

Fejlesztői Dokumentáció: WPF Vérnyomáskereső
Program

Vérnyomáskereső

Verzió:1.0

2024.10.22

Kontakt információk:

Telefonszám: +36 20 284 5178

Email: horvathi@kkszki.hu

Iroda:3500 Miskolc, Palóczy László utca 5

Tartalomjegyzék

1. Bevezetés

- a. Program célja
- b. Áttekintés

2. Rendszerarchitektúra

2.1 Technológiai Halmaz

2.2 Fájlstruktúra

3. Kódstruktúra és Funkcionalitás

3.1 Adatok osztály (Adatok.cs)

3.2 Főablak vezérlője (MainWindow.xaml.cs)

3.3 Adatmódosító ablak (Modosit.xaml.cs)

4. Felhasználói felület (UI)

- a. Felület felépítése
- b. Működési elvek

5. Mentés és Adatkezelés

- a. Memóriában történő adatkezelés
- b. Jövőbeli fájlkezelési lehetőségek

6. Hibaelhárítás és Karbantartás

6.1 Gyakori Hibák

6.2 Jövőbeli Fejlesztési Lehetőségek

7. Összegzés

- a. Program összefoglalása
- b. További fejlesztési irányok

1. Bevezetés

A WPF-alapú vérnyomásnaplózó alkalmazás lehetővé teszi a felhasználók számára, hogy nyomon kövessék vérnyomásadataikat, kiszámítsák a vérnyomás típusát (normál, prehipertenzió, hipertónia), és kezeljék a felvitt adatokat. A program használható új adatok hozzáadására, meglévő adatok módosítására, és egyszerű adatmentésre.

2. Rendszerarchitektúra

2.1 Technológiai Halmaz

- Fejlesztési környezet: Visual Studio 2022
- Nyelv: C#
- UI Framework: Windows Presentation Foundation (WPF)
- Adatbázis/Adattárolás: Lokális tárolás (memóriában)
- .NET keretrendszer: .NET 6.0

2.2 Fájlstruktúra

- **Adatok.cs** – Az adatok kezeléséért és a vérnyomásmérés típusának kiszámításáért felelős osztály.
- **MainWindow.xaml.cs** – A fő ablak vezérlője, ahol a felhasználó interakciókat hajt végre, mint pl. új mérés felvétele vagy meglévő mérés szerkesztése.

- **Modosit.xaml.cs** – Az adatok módosítására szolgáló ablak vezérlője, amely lehetőséget biztosít új mérés felvételére és meglévő adat módosítására.

3. Kódstruktúra és Funkcionalitás

3.1 Adatok osztály (Adatok.cs)

Az **Adatok** osztály a vérnyomásmérések adatmodelljét tartalmazza. Az alábbi mezőket tartalmazza:

- **Datum:** A mérés dátuma.
- **Ido:** A mérés pontos időpontja.
- **Szisztoles:** Szisztolés vérnyomásérték (felső érték).
- **Diasztoles:** Diasztolés vérnyomásérték (alsó érték).
- **Pulzus:** Pulzusérték.
- **Tipus:** A vérnyomás típusa, amelyet a program automatikusan kiszámol az alábbi logika alapján:

```
1 reference
public String VERNYOMAS_TIPUS()
{
    if (Szisztoles < 120 && Diasztoles < 80)
    {
        return "Normál";
    }
    else if ((Szisztoles >= 120 && Szisztoles < 140) && (Diasztoles >= 80 && Diasztoles < 90))
    {
        return "Prehipertenzió";
    }
    else if ((Szisztoles >= 140 && Szisztoles < 160) && (Diasztoles >= 90 && Diasztoles < 100))
    {
        return "1.fokú hipertónia";
    }
    else if ((Szisztoles >= 160 && Szisztoles < 180) && (Diasztoles >= 100 && Diasztoles < 110))
    {
        return "2.fokú hipertónia";
    }
    else if (Szisztoles >= 180 && Diasztoles >= 110)
    {
        return "Magas vérnyomás krízis";
    }
    return "Nincs ilyen típus!";
}
```

A program a mérések rögzítésekor automatikusan meghatározza a vérnyomás típusát (pl. normál, prehipertenzió stb.).

3.2 Főablak vezérlője (MainWindow.xaml.cs)

A fő ablak kezeli a felhasználói interakciókat, beleértve a mérések hozzáadását és módosítását.

- Az adatok kezelése listában történik, ahol minden mérés objektumként van tárolva.
- A **MainWindow** osztály tartalmazza az eseménykezelőket, például az adatok mentését és visszatöltését.

```
1 reference
private void btnFelvetel_Click(object sender, RoutedEventArgs e)
{
    Modosit modosit = new Modosit();
    modosit.Title = "Felvétel";
    modosit.ShowDialog();
}
```

- A felhasználó által bevitt adatokat az **AdatokLista** nevű lista tárolja, amely az összes mérés adatait tartalmazza.

3.3 Adatmódosító ablak (Modosit.xaml.cs)

Ez az ablak lehetőséget biztosít a felhasználóknak adatok módosítására vagy új mérés felvételére. A logika a következőképpen működik:

```
1 reference
public Modosit(int id)
{
    InitializeComponent();
    this.id = id;
    Adatok adatok = MainWindow.AdatokLista[id];
    tbDatum.Text = adatok.Datum;
    tbIdo.Text = adatok.Ido;
    tbSzisztoles.Text = Convert.ToString(adatok.Szisztoles);
    tbDiasztoles.Text = Convert.ToString(adatok.Diaszteles);
    tbPulzus.Text = Convert.ToString(adatok.Pulzus);
}
```

- Az ablak azonosítja a szerkesztendő adatot az ID alapján, majd betölti az adott mérés részleteit.
- A felhasználó módosíthatja a méréseket, majd az új adatokat elmenti.

-

Ez az osztály két fő funkciót lát el:

- Új mérés felvétele.
- Meglévő mérés módosítása.

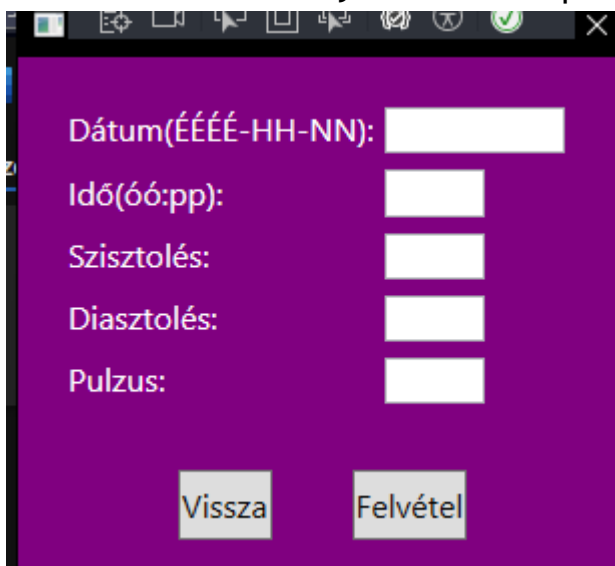
4. Felhasználói felület (UI)

A WPF használatával egy egyszerű, de hatékony felhasználói felület készült. A felhasználói felület a következőket tartalmazza:

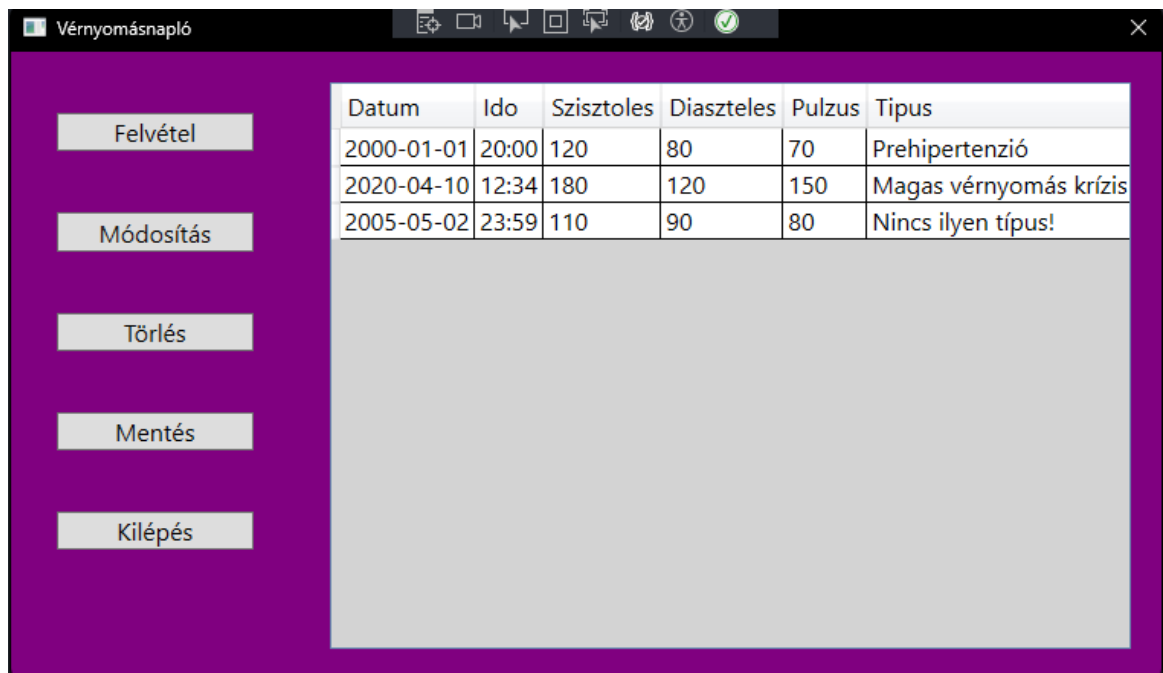
- **Mérések listája:** Megjeleníti a korábban rögzített vérnyomásméréseket.

Datum	Idő	Szisztolés	Diasztolés	Pulzus	Típus
2000-01-01	20:00	120	80	70	Prehipertenzió
2020-04-10	12:34	180	120	150	Magas vérnyomás krízis
2005-05-02	23:59	110	90	80	Nincs ilyen típus!
2060-12-12	23:59	120	120	100	Nincs ilyen típus!

-
- **Adatbeviteli mezők:** Dátum, idő, szisztolés és diasztolés érték, valamint pulzusérték bevitele.



-
- **Gombok:** Új mérés hozzáadása, meglévő adatok módosítása.



5. Mentés és Adatkezelés

- **Memóriában történő tárolás:** Az adatok a program futása közben a memóriában vannak tárolva. Az adatok nem kerülnek automatikusan fájlba mentésre, de a későbbi fejlesztések során fájlkezelés (pl. CSV, XML export) hozzáadható.
- **Adatok módosítása:** A felhasználó módosíthatja a korábban rögzített adatokat az ID alapján.

6. Hibaelhárítás és Karbantartás

6.1 Gyakori Hibák

Hibás adatbevitel a vérnyomásmérésnél:

- **Leírás:** Ha a felhasználó nem megfelelő formátumú adatokat visz be a szisztolés, diasztolés vagy pulzus mezőkbe (például betűket számok helyett), a program hibaüzenetet ad vissza, mivel ezek a mezők numerikus adatokat várnak.
- **Megoldás:** Biztosítsd, hogy a mezők validálása megfelelően működik, és a nem numerikus karaktereket

vagy hibás formátumokat (pl. túl alacsony vagy magas értékek) már az adatbevitel során blokkolja a program.

```
if (!int.TryParse(tbSzisztoles.Text, out int szisztoles) ||
    !int.TryParse(tbDiasztoles.Text, out int diasztoles) ||
    !int.TryParse(tbPulzus.Text, out int pulzus))
{
    MessageBox.Show("Kérjük, csak számokat adjon meg a vérnyomás értékekhez.", "Hiba", MessageBoxButtons.OK, MessageBoxIcon.Error);
    return;
}
```

Adatok mentésének hibája:

- **Leírás:** A program az adatok memóriában történő tárolására van optimalizálva. Ha a felhasználó új mérés rögzítése után bezárja a programot, az adatok elveszhetnek, mivel nincs fájlba vagy adatbázisba történő automatikus mentés.
- **Megoldás:** A későbbi fejlesztések során hozzáadható fájlba mentési opció (pl. CSV vagy XML fájl), amely az adatok tartós tárolását biztosítja.

Adatmódosítás közbeni hiba:

- **Leírás:** Ha a felhasználó egy már meglévő mérés módosítása közben hibás adatokat ad meg, az alkalmazás helytelen adatokat menthet. Ez főként akkor történik, ha nincs megfelelően implementálva az adatok ellenőrzése (pl. null értékek vagy hibás formátum).
- **Megoldás:** Az adatbevitel minden esetben validálásra kell kerüljön a mentés előtt. Ezenkívül az üres mezők esetében figyelmeztetni kell a felhasználót.

6.2 Jövőbeli Fejlesztési Lehetőségek

- **Fájlba mentés:** Lehetőség CSV, XML vagy adatbázis (pl. SQLite) használatára.

- **Adatszinkronizálás:** Felhőalapú adatszinkronizálás a felhasználói élmény javításához.

7. Összegzés

Ez a vérnyomásnaplózó program egyszerű, de hatékony megoldást kínál a vérnyomásmérések nyomon követésére. A program a WPF segítségével tiszta felhasználói felületet biztosít, amely támogatja a vérnyomásmérések bevitelét, kategorizálását és módosítását. A jövőbeli fejlesztések során kiegészíthető fájlba mentési és adatbázis-kezelési funkciókkal.