**Dokumentáció a Számológép Projekthez – Bankó Olivér és Hoffer Tamás**

**A program célja**

A projekt célja egy egyszerű számológép készítése Python programozási nyelvben a Tkinter könyvtár segítségével, amely lehetővé teszi alapvető matematikai műveletek (összeadás, kivonás, szorzás, osztás) elvégzését egy grafikus felhasználói felületen keresztül. A számológép egyszerűsített verzió, mely elsősorban oktatási célokat szolgál, és bemutatja, hogyan lehet egy egyszerű GUI alapú alkalmazást megvalósítani Pythonban.

**Megvalósított technológia**

A program Pythonban íródott, és a Tkinter könyvtárat használja a grafikus felület létrehozásához. A Tkinter lehetővé teszi különféle GUI elemek, például gombok, címkék és szövegdobozok egyszerű kezelését és megjelenítését. A Tkinter előnye, hogy könnyen elsajátítható, és beépített Python csomagként elérhető, így kezdőknek is ideális választás.

**Felhasználói interakciók**

A program egyszerű gombokkal vezérelhető:

* **Számgombok (0-9)**: Ezek a gombok a számok bevitelére szolgálnak.
* **Műveleti gombok (+, -, \*, /)**: Lehetővé teszik a műveletek kiválasztását a megadott számok között.
* **Clear**: A Clear gomb törli az aktuális bevitt számot és a részeredményeket.
* **Egyenlőségjel (=)**: Az egyenlőségjel megjeleníti az eredményt az aktuálisan kijelölt művelet és a beírt szám alapján.

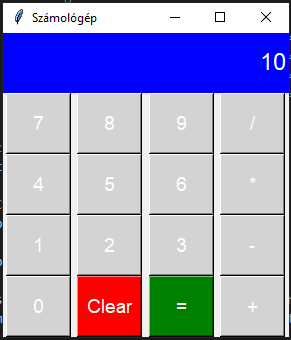
**Kód**

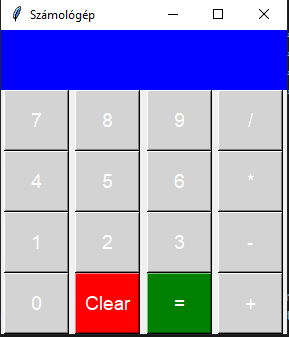
A kódot és a hozzá tartozó magyarázatokat az alábbiakban találod.

* import tkinter as tk
* A tkinter könyvtárat importáljuk, amely biztosítja a Python alapú grafikus elemek (pl. ablak, gombok) használatát.
* class Szamologep:
* def \_\_init\_\_(self, master):
* self.master = master
* self.master.title("Számológép")
* A Szamologep osztály tartalmazza a számológép teljes funkcionalitását. Az osztály az \_\_init\_\_ konstruktorban inicializálja az alapvető beállításokat és a felhasználói felület elemeit.
* A self.master az alkalmazás főablakára hivatkozik, amit a master változón keresztül adunk át a konstruktorba.
* self.kijelzo = tk.Label(master, text="", width=20, height=2, font=("Arial", 18), anchor='e')
* self.kijelzo.grid(row=0, column=0, columnspan=4)
* Létrehozunk egy Label objektumot, ami a számológép kijelzője lesz, és a bevitt értékeket, valamint az eredményeket jeleníti meg.
* A grid metódus meghatározza az elemek elrendezését a felületen, a row és column paraméterekkel állíthatjuk be, hogy melyik sorban és oszlopban helyezkedjenek el.
* buttons = [
* ('7', 1, 0), ('8', 1, 1), ('9', 1, 2), ('/', 1, 3),
* ('4', 2, 0), ('5', 2, 1), ('6', 2, 2), ('\*', 2, 3),
* ('1', 3, 0), ('2', 3, 1), ('3', 3, 2), ('-', 3, 3),
* ('0', 4, 0), ('Clear', 4, 1), ('=', 4, 2), ('+', 4, 3)
* ]
* Létrehozunk egy listát, amely a számológép összes gombját definiálja. Minden gomb egy tuple, amely tartalmazza a megjelenített szöveget, valamint a sor- és oszlopszámot.
* for (text, row, col) in buttons:
* tk.Button(master, text=text, width=5, height=2, font=("Arial", 14),
* command=lambda t=text: self.gomb\_nyomas(t)).grid(row=row, column=col)
* A for ciklusban létrehozunk minden gombot. A command paraméter biztosítja, hogy a gomb lenyomásakor meghívódjon a gomb\_nyomas metódus a gombhoz tartozó szöveggel (t), így minden gombnak saját funkciója lesz.
* self.bevitel = ""  # A kijelzőre írandó érték
* self.szam1 = None  # Az első operandus
* self.muvelet = None  # A kijelölt művelet
* A self.bevitel a kijelzőn megjelenő aktuális értéket tárolja, míg a self.szam1 az első számot, és a self.muvelet az aktuális műveletet tárolja.
* def gomb\_nyomas(self, gomb):
* if gomb.isdigit():
* # Szám hozzáadása a bevitelhez
* self.bevitel += gomb
* self.kijelzo.config(text=self.bevitel)
* Ha számgombot nyomunk meg, azt a bevitel változóhoz fűzi, és frissíti a kijelzőn megjelenő szöveget.
* elif gomb in "+-\*/":
* # Első szám rögzítése és művelet kiválasztása
* if self.bevitel:
* self.szam1 = float(self.bevitel)
* self.muvelet = gomb
* self.bevitel = ""
* self.kijelzo.config(text=self.muvelet)
* Ha egy műveleti gombot nyomunk meg, először ellenőrzi, hogy van-e érvényes bevitt érték (bevitel). Ha igen, akkor elmenti azt, mint szam1, elmenti a művelet típusát, és kiüríti a kijelzőt az új számhoz.
* elif gomb == "=":
* # Számítás végrehajtása és eredmény megjelenítése
* if self.bevitel and self.szam1 is not None and self.muvelet:
* szam2 = float(self.bevitel)
* eredmeny = 0
* if self.muvelet == "+":
* eredmeny = self.szam1 + szam2
* elif self.muvelet == "-":
* eredmeny = self.szam1 - szam2
* elif self.muvelet == "\*":
* eredmeny = self.szam1 \* szam2
* elif self.muvelet == "/":
* if szam2 != 0:
* eredmeny = self.szam1 / szam2
* else:
* eredmeny = "Hiba (0-val osztás)"
* self.kijelzo.config(text=str(eredmeny))
* self.bevitel = str(eredmeny)
* self.szam1 = None
* self.muvelet = None
* Az = gomb lenyomásakor az elmentett első szám (szam1), a második szám (szam2), és a művelet alapján kiszámítja az eredményt.
* Ha az osztó nulla, hibaüzenetet ad. Az eredményt ezután megjeleníti a kijelzőn.
* elif gomb == "Clear":
* # Minden törlése
* self.bevitel = ""
* self.szam1 = None
* self.muvelet = None
* self.kijelzo.config(text="")
* A Clear gomb törli az összes korábban bevitt adatot és alaphelyzetbe állítja a számológépet.
* root = tk.Tk()
* app = Szamologep(root)
* root.mainloop()
* A program főablaka (root) létrejön, és a Szamologep osztályhoz rendeli. A mainloop tartja nyitva az ablakot, amíg a felhasználó be nem zárja.

**Továbbfejlesztési lehetőségek**

* **Bővített műveletek**: Négyzetgyök, hatványozás, vagy százalékszámítás bevezetése.
* **Kivételkezelés**: Pontosabb hibaüzenetek, például nulla osztásnál vagy érvénytelen műveleteknél.
* **Bővített kijelző**: Műveletek és számok folyamatos megjelenítése

**Illusztráció**

****

****