



T.C

**KOCaeli SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ LİSANS
EęİTİM ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİęİ PROGRAMI

ÖDEV KONUSU
LİMAN OTOMASYONU

Hazırlayan
ŞEVVAL ÖZEREN

220501028

ZEYNEP ÇALIK

220501008

DERS SORUMLUSU
PROF. DR. HÜSEYİN TARIK DURU

13.12.2023

İÇİNDEKİLER

1. ÖZET	3
2. GİRİŞ	3
3. YÖNTEM.....	3
3.1 Menü Ekranı Oluşturma	3
3.2 Liman Otomasyonunu Oluşturma	4
4. SONUÇ VE ÖĞRENİLEN DERSLER.....	7
5. KAYNAKÇA.....	7

1. ÖZET

Bu projemizi PyCharm IDE'si üzerinden hazırladık. Ödevin amacı bir liman otomasyonu sürecini yani yüklü tırların limana gelişi ve yüklerin gemilere yüklenerek gemilerin limandan ayrılışını kod ile göstermekti. Biz bunu Tkinter ile arayüz oluşturarak yaptık.

2. GİRİŞ

Projemizde Tkinter ile pencere oluşturduktan sonra pencere içine ilk satıra bir saniye sayacı ekleyerek kod işleyiş sürecini gösterdik daha sonra ise 4 farklı label frame oluşturduk. Bunlar "Tır Listesi", "1. İstif Alanı", "2. İstif Alanı" ve "Gemi Listesi" idi. Tır ve Gemi bilgisi sorgulayabilmek için de Entry Box ve Button ekledik.

3. YÖNTEM

Liman Otomasyonunda yüklerin gelişi, istiflenişi ve gemilere yüklenişi sürecinin görsel olarak ifade edebilmek amacıyla arayüz oluştururken Tkinter kütüphanesini kullandık. Oluşturulan tek bir pencere üzerinde Tır, Gemi ve yük bilgilerinin yer aldığı Label Frame'ler oluşturuldu. Yüklerin gelişi vb. Diğer işlemlerin t zamanda gerçekleştiğini göstermek amacıyla da bir saniye sayacı eklendi.

Tırlarla ilgili işlemlerin olduğu ve bilgilerin tutulduğu bir sınıf hazırlandı, Gemilerin işlemlerinin bulunduğu aynı zamanda bilgilerinin tutulduğu sınıflar hazırlandı. Ayrıca Label Frame'lerin içine Tır, Gemi, Yük bilgilerinin yazdırırken zamana duyarlı işlem yapılması adına saniye sayacıyla birlikte çalışması sağlandı.

3.1 GitHub Bağlantıları

Şevval Özeren

<https://github.com/sevvalozrn>

Zeynep Çalık

<https://github.com/zeynepcallkk>

3.2 Menü Ekranı Oluşturma

Öncelikle "TK()" modülü ile bir pencere oluşturuldu. ".title()" ile bir başlık verildi. Ardından pencerenin tam ekran gözükmesi için ".attributes('-fullscreen', True)" komutu kullanıldı.

Pencereye bir kapatma tuşu tanımlamak için de önce bir fonksiyon ile pencereyi kapatma işi tanımlandı ardından ".bind("<Escape>", kapat)" komutuyla pencere kapatma tuşu "ESC" tuşu olarak ayarlanmış oldu.

Update_saniye() adında bir fonksiyon tanımladı. Bu fonksiyon kendisini 1000 milisaniyede bir çağıran sonsuz döngülü bir özyinelemeli fonksiyondur. Bu fonksiyon saniye sayacı işlevi görmektedir ve her saniye yenilendiğinde sayac kendisini 1 arttırmaktadı ve pencerenin sol üst köşesinde oluşturulan sayaca yazılmaktadır, böylece zamanı takip edebileceğimiz bir sayacımız olmuş olmaktadır. Tır, Gemi ve İstif Alanı verileri de zamana duyarlı çalıştığından hepsi bu fonksiyonun içine eklenmiş durumdadır ve bu fonksiyon ile tutulan sayaç ile çalışmaktadırlar.

```
sayac = tk.Label(pencere, text="0", font=("Times New Roman", 12),  
fg="black")  
sayac.grid(row=0, column=0, padx=5, pady=0)
```

Sayaç için üstte gösterildiği gibi konumu 0. satır ve 0. sütun olmak üzere yani pencerenin sol üste köşesinde yer almaktadır. “tk.Label()” ile sayaç için bir etiket oluştururuz. Oluşturduğumuz pencere içinde olacak, yazı stili ve rengi ayarladığımız şekilde olacak ve 0 dan kendini başlatacak bir saniye sayacı oluşturduk.

Label Framelerin oluşum sürecine örnek olarak da Tır Listesi Label Framei üzerinden gidilerek anlatılacaktır.

```
tir_label_frame = tk.LabelFrame(pencere, text="Tır Listesi")  
tir_label_frame.grid(row=1, column=0, padx=10, pady=10)
```

Tır için başlığı “Tır Listesi” olan bir label frame oluşturulur. Ardından “.grid()” ile 1. satırda 0. sütunda olacak şekilde konumu ayarlanır. Kenar boşlukları ise 10’ar pikseldir.

```
tir_scrollbar = tk.Scrollbar(tir_label_frame)  
tir_scrollbar.pack(side=tk.RIGHT, fill=tk.Y)
```

Ardından bu label frame için sağ tarafında dikey konumda olacak şekilde bir kaydırma çubuğu eklenir. Bu sayede eklenen veriler fazla olsa bile kaydırarak aşağıda kalan veriler de gözükebilecektir.

Daha sonra kaydırma çubuğu ile birlikte bir canvas oluşturulur.

```
tir_canvas = tk.Canvas(tir_label_frame,  
yscrollcommand=tir_scrollbar.set)  
tir_canvas.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)
```

Label Frame’in içinde sola dayalı, kaydırma çubuğu barındıran bir canvas oluşturulur.

```
tir_label_inside = tk.Frame(tir_canvas)  
tir_canvas.create_window((0, 0), window=tir_label_inside,  
anchor=tk.NW)
```

Kanvasın içine eklenen etiketlerin dinamik olarak yazdırılacağı bir pencere oluşturulur.

Gemi ve İstif Alanı kısımları için olan Label Framelerin oluşumu ve işleyişi de aynen böyledir.

3.3 Liman Otomasyonunu Oluşturma

Tır İşlemleri

Tır işlemlerini barındıran bir sınıf oluşturuldu ve içinde birbirinden farklı fonksiyonlar tanımlandı. Öncelikle tır listesinin bulunduğu dosya açıldı ve satır satır okunarak, virgül(“,”) ifadesinden ayrılarak bir liste haline getirildi. Bu ayırım sırasında satır sonundaki elemanda (“\n”) ifadesi kaldığından oluşturulan bir döngü ile o kısım silindi. Ayrıca listenin ilk elemanı olan dosyanın ilk satırı yani başlıkların olduğu kısmı atmak için de “.pop()” ile 0. indexteki elemanı çıkarılır.

Düzenlenmiş tır listesine ulaşabilmek için listeyi döndüren bir fonksiyon yazılır. Ardından plakasına göre tır bilgilerinin sorgulanacağı “tır_sorgulama()” fonksiyonu yazılır.

Bu fonksiyon için bir entry box oluşturuldu. Entry boxa yazılan plaka, butona tıklandığında çağırılan fonksiyon ile istenen plakanın sahibi tırın bildilerini sözlük veri tipinde hazırlayarak ekrana yazdırır.

Ödev No: 2	Tarih 13.12.2023	4/7
------------	------------------	-----

En başta oluşturulan tır listesi alınarak bir döngü oluşturulur bu döngü de istenen plaka bulunur ve bulunduğu liste elemanından tırın diğer bilgilerine erişim sağlanır. Ardından başta verilen boş sözlüğe satır satır anahtarları hazır verilmiş şekilde değerler yazdırılır ve oluşturulan bu sözlük oluşturulmuş Entry Box ve Button'un altında satır satır yazdırılır. Bu yazdırma işleminde ise sözlük anahtar ve değerlerini tutan bi döngü oluşturulur.

```
yazi = f"{anahtar} : {deger}"
```

Yazı değişkenine anahtar ve değeri verilir.

```
label = tk.Label(pencere, text=yazi)
```

Label'a pencere üstünde gözükmeleri için text eklenir.

```
label.grid(row=row_index, column=0, padx=0, pady=0)
```

Label konumu başta verilen row indexe göre ayarlanarak yazdırılır.

```
label_listesi.append(label)
```

Ve son olarak boş olarak oluşturulmuş label listesine eklenir.

“gelis_zamani_t()” fonksiyonunda ise tırlar geliş zamanına göre bir sözlük içinde tutulur. T zamanında gelen tırlar için sözlükteki anahtar t olur karşısındaki değer ise gelmiş tırların liste hali vardır. Bu sözlük kullanılarak tırlar geliş zamanına göre label frame içine yazdırılır.

```
sirali_deger = sorted(deger, key=lambda x: int(((x[1]).split("_"))[-1]))
```

Aynı zamanda gelmiş tırlar plakalarındaki sayılara göre küçükten büyüğe sıralanarak yazdırılmaktadır. Ve bu fonksiyonda sıralı tır listesini döndürmektedir.

Ardından sınıf dışında “tır_gelisi()” anında bir fonksiyon tanımladık. Bu fonksiyon ise üstte oluşturulan fonksiyonun döndürdüğü sözlüğü alarak Tır Listesi label frame'inin içine zamana bağlı olarak gelen tırları satır satır yazdırmaktadır. Bu fonksiyonda da labellar oluşturulur ve text olarak tanımlanan tırlar label listesine eklenir ve liste ekrana satır satır yazdırılır.

```
tir_label_inside.update_idletasks()
```

Metodu kullanılarak da label güncellenir.

```
tir_canvas.update_idletasks()
```

```
tir_canvas.config(scrollregion=tir_canvas.bbox("all"))
```

Son olarak da canvas güncellenir ve canvas içeriği güncellendikçe kaydırma alanının da boyutu ayarlanır.

İstif Alanı İşlemleri

Tırların getirdiği yükleri İstif Alanı başlıklı label frame'in içine yazdıran bir fonksiyon oluşturulur. Liste olarak Tır sınıfından “gelis_zamani_t()” çağırılır. Alınan sözlük ile geliş zamanına göre tırlara erişilir ve geliş zamanı saniyeye eşit olduğunda tırın getirdiği yük istif kısmına yazılır.

Label içine yazdırılırken de yine aynı işlem yapılır. Boş label listesi oluşturulur, text olarak t anında yazılacak yük alınır ve labela eklenir ardından oluşturulan label ise label listesine alınır ve canvasa yazdırılır.

```
istif1_label_inside.update_idletasks()
```

Label boyutu güncellenir.

```
istif1_canvas.update_idletasks()
```

Kanvas boyutu güncellenir.

```
istif1_canvas.config(scrollregion=istif1_canvas.bbox("all"))
```

Kanvas güncellenir ve kanvas içeriği güncellendikçe kaydırma alanının da boyutu ayarlanır.

```
istif1_canvas.yview_moveto(0)
```

Kanvas en üstteki etiketi gösterecek şekilde ayarlanır.

Gemi İşlemleri

Gemi işlemlerini barındıran bir sınıf oluşturuldu ve içinde birbirinden farklı fonksiyonlar tanımlandı. Öncelikle gemi listesinin bulunduğu dosya açıldı ve satır satır okunarak, virgül(“,”) ifadesinden ayrılarak bir liste haline getirildi. Bu ayırım sırasında satır sonundaki elemanda (“\n”) ifadesi kaldığından oluşturulan bir döngü ile o kısım silindi. Ayrıca listenin ilk elemanı olan dosyanın ilk satırı yani başlıkların olduğu kısmı atmak için de “.pop()” ile 0. indexteki elemanı çıkarılır.

Düzenlenmiş gemi listesine ulaşabilmek için listeyi döndüren bir fonksiyon yazılır. Ardından numarasına göre gemi bilgilerinin sorgulanacağı “gemi_sorgulama()” fonksiyonu yazılır.

Bu fonksiyon için bir entry box oluşturuldu. Entry boxa yazılan numara, buttona tıklandığında çağırılan fonksiyon ile istenen numaranın sahibi geminin bilgilerini sözlük veri tipinde hazırlayarak ekrana yazdırır.

En başta oluşturulan gemi listesi alınarak bir döngü oluşturulur bu döngü de istenen numara bulunur ve bulunduğu liste elemanından geminin diğer bilgilerine erişim sağlanır. Ardından başta verilen boş sözlüğe satır satır anahtarları hazır verilmiş şekilde değerler yazdırılır ve oluşturulan bu sözlük oluşturulmuş Entry Box ve Button’un altında satır satır yazdırılır. Bu yazdırma işleminde ise sözlük anahtar ve değerlerini tutan bir döngü oluşturulur.

```
yazi = f"{anahtar} : {deger}"
```

Yazı değişkenine anahtar ve değeri verilir.

```
label = tk.Label(pencere, text=yazi)
```

Label’a pencere üstünde gözükmesi için text eklenir.

```
label.grid(row=row_index, column=0, padx=0, pady=0)
```

Label konumu başta verilen row indexe göre ayarlanarak yazdırılır.

```
label_listesi.append(label)
```

Ve son olarak boş olarak oluşturulmuş label listesine eklenir.

“gelis_zamanı_g()” fonksiyonunda ise gemiler geliş zamanına göre bir sözlük içinde tutulur. T zamanında gelen gemiler için sözlükteki anahtar t olur karşısındaki değer ise gelmiş gemilerin liste hali vardır. Bu sözlük kullanılarak gemiler geliş zamanına göre label frame içine yazdırılır. Tırlar zaten numaraların göre büyükten küçüğe olacak şekilde geldiğinden tekrar bir düzenleme ile sıralama yapılma gereksinimi duyulmamıştır.

Ardından sınıf dışında “gemi_gelisi()” anında bir fonksiyon tanımladık. Bu fonksiyon ise üstte oluşturulan fonksiyonun döndürdüğü sözlüğü alarak Limana Gelen Gemi Listesi label frame’inin içine zamana bağlı olarak gelen gemileri satır satır yazdırmaktadır. Bu fonksiyonda da labellar oluşturulur ve text olarak tanımlanan gemiler label listesine eklenir ve liste ekrana

satır satır yazdırılır.

```
gemi_label_inside.update_idletasks()
```

Metodu kullanılarak da label güncellenir.

```
gemi_canvas.update_idletasks()  
gemi_canvas.config(scrollregion=gemi_canvas.bbox("all"))
```

Son olarak da canvas güncellenir ve kanvas içeriği güncellendikçe kaydırma alanının da boyutu ayarlanır.

Ayrı olarak bir de Yüklendi Gemi Listesi başlıklı bir label frame'imiz var. Burda ise tırlar ile gelip, istiflenen yüklerin daha sonra gideceği yere göre gemilere yüklenişi ve yüklendi bu gemilerin bilgileri ile ne kadar ton yüklendiği ve kaçınıcı saniyede yüklemenin bittiğini yazdıran bir fonksiyonumuz vardır. Bunun için öncelikle "gemi_yuklenisi()" adlı bir fonksiyon oluşturduk.

4. SONUÇ VE ÖĞRENİLEN DERSLER

Bir Tkinter penceresi üzerinde nasıl sayaç oluşturabileceği ve bu sırada zamana dayalı işlemleri gerçekleştiren fonksiyonların nasıl oluşturulabileceğini öğrendik. Label Frame'ler ile işlem yapmayı ve Entry Box ile Button kullanımını deneyimlemiş olduk.

5. KAYNAKÇA

- https://youtube.com/playlist?list=PLSmHiN0iazq_qX_6Tmecj4tTOefqh2-m2&si=EABqpoKPoyJ0IAMC
- https://youtube.com/playlist?list=PLDWd7pi6Qk7rpa9NiegdXAmkM9NCFs_9i&si=nSe6iK3ACs_2COTj
- <https://www.sadikturan.com/python-objeleri-ve-veri-yapilari/python-liste-metotlari/1376>
- <https://python-istihza.yazbel.com/sozlukler.html>

Ödev No: 2	Tarih 13.12.2023	7/7
------------	------------------	-----