1

TC
İZMİR BAKIRÇAY ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PROGRAMLAMA DERSİ
BAHAR DÖNEMİ

FİNAL PROJESİ ÖDEV RAPORU

HAZIRLAYAN
AD-SOYAD: SEVDE IŞIL BODUR
OKUL NUMARASI: 220601026

İÇİNDEKİLER

KOD AÇIKLAMASI

Bu proje, bir sağlık kuruluşunda çalışan personel ve hastaların yönetimi için geliştirilmiş bir Python uygulamasıdır. Uygulama, Personel, Doktor, Hemşire ve Hasta sınıflarını kullanarak çeşitli işlemler gerçekleştirir. Projenin temel amacı, sağlık personelinin ve hastaların bilgilerini düzenli bir şekilde yönetmek ve analiz etmektir.

Personel sınıfı; Personelin adı, soyadı, departmanı ve maaşı gibi bilgileri tutar ve __str__ metodu ile personel bilgilerini yazdırır.

Doktor sınıfı; Personel sınıfından türetilmiştir ve uzmanlık alanı, deneyim yılı, çalıştıkları hastane gibi bilgileri tutar. Ayrıca maaş artırımı için maas_arttir metoduna sahiptir.

Hemşire sınıfı; Personel sınıfından türetilmiştir ve çalışma saatleri, sertifikaları, çalıştıkları hastane gibi bilgileri tutar ve maaş artırımı için maas_arttir metoduna sahiptir.

Hasta sınıfı; Hastaların hasta numarası, adı, soyadı, doğum tarihi, hastalığı ve tedavi bilgilerini tutar ve tedavi süresini hesaplamak için tedavi_suresi_hesapla metoduna sahiptir.

Uygulamanın işleyişi; İlk olarak, Personel, Doktor, Hemşire ve Hasta sınıflarından nesneler oluşturulur ve bu nesnelerin bilgileri ekrana yazdırılır. Daha sonra, doktor ve hemşirelerin maaşları belirli bir oranla artırılır ve güncellenmiş maaşlar yazdırılır. Tedavi tarihleri verilerek hastaların tedavi süreleri hesaplanır ve bu bilgiler hasta nesnelerine kaydedilir.

Tüm nesnelerden elde edilen veriler kullanılarak bir DataFrame oluşturulur. DataFrame'de eksik değerler kontrol edilir ve NaN değerler 0 ile doldurulur.

Oluşturulan DataFrame'den Doktorlar uzmanlık alanlarına göre gruplandırılır ve 5 yıldan fazla deneyime sahip doktorların sayısı hesaplanır. Hasta bilgileri alfabetik olarak sıralanır ve maaşı 7000 TL üzerinde olan personeller listelenir. Ayrıca, doğum tarihi 1990 ve sonrası olan hastalar filtrelenir.

Son olarak, ad, soyad, departman, maaş, uzmanlık, deneyim yılı, hastalık ve tedavi bilgilerini içeren yeni bir DataFrame oluşturulur ve yazdırılır.

PYTHON KODU

```
import pandas as pd
import datetime
from Personel import Personel
from Doktor import Doktor
from Hemsire import Hemsire
from Hasta import Hasta
try:
  # Personel nesnelerini oluşturma ve yazdırma
  print("Personel Bilgileri:")
  personel1 = Personel("1001", "Ege", "Yılmaz", "Sekreter", 3000)
  personel2 = Personel("1002", "Mira", "Çelik", "İnsan Kaynakları",3500)
  print(personel1)
  print(personel2)
  print("-----------------------------------")
  # Doktor nesnelerini oluşturma ve yazdırma
  print("Doktor Bilgileri:")
  doktor1 = Doktor("1003", "Hakan", "Kaya", "Doktor", "Dahiliye", 8, 7500.00, "Şehir
Hastanesi" )
  doktor2 = Doktor("1004", "Rasim", "Bodur", "Doktor", "Ortopedi", 5, 6000.00, "Eğitim
Hastanesi")
  doktor3 = Doktor("1005", "Ahmet", "Yıldız", "Doktor", "Nöroloji", 10, 9000.00, "Özel
Hastane")
  print(doktor1)
  print(doktor2)
  print(doktor3)
  print("-----------------------------------")
  # Doktor maaşlarını %10 artırma
  doktor1.maas_arttir(10) # %10 arttırmak için oran 10 olarak geçilir
  doktor2.maas_arttir(10)
  doktor3.maas_arttir(10)
  # Güncellenmiş doktor maaşlarını yazdırma
  print("Güncellenmiş Doktor Maaşları:")
  print(doktor1)
  print(doktor2)
  print(doktor3)
  print("-----------------------------------")
```

```
 # Hemşire nesnelerini oluşturma ve yazdırma
  print("Hemşire Bilgileri:")
  hemsire1 = Hemsire("1006", "Rüzgar", "Gül", "Pediatri", 4500.00, 36, "A Sertifikası",
"Eğitim Hastanesi")
  hemsire2 = Hemsire("1007", "Mehmet", "Can", "Diyaliz", 4000.00, 28, "B Sertifikası",
"Özel Hastane")
  hemsire3 = Hemsire("1008", "Hatice", "Örs", "Onkoloji", 5000.00, 40, "C Sertifikası",
"Şehir Hastanesi")
  print(hemsire1)
  print(hemsire2)
  print(hemsire3)
  print("-----------------------------------")
  # Hemşire maaşlarını %10 artırma
  hemsire1.maas_arttir(10) # %10 arttırmak için oran 10 olarak geçilir
  hemsire2.maas_arttir(10)
  hemsire3.maas_arttir(10)
  # Güncellenmiş hemşire maaşlarını yazdırma
  print("Güncellenmiş Hemşire Maaşları:")
  print(hemsire1)
  print(hemsire2)
  print(hemsire3)
  print("-----------------------------------")
  # Hasta nesnelerini oluşturma
  hasta1 = Hasta("H-0001", "Ümmü", "Özdemir", "1979-04-20", "Grip", "Antibiyotik", 0)
  hasta2 = Hasta("H-0002", "Ayşe", "Demir", "1994-11-15", "Migren", "Ağrı kesici", 0)
  hasta3 = Hasta("H-0003", "Fatma", "Kara", "1982-03-05", "Diyabet", "İnsülin", 0)
  # Başlangıç ve bitiş tarihlerini içeren bir liste oluşturma
  tedavi_tarihleri = [
    ("2024-05-01", "2024-05-08"),
    ("2024-05-02", "2024-05-30"),
    ("2024-05-03", "2024-06-14") ]
```

```
 # Tedavi Süreleri:
 print("Tedavi Süreleri:")
 for i, (baslangic, bitis) in enumerate(tedavi_tarihleri):
 if i == 0:
 hasta = hasta1
elif i == 1:
 hasta = hasta2
 else:
 hasta = hasta3
 tedavi_suresi = hasta.tedavi_suresi_hesapla(baslangic, bitis)
 hasta.set_tedavi_suresi(tedavi_suresi) # Tedavi süresini hasta nesnesine kaydet
 print(f"Başlangıç Tarihi: {baslangic}, Bitiş Tarihi: {bitis}")
 print(f"Hasta {i+1} Tedavi Süresi: {tedavi_suresi} gün")
 print("----------------------")
 # Hasta Bilgilerini Yazdırma
 print("Hasta Bilgileri:")
 print(hasta1)
 print(hasta2)
 print(hasta3)
 print("----------------------")
 # Tüm nesnelerden DataFrame oluşturma
  data = {
    "nesne": ["personel1", "personel2", "doktor1", "doktor2", "doktor3", "hemsire1",
"hemsire2", "hemsire3", "hasta1", "hasta2", "hasta3"],
     "personel_no": [personel1.get_personel_no(), personel2.get_personel_no(),
doktor1.get_personel_no(), doktor2.get_personel_no(), doktor3.get_personel_no(),
hemsire1.get_personel_no(), hemsire2.get_personel_no(), hemsire3.get_personel_no(), 0, 0,
0],
    "ad": [personel1.get_ad(), personel2.get_ad(), doktor1.get_ad(), doktor2.get_ad(),
doktor3.get_ad(), hemsire1.get_ad(), hemsire2.get_ad(), hemsire3.get_ad(),
hasta1.get_ad(), hasta2.get_ad(), hasta3.get_ad()],
    "soyad": [personel1.get_soyad(), personel2.get_soyad(), doktor1.get_soyad(),
doktor2.get_soyad(), doktor3.get_soyad(), hemsire1.get_soyad(), hemsire2.get_soyad(),
hemsire3.get_soyad(), hasta1.get_soyad(), hasta2.get_soyad(), hasta3.get_soyad()],
    "departman": [personel1.get_departman(), personel2.get_departman() ,
doktor1.get_departman(), doktor2.get_departman(), doktor3.get_departman(),
hemsire1.get_departman(), hemsire2.get_departman(), hemsire3.get_departman(), 0, 0, 0],
```

```
  "maas": [personel1.get_maas(), personel2.get_maas(), doktor1.get_maas(),
doktor2.get_maas(), doktor3.get_maas(), hemsire1.get_maas(), hemsire2.get_maas(),
hemsire3.get_maas(), 0, 0, 0],
    "uzmanlik": [0, 0, doktor1.get_uzmanlik(), doktor2.get_uzmanlik(),
doktor3.get_uzmanlik(), 0, 0, 0, 0, 0, 0],
    "deneyim_yili": [0, 0, doktor1.get_deneyim_yili(), doktor2.get_deneyim_yili(),
doktor3.get_deneyim_yili(), 0, 0, 0, 0, 0, 0],
    "hastane": [0, 0, doktor1.get_hastane(), doktor2.get_hastane(), doktor3.get_hastane(),
hemsire1.get_hastane(), hemsire2.get_hastane(), hemsire3.get_hastane(), 0, 0, 0],
    "calisma_saati": [0, 0, 0, 0, 0, hemsire1.get_calisma_saati(),
hemsire2.get_calisma_saati(), hemsire3.get_calisma_saati(), 0, 0, 0],
    "sertifika": [0, 0, 0, 0, 0, hemsire1.get_sertifika(), hemsire2.get_sertifika(),
hemsire3.get_sertifika(), 0, 0, 0],
    "hasta_no": [0, 0, 0, 0, 0, 0, 0, 0, hasta1.get_hasta_no(), hasta2.get_hasta_no(),
hasta3.get_hasta_no()],
    "dogum_tarihi": [0, 0, 0, 0, 0, 0, 0, 0, hasta1.get_dogum_tarihi(),
hasta2.get_dogum_tarihi(), hasta3.get_dogum_tarihi()],
    "hastalik": [0, 0, 0, 0, 0, 0, 0, 0, hasta1.get_hastalik(), hasta2.get_hastalik(),
hasta3.get_hastalik()],
    "tedavi": [0, 0, 0, 0, 0, 0, 0, 0, hasta1.get_tedavi(), hasta2.get_tedavi(),
hasta3.get_tedavi()],
    "tedavi_suresi": [0, 0, 0, 0, 0, 0, 0, 0, hasta1.get_tedavi_suresi(),
hasta2.get_tedavi_suresi(), hasta3.get_tedavi_suresi()]
}
 df = pd.DataFrame(data)
  # Eksik değerleri içeren satırları silme
 df.dropna(inplace=True)
 # DataFrame'de NaN değerlerini kontrol etme
 nan_values = df.isna().sum().sum()
 if nan_values > 0:
   print("DataFrame'de NaN değerleri var.")
 else:
   print("DataFrame'de NaN değerleri yok.")
 print("DataFrame:")
 print(df)
 print("----------------------------------")
 # Boş olan değişken değerlerine 0 atama
 df.fillna(0, inplace=True)
```

```
  # Doktorları uzmanlık alanlarına göre gruplandırarak toplam sayısını hesaplama ve
yazdırma
  doktor_uzmanlik_gruplari = df[df['uzmanlik'] != 0].groupby('uzmanlik').size()
  print("Doktor Uzmanlık Grupları ve Sayıları:")
  print(doktor_uzmanlik_gruplari)
  print("-----------------------------------")
  # 5 yıldan fazla deneyime sahip doktorların toplam sayısını bulma
  deneyimli_doktorlar = df[(df['deneyim_yili'] > 5) & (df['deneyim_yili'] != 0)].shape[0]
  print("5 yıldan fazla deneyime sahip doktorların sayısı:", deneyimli_doktorlar)
  print("-----------------------------------")
  # Hasta adına göre DataFrame'i alfabetik olarak sıralama ve yazdırma
  df_hasta_sirali = df[df['hasta_no'] != 0].sort_values(by='ad')
  print("Alfabetik Sıralanmış Hasta Bilgileri:")
  print(df_hasta_sirali)
  print("-----------------------------------")
  # Maaşı 7000 TL üzerinde olan personelleri bulup ve yazdırın
  maas_7000_uzeri = df[df['maas'] > 7000]
  print("Maaşı 7000 TL üzerinde olan personeller:")
  print(maas_7000_uzeri)
  print("-----------------------------------")
  # Doğum tarihi 1990 ve sonrası olan hastaları gösterme
  df['dogum_tarihi'] = pd.to_datetime(df['dogum_tarihi'], errors='coerce') # Hatalı
tarihleri NaT yapar
  df_dogum_tarihi_1990_sonrasi = df[(df['dogum_tarihi'] >= '1990-01-01') &
(df['dogum_tarihi'].notna())]
  print("Doğum Tarihi 1990 ve Sonrası Olan Hastalar:")
  print(df_dogum_tarihi_1990_sonrasi)
  print("-----------------------------------")
  # Ad, soyad, departman, maas, uzmanlik, deneyim_yili, hastalik, tedavi bilgilerini
içeren yeni bir DataFrame oluşturma
  yeni_df = df[["ad", "soyad", "departman", "maas", "uzmanlik", "deneyim_yili",
"hastalik", "tedavi", "tedavi_suresi"]]
  print("Yeni DataFrame:")
  print(yeni_df)
except Exception as e:
  print(str(e))
```

KOD ÇIKTILARI

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\HP\Desktop\DENEME> & C:/Users/HP/AppData/Local/Programs/Python/Python312/python.exe c:/Users/HP/Desktop/DENEME/Main.py
Personel Bilgileri:
Personel No: 1001, Ad: Ege, Soyad: Yılmaz, Departman: Sekreter, Maaş: 3000
Personel No: 1002, Ad: Mira, Soyad: Çelik, Departman: İnsan Kaynakları, Maaş: 3500
-----------------------------------
Doktor Bilgileri:
Personel No: 1003, Ad: Hakan, Soyad: Kaya,  Departman: Doktor, Uzmanlık: Dahiliye, Deneyim Yılı: 8, Maaş: 7500.0, Hastane: Şehir Hastanesi
Personel No: 1004, Ad: Rasim, Soyad: Bodur,  Departman: Doktor, Uzmanlık: Ortopedi, Deneyim Yılı: 5, Maaş: 6000.0, Hastane: Eğitim Hastanesi
Personel No: 1005, Ad: Ahmet, Soyad: Yıldız,  Departman: Doktor, Uzmanlık: Nöroloji, Deneyim Yılı: 10, Maaş: 9000.0, Hastane: Özel Hastane
-----------------------------------
Güncellenmiş Doktor Maaşları:
Personel No: 1003, Ad: Hakan, Soyad: Kaya,  Departman: Doktor, Uzmanlık: Dahiliye, Deneyim Yılı: 8, Maaş: 8250.0, Hastane: Şehir Hastanesi
Personel No: 1004, Ad: Rasim, Soyad: Bodur,  Departman: Doktor, Uzmanlık: Ortopedi, Deneyim Yılı: 5, Maaş: 6600.000000000001, Hastane: Eğitim Hastanesi
Personel No: 1005, Ad: Ahmet, Soyad: Yıldız,  Departman: Doktor, Uzmanlık: Nöroloji, Deneyim Yılı: 10, Maaş: 9900.0, Hastane: Özel Hastane
-----------------------------------
Hemşire Bilgileri:
Personel No: 1006, Ad: Rüzgar, Soyad: Gül, Departman: Pediatri, Maaş: 4500.0, Çalışma Saati: 36, Sertifika: A Sertifikası, Hastane: Eğitim Hastanesi
Personel No: 1007, Ad: Mehmet, Soyad: Can, Departman: Diyaliz, Maaş: 4000.0, Çalışma Saati: 28, Sertifika: B Sertifikası, Hastane: Özel Hastane
Personel No: 1008, Ad: Hatice, Soyad: Örs, Departman: Onkoloji, Maaş: 5000.0, Çalışma Saati: 40, Sertifika: C Sertifikası, Hastane: Şehir Hastanesi
-----------------------------------
Güncellenmiş Hemşire Maaşları:
Personel No: 1006, Ad: Rüzgar, Soyad: Gül, Departman: Pediatri, Maaş: 4950.0, Çalışma Saati: 36, Sertifika: A Sertifikası, Hastane: Eğitim Hastanesi
Personel No: 1007, Ad: Mehmet, Soyad: Can, Departman: Diyaliz, Maaş: 4400.0, Çalışma Saati: 28, Sertifika: B Sertifikası, Hastane: Özel Hastane
Personel No: 1008, Ad: Hatice, Soyad: Örs, Departman: Onkoloji, Maaş: 5500.0, Çalışma Saati: 40, Sertifika: C Sertifikası, Hastane: Şehir Hastanesi
-----------------------------------
```

Personel, doktor , hemşire bilgileri yazdırıldı. Doktor ve hemşirelerin maaşları güncellendi.

```
Tedavi Süreleri:
Başlangıç Tarihi: 2024-05-01, Bitiş Tarihi: 2024-05-08
Hasta 1 Tedavi Süresi: 7 gün
-----------------------
Başlangıç Tarihi: 2024-05-02, Bitiş Tarihi: 2024-05-30
Hasta 2 Tedavi Süresi: 28 gün
-----------------------
Başlangıç Tarihi: 2024-05-03, Bitiş Tarihi: 2024-06-14
Hasta 3 Tedavi Süresi: 42 gün
-----------------------
Hasta Bilgileri:
Hasta No: H-0001, Ad: Ümmü, Soyad: Özdemir, Doğum Tarihi: 1979-04-20, Hastalık: Grip, Tedavi: Antibiyotik, Tedavi Süresi: 7
Hasta No: H-0002, Ad: Ayşe, Soyad: Demir, Doğum Tarihi: 1994-11-15, Hastalık: Migren, Tedavi: Ağrı kesici, Tedavi Süresi: 28
Hasta No: H-0003, Ad: Fatma, Soyad: Kara, Doğum Tarihi: 1982-03-05, Hastalık: Diyabet, Tedavi: İnsülin, Tedavi Süresi: 42
-----------------------
```

Hasta bilgileri ve hesaplanan tedavi günleri yazdırıldı.

```
DataFrame:
      nesne personel_no     ad    soyad        departman    maas uzmanlik deneyim_yili          hastane calisma_saati      sertifika hasta_no dogum_tarihi hastalik       tedavi tedavi_suresi
0  personel1        1001    Ege   Yılmaz          Sekreter  3000.0        0           0                0            0              0        0            0         0            0             0
1  personel2        1002   Mira    Çelik  İnsan Kaynakları  3500.0        0           0                0            0              0        0            0         0            0             0
2    doktor1        1003  Hakan     Kaya            Doktor  8250.0 Dahiliye           8   Şehir Hastanesi            0              0        0            0         0            0             0
3    doktor2        1004  Rasim    Bodur            Doktor  6600.0 Ortopedi           5  Eğitim Hastanesi            0              0        0            0         0            0             0
4    doktor3        1005  Ahmet   Yıldız            Doktor  9900.0 Nöroloji          10      Özel Hastane            0              0        0            0         0            0             0
5   hemsire1        1006 Rüzgar      Gül          Pediatri  4950.0        0           0  Eğitim Hastanesi           36 A Sertifikası1        0            0         0            0             0
6   hemsire2        1007 Mehmet      Can          Diyaliz  4400.0        0           0      Özel Hastane           28 B Sertifikası1        0            0         0            0             0
7   hemsire3        1008 Hatice      Örs          Onkoloji  5500.0        0           0   Şehir Hastanesi           40 C Sertifikası1        0            0         0            0             0
8     hasta1           0  Ümmü Özdemir                 0     0.0        0           0                0            0              0   H-0001   1979-04-20      Grip  Antibiyotik             7
9     hasta2           0  Ayşe    Demir                 0     0.0        0           0                0            0              0   H-0002   1994-11-15    Migren  Ağrı kesici            28
10    hasta3           0 Fatma     Kara                 0     0.0        0           0                0            0              0   H-0003   1982-03-05   Diyabet      İnsülin            42
-----------------------------------
```

Personel, doktor, hemşire, hasta nesnelerinin özelliklerinden DataFrame oluşturuldu.

```
----------------------------------
Doktor Uzmanlık Grupları ve Sayıları:
uzmanlik
Dahiliye    1
Nöroloji    1
Ortopedi    1
dtype: int64
----------------------------------
5 yıldan fazla deneyime sahip doktorların sayısı: 2
----------------------------------
Alfabetik Sıralanmış Hasta Bilgileri:
     nesne personel_no     ad    soyad departman  maas uzmanlik  deneyim_yili hastane  calisma_saati sertifika hasta_no dogum_tarihi hastalik       tedavi  tedavi_suresi
9   hasta2           0   Ayşe    Demir         0   0.0        0             0       0              0         0  H-0002   1994-11-15   Migren  Ağrı kesici             28
10  hasta3           0  Fatma     Kara         0   0.0        0             0       0              0         0  H-0003   1982-03-05  Diyabet      İnsülin             42
8   hasta1           0   Ümmü  Özdemir         0   0.0        0             0       0              0         0  H-0001   1979-04-20     Grip  Antibiyotik              7
----------------------------------
Maaşı 7000 TL üzerinde olan personeller:
     nesne personel_no     ad  soyad departman    maas  uzmanlik  deneyim_yili         hastane  calisma_saati sertifika hasta_no dogum_tarihi hastalik tedavi  tedavi_suresi
2  doktor1        1003  Hakan   Kaya    Doktor  8250.0  Dahiliye             8  Şehir Hastanesi              0         0        0            0       0     0              0
4  doktor3        1005  Ahmet  Yıldız    Doktor  9900.0  Nöroloji            10     Özel Hastane              0         0        0            0       0     0              0
----------------------------------
Doğum Tarihi 1990 ve Sonrası Olan Hastalar:
    nesne personel_no    ad  soyad departman  maas uzmanlik  deneyim_yili hastane  calisma_saati sertifika hasta_no dogum_tarihi hastalik       tedavi  tedavi_suresi
9  hasta2           0  Ayşe  Demir         0   0.0        0             0       0              0         0  H-0002   1994-11-15   Migren  Ağrı kesici             28
----------------------------------
```

Oluşturulan DataFrame' den yararlanarak; Doktorları uzmanlık alanlarına göre gruplandırarak toplam sayısını, 5 yıldan fazla deneyime sahip doktorların toplam sayısını, hasta adına göre DataFrame'i alfabetik olarak sıralanmasını, maaşı 7000 TL üzerinde olan personellerin bulunması, doğum tarihi 1990 ve sonrası olan hastaları yazdırıldı.

```
----------------------------------
Yeni DataFrame:
       ad    soyad         departman    maas  uzmanlik  deneyim_yili hastalik       tedavi  tedavi_suresi
0     Ege   Yılmaz          Sekreter  3000.0         0             0        0            0              0
1    Mira    Çelik  İnsan Kaynakları  3500.0         0             0        0            0              0
2   Hakan     Kaya            Doktor  8250.0  Dahiliye             8        0            0              0
3   Rasim    Bodur            Doktor  6600.0  Ortopedi             5        0            0              0
4   Ahmet   Yıldız            Doktor  9900.0  Nöroloji            10        0            0              0
5  Rüzgar      Gül          Pediatri  4950.0         0             0        0            0              0
6  Mehmet      Can           Diyaliz  4400.0         0             0        0            0              0
7  Hatice      Örs          Onkoloji  5500.0         0             0        0            0              0
8    Ümmü  Özdemir                 0     0.0         0             0     Grip  Antibiyotik              7
9    Ayşe    Demir                 0     0.0         0             0   Migren  Ağrı kesici             28
10  Fatma     Kara                 0     0.0         0             0  Diyabet      İnsülin             42
PS C:\Users\HP\Desktop\DENEME>
```

Var olan DataFrame'den ad, soyad, departman, maas, uzmanlik, deneyim_yili, hastalik, tedavi bilgilerini içeren yeni bir DataFrame elde edildi.

SÖZDE KODLAR

PERSONEL.PY:

```
SINIF OLUŞTUR(Personel)

FONKSİYON(__init__(self, ad, soyad, departman, maas))
  ATA(self.__ad = ad, self.__soyad = soyad, self.__departman = departman, self.__maas = maas)

FONKSİYON(get_ad(self))
  DÖNDÜR(self.__ad)

FONKSİYON(set_ad(self, ad))
  ATA(self.__ad = ad)

FONKSİYON(get_soyad(self))
  DÖNDÜR(self.__soyad)

FONKSİYON(set_soyad(self, soyad))
  ATA(self.__soyad = soyad)

FONKSİYON(get_departman(self))
  DÖNDÜR(self.__departman)

FONKSİYON(set_departman(self, departman))
  ATA(self.__departman = departman)

FONKSİYON(get_maas(self))
  DÖNDÜR(self.__maas)

FONKSİYON(set_maas(self, maas))
  ATA(self.__maas = maas)

FONKSİYON(__str__(self))
  DÖNDÜR(f"Ad: {self.__ad}, Soyad: {self.__soyad}, Departman: {self.__departman}, Maaş: {self.__maas}")
```

DOKTOR.PY:

```
AKTAR(Personel import Personel)

SINIF OLUŞTUR(Doktor(Personel))
FONKSİYON(__init__(self, ad, soyad, departman, maas, uzmanlik, deneyim_yili, hastane))
  ÜST SINIF METOT ÇAĞIRMA(Personel.__init__(ad, soyad, departman, maas))
  ATA(self.__uzmanlik = uzmanlik, self.__deneyim_yili = deneyim_yili, self.__hastane = hastane)

FONKSİYON(get_uzmanlik(self))
  DÖNDÜR(self.__uzmanlik)

FONKSİYON(set_uzmanlik(self, uzmanlik))
  ATA(self.__uzmanlik = uzmanlik)

FONKSİYON(get_deneyim_yili(self))
  DÖNDÜR(self.__deneyim_yili)

FONKSİYON(set_deneyim_yili(self, deneyim_yili))
  ATA(self.__deneyim_yili = deneyim_yili)

FONKSİYON(get_hastane(self))
  DÖNDÜR(self.__hastane)

FONKSİYON(set_hastane(self, hastane))
  ATA(self.__hastane = hastane)

FONKSİYON(maas_arttir(self, oran))
  ATA(self.__maas = self.__maas * (1 + oran / 100))

FONKSİYON(__str__(self))
  DÖNDÜR(f"Ad: {self.get_ad()}, Soyad: {self.get_soyad()}, Uzmanlık: {self.__uzmanlik}, Deneyim Yılı: {self.__deneyim_yili},  Hastane: {self.__hastane}, Maaş: {self.get_maas()}")
```

HEMSİRE.PY:

```
AKTAR(Personel import Personel)

SINIF OLUŞTUR(Hemsire(Personel))
FONKSİYON(__init__(self, ad, soyad, departman, maas, vardiya, calisma_yili, hastane))
  ÜST SINIF METOT ÇAĞIRMA(Personel.__init__(ad, soyad, departman, maas))
  ATA(self.__vardiya = vardiya, self.__calisma_yili = calisma_yili, self.__hastane = hastane)

FONKSİYON(get_vardiya(self))
  DÖNDÜR(self.__vardiya)

FONKSİYON(set_vardiya(self, vardiya))
  ATA(self.__vardiya = vardiya)

FONKSİYON(get_calisma_yili(self))
  DÖNDÜR(self.__calisma_yili)

FONKSİYON(set_calisma_yili(self, calisma_yili))
  ATA(self.__calisma_yili = calisma_yili)

FONKSİYON(get_hastane(self))
  DÖNDÜR(self.__hastane)

FONKSİYON(set_hastane(self, hastane))
  ATA(self.__hastane = hastane)

FONKSİYON(maas_arttir(self, oran))
  ATA(self.__maas = self.__maas * (1 + oran / 100))

FONKSİYON(__str__(self))
  DÖNDÜR(f"Ad: {self.get_ad()}, Soyad: {self.get_soyad()}, Vardiya: {self.__vardiya},
Çalışma Yılı: {self.__calisma_yili}, Hastane: {self.__hastane}, Maaş: {self.get_maas()}")
```

HASTA.PY:

```
SINIF OLUŞTUR(Hasta)

FONKSİYON(__init__(self, ad, soyad, yas, cinsiyet, hastalik, hastane))
  ATA(self.__ad = ad, self.__soyad = soyad, self.__yas = yas, self.__cinsiyet = cinsiyet,
self.__hastalik = hastalik, self.__hastane = hastane)

FONKSİYON(get_ad(self))
  DÖNDÜR(self.__ad)

FONKSİYON(set_ad(self, ad))
  ATA(self.__ad = ad)

FONKSİYON(get_soyad(self))
  DÖNDÜR(self.__soyad)

FONKSİYON(set_soyad(self, soyad))
  ATA(self.__soyad = soyad)

FONKSİYON(get_yas(self))
  DÖNDÜR(self.__yas)

FONKSİYON(set_yas(self, yas))
  ATA(self.__yas = yas)

FONKSİYON(get_cinsiyet(self))
  DÖNDÜR(self.__cinsiyet)

FONKSİYON(set_cinsiyet(self, cinsiyet))
  ATA(self.__cinsiyet = cinsiyet)
```

HASTA.PY (SÖZDE KODUN DEVAMI):

```
 FONKSİYON(get_hastalik(self))
   DÖNDÜR(self.__hastalik)

  FONKSİYON(set_hastalik(self, hastalik))
    ATA(self.__hastalik = hastalik)

  FONKSİYON(get_hastane(self))
    DÖNDÜR(self.__hastane)

  FONKSİYON(set_hastane(self, hastane))
    ATA(self.__hastane = hastane)

  FONKSİYON(__str__(self))
    DÖNDÜR(f"Ad: {self.__ad}, Soyad: {self.__soyad}, Yaş: {self.__yas}, Cinsiyet:
{self.__cinsiyet}, Hastalık: {self.__hastalik}, Hastane: {self.__hastane}")
```

MAİN.PY

1.BAŞLA
2.İÇE AKTAR (import pandas as pd, from Personel import Personel, from Doktor import Doktor, from Hemsire import Hemsire, from Hasta import Hasta)
3.DENE
    3.1. YAZDIR("Personel Bilgileri:")
    3.2. NESNE OLUŞTUR(personel1 = Personel("Ali", "Veli", "Yönetim", 5000), personel2 = Personel("Ayşe", "Fatma", "Muhasebe", 4000))
    3.3. YAZDIR(personel1)
    3.4. YAZDIR(personel2)
    3.5. YAZDIR("Doktor Bilgileri:")
    3.6. NESNE OLUŞTUR(doktor1 = Doktor("Mehmet", "Kara", "Cerrahi", 15000, "Kalp", 10, "Acıbadem"), doktor2 = Doktor("Hakan", "Yılmaz", "Ortopedi", 12000, "Kemik", 7, "Medicana"))
    3.7. YAZDIR(doktor1)
    3.8. YAZDIR(doktor2)
    3.9. YAZDIR("Hemşire Bilgileri:")
    3.10. NESNE OLUŞTUR(hemsire1 = Hemsire("Selin", "Öztürk", "Acil", 6000, "Gece", 5, "Şişli Etfal"), hemsire2 = Hemsire("Melis", "Kaya", "Pediatri", 5500, "Gündüz", 3, "Florence Nightingale"))
    3.11. YAZDIR(hemsire1)
    3.12. YAZDIR(hemsire2)
    3.13. YAZDIR("Hasta Bilgileri:")
    3.14. NESNE OLUŞTUR(hasta1 = Hasta("Cem", "Demir", 45, "Erkek", "Kalp Hastalığı", "Acıbadem"), hasta2 = Hasta("Zeynep", "Yıldız", 30, "Kadın", "Astım", "Medicana"))
    3.15. YAZDIR(hasta1)
    3.16. YAZDIR(hasta2)
4. ATA(yeni_df = df[["personel_no", "ad", "soyad", "departman", "maas", "uzmanlik", "deneyim_yili", "hastane", "calisma_saati", "sertifika", "hasta_no", "dogum_tarihi", "hastalik", "tedavi", "tedavi_suresi"]])
5. SÖZLÜK OLUŞTUR(data)
6. ATA(df = pd.DataFrame(data))

MAİN.PY (SÖZDE KODUN DEVAMI):

```
8.  df.dropna(inplace=True)
9.  nan_values = df.isna().sum().sum()
10.  IF nan_values > 0 THEN
11.      YAZDIR("DataFrame'de NaN değerleri var.")
12.  ELSE
13.      YAZDIR("DataFrame'de NaN değerleri yok.")
14.  END IF
15. YAZDIR("DataFrame:")
16. YAZDIR(df)
17. ATA(df.fillna(0, inplace=True))
18. ATA(doktor_uzmanlik_gruplari = df[df['uzmanlik'] != 0].groupby('uzmanlik').size())
19. YAZDIR("Doktor Uzmanlık Grupları ve Sayıları:")
20. YAZDIR(doktor_uzmanlik_gruplari)
21. ATA(deneyimli_doktorlar = df[(df['deneyim_yili'] > 5) & (df['deneyim_yili'] != 0)].shape[0])
22. YAZDIR("5 yıldan fazla deneyime sahip doktorların sayısı:", deneyimli_doktorlar)
23. ATA(df_hasta_sirali = df[df['hasta_no'] != 0].sort_values(by='ad'))
24. YAZDIR("Alfabetik Sıralanmış Hasta Bilgileri:")
25. YAZDIR(df_hasta_sirali)
26. ATA(maas_7000_uzeri = df[df['maas'] > 7000])
27. YAZDIR("Maaşı 7000 TL üzerinde olan personeller:")
28. YAZDIR(maas_7000_uzeri)
29. ATA(df['dogum_tarihi'] = pd.to_datetime(df['dogum_tarihi'], errors='coerce'))
30. ATA(df_dogum_tarihi_1990_sonrasi = df[(df['dogum_tarihi'] >= '1990-01-01') &
(df['dogum_tarihi'].notna())])
31. YAZDIR("Doğum Tarihi 1990 ve Sonrası Olan Hastalar:")
32. YAZDIR(df_dogum_tarihi_1990_sonrasi)
33. ATA(yeni_df = df[["ad", "soyad", "departman", "maas", "uzmanlik", "deneyim_yili",
"hastalik", "tedavi", "tedavi_suresi"]])
34. YAZDIR("Yeni DataFrame:")
35. YAZDIR(yeni_df)
36. SONRA
37.  HATA VARSA(Exception as e)
38.      YAZDIR(str(e))
39. SON
```

UML DİYAGRAMI