ve iş birliğini kolaylaştırır.
4. Süreçleri optimize ederek zaman ve kaynak tasarrufu sağlar.

6.3. Akış Diyagramında Kullanılan Çizimler

[alternatif metin](#)

6.4. Akış Diyagramı Oluşturma Adımları

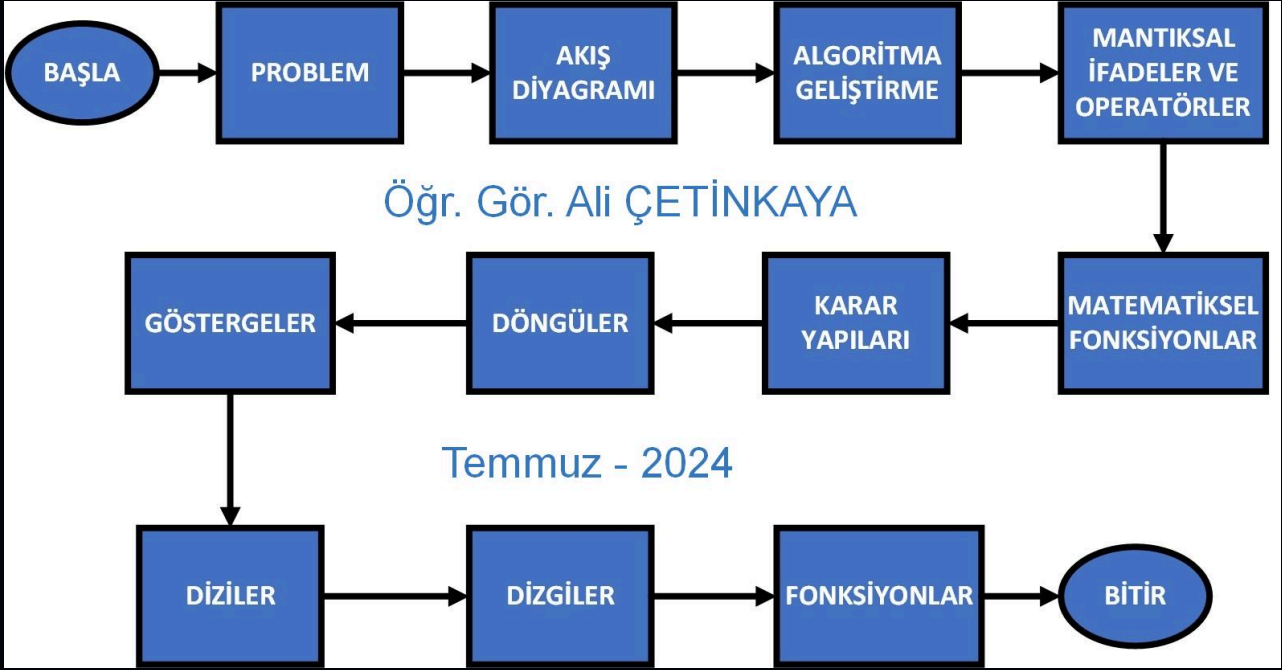
if Koşul Yapısının Akış Şemasında Gösterimi->



For Döngü Yapısının Akış Şemasında Gösterimi->



Programlama Dili Öğreniminde Öğrenme Aşamalarının Akış Şemasında Gösterimi->

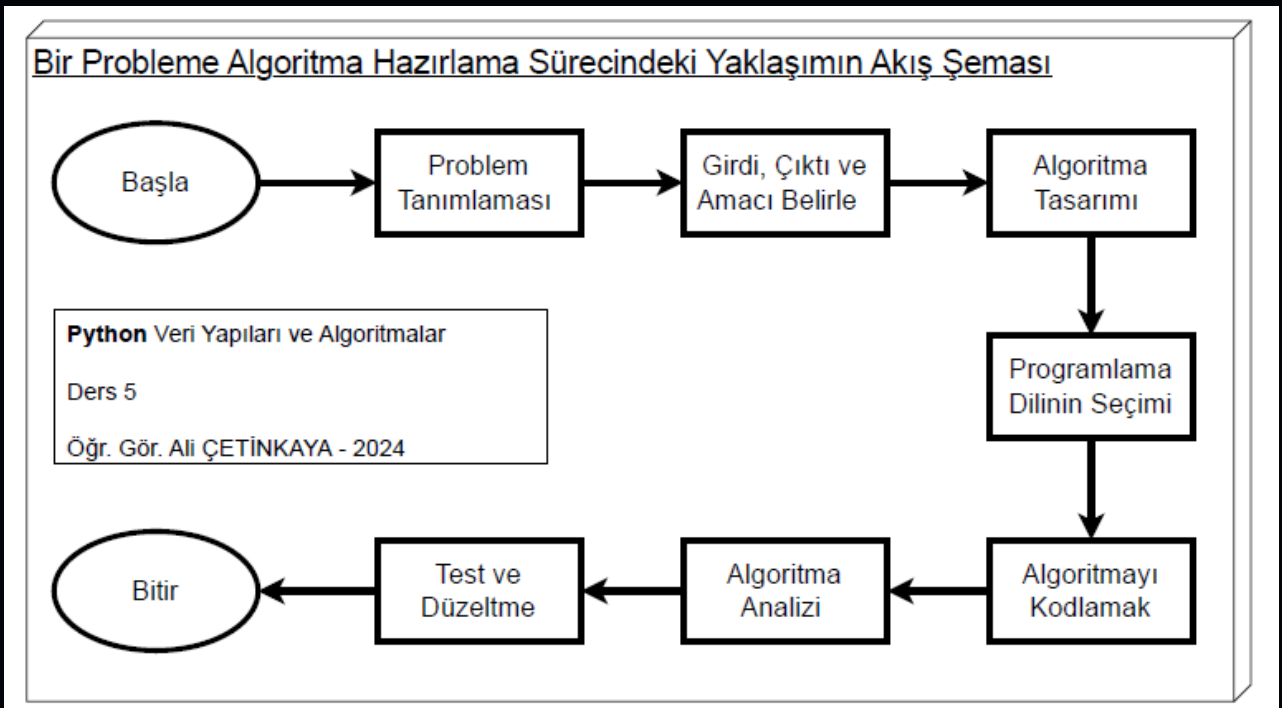


6.5. Akış Diyagramı Çizim Araçları

- 6.5.1. Draw.io
- 6.5.2. Diagrams.net
- 6.5.3. Microsoft Visio
- 6.5.4. Lucidchart

Draw.io ve Diagrams.net kullanılarak Akış Şeması Oluşturulması

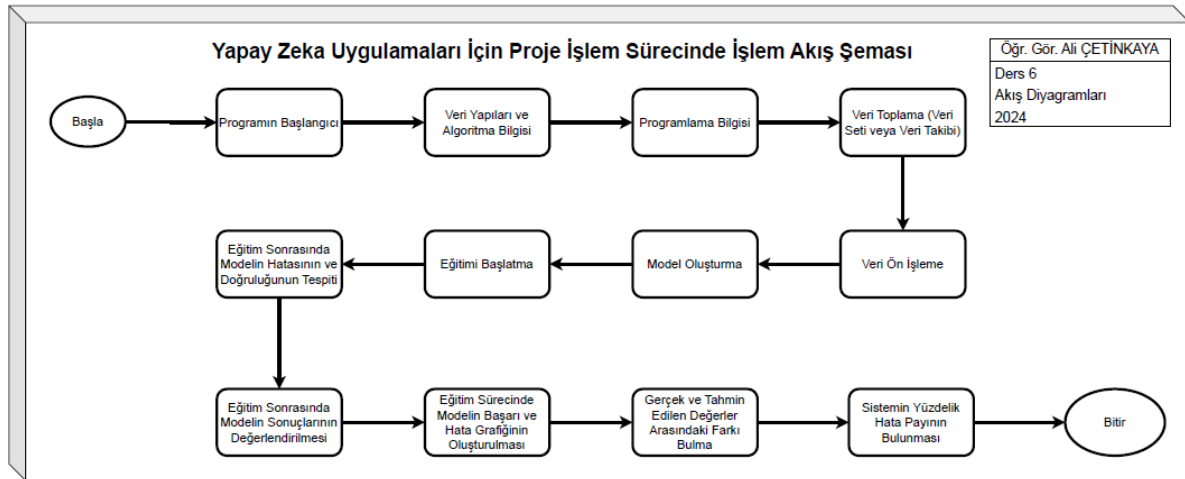
Örnek Çizimler - 1



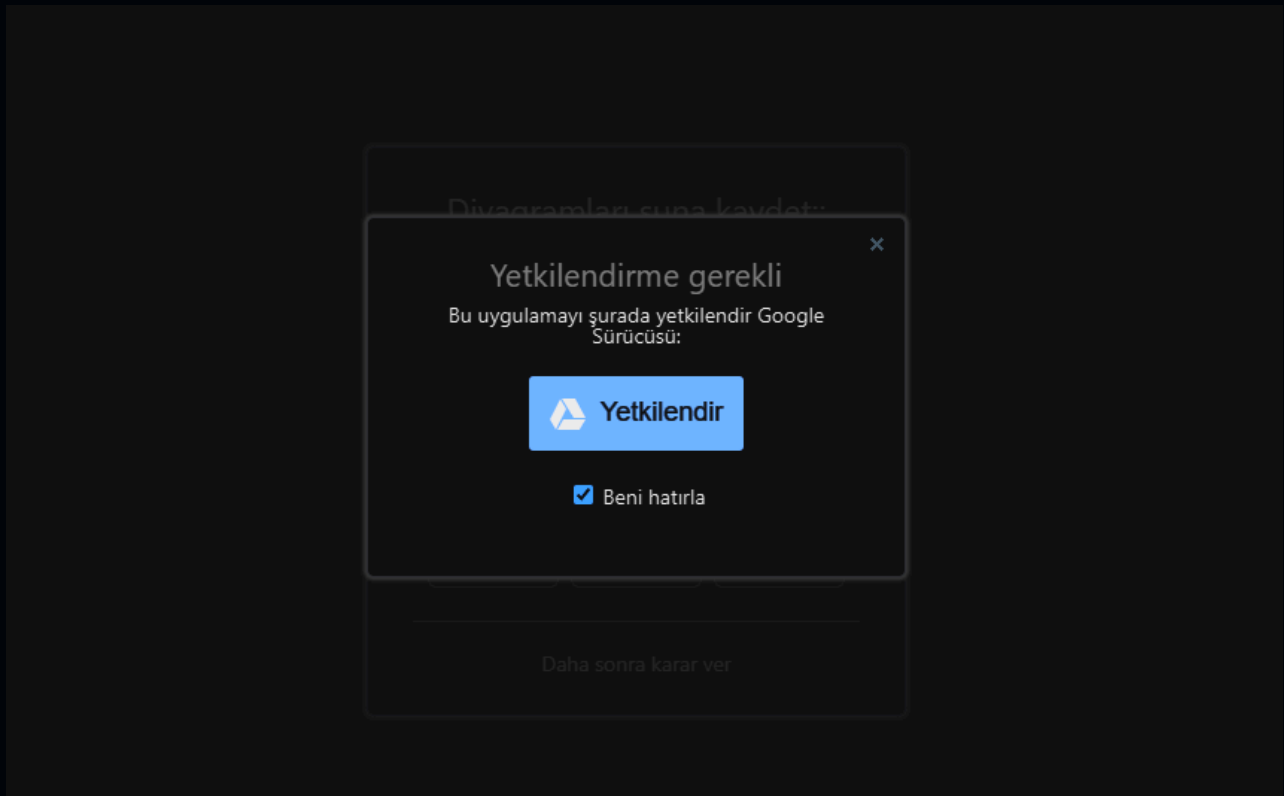
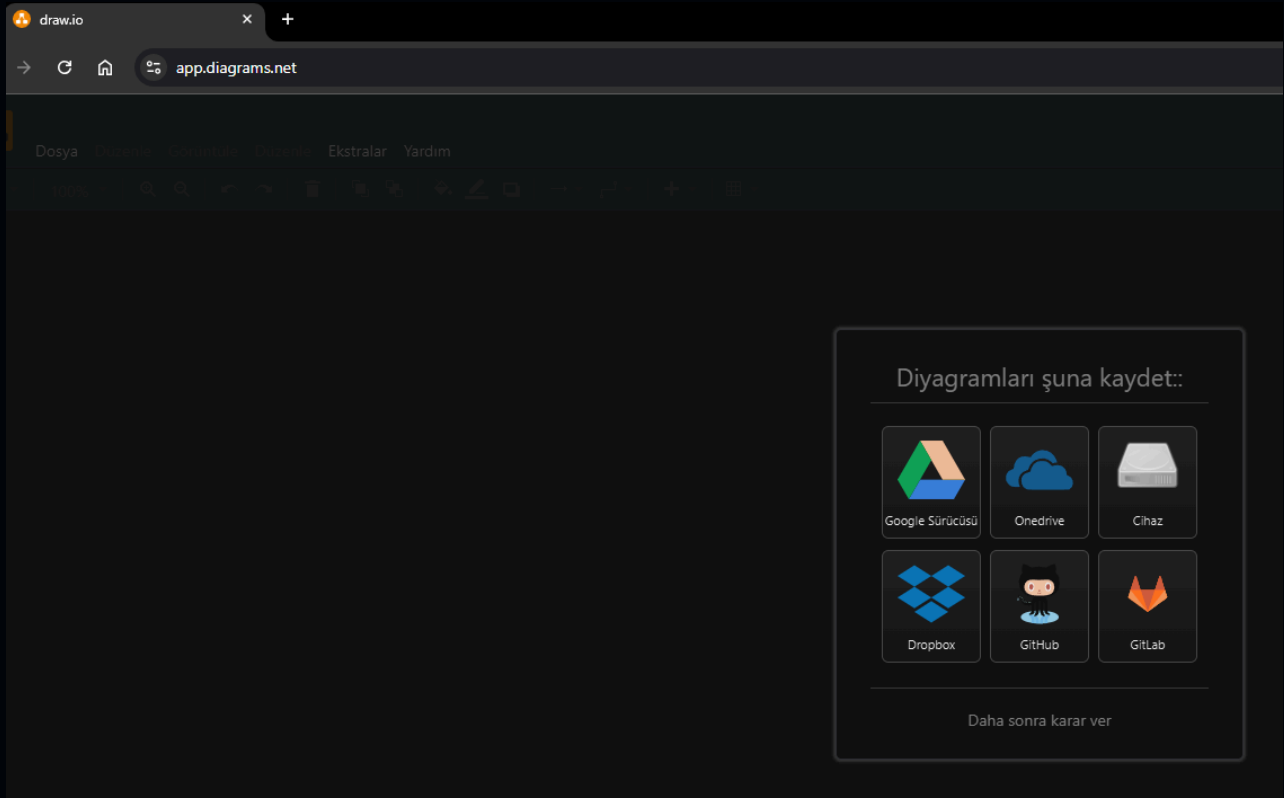
1. Başla

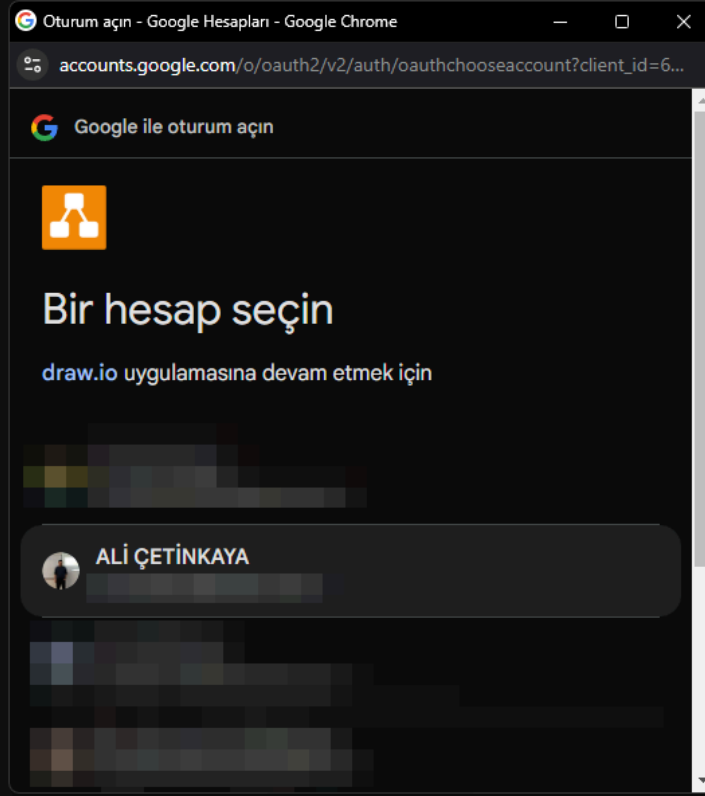
2. Problem Tanımlaması
3. Girdi, Çıktı ve Amacı Belirle
4. Algoritma Tasarımı :-> Adım adım çözüm sürecini tasarlamak. Burada kullanılan yöntemler: Pseudocode ve Akış Diyagramları'dır.
5. Programlama Dilinin Seçimi
6. Algoritmayı Kodlamak
7. Algoritma Analizi: Algoritmanın zaman ve alan karmaşıklığını analiz edin.
8. Test ve Düzeltme: Algoritmanın çeşitli test senaryolarında doğruluğunu test etmek ve gerekirse düzeltmeler yapmak.
9. Bitir

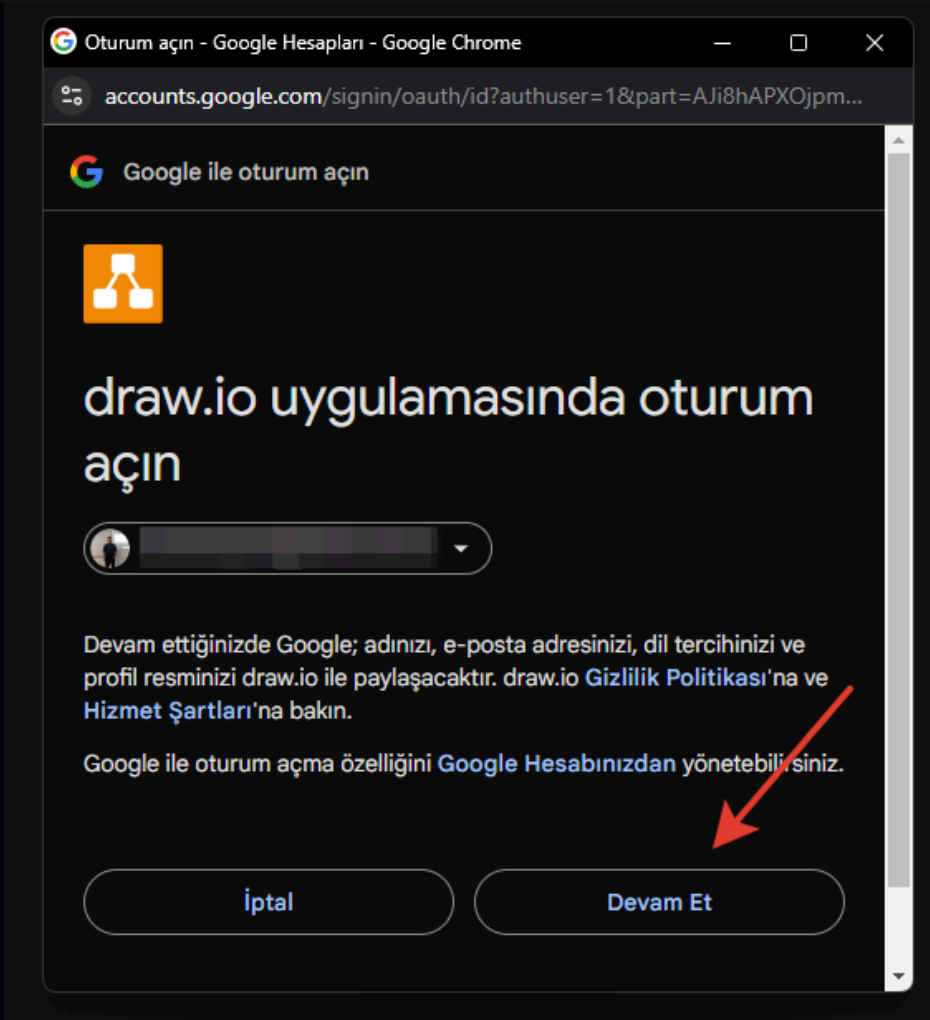
Örnek Çizimler - 2

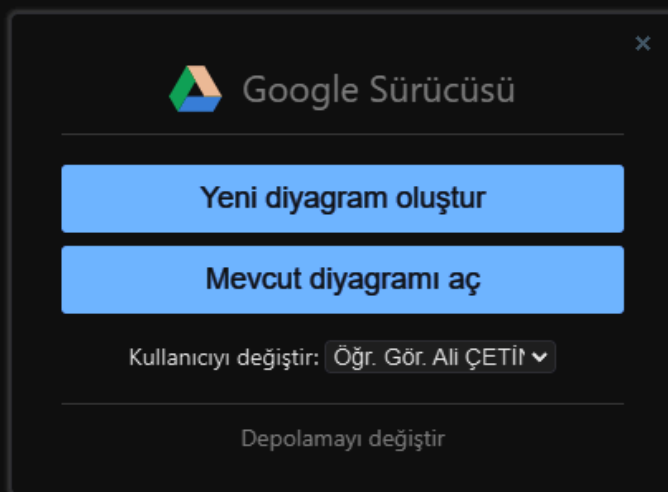
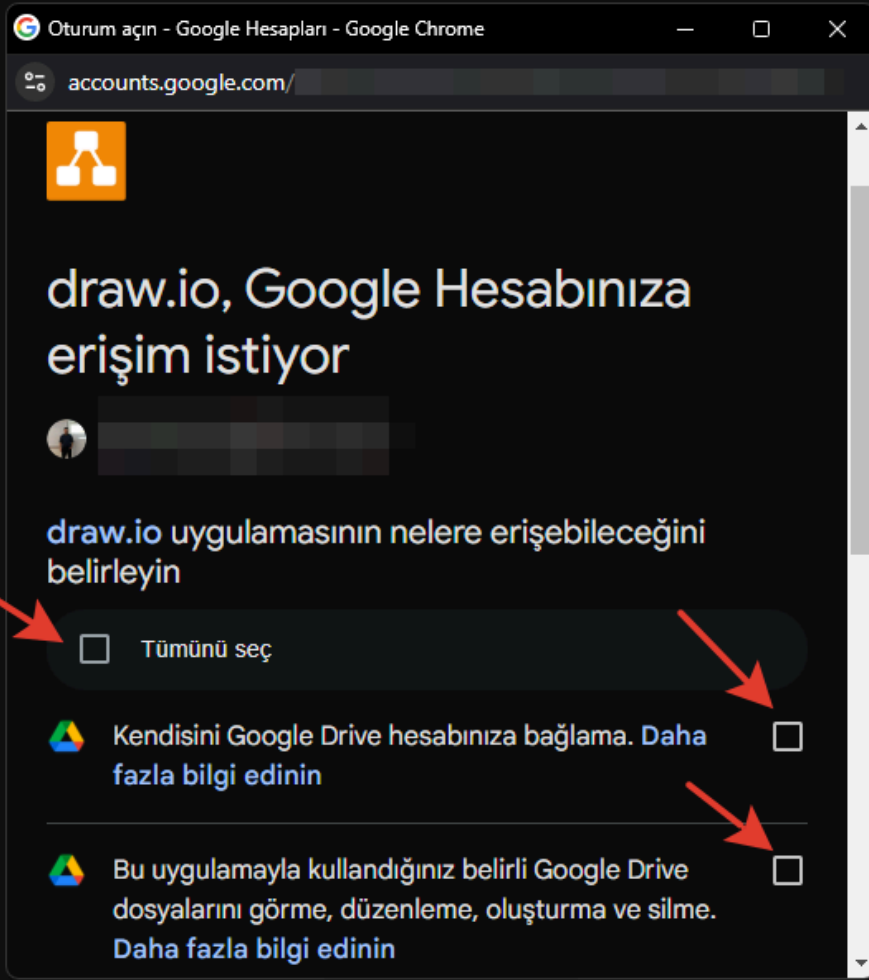


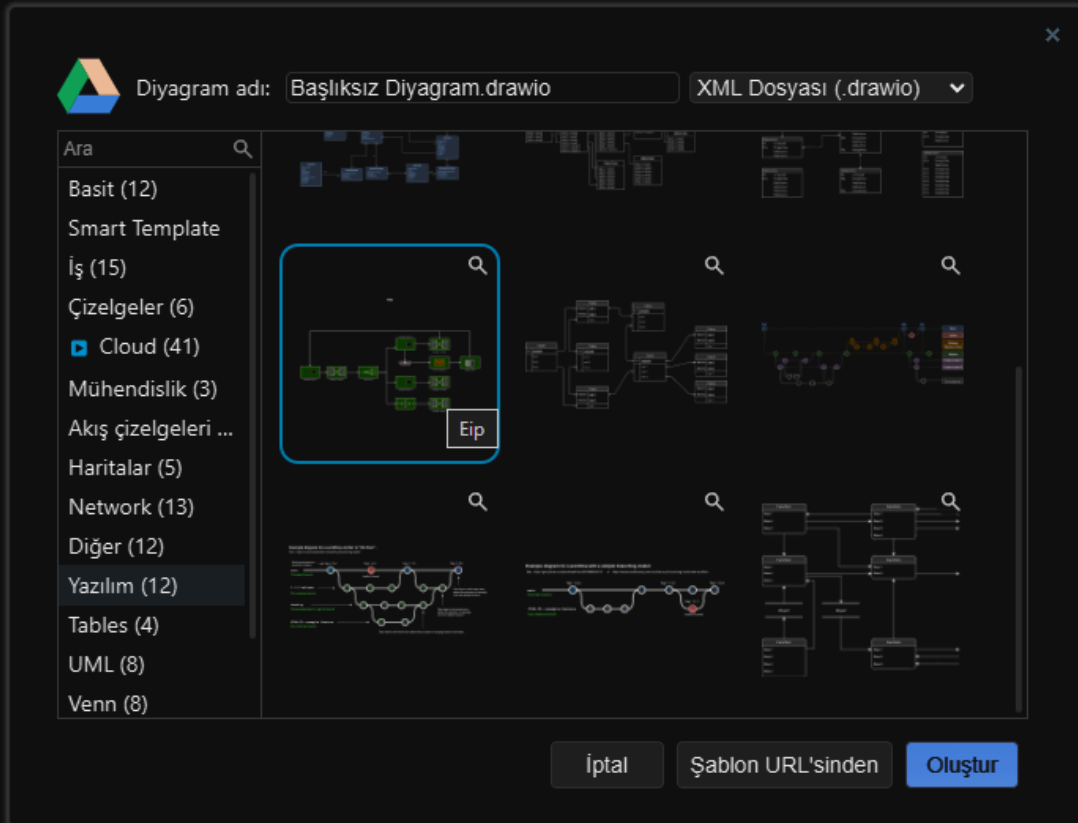
Draw.io ve Diagrams.net Genel Kullanımı

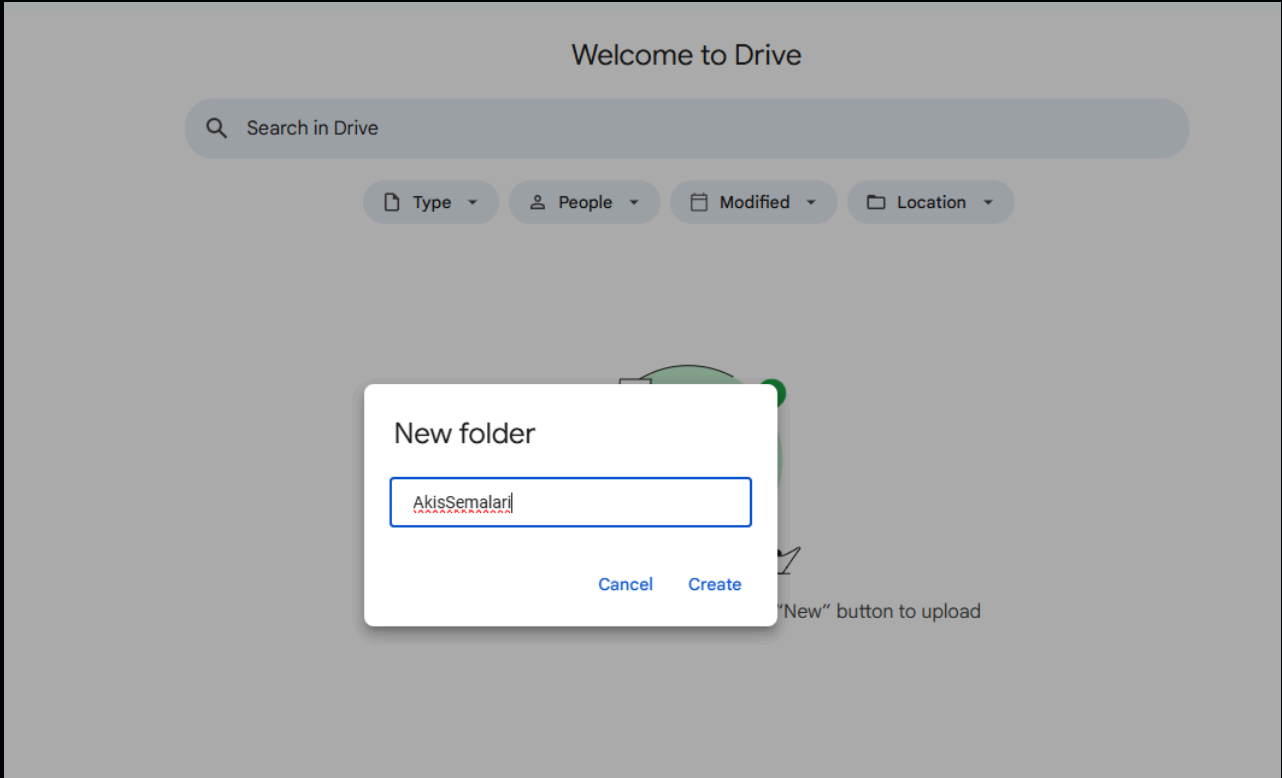
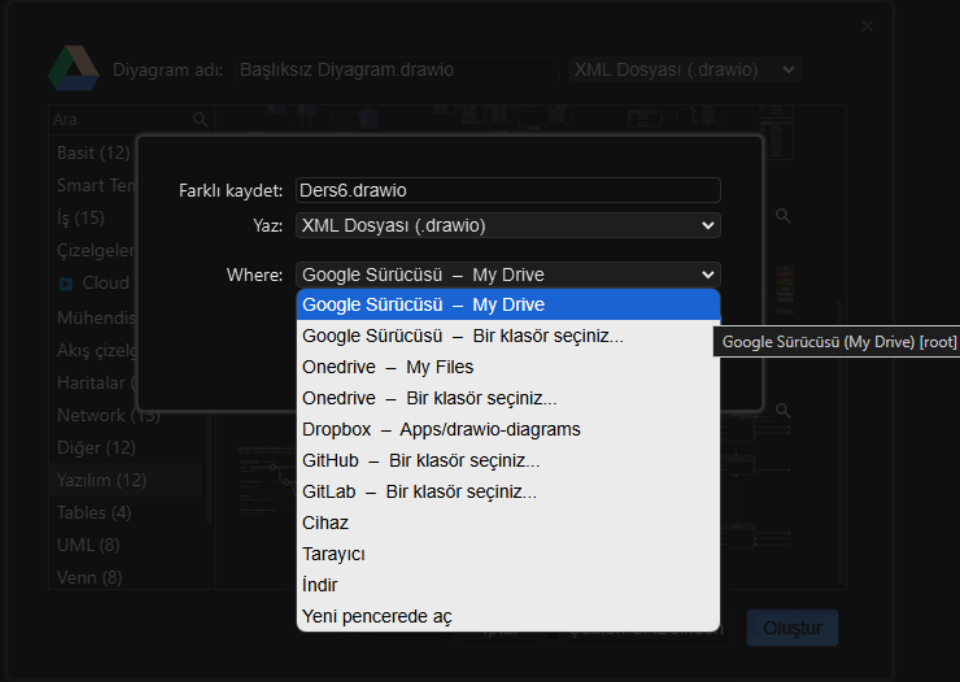


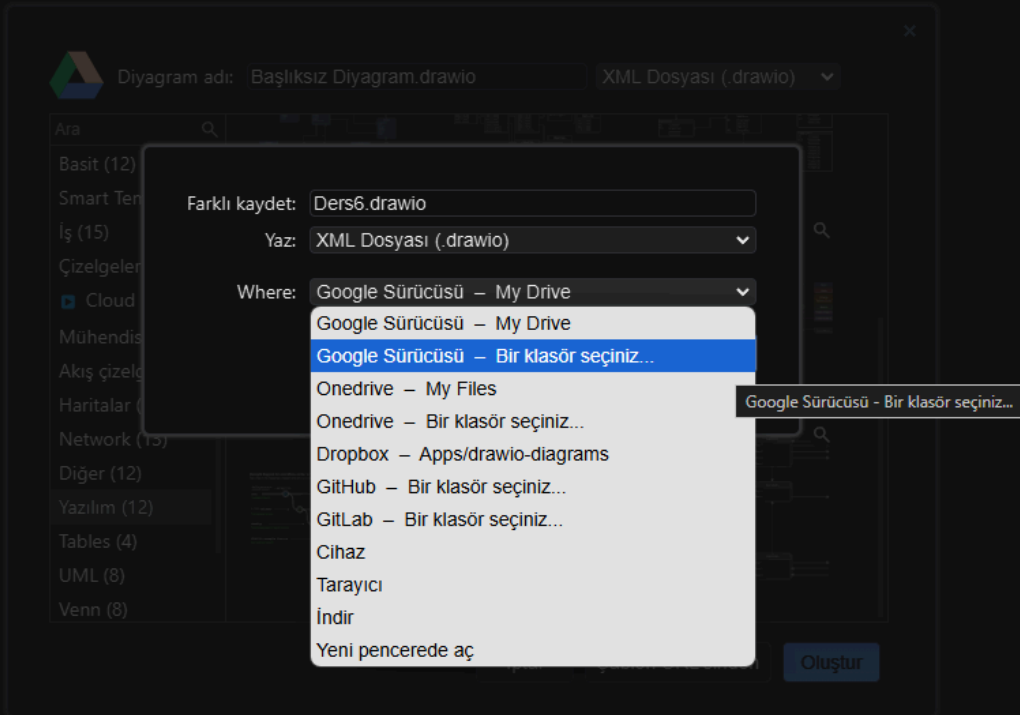
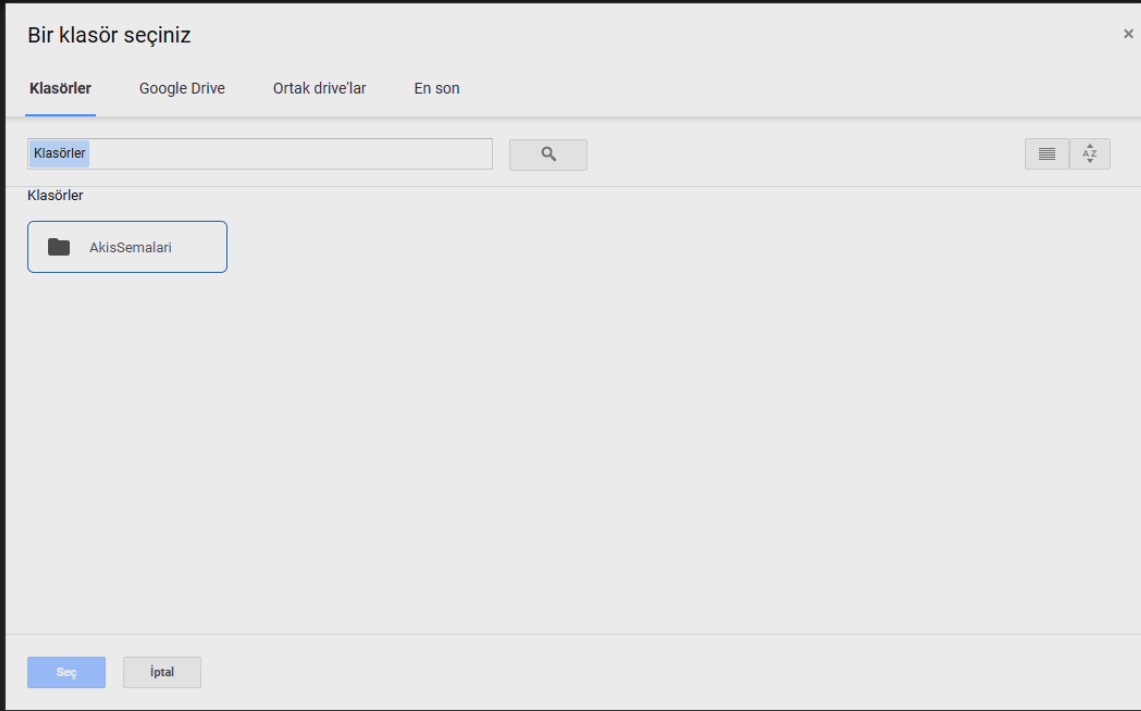


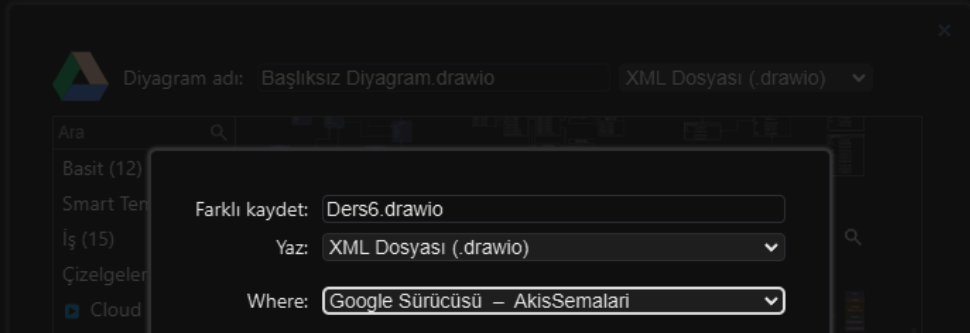
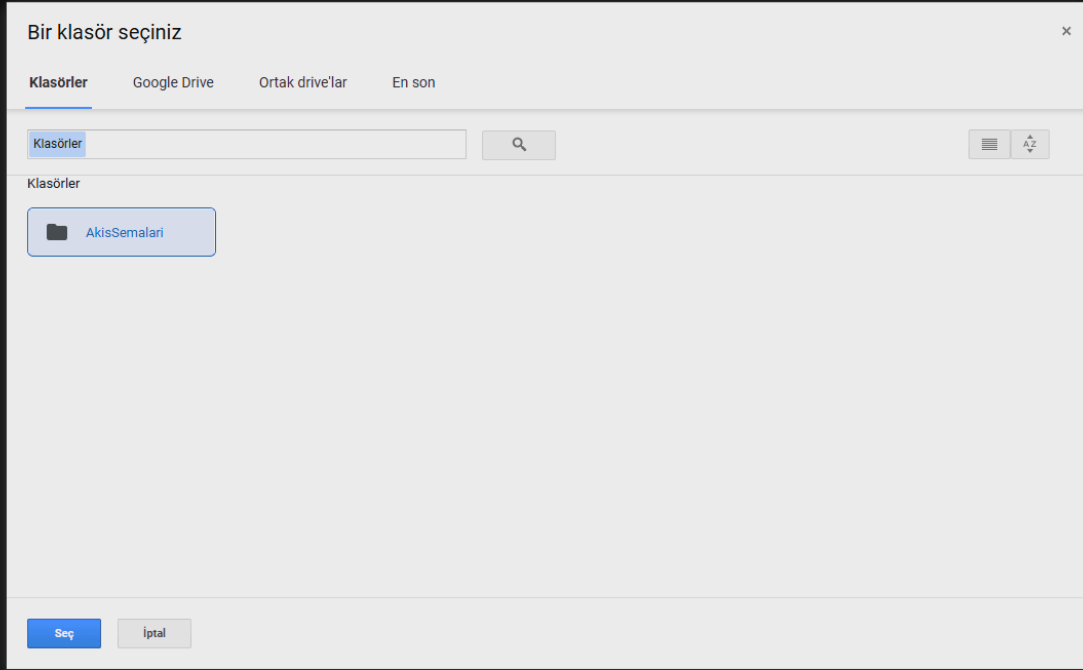












Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)