**Projekt Leírás**  
A projekt célja egy olyan modern időpontfoglalási rendszer létrehozása, amely hatékonyan támogatja egy férfi fodrászat mindennapi működését. Az applikáció fő feladata az időpontfoglalások nyilvántartása és kezelése egy mátrix-alapú megoldással, amely három napra osztva, félórás bontásban tárolja az ügyfelek foglalásait.

**Rendszer Jellemzői**

1. **Időpontfoglalás kezelése**  
   A rendszer lehetővé teszi az ügyfelek számára, hogy szabad időpontokat keressenek, foglaljanak, és megtekintsék a jelenlegi foglalási állapotot.
2. **Fájlalapú tárolás**  
   A foglalásokat egyszerű szövegfájlban tárolja és onnan tölti be, így könnyű a rendszer integrációja és karbantartása.
3. **Felhasználói interakció**  
   Az alkalmazás egyszerű konzolos felületen keresztül működik, amely biztosítja az ügyfélbarát használatot. A felhasználók névvel regisztrálhatják foglalásaikat.
4. **Foglalások megjelenítése**  
   A foglalási táblázat vizuális megjelenítése segíti a felhasználót az átlátható időpontválasztásban.

**Alkalmazás Felépítése**

* **FoglalasKezelo osztály:** Kezeli az időpontfoglalásokat, valamint a foglalások betöltését, mentését és megjelenítését. Ez a program logikai központja.
* **FodraszatApp osztály:** Ez a program belépési pontja, amely összekapcsolja a felhasználói interakciót a FoglalasKezelo osztály funkcióival.

**Diagramok Részletezése**

**1. Osztálydiagram (Class Diagram)**

Az osztálydiagram bemutatja a rendszer osztályainak felépítését, azok attribútumait és műveleteit, valamint az osztályok közötti kapcsolatokat. Az osztálydiagram szerint:

* **FoglalasKezelo osztály**
  + **Attribútumok:**
    - foglalasok: Egy kétdimenziós tömb, amely a foglalásokat tárolja napokra és időpontokra bontva.
  + **Műveletek:**
    - betoltes(String fajlEleres): Betölti a foglalásokat egy fájlból.
    - mentes(String fajlEleres): Elmenti a foglalásokat fájlba.
    - megjelenitFoglalasok(): Megjeleníti a foglalási táblázatot a konzolon.
    - foglal(int sor, int oszlop, String felhasznaloNev): Foglal egy szabad időpontot a felhasználó nevére.
* **FodraszatApp osztály**
  + **Műveletek:**
    - main(String[] args): A program belépési pontja, amely a felhasználói interakciót és a FoglalasKezelo hívásait koordinálja.

**Kapcsolatok:**

A FodraszatApp használja a FoglalasKezelo osztályt a foglalások kezelésére.

**2. Használati eset diagram (Use Case Diagram)**

Ez a diagram az applikáció főbb funkcióit szemlélteti, és bemutatja a felhasználók (ügyfelek) és a rendszer közötti interakciókat. Főbb használati esetek:

* **Időpont foglalása:** Az ügyfél egy szabad időpontot foglal a kívánt napon és időben.
* **Foglalások megtekintése:** Az ügyfél megtekintheti az aktuális foglalási táblázatot.
* **Foglalások mentése és betöltése:** A rendszer automatikusan kezeli a foglalások fájlba mentését és betöltését.

**Résztvevők:**

* **Felhasználó:** Interakcióban áll a rendszerrel a foglalások kapcsán.
* **Rendszer:** Kezeli a foglalásokat és azok állapotát.

**3. Interakciós diagram (Interaction Diagram)**

Az interakciós diagram azt mutatja be, hogy az osztályok hogyan kommunikálnak egymással egy adott művelet során. Például:

**Időpont foglalása művelet:**

* **Felhasználó** -> **FodraszatApp**: Kérdés az ügyfél nevéről, majd az időpont koordinátáiról.
* **FodraszatApp** -> **FoglalasKezelo**: Foglalási kérés (foglal() metódus meghívása).
* **FoglalasKezelo**: Ellenőrzi, hogy az adott időpont szabad-e. Ha igen, rögzíti a foglalást.
* **FoglalasKezelo** -> **FodraszatApp**: A foglalás sikerességének visszajelzése.
* **FodraszatApp** -> **FoglalasKezelo**: A foglalások mentésének kérése.
* **FoglalasKezelo**: A foglalási mátrix mentése fájlba.