# Atomreaktor működés Dokumentáció

# Szerzők

**Név**: **Kecskés Roland, Csillik Gergely**

**Email**: [kecskesr@kkszki..hu](mailto:kecskesr@kkszki..hu), csillikg@kkszki.hu

# Feladat: Atomreaktor

**Tartalomjegyzék**

# Felhasználói Dokumentáció

## Bevezetés

Az Atomreaktor alkalmazás egy szimulációs program, amely lehetővé teszi egy atomreaktor működésének szimulálását. Ez az alkalmazás lehetővé teszi a felhasználók számára, hogy interaktív módon vegyenek részt egy virtuális reaktor működtetésében, megfigyeljék annak hőmérsékletét és generált energiáját, valamint kezeljék a kritikus helyzeteket.

## Környezet

PC, Visual Studio futtatására alkalmas operációs rendszer (pl. Windows 10).Nem igényel egeret.

## Telepítés és Indítás

Az alkalmazás használatához nincs szükség telepítésre, egyszerűen elég csak letölteni és futtatni.

1. Töltse le a programot a GitHub repository-ból vagy klónozza le a forráskódot a saját számítógépére.

2. Nyissa meg a letöltött könyvtárat vagy kicsomagolt mappát.

3. Indítsa el az alkalmazást a `SzinhaziJegyErtekesitoProgram.exe` fájlra kattintva.

## Hibák

Együttműködő csapatunk elméleti szinten fixált minden lehetséges hibát vagy gondolt rá és ezzel megelőzzük őket program futtatása közben. Ha esetlegesen találkozik hibával kérem értesítsen minket minnél hamarabb ezeken az email címeken: [kecskesr@kkszki..hu](mailto:kecskesr@kkszki..hu), [csillikg@kkszki.hu](mailto:csillikg@kkszki.hu).

Tesztelések során találkozott hibákkor a program megfagyott és megállt. Ne ijedjen meg indítsa újra teljesen a programot és próbálja elkerülni azt a mozdulatot amivel előhozta a hibát!

## Használat(elméleti)

Az alkalmazás használata egyszerű és intuitív. Az alábbi lépések segítségével lehet elindítani és használni:

Indítsa el az alkalmazást a Program.cs fájlban lévő Main metódus segítségével.

Az alkalmazás elindítása után megjelenik a menü, amely különböző műveleteket kínál az atomreaktorral kapcsolatban. A menü megjelenése a következő:

#### Menü:

1. Beindítás

2. Leállítás

3. Generált adatok megjelenítése

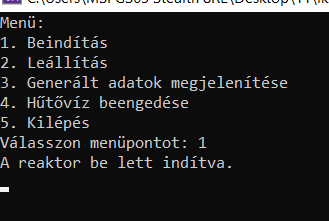
4. Hűtővíz beengedése

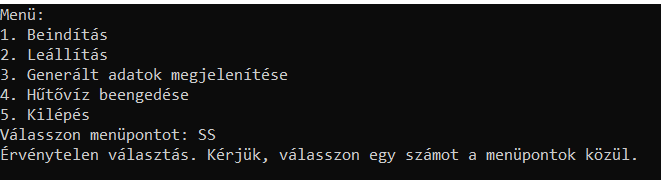
5. Kilépés

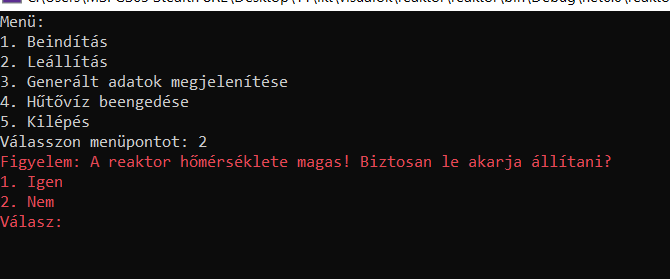
Válassza ki a kívánt műveletet a megfelelő szám beírásával.

Az egyes műveletek végrehajtása után az alkalmazás újra megjeleníti a menüt, hogy újabb műveletet választhasson.

## Használat(gyakorlati)

A program sikerelindítása utána ez a menü fog fogadni bennünket:





Felhasználói Interfész

Az alkalmazás konzolos felhasználói felülettel rendelkezik, amely egyszerű és intuitív navigációt biztosít a különböző funkciók között. A képernyőn megjelenő utasítások segítségével könnyen használhatja az alkalmazást.

## Fájlok

Az alkalmazás használata során a következő fájlok jönnek létre és módosulnak:

- `Reaktor.exe`: Az alkalmazás végrehajtható fájlja.

## További Segítség

Ha további segítségre van szüksége az alkalmazás használata során, olvassa el a mellékelt dokumentációt vagy forduljon a fejlesztőkhöz.

## Összefoglalás Felhasználói Szempontból

Az Atomreaktor alkalmazás segítségével könnyedén megismerheti egy atomreaktor működését, valamint interaktívan részt vehet a szimulációban. Kérem, vegye figyelembe az alkalmazás figyelmeztetéseit és ügyeljen a biztonságra!

Ha bármilyen kérdése vagy problémája merülne fel az alkalmazás használata során, kérjük, ne habozzon kapcsolatba lépni az alkalmazás fejlesztőjével.

# Fejlesztői Dokumentáció

## Fejlesztői Előfeltételek:

- C# programozási nyelv ismerete.

- Visual Studio vagy más C# fejlesztőeszköz telepítése és konfigurálása.

- Git vagy más verziókezelő rendszer ismerete és konfigurálása a forráskód verziókezeléséhez.

## Bevezetés

A Reaktor kezelő program egy egyszerű, konzolos alkalmazás, amely lehetővé teszi a felhasználók számára egy reaktor működésének kezelését. Ez a dokumentáció részletes útmutatást nyújt a program felépítéséről, a forráskódról, valamint a fejlesztőknek szóló hasznos információkról.

## Környezet

PC, Visual Studio futtatására alkalmas operációs rendszer (pl. Windows 10).Nem igényel egeret.

## Program Felépítése

A program a következő fő részekből áll:

1. Main metódus: A program belépési pontja. Itt történik a felhasználói interakciók kezelése és a menü megjelenítése.
2. Atomreaktor osztály: Ez az osztály felelős az atomreaktor működésének modellezéséért. Tartalmazza az aktuális hőmérsékletet, a generált energiát, valamint az atomreaktor állapotát (működés vagy leállás).
3. Program osztály: Ez az osztály tartalmazza a Main metódust, amely inicializálja az atomreaktor objektumot és vezérli az alkalmazás futását a felhasználói interakciók alapján.

## Programstruktúra

Az alkalmazás forráskódja C# nyelven íródott és két osztályból és több metódus, függvényből áll:

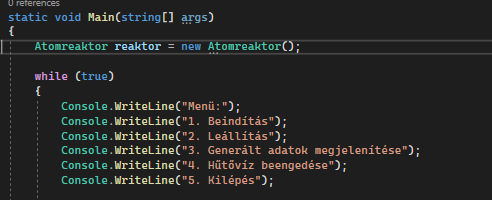
### Osztályok:

Atomreaktor osztály: Ez az osztály modellezi egy atomreaktor működését és kezeli annak állapotát.

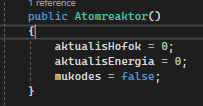
Program osztály: Ez az osztály tartalmazza a Main metódust, ami az alkalmazás belépési pontja.

### Metódusok:

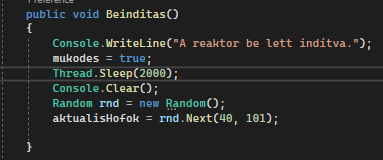
static void Main(string[] args): A program belépési pontja, vezérli a felhasználói interakciót a konzolon keresztül.



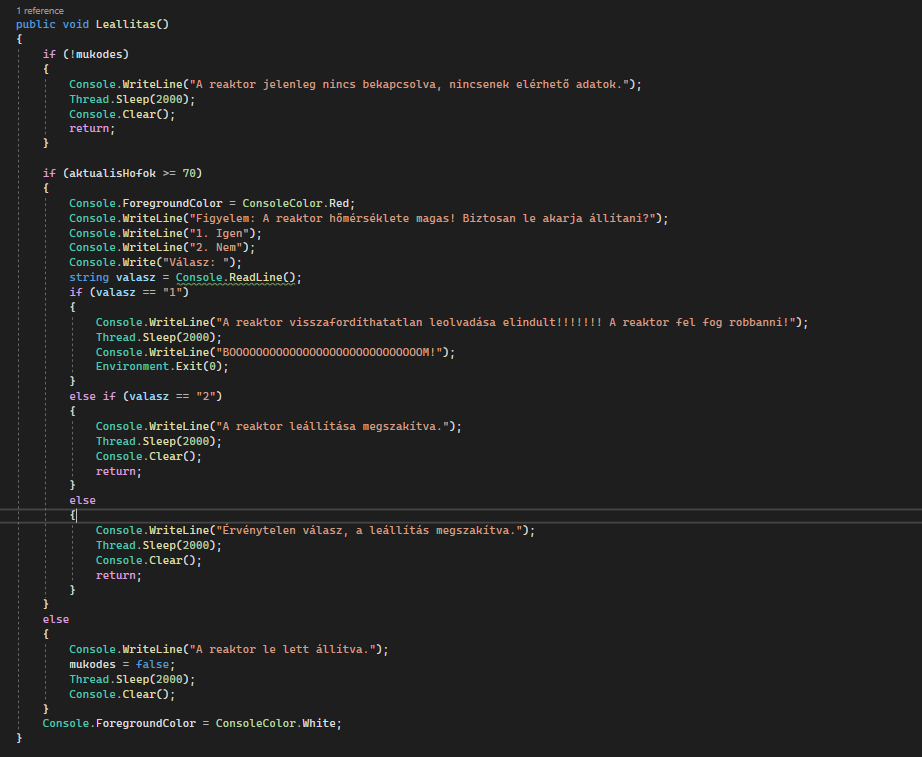
public Atomreaktor(): Konstruktor inicializálja az Atomreaktor objektumot.



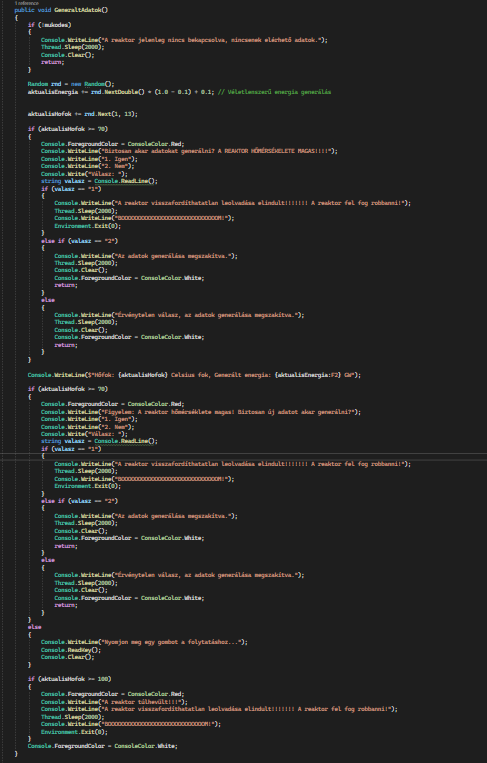
public void Beinditas(): Metódus a reaktor beindításához.



public void Leallitas(): Metódus a reaktor leállításához.



public void GeneraltAdatok(): Metódus a generált adatok megjelenítéséhez.



public void HutovizBeengedese(): Metódus a hűtővíz beengedéséhez.

## 

## Hibakeresés és Hiba javítás

A program különböző hibákat kezel, például helytelen bemeneteket vagy problémákat a fájl olvasása vagy írása közben. Hibás bemenet esetén megjeleníti az erre vonatkozó üzenetet, majd lehetőséget ad a helyes bemenet megadására.

Az alkalmazás hibáinak javítása és a hibák elhárítása a következő lépésekből áll:

**1. Hiba Felfedezése**: A felhasználói visszajelzések, hibajelentések és tesztek alapján a hibák felfedezése.

**2. Hibakeresés**: A hiba forrásának meghatározása a forráskód és a tesztek alapján.

**3. Hibajavítás**: A hiba javítása a forráskód módosításával.

**4. Tesztelés**: Az alkalmazás újbóli tesztelése a hiba javítása után.

## Korlátok:

* Adatok tárolása nem lehetséges

## Fejlesztési Folyamat

Az alkalmazás fejlesztési folyamata az alábbi lépésekből áll:

**1. Követelmények Felülvizsgálata:** Az alkalmazás funkcionalitásának és követelményeinek felülvizsgálata.

**2. Tervezés:** Az alkalmazás tervezése, beleértve az osztályokat, a metódusokat és az interfészt.

**3. Implementáció:** Az alkalmazás kódolása a tervezés alapján.

**4. Tesztelés:** Az alkalmazás funkcióinak tesztelése, hibák keresése és javítása.

**5. Dokumentáció:** A fejlesztői és felhasználói dokumentáció elkészítése az alkalmazás működéséről és használatáról.

## Fejleszthetőségi lehetőségek

* Program esztétikusabbá tétele.
* Értelmes ötletek megvalósítása.

### Néhány ötlet:

1. Felhasználói felület fejlesztése: További funkciók hozzáadása, grafikus vagy webes felület létrehozása.
2. Bővített funkciók: Több paraméter és állapot figyelembevétele a reaktor működésének részletesebb modellezéséhez.
3. Adatok tárolása

## Verziókezelés

Az alkalmazás forráskódjának verziókezelése Git segítségével történik. A verziókezelés lehetővé teszi a kódbázis változásainak nyomon követését, a különféle verziók létrehozását és a csapatmunkát.

## Összefoglalás Fejlesztői Szempontból

A Reaktor kezelő program C# nyelven íródott és könnyen fejleszthető. Az alkalmazás strukturált és jól dokumentált forráskóddal rendelkezik, amely lehetővé teszi az egyszerű karbantartást és bővítést. A fejlesztőknek széles körű fejlesztői dokumentáció és támogatás áll rendelkezésre, így könnyen megérthetik az alkalmazás működését és fejleszthetik azt tovább.

# Teljes összefoglalás

Az Atomreaktor alkalmazás egy szimulációs program, amely lehetővé teszi egy atomreaktor működésének szimulálását. A jól strukturált forráskód, a dokumentáció és a támogatás segítséget nyújt mind a felhasználók, mind a fejlesztők számára.

## Köszönetnyilvánítás

Szeretném megköszönni mindazoknak, akik támogattak és inspiráltak a projekt során. Külön köszönet akik türelmesen álltak mellém, miközben ezen a projekten dolgoztunk. Hálás vagyok a szakmai támogatásért és a tanácsokért, amiket kaptam, és mindazoknak, akik hozzájárultak az alkalmazás fejlesztéséhez közvetlenül vagy közvetve.

Az elkészült alkalmazás az én és társam, barátom Csillik Gergely szakmai fejlődésemet is segítette, és bízom benne, hogy másoknak is hasznos eszköz lesz.