

# Program İşlevleri ve Satır Algoritmaları

---

Bu doküman, program işlevlerinin amacını, algoritmalarını ve programın genel tasarım prensiplerini detaylı bir şekilde açıklamaktadır. Her bir sınıf yönteminin ve işlevin görevleri ile nasıl çalıştıkları adım adım ele alınmıştır.

## 1 [FileValidator Sınıfı](#)

### 1.1 [Yöntemler](#)

- 1.1.1 validate\_path (None)
- 1.1.2 validate\_file (None)
- 1.1.3 validate\_read\_permission (None)
- 1.1.4 validate\_file\_name (None)

## 2 [FilePathManager Sınıfı](#)

### 2.1 [Yöntemler](#)

- 2.1.1 get\_py\_or\_exe\_dir (pathlib.Path)
- 2.1.2 convert\_f\_name\_to\_path (pathlib.Path)
- 2.1.3 save\_f\_path (None)
- 2.1.4 read\_f\_path (str)

## 3 [ImageResizer Sınıfı](#)

### 3.1 [Yöntemler](#)

- 3.1.1 \_validate\_byte\_len (None)
- 3.1.2 \_split\_bytes (int)
- 3.1.3 \_convert\_to\_int (int)
- 3.1.4 \_convert\_color\_to\_byte (bytearray)
- 3.1.5 read\_image (bytearray)
- 3.1.6 save\_image (None)
- 3.1.7 add\_grid (bytearray)
- 3.1.8 resize\_image (bytearray)

## 4 [Ek İşlevler](#)

- 4.1 get\_input (str)
- 4.2 getting\_log\_f\_path (pathlib.Path)
- 4.3 resetting\_log\_f\_path (None)
- 4.4 selecting\_file\_to\_process (None)
- 4.5 image\_gridding (None)
- 4.6 image\_resizing (None)

## 5 [Menü İşlevleri](#)

- 5.1 continue\_processing (None)
- 5.2 main\_menu (None)
- 6 [Versiyon Takip Tablosu](#)

## FileValidator Sınıfı

Açıklama: Bir dosya yolunun doğruluğunu, varlığını ve erişilebilirliğini kontrol etmek için işlevler sağlar.

### Yöntemler:

- **validate\_path (None):** Bir yolun boş veya geçersiz olup olmadığını doğrular. Yol boş veya geçersizse bir istisna fırlatır.

Satır Algoritması:

1. Bir dosya yolu al.
2. Alınan dosya yolu, boş ("None" veya "") veya geçersiz bir türde mi (type) kontrol et.
  1. Eğer boş veya geçersiz bir türdeyse, bir istisna (ValueError) fırlat ve işlevi sonlandır.
  2. Değilse işlevi sessizce sonlandır.

- **validate\_file (None):** Bir dosyanın belirtilen konumda bulunup bulunmadığını doğrular. Belirtilen konumda yoksa bir istisna fırlatır.

Satır Algoritması:

1. Bir dosya yolu al.
2. Dosya belirtilen konumda mı kontrol et.
  1. Eğer belirtilen konumda yoksa bir istisna (FileNotFoundError) fırlat ve işlevi sonlandır.
  2. Belirtilen konumdaysa işlevi sessizce sonlandır.

- **validate\_read\_permission (None):** Bir dosyanın okuma izinlerini kontrol eder. Dosya, okuma izinlerine sahip değilse bir istisna fırlatır.

Satır Algoritması:

1. Bir dosya yolu al.
2. Dosyanın okuma izni var mı kontrol et.
  1. Eğer dosyanın okuma izni yoksa bir istisna (PermissionError) fırlat ve işlevi sonlandır.
  2. Okuma izni varsa işlevi sessizce sonlandır.

- **validate\_file\_name (None):** Bir dosya adının boş bir metin olup olmadığını kontrol eder. Boşsa bir istisna fırlatır.

Satır Algoritması:

1. Bir dosya ismi al.
2. Dosya ismi boş bir metin mi kontrol et.
  1. Eğer boş bir metinse bir istisna (ValueError) fırlat ve işlevi sonlandır.
  2. Boş bir metin değilse işlevi sessizce sonlandır.

# FilePathManager Sınıfı

**Açıklama:** Dosya yollarını oluşturmak, yönetmek ve saklamak gibi sıkça kullanılan işlemler için işlevler sağlar.

## Yöntemler:

- `get_py_or_exe_dir (pathlib.Path)`: Uygulamanın bir betik dosyası veya executable olarak çalışmasına göre çalıştığı dizini alır.

Satır Algoritması:

1. Uygulama bir executable (örneğin .exe) olarak mı paketlenmiş kontrol et.
  1. Eğer bir executable ise çalıştırılabilir dosyanın bulunduğu dizini al.
  2. Bir executable değilse yani bir betik dosyası olarak çalıştırılmışsa, betiğin bulunduğu dizini al.
2. Değer olarak alınan dizini döndür.

- `convert_f_name_to_path (pathlib.Path)`: Bir dosya adına ve hedef klasöre göre tam dosya yolunu oluşturur. Dosya adı boş bir metinse bir istisna fırlatır. (Bağımlılıklar: 'FileValidator')

Satır Algoritması:

1. Dosyanın adını al.
2. İsteğe bağlı hedef klasör adını al.
3. Dosya adı boş bir metin mi kontrol et.
  1. Eğer boş bir metinse bir istisna fırlat ve işlevi sessizce sonlandır.
4. Uygulamanın çalıştığı dizini al.
5. Hedef klasör adı alınmış mı kontrol et.
  1. Eğer bir klasör ismi alınmışsa, değer olarak ana dizin, klasör ismi ve dosya adını birleştirip döndür.
6. Değer olarak ana dizin ve dosya adını birleştirip döndür.

- `save_f_path (None)`: Bir dosya yolunun başka bir dosyada saklanmasını sağlar. Verinin saklanacağı dosya belirtilen yolda zaten mevcutsa, yeni veriyi eski verinin üzerine yazar. (Bağımlılıklar: 'FileValidator')

Satır Algoritması:

1. Kayıt dosyasının yolunu al.
2. Kayıt edilecek yol bilgisini al.
3. İsteğe bağlı yol bilgisinin dosyaya ait olup olmadığına bakma. Varsayılan olarak dosyanın kayıt edilecek yolda olup olmadığına bakılır.
4. Alınan kayıt dosyası yolu boş veya geçersiz mi kontrol et.
  1. Eğer boş veya geçersizse bir istisna (ValueError) fırlat ve işlevi sonlandır.
5. Dosya kayıt edilecek yolda mı kontrol et.
  1. Eğer dosya kayıt edilecek yolda yoksa, bir hata (FileNotFoundError) fırlat ve işlevi sonlandır.
6. Belirtilen kayıt dosyasına yol bilgisini yazdır.

1. Bu işlem sırasında bir hata oluşursa bir istisna (RuntimeError) fırlat ve işlevi sonlandır.
  2. Bir hata oluşmazsa işlevi sessizce sonlandır.
- `read_f_path (str)`: Belirtilen kayıt dosyasında saklanan yol bilgisini okur. (Bağımlılıklar: 'FileValidator')

#### Satır Algoritması:

1. Kayıt dosyasının yolunu al.
2. Alınan kayıt dosyası yolu boş veya geçersiz mi kontrol et.
  1. Eğer boş veya geçersizse bir istisna (ValueError) fırlat ve işlevi sonlandır.
3. Dosya belirtilen konumda mı kontrol et.
  1. Eğer belirtilen konumda yoksa bir istisna (FileNotFoundError) fırlat ve işlevi sonlandır.
4. Dosyanın okuma izni var mı kontrol et.
  1. Eğer dosyanın okuma izni yoksa bir istisna (PermissionError) fırlat ve işlevi sonlandır.
5. Belirtilen kayıt dosyasında saklanan yol bilgisini oku.
  1. Bu işlem sırasında bir hata oluşursa bir istisna (RuntimeError) fırlat ve işlevi sonlandır.
  2. Bir hata oluşmazsa işlevi sessizce sonlandır.

## ImageResizer Sınıfı

Açıklama: 'BMP' uzantılı bir dosyaya ızgara eklemek ve yeniden boyutlandırmak için işlevler sağlar.

### Yöntemler:

- `_validate_byte_len (None)`: Mevcut bayt sayısı ile beklenen bayt sayısını karşılaştıran, özel metot. Bayt sayıları eşit değilse bir istisna fırlatır.

#### Satır Algoritması:

1. Mevcut bayt sayısını al.
2. Beklenen bayt sayısını al.
3. Mevcut bayt sayısı ile beklenen bayt sayısını karşılaştır.
  1. Eğer bayt sayıları eşit değilse bir istisna (ValueError) fırlat ve işlevi sonlandır.
  2. Eşitse işlevi sessizce sonlandır.

- `_split_bytes (int)`: Bit derinliğini baza olarak, piksel başına bayt sayısını belirleyen, özel metot. Bit derinliği desteklenmiyorsa bir istisna fırlatır.

#### Satır Algoritması:

1. Bit derinliğini al.
2. Bit değeri tanımlı değerlere (32, 24, 16, 8) eşit mi kontrol et.
  1. Eğer 32'ye eşitse, değer olarak 4'ü döndür.
  2. Değilse, eğer 24'e eşitse değer olarak 3'ü döndür.
  3. Değilse, eğer 16'ya eşitse değer olarak 2'yi döndür.
  4. Değilse, eğer 8'e eşitse değer olarak 1'i döndür.
  5. Hiçbiri değilse, bir istisna (ValueError) fırlat ve işlevi sonlandır.

- `_convert_to_int (int)`: Sayısal bir metin değerini "int" bir değere dönüştüren, özel metot. Değer sayısal bir metin değilse bir istisna fırlatır.

#### Satır Algoritması:

1. Bir metin değeri al.
  2. Alınan değeri 'int' bir değere dönüştür ve değer olarak döndür.
    1. Bu işlem sırasında alınan metin 'str' türünde değilse bir istisna (TypeError) fırlat ve işlevi sonlandır.
    2. Bu işlem sırasında alınan metin sayısal bir değer içermiyorsa, bir istisna (ValueError) fırlat ve işlevi sonlandır.
    3. Bir hata oluşmazsa işlevi sessizce sonlandır.
- `_convert_color_to_byte (bytearray)`: Tanımlı renk adını, RGB formatında bir "bytearray"'e dönüştüren, özel metot. Renk tanımlı değilse bir istisna fırlatır.

#### Satır Algoritması:

1. Bir renk adı al.
  2. Alınan renk adı:
    1. "color\_name1" ise değer olarak "bytearray1" döndür.
    2. "color\_name2" ise değer olarak "bytearray2" döndür.
    3. Alınan renk adı hiçbirisi değilse, bir istisna (ValueError) fırlat ve işlevi sonlandır.
- `read_image (bytearray)`: Görüntü dosyasını "binary" olarak okur. (Bağımlılıklar: 'FileValidator')

#### Satır Algoritması:

1. Görüntü dosyasının yolunu al.
  2. Alınan görüntü dosyası yolu boş veya geçersiz mi kontrol et.
    1. Eğer boş veya geçersizse bir istisna (ValueError) fırlat ve işlevi sonlandır.
  3. Dosya belirtilen konumda mı kontrol et.
    1. Eğer belirtilen konumda yoksa bir istisna (FileNotFoundError) fırlat ve işlevi sonlandır.
  4. Dosyanın okuma izni var mı kontrol et.
    1. Eğer dosyanın okuma izni yoksa bir istisna (PermissionError) fırlat ve işlevi sonlandır.
  5. Belirtilen görüntü dosyasını 'bytearray' olarak oku ve değer olarak döndür.
    1. Bu işlem sırasında bir hata oluşursa bir istisna (RuntimeError) fırlat ve işlevi sonlandır.
    2. Bir hata oluşmazsa işlevi sessizce sonlandır.
- `save_image (None)`: Görüntü dosyasının güncellenmiş binary içeriğini belirtilen konuma kaydeder. (Bağımlılıklar: 'FileValidator')

#### Satır Algoritması:

1. Görüntü dosyasının kaydedileceği yolu al.
  2. Kaydedilecek görüntüye ait 'bytearray' verisini al.
  3. Alınan kayıt dosyası yolu boş veya geçersiz mi kontrol et.
    1. Eğer boş veya geçersizse bir istisna (ValueError) fırlat ve işlevi sonlandır.
  4. Alınan 'bytearray' verisini belirtilen konuma yazdır.
    1. Bu işlem sırasında bir hata oluşursa bir istisna (RuntimeError) fırlat ve işlevi sonlandır.
    2. Bir hata oluşmazsa işlevi sessizce sonlandır.
- `add_grid (bytearray)`: Resme ızgara ekler.

#### Satır Algoritması:

1. İşlenecek görsele ait 'bytearray' verisini al.
2. Izgara kareleri için bir boyut al.
3. Izgara çizgileri için bir renk al.
4. Alınan verideki, piksel verisinin başlangıç indeksini bul.
5. Başlangıç indeksini kullanarak piksel verisini alınan (ana) veriden ayır.
6. Ana verideki, görsele ait genişlik bilgisini al.
7. Ana verideki, görsele ait yükseklik bilgisini al.
8. Ana verideki, görsele ait bit derinliği bilgisini al.
9. Alınan bit derinliğine göre piksel başına düşecek bayt sayısını belirle.
10. Izgara karelerinin boyutu için alınan 'str' değeri 'int' bir değere dönüştür.
11. Izgara çizgileri için alınan renk adını bir 'bytearray'e dönüştür.
12. Satır ve sütun döngüleri başlat.
  1. Dış döngü: Satırların (y) üzerinde gezin.
  2. İç döngü: Her satırdaki sütunların (x) üzerinde gezin.
    1. Döngü ile alınan her piksel verisinin (baytlarının) başlangıç indeksini belirle.
    2. Piksel izgara çizgisi üzerinde mi kontrol et.
      1. Eğer izgara çizgisi üzerindeyse başlangıç indeksini kullanarak ilgili piksel baytlarını, dönüştürülen renk baytlarıyla değiştir.
13. Ana verideki piksel verisini yeni piksel verisiyle değiştir.
14. Değer olarak güncellenmiş görsel verisini döndür.

- `resize_image (bytearray)`: Resmi yeniden boyutlandırır.

#### Satır Algoritması:

1. İşlenecek görsele ait 'bytearray' verisini al.
2. Yeni bir genişlik al.
3. Yeni bir yükseklik al.
4. Boyutlandırmanın başlayacağı yatay konumu (x) al.
5. Boyutlandırmanın başlayacağı dikey konumu (y) al.
6. Alınan verideki, piksel indeksinin başlangıç indeksini bul.
7. Başlangıç indeksini kullanarak piksel verisini alınan (ana) veriden ayır.
8. Ana verideki, görsele ait genişlik bilgisini al.
9. Ana verideki, görsele ait yükseklik bilgisini al.
10. Ana verideki, görsele ait bit derinliğini al.
11. Alınan bit derinliğine göre piksel başına düşecek bayt sayısını belirle.
12. Yeni genişlik boyutu için alınan 'str' değeri 'int' bir değere dönüştür.
13. '4 <= yeni genişlik <= genişlik' durumu sağlanıyor mu kontrol et.
  1. Eğer durum sağlanmıyorsa bir istisna (ValueError) fırlat ve işlevi sonlandır.
14. Yeni yükseklik boyutu için alınan 'str' değeri 'int' bir değere dönüştür.
15. '4 <= yeni yükseklik <= yükseklik' durumu sağlanıyor mu kontrol et.
  1. Eğer durum sağlanmıyorsa bir istisna (ValueError) fırlat ve işlevi sonlandır.
16. Yataydaki başlangıç konumu için alınan 'str' değeri 'int' bir değere dönüştür.
17. 'başlangıç x >= 0' durumu sağlanıyor mu kontrol et.
  1. Eğer durum sağlanmıyorsa bir istisna (ValueError) fırlat ve işlevi sonlandır.
18. Dikeydeki başlangıç konumu için alınan 'str' değeri 'int' bir değere dönüştür.
19. 'başlangıç y >= 0' durumu sağlanıyor mu kontrol et.
  1. Eğer durum sağlanmıyorsa bir istisna (ValueError) fırlat ve işlevi sonlandır.
20. Yeni görsele dahil olacak pikselleri tutmak için bir boş bir koleksiyon oluştur.
21. Satır ve sütun döngüleri başlat.

1. Dış döngü: Satırların (y) üzerinde gezin.
  1. Şu anki y değeri görüntüye dahil mi kontrol et.
    1. Eğer 'y < başlangıç y' ise dış döngüye devam et.
    2. Değilse, eğer 'y >= başlangıç y + yeni yükseklik' ise dış döngüye devam et.
  2. İç döngü: Her satırdaki sütunların (x) üzerinde gezin
    1. Şu anki x değeri görüntüye dahil mi kontrol et.
      1. Eğer 'x < başlangıç x' ise iç döngüye devam et.
      2. Değilse, eğer 'x >= başlangıç x + yeni genişlik' ise iç döngüye devam et.
    2. Döngü ile alınan her piksel verisinin (baytlarının) başlangıç indeksini belirle.
    3. Oluşturulan boş koleksiyona, piksel verisi içindeki yeni görsele dahil olan pikselleri ekle.
22. Yeni piksel verisindeki bayt sayısı, beklenen bayt sayısına (yeni genişlik \* yeni yükseklik \* piksel başına düşen bayt sayısı) eşit mi kontrol et.
  1. Eğer eşit değilse, bir istisna (ValueError) fırlat ve işlevi sonlandır.
23. Ana verideki, görsele ait genişlik bilgisini yeni genişlik bilgisiyle değiştir.
24. Ana verideki, görsele ait yükseklik bilgisini yeni yükseklik bilgisiyle değiştir.
25. Değer olarak güncellenmiş görsel verisini döndür.

## Ek İşlevler

Açıklama: Programın çalışması için gerekli süreçleri yöneten işlevleri sağlar.

- `get_input (str)`: Kullanıcıdan özel bir metin aracılığıyla veri alır.

Satır Algoritması:

1. Kullanıcıya gösterilmek istenen metni al.
2. Alınan metni kullanarak kullanıcıdan bir veri al ve değer olarak alınan veriyi döndür.

- `getting_log_f_path (pathlib.Path)`: Kayıt dosyası yolunu oluşturur. (Bu işlem, bakımı daha kolay olması için ayrı bir işlev olarak tanımlanmıştır.)

Satır Algoritması:

1. Kayıt dosyası yolunu dinamik olarak oluştur ve değer olarak bu yolu döndür.

- `resetting_log_f_path (None)`: Kayıtlı dosya yolunu sıfırlamak için gereken süreci işler.

Satır Algoritması:

1. Kayıt dosyası yolunu al.
2. Kayıtlı dosya yolunu oku.
  1. Bu işlem sırasında bir hata oluşursa:
    1. Hatayı yakala ve hataya dair bir mesaj yazdır.
    2. Boş bir değer (None) döndür ve işlevi sonlandır.
  2. Bir hata oluşmazsa işleve devam et.
3. Okunan yolu yazdır.
4. Kullanıcıya mevcut dosya yolunun sıfırlanmak üzere olduğunu bildiren bir mesaj yazdır.
5. Kullanıcıya işleme devam edip, etmek istemediğini sor.
  1. Devam etmek istemezse ana menüye dön.
  2. Eğer devam etmek isterse sürece devam et.

6. Kayıt dosyasına "Kayıtlı dosya yolu yok." yazdır.
    1. Bu işlem sırasında bir hata oluşursa:
      1. Hatayı yakala ve hataya dair bir mesaj yazdır.
      2. Boş bir değer (None) döndür ve işlevi sonlandır.
    2. Bir hata oluşmazsa işleve devam et.
  7. Sıfırlama işleminin başarılı olduğuna dair bir mesaj yazdır ve işlevi sonlandır.
- `selecting_file_to_process (None)`: İşlenecek dosyanın yolunu oluşturmak ve saklamak için gereken süreci işler.

#### Satır Algoritması:

1. Kayıt dosyası yolunu al.
  2. Kayıtlı dosya yolunu oku.
    1. Bu işlem sırasında bir hata oluşursa:
      1. Hatayı yakala ve hataya dair bir mesaj yazdır.
      2. Boş bir değer (None) döndür ve işlevi sonlandır.
    2. Bir hata oluşmazsa işleve devam et.
  3. Okunan yolu yazdır.
  4. Düzenlenecek görselin tam adını (uzantı dahil) al.
  5. Kullanıcıya işleme devam edip, etmek istemediğini sor.
    1. Devam etmek istemezse ana menüye dön.
    2. Eğer devam etmek isterse sürece devam et.
  6. Alınan görsel ismini kullanarak görselin tam yolunu oluştur.
  7. Oluşturulan dosya yolunu 'str' olarak kayıt dosyasına yazdır.
    1. Bu işlem sırasında bir hata oluşursa:
      1. Hatayı yakala ve hataya dair bir mesaj yazdır.
      2. Boş bir değer (None) döndür ve işlevi sonlandır.
  8. Görüntü dosyası seçme işleminin başarılı olduğuna dair bir mesaj yazdır ve işlevi sonlandır.
- `image_gridding (None)`: Seçilen görsele ızgara eklemek ve yeni bir dosya olarak kaydetmek için gereken süreci işler.

#### Satır Algoritması:

1. Kayıt dosyasının yolunu al.
2. Kayıt dosyasından görselin bulunduğu yol bilgisini oku.
  1. Bu işlem sırasında bir hata oluşursa:
    1. Hatayı yakala ve hataya dair bir mesaj yazdır.
    2. Boş bir değer (None) döndür ve işlevi sonlandır.
3. Okunan yol bilgisini 'str'den 'pathlib.Path'e çevir. (read\_image(): validate\_path() metotlarının hata vermemesi için.)
4. Görsel dosyasını oku.
  1. Bu işlem sırasında bir hata oluşursa:
    1. Hatayı yakala ve hataya dair bir mesaj yazdır.
    2. Boş bir değer (None) döndür ve işlevi sonlandır.
5. Yeni görselin kaydedileceği ismi al.
6. Alınan ismi bir yola dönüştür.
  1. Bu işlem sırasında bir hata oluşursa:
    1. Hatayı yakala ve hataya dair bir mesaj yazdır.



2. Boş bir değer (None) döndür ve işlevi sonlandır.
  7. Izgara kareleri için bir büyüklük al.
  8. Izgara çizgileri için bir renk al ve küçük harflere dönüştür.
  9. Alınan renk boş string veya boşluk mu kontrol et.
    1. Boş string veya boşluksa varsayılan değer olarak 'beyaz' al.
  10. Kullanıcıya işleme devam edip, etmek istemediğini sor.
    1. Devam etmek istemezse ana menüye dön.
    2. Eğer devam etmek isterse sürece devam et.
  11. Görsele ızgara ekle.
    1. Bu işlem sırasında bir hata oluşursa:
      1. Hatayı yakala ve hataya dair bir mesaj yazdır.
      2. Boş bir değer (None) döndür ve işlevi sonlandır.
  12. Izgara eklenen yeni görseli, oluşturulan yola alınan isimde kaydet.
  13. Görüntüye ızgara ekleme işleminin başarılı olduğuna dair bir mesaj yazdır ve işlevi sonlandır.
- **image\_resizing (None):** Seçilen görseli yeniden boyutlandırmak ve yeni bir dosya olarak kaydetmek için gereken süreci işler.

#### Satır Algoritması:

1. Kayıt dosyasının yolunu al.
2. Kayıt dosyasından görselin bulunduğu yol bilgisini oku.
  1. Bu işlem sırasında bir hata oluşursa:
    1. Hatayı yakala ve hataya dair bir mesaj yazdır.
    2. Boş bir değer (None) döndür ve işlevi sonlandır.
3. Okunan yol bilgisini 'str'den 'pathlib.Path'e çevir. (read\_image(): validate\_path() metotlarının hata vermemesi için.)
4. Görsel dosyasını oku.
  1. Bu işlem sırasında bir hata oluşursa:
    1. Hatayı yakala ve hataya dair bir mesaj yazdır.
    2. Boş bir değer (None) döndür ve işlevi sonlandır.
5. Yeni görselin kaydedileceği ismi al.
6. Alınan ismi bir yola dönüştür.
  1. Bu işlem sırasında bir hata oluşursa:
    1. Hatayı yakala ve hataya dair bir mesaj yazdır.
    2. Boş bir değer (None) döndür ve işlevi sonlandır.
7. Görsel için istenilen yeni genişliği al.
8. Görsel için istenilen yeni yüksekliği al.
9. Boyutlandırmanın başlayacağı yatay konumu (x) al.
10. Alınan yatay konuma bir değer verilmiş mi kontrol et.
  1. Verilmemişse varsayılan olarak '0' al
11. Boyutlandırmanın başlayacağı dikey konumu (y) al.
12. Alınan dikey konuma bir değer verilmiş mi kontrol et.
  1. Verilmemişse varsayılan olarak '0' al
13. Kullanıcıya işleme devam edip, etmek istemediğini sor.
  1. Devam etmek istemezse ana menüye dön.
  2. Eğer devam etmek isterse sürece devam et.
14. Görseli yeniden boyutlandır.
  1. Bu işlem sırasında bir hata oluşursa:

1. Hatayı yakala ve hataya dair bir mesaj yazdır.
  2. Boş bir değer (None) döndür ve işlevi sonlandır.
15. Yeniden boyutlandırılan yeni görseli, oluşturulan yola alınan isimde kaydet.
16. Yeniden boyutlandırma işleminin başarılı olduğuna dair bir mesaj yazdır ve işlevi sonlandır.

## Menü İşlevleri

Açıklama: Program akışını yönetmek ve komut satır arayüzü (CLI) için gereken menü işlevlerini sağlar.

- `continue_processing` (None): Kullanıcıya mevcut işleme devam edilip, edilmeyeceği sorulur. Cevap evetse işleme devam edilir, hayırsa ana menüye dönülür.

Satır Algoritması:

1. Sonsuz döngü başlat.
  1. Kullanıcıdan "İşleme devam et (e/h)" için bir giriş al ve küçük harflere dönüştür.
    1. e: Evet, işleme devam et.
    2. h: Hayır, ana menüye dön.
  2. Alınan girişi kontrol et.
    1. Eğer cevap hayırsa:
      1. "Ana menüye dönülüyor..." yazdır.
      2. Bir süre bekle (örneğin 1sn).
      3. Ana menüye dön ve döngüyü sonlandır.
    2. Değilse, eğer cevap evetse, döngüden çık ve işlevi sessizce sonlandır.
    3. Hiçbiri değilse (cevap geçersizse) döngüye devam et ve tekrar giriş al.

- `main_menu` (None): Uygulamanın ana menüsünü başlatır. Kullanıcıdan bir işlem seçmesini ister ve seçilen işleme göre ilgili süreci başlatır.

Satır Algoritması:

1. Programa ait işlemler ve anahtarlarının olduğu bir koleksiyon oluştur (örneğin liste).
2. Sonsuz döngü başlat.
  1. Program başlığını yazdır.
  2. Koleksiyon elemanları için bir döngü başlat.
    1. Koleksiyon elemanlarını ekrana yazdır.
  3. Kullanıcıdan istenilen işleme ait anahtar için bir giriş al ve küçük harflere dönüştür.
  4. Alınan girişi kontrol et.
    1. Eğer giriş 'exit' ise:
      1. "Uygulamadan çıkış yapılıyor..." yazdır.
      2. Bir süre bekle (örneğin 1sn).
      3. Uygulamadan çık.
    2. Değilse, eğer giriş '0' ise:
      1. Sürecin başlığını yazdır.
      2. Seçili dosya yolu sıfırlama sürecini başlat.
      3. Bir süre bekle (örneğin 1sn).
      4. Döngüye devam et ve tekrar listeyi yazdırıp giriş al.
    3. Değilse, eğer '1' ise:

1. Sürecin başlığını yazdır.
2. Görüntü dosyası seçme sürecini başlat.
3. Bir süre bekle (örneğin 1sn).
4. Döngüye devam et ve tekrar listeyi yazdırıp giriş al.
4. Değilse, eğer '2' ise:
  1. Sürecin başlığını yazdır.
  2. Resim ızgaralama sürecini başlat.
  3. Bir süre bekle (örneğin 1sn).
  4. Döngüye devam et ve tekrar listeyi yazdırıp giriş al.
5. Değilse, eğer '3' ise:
  1. Sürecin başlığını yazdır.
  2. Görüntüyü yeniden boyutlandırma sürecini başlat.
  3. Bir süre bekle (örneğin 1sn).
  4. Döngüye devam et ve tekrar listeyi yazdırıp giriş al.
6. Hiçbiri değilse (cevap geçersizse):
  1. Doğru anahtar girilmediğine dair bir mesaj yazdır.
  2. Bir süre bekle (örneğin 1sn).
  3. Döngüye devam et ve tekrar giriş al.

## Versiyon Takip Tablosu

Versiyon:	Tarih:	Destekçi:	Açıklama:
0.0.0	04.01.2025	Uruz	Belgenin ham versiyonu.
0.1.0	08.01.2025	Uruz	Belgenin ilk versiyonu.