**Kaposvári Szakképzési Centrum**

**Eötvös Loránd Műszaki Technikum és Kollégium**

**7400 Kaposvár, Pázmány Péter u. 17.**

****

**Projektfeladat 4.**

**2023 / 2024-es tanév**

Készítette: Szekeres Benjámin

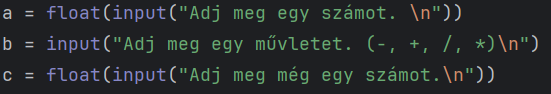
A feladat egy alap kis számológép megírása volt Python-ban. A számológép csak az alap 4 műveletet ismeri, amik a szorzás, osztás, össze adás, kíván. Illetve csak is 2 darab számot lehet megadni.

Kezdetek kezdetén át kellett gondolni, hogy hogyan kezdeném el a feladatot kivitelezni.

Hosszan gondolkozások után arra jutottam, hogy kelleni fog 3 darab változó amiben fogom tárolni a bekért adatokat. Ezeket az egyszerűség kedvéért a, b, c névvel láttam el, így sokkal gyorsabban és egyszerűbben tudtam hivatkozni rá a kód későbbi részeiben.

Mivel én azt szeretném, hogy ne csak egész számokkal tudjon a számológépen számolni ezért floatként kértem be a 2 számot az a és c változóba. Majd a b string típusú változóba kértem be.

És mivel tetszik az, hogy ne egy sorba kelljen írni a megadott értéket a kéréssel ezért egy „\n” jellel sort törtem, tehát új sorba fogja kérni a megadni való értéket.



Miután megtörtént az adatok bekérése és nagyszerű módú tárolása, mind az első számnak, mind a második számnak, mind a műveleti jelnek. Így fog kinézni gyakorlatban a bekérés, mikor a felhasználónak meg kell adnia egy értéket és meg is ad egy értéket.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Mindazok után, hogy bevittem az adatokat a számológép működtetéséhez, létre kell hoznom magát a működési folyamatot. Ami pedig úgy néz ki, hogy az átláthatóság kedvéért és az órán gyakorlott feladatokból inspirálódva kezdtem bele magába a logikai részébe, amitől össze áll a számológép. Legeslegelső sorban létrehoztam egy def függvényt amit „szorzas” névvel láttam el. Ez a def függvény bekéri a kettő megadott szám adatot és azokat össze szorozza, majd ahelyett, hogy vissza adom az eredményt a „return” segítségével, inkább egyből ki íratom a konzolra az eredményt. Megcsináltam még az osztást is ugyan ilyen módon, hogy egy def-ben bekérem a kettő megadott számot és elosztom egymással. Itt is az egyszerűségért egyből ki lesz íratva az eredmény a konzolra. Majd ezután következett az össze adás, amit szintén ugyan olyan módszerrel csináltam meg, hogy létrehoztam egy def függvényt, és abba kértem bele a kettő előre bekért szám értéket, amit a felhasználó megadott. Majd az össze adás elvégzése után ez a függvény is a return-t kihagyva egyből kiírja az eredményt a konzol felületre. Végül létrehoztam a kivonás műveletnek is egy def függvényt, ahol ugyan csak bekérem a felhasználó által már megadott szám értékeket és a művelet elvégzése után itt is egyből az eredmény ki írattam a konzolra.

Szorzás:

A képen Betűtípus, szöveg, képernyőkép, Grafika látható

Automatikusan generált leírás

Osztás:

A képen Betűtípus, szöveg, kézírás, képernyőkép látható

Automatikusan generált leírás

Összeadás:

A képen Betűtípus, szöveg, képernyőkép, tipográfia látható

Automatikusan generált leírás

Kivonás:

A képen Betűtípus, szöveg, képernyőkép, szám látható

Automatikusan generált leírás

A def függvények kialakítása után fel is kell ismerni, hogy a felhasználó milyen műveleti jelet adott meg, tehát, hogy pontosan mi is van a b változóban. Ezt if függvényekkel tettem meg. Pontosabban egy darab if függvénnyel és három darab elif függvénnyel, majd arra is kellett gondolni, hogy a felhasználó nem a megadott műveleti jelek egyikéből fog kiválasztani egyet és valami teljesen más értéket fog megadni. Erre létre hoztam egy else függvényt is. Így ennek a függvény sokaság pontos lényege, hogy első sorban megvizsgálja nekem, hogy a felhasználó által megadott érték az egy „\*” e. Ha igen, akkor a már előre létrehozott megfelelő def függvényt fogja meghívni az a és a c változót használva. Ha nem a csillag jelet adta meg a felhasználó akkor megy tovább a kód az elifekre, első sorban az osztásra. Itt egy „/” jelet fog keresni a kód, ha a b nevű változó egyenlő lesz a „/” jellel akkor a felhasználó osztást kívánt végezni és a kód nem fog tovább menni a következő elifre, hanem meghívja a megfelelő def fügvényt ami jelen esetben az osztáshoz tartozó függvény lesz. Amennyiben a felhasználó nem a „\*” jelet és nem a „/” jelet adta meg, akkor a kód folytatódni fog egy újabb elif-el, mégpedig az összeadással. Ha a felhasználó a „+” jelet adta meg akkor ez az elif függvény ezt érzékeli és ha úgy látja, hogy a felhasználó az össze szeretné adni az a kettő számot amit megadott, akkor meghívja az összeadásra szolgáló de függvényt, ami az művelet elvégzése során kiírja a konzolra az összeadás eredményét. De mégis mi van akkor ha a felhasználó nem „\*” jelet, nem „/” jelet és nem „+” jelet adott meg? Van még esély rá, hogy a felhasználó kivonást szeretne végezni, tehát ezt a következő elif függvény fogja ellenőrizni. Hogy ha felhasználó tényleg kivonni szeretne és a „-” jelet adta meg mint műveleti jel, akkor a kód meghívja az utolsó lehetséges eseményt, a kivonásra szolgáló def függvényt, ami elvégzi a műveletet és az elvégzés után kiírja a konzolra az eredményt. Na de abban az esetben mi a helyzet, hogy ha felhasználó nem a „\*” jelet, nem a „/” jelet, nem a „+” jelet, de nem is a „-” jelet adta meg műveleti jelként hanem valami teljesen mást. Erre van az else függvény, ami akkor lép érvénybe hogy ha felhasználó által megadott műveleti jel nincs a felsoroltakban. Az else függvény van a legutolsó helyen és csak annyit csinál, hogy megszakítja a kódot és egy hiba üzenetet fog vissza adni a konzolra a felhasználó részének.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Maga az egész kód az ennyiből áll. 4 fajta módon lehet variálni, hogy milyen műveletet szeretnénk elvégezni.

Lehet az összeadás

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Lehet az kivonás

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Lehet az szorzás

A képen szöveg, Betűtípus, képernyőkép látható

Automatikusan generált leírás

Lehet az osztás

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

És amennyiben a műveleti jel nem valós

A képen szöveg, Betűtípus, képernyőkép látható

Automatikusan generált leírás