

**Vizuális\_programozás\_L-K-GINFBAL-VIZUPROG-1**

**MIT App Inventor – Calculator applikáció**

**Schiller Viktor – GWOQXX**

**Második Évfolyam – 2. Szemeszter**

**NJE GAMF Műszaki és Informatikai Kar**

Tartalom

[1. MIT App Inventor 2](#_Toc129462800)

[1.1. Egyszerű mobilalkalmazás készítése 2](#_Toc129462801)

[2. Elrendezés kialakítása 3](#_Toc129462802)

[2.2 Kijelző ablak létrehozása 3](#_Toc129462803)

[2.3 Gombsor létrehozása 4](#_Toc129462804)

[3. A programkód 6](#_Toc129462805)

[3.1 Globális változók 6](#_Toc129462806)

[3.2 A Metódus 6](#_Toc129462807)

[3.3 A nyomógombok 7](#_Toc129462808)

[3.4 A float\_gomb 7](#_Toc129462809)

[3.5 Műveleti jelek 8](#_Toc129462810)

[3.6 A gyökölő 9](#_Toc129462811)

[3.7 Az előjelváltás 10](#_Toc129462812)

[3.8 A törlés (Clear) billentyű 11](#_Toc129462813)

[3.9 Az egyenlőség gomb 11](#_Toc129462814)

[4. Összegzés 12](#_Toc129462815)

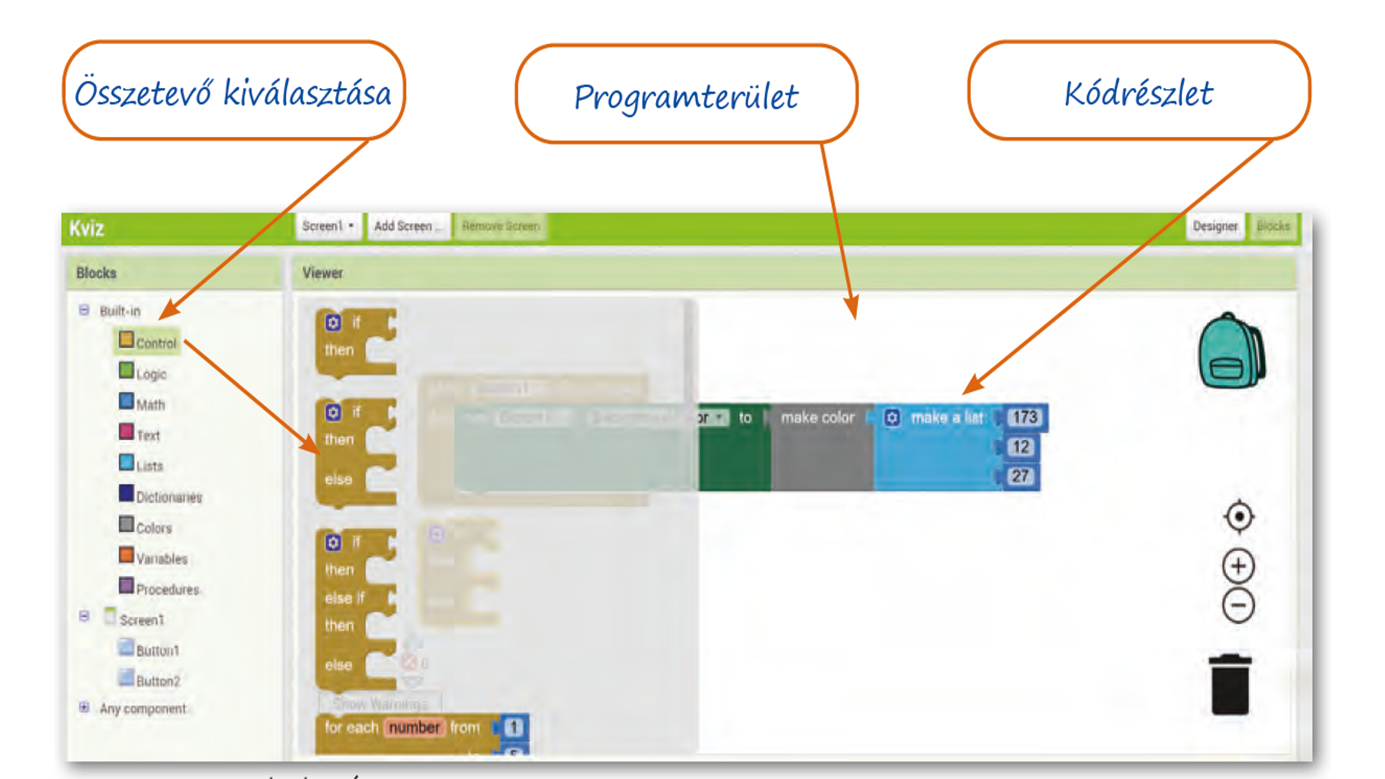
[FELHASZNÁLT IRODALOM & KÉPEK 14](#_Toc129462816)

# 1. MIT App Inventor

## **1.1. Egyszerű mobilalkalmazás készítése**

A mobil eszközök programozása szakértelmet igényel, de vannak olyan eszközök, amelyek ezt egy átlagos felhasználó számára is könnyen érthetővé teszik. Ilyen például a MIT App Inventor alkalmazás, amelynek segítségével egyszerűen készíthetünk programokat Android operációs rendszerre. Az alkalmazásokat online felületen lehet létrehozni. Az ehhez szükséges weboldalt a MIT (Massachusetts Institute of Technology) tartja fenn. Az oldal használatához be kell jelentkezni, amelyhez a Google-fiókunk adatait kell megadnunk. Miután regisztráltam és elfogadtam a feltételeket, új projektet indítottam.

Az applikációm elkészítése során két lényegesen különböző felületen dolgoztam. Az első felület a Designer ablak: itt állíthatjuk be a képernyő és a képernyőn megjelenő egyes elemek kinézetét. A projekt nevének megadása után ebbe a nézetbe lépünk be. A másik a Blocks ablak: ott készíthetjük el az alkalmazásunk kódját. A két nézet között a jobb felső sarokban lévő gombokkal válthatunk.

****

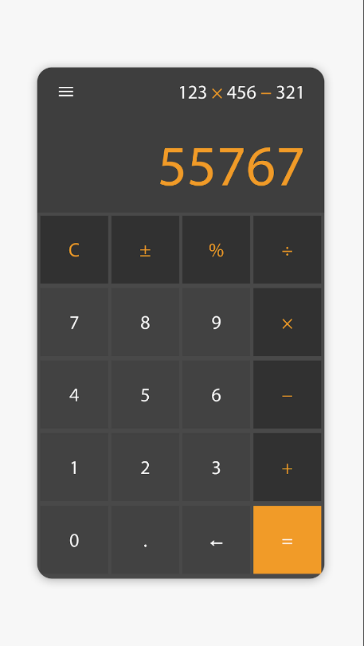
*1. ábra. App Inventor „Block” nézete*

A középen látható telefon képernyőjén jelennek meg az általam kiválasztott elemek. A beilleszthető objektumok listája (Palette) a telefon képétől balra helyezkedik el, a másik oldalon pedig a tulajdonságaikat állíthatjuk be. Az elemeket egyszerűen behúzhatjuk a telefon képernyőjére, majd a jobb oldalon beállítjuk a kinézetüket.

A Blocks nézetben a beállított elemeknek megfelelően csoportosítva jelennek meg a program egyes lehetséges utasításai. Ezeket az utasításokat kiválasztva és a kódterületre (Viewer) behúzva állíthatjuk össze a programunkat.

# 2. Elrendezés kialakítása

Legelső lépésben kerestem egy minta számológép alkalmazás kinézetet az internet képkeresőjében, ami sablonként fog szolgálni az elemek, gombok, kijelző, a komplett GUI elrendezését illetően, amolyan beépített Windows-osat, letisztult design-al, használható felülettel. Íme:

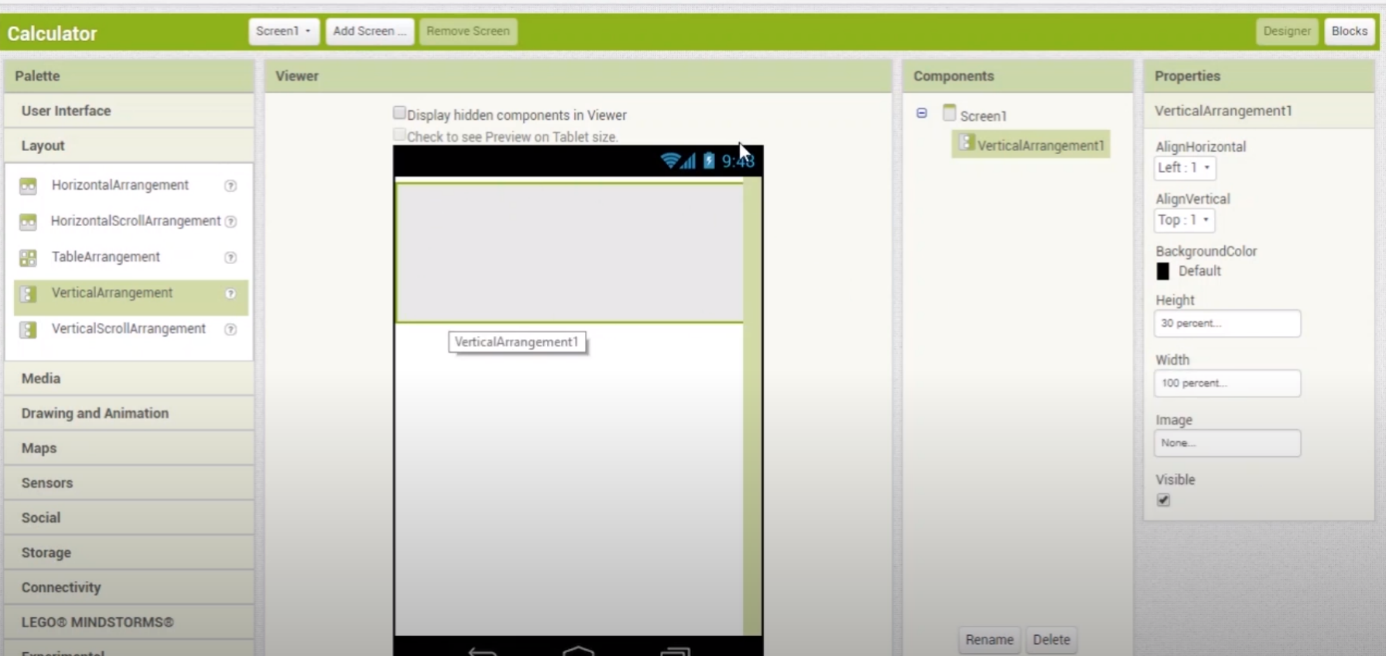


*2. ábra. A számológép-minta*

Először létrehoztam az új projektet a menüsorban a *My Projects > New project* menüpontnál. Legyen a projekt neve Calculator. Ezután a képernyő kinézetét kellett felépítenem a *Designer* nézetben.

## **2.2 Kijelző ablak létrehozása**

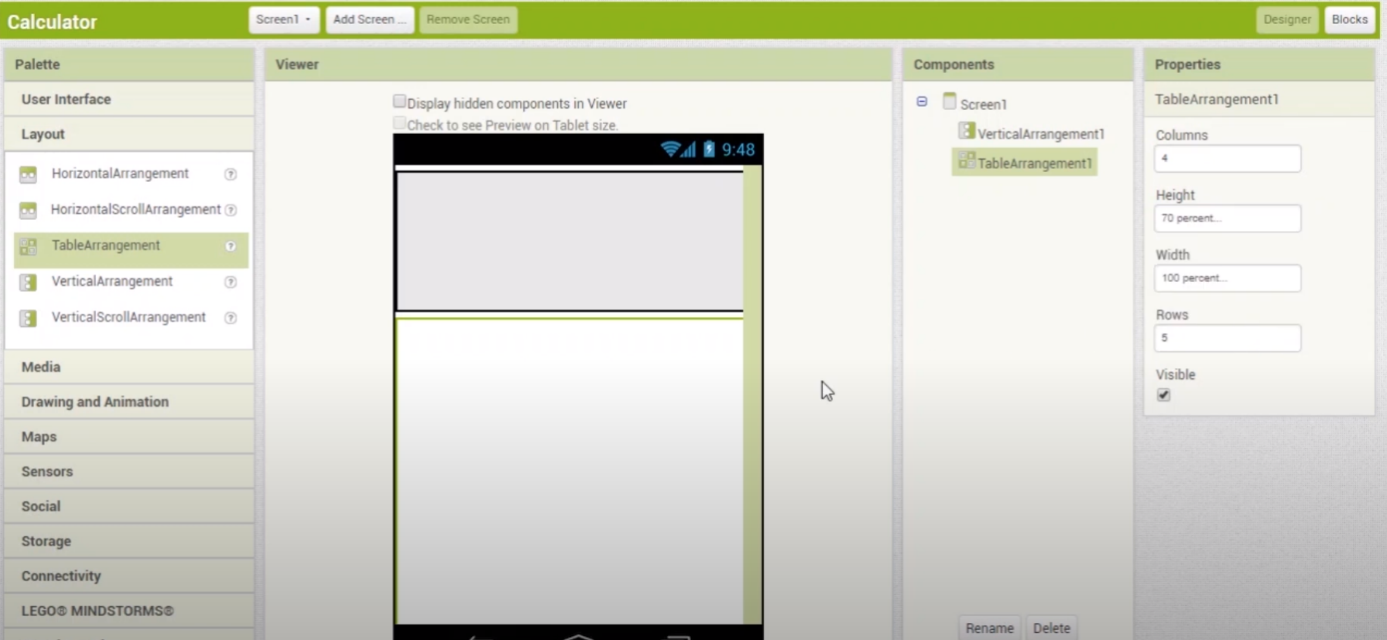
Kialakítom a képernyő tartalmát. A jobb oldali részben először az összetevőt kell kiválasztanom *(Components)*, utána a komponens tulajdonságait állítom be *(Properties)*. Először a kezdőképernyő *(Screen1)* igazítását adtam meg, ezen később beállíthatom a háttérszínt. Az első képernyő kialakításához a bal oldali elemek közül szükség volt egy „Függőleges Elrendezés” elemre, ez lesz a számológép kijelzője. Behúztam ezeket a megfelelő sorrendben a képernyőre (Screen1). A leendő kijelző elrendezése a *screen1*-en vertikálisan: balra, horizontálisan: fel állapotban van. A számológép-kijelző szélességben 100%-a, magasságban 30%-a kell legyen a később kialakítandó gombsorhoz képest. Természetesen reszponzív formában megadva (a többi összetevővel együtt) mobil alkalmazásról lévén szó.



*3. ábra. A számológép-kijelző kialakítása (felső rész)*

## **2.3 Gombsor létrehozása**

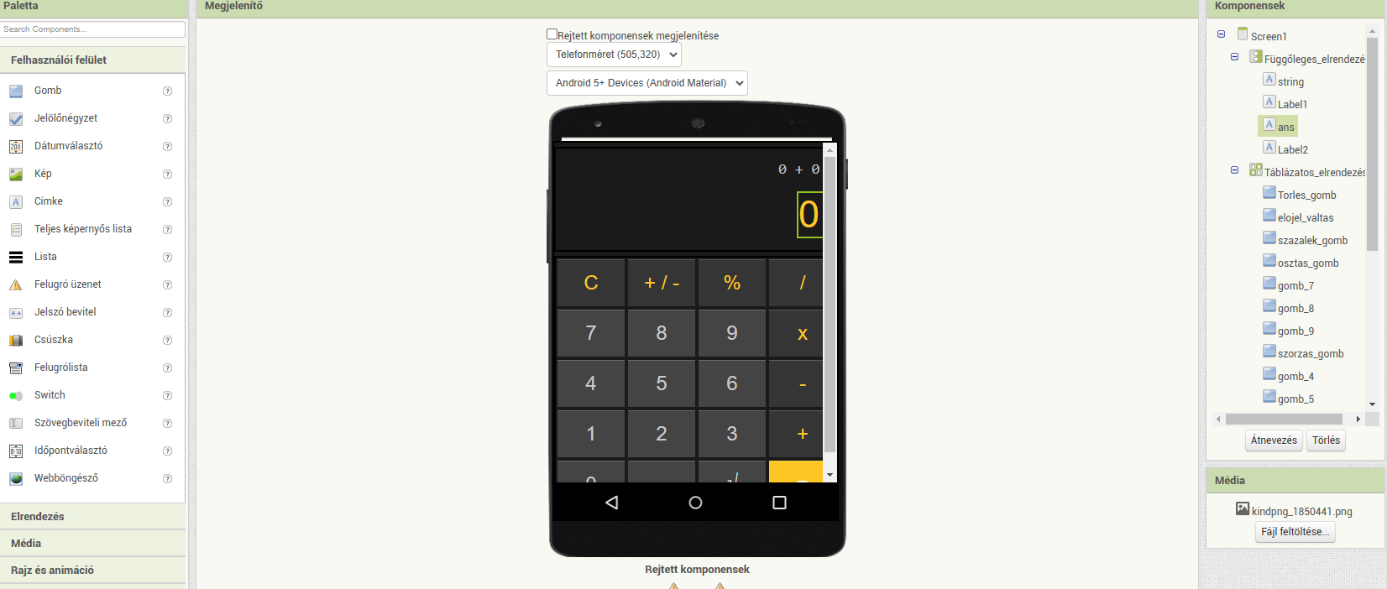
Ezek után létrehoztam a kijelző alatti területen egy „táblázatos elrendezés” -t a számológép gombjainak helyt adva. A tabella vertikálisan 70%-ra (kijelző volt 30%) kellett hogy nyúljon, szélességben szintén 100% volt az értéke. Az oszlopok száma: 4, a sorok száma: 5 lett, a minta alapján.



*4. ábra. A számológép-gombsor-táblaelrendezés kialakítása (alsó rész)*

A kijelzőn folytatva a munkafolyamatot címkéket helyeztem el, függőlegesen jobbra, vízszintesen középre zárt formában, sorközökkel ellátva. Ezek fogják a műveletet (felső sor) és az eredményt (alsó sor) a kijelzőn kijelezni. Az eredmény mezőt 50-es betűmérettel, a jobb olvashatóságért. Beállításra került a kijelző háttérszíne, illetve a kijelzőn megjelenő későbbi sztring adatok háttérszíne a „Tulajdonságok” menü alatt.

Elérkezett az idő a gombsor kialakítására is. 4 x 5 -ös táblázatba lett beágyazva a 20 db gomb a „Felhasználói felület” menüből drag & drop megoldással. Szélességnek 25% lett oszloponként megadva, magasságban pedig 14% az 5 sornak. A gombok elnevezésében törekedtem arra, hogy a kódolásnál egyértelmű legyen a behelyettesítés, kivételesen magyar nyelven. (angol a hivatalos😊). A gombsor kialakítás után a gombok háttérszínének és „FontColor”-jának, azaz betűszínének beállítása, megadása volt hátra.



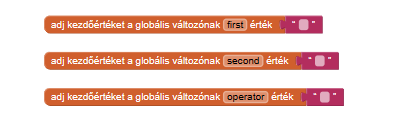
*5. ábra. A számológép-gombsor háttérszín-betűszín beállítása*

# 3. A programkód

A MIT App Inventor kezelőfelület „Parancsok” nézetében folytattam tovább a munkát a gombsorhoz tartozó kód megalkotásával, működésre késztetve a számológépet.

## **3.1 Globális változók**

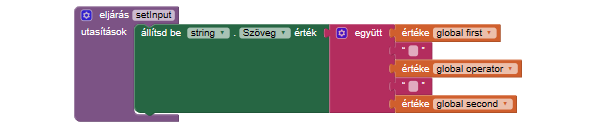
Definiáltam három darab globális változót, üres szöveges (String) értékkel inicializálva. Ezek a változók fogják a mindig aktuális begépelt értéket tárolni, majd egy eljárással kiírni, megjeleníteni a kis kalkulátoron. Az operandus első fele a „first”, a második fele „second”, a műveleti jel pedig az „operator” nevet kapta. Mivel csak két változón történik egyszerre művelet, ez picit be is határolja a ’Calculator’ működését, de a kód későbbi részében találtam rá megoldást, hogy egy számítás tovább fűzhető, adoptálható legyen, további műveletekkel.



*6. ábra: Globális változók*

## **3.2 A Metódus**

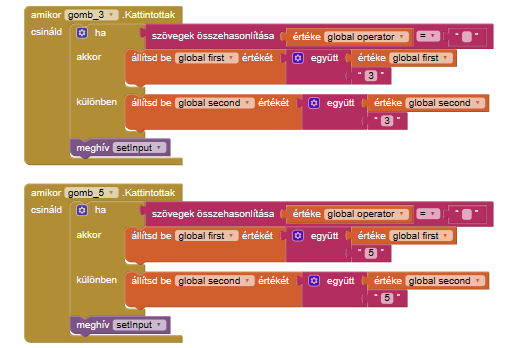
Neve: setinput, egészen egyszerűen a szövegek összefűzéséért, a képi megjelenítésért felel, kiírja a „string” nevezetű cimkémre a számológép kijelzőjének felső kisebb méretű sorába az éppen begépelt matematikai példát. Ehhez ez az eljárás összefűzött string szöveget hoz létre, szóközökkel elválasztva a 3. bekezdésben tárgyalt globális változók aktuális értékeit veszi át, jeleníti meg.



*7. ábra: A ’setinput’ metódus*

## **3.3 A nyomógombok**

Először a numerikus billentyűzet kódolása történt ugyanis ez a legegyszerűbb. 1-től 9-ig történt a szám gombok megadása. Ha adott számra kattintanak, akkor a kód tegye (do parancs), mégpedig abban az esetben, ha a felhasználó még nem adott meg operátor jelet (műv.jelet), akkor állítsa be a megadott számjegyet (v.számjegyeket) az első (first) helyre. Különben ha van műveleti jel, akkor a második (second) helyére íródjon be a számjegy. A kódrészlet végén mindegyiken ott van a ’setinput’ eljárásom, a kijelzőre kiíratás érdekében.



*8. ábra: 3-as és 5-ös gombok kódjai*

## **3.4 A float\_gomb**

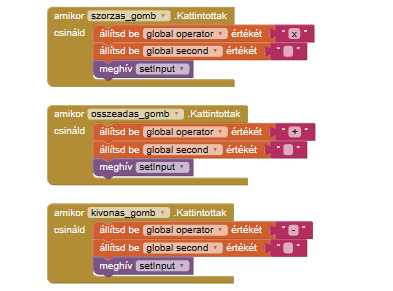
Kellett a számológéphez egy úgynevezett „tizedesvessző” billentyű is, amivel a nem egész, azaz valós számok (float v. double) műveleteit tudjuk a segítségével elvégezni, megjeleníteni. Szisztémájában ugyanaz, mint a számbillentyűk kódja. A változóba pontot tesz az utolsó megadott szám után, ami után természetesen újabb számok megadása következhet. (pl.: 3.14) A végén meghívjuk a ’setinput’ metódust, hogy egy szöveges formában összefűzi és a kijelzőnkön megjelenjen, mint double vagy float szám.



*8. ábra: A nem egész számok megadásához használt tizedesvessző gomb*

## **3.5 Műveleti jelek**

Öt darab művelet azonos apró programkódot kapott, ezek: az osztás, szorzás, összeadás, kivonás, és a százalék műveletek. Sima „do” parancs, ha az adott gombra kattintanak, állítsa be az ’operator’ globális változó értékét az adott műveleti jelnek megfelelően. A kiegészítés csak annyi, hogy a ’second’ változó helyét stornózza, majd jön a szokásos setinput eljárás hívás. Hogy miért kell előtte törölni a második változót? Azért, mert ha már nem az első műveletet végezzük, hanem pl.: a második szorzást írjuk be, akkor a már korábban beírt második változót nem viszi tovább. majd az egyenlőségjelnél látható, hogy ő az eredményt átadja az első (first) változónak, ergo az újabb művelet már tisztán végrehajtható egy újabb (second) változó megadásával, és így tovább-és így tovább, folytathatóak a műveletek bármeddig. (vagy amíg nem nyomjuk meg a törlés gombot)



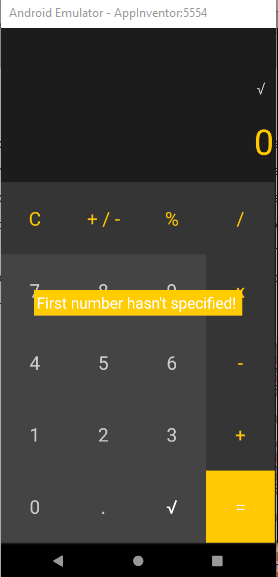
*9. ábra: Matematikai műveletek kódjai, itt összeadás-kivonás-szorzás.*

## **3.6 A gyökölő**

A gyök gombra való kattintáskor a folyamat némileg eltér a hagyományos operátorokétól. Először beállítja a gyök jelet operátorként, kiírja azt a ’setinput’ metódussal, ebben semmi meglepő nincs. Utána azonban jön egy kivételkezelés. Ugyanis itt nincs második változó, nincs egyenlőség-jel sem, a négyzetgyök gomb megnyomása után egyből kiírja az eredményt a kijelzőre string alakban (answer). a kivételkezelés annyiban merül ki, hogy amennyiben az első változó, tehát maga a szám hiányzik és a felhasználó egyből a gyök gombot nyomja le, egy hibaüzenetet generál ami figyelmeztet a helyes sorrendre. Abban az esetben, ha a sorrend jó, egy sima matematikai (négyzetgyök) műveletet futtat le a kód, azt egyből az „answer” szövegbe teszi (erről majd többet az egyenlőségjelnél), illetve eltárolja az eredményt első (first) változónak. Így, ha a felhasználó többször megnyomja a gyök gombot, mindannyiszor az éppen aktuálisan kiszámított eredmény gyökét is kiírja.



*10. ábra: A négyzetgyök művelet.*



*11. ábra: A hibakód, rossz begépelési sorrend esetén.*

## **3.7 Az előjelváltás**

Hasonlóan a már látott négyzetgyök gomb folyamathoz, abban az esetben, ha a felhasználó egyből az előjel váltás gombot nyomja le, egy hibaüzenetet generál, ami figyelmeztet a helyes sorrendre. Itt viszont lehet művelet is előjeles számokkal, tehát a hibaüzenetet ki kellett bővíteni arra az esetre is, ha a második változót nem adja meg a felhasználó, de közben előjelet meg adna hozzá (boolean segítségével). Ha a változó adott de nincs műveleti jel (azaz az első változó be van írva) akkor azon hajtja végre az előjel váltást, ha van műveleti jel, akkor a másodikon. Minden sikeres esetben ’setinput’ eljárás meghívása.



*12. ábra: Az előjelváltás folyamata, kódja*

## **3.8 A törlés (Clear) billentyű**

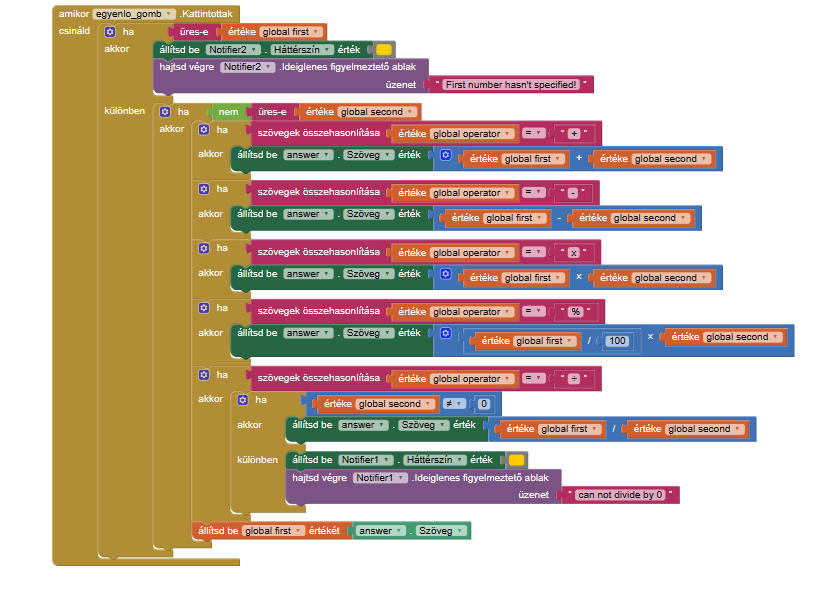
A törlés billentyű – hasonlóan a valódi számológép C billentyűjéhez – törli az addigi műveleteket, a kijelzőt alapállapotba (default) hozza, a felső címke eltűnik, üressé válik, az „answer” azaz ’válasz’ címke pedig 0-ra áll vissza. Ehhez a kód üres szöveg adatokkal tölti fel a „first”, „second” és „operator” változókat, illetve az „answer” értéket 0-ra állítja. A „setinput” eljárással írja ki mindezt.



*13. ábra: A törlés funkció kódja*

## **3.9 Az egyenlőség gomb**

A legtestesebb kódrészlet a programban. A kód elején rögtön, ha a felhasználó véletlenül műveleti jelet kezdett begépelni, és aztán megadott egy változót vagy sem, akkor hibaüzenetet dob a számológép, az értelmezhetetlen műveleti sorrend miatt. Különben, ha a „second” változó értéke NEM üres (!=), akkor már végrehajtható a művelet. Ennél a résznél jönnek tételesen a számítások matematikai kódolásai. Például, ha az „operator” változó „+” jel, akkor állítsd be az „answer” szövegértéket (kijelző nagy, eredmény számjegye) az első (first) és a második (second) változó összegének visszatérési értékével. A folyamat ugyanaz a kivonás, összeadás, szorzás, százalékolás esetében, ez utóbbi esetében az App Inventor matematikai készletében azonban nincs külön százalékszámítás operandus, így a klasszikus „osztom százzal az első változót és megszorzom a második változóval” számítás marad. A matematikai osztás az, ami picit bővebb lett, itt ugyanis hibaüzenetet kellett generálni a 0-val történő osztás esetén, ami ugye nem értelmezhető. A tényleges műveleti kód legvégén pedig az „answer” cimkébe történő eredmény kiírás található, hogy a felhasználó láthassa az eredményt. Pont ez miatt, mivel az egyenlőség gomb már csak az „answer” label-t befolyásolja, itt már nem kellett a „setinput” eljárást meghívnom. A kód:



*14. ábra: Az „egyenlo\_gomb” működése*

Biztos vagyok benne, hogy több eljárással adott esetben tovább egyszerűsíthető lenne a kód, illetve további fejlesztéssel és adaptációval akár „haladó” tudományos géppé is fejleszthető, sin, cos, tan, rad, log, bin és egyéb funkciókkal kiegészítve. Alapnak viszont tökéletes😊.

**4. Összegzés**

A MIT App Inventor segítségével a program .apk fájllá alakítható, letölthető, ikon adható hozzá, akár a Google Play-en is publikálható, illetve rátölthető az arra alkalmas operációs rendszert futtató okostelefonra, mint alkalmazás. Én is így tettem, egy SAMSUNG Galaxy A40-es szolgált tesztalanyként. A program futása az emulátorhoz képest is sokkal jobb volt, teljesen zökkenőmentes, hibamentes.

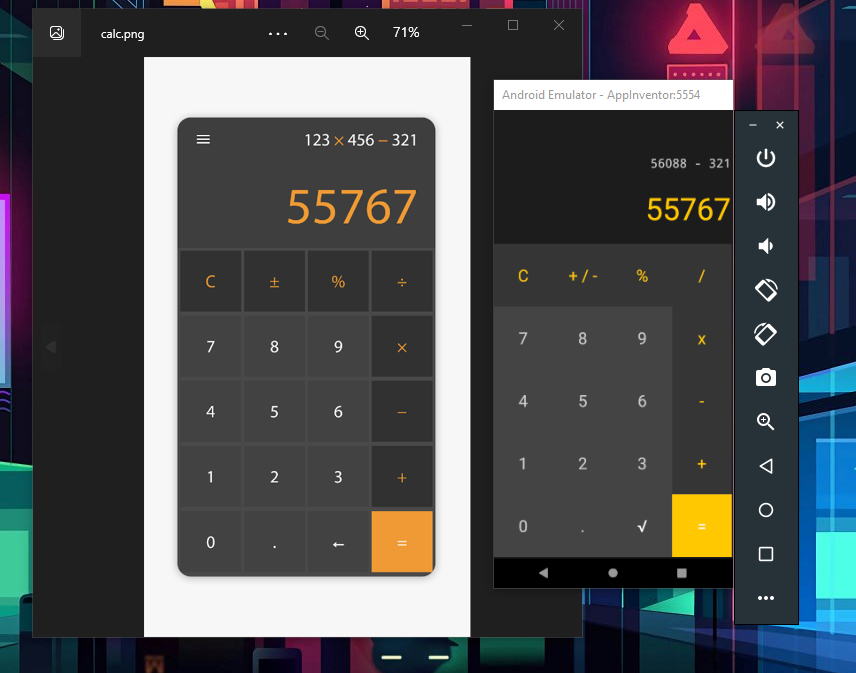






*15. ábra: A Mobil verzió 1.0 és működése Galaxy A40-en, és a program ikonja*

Az App Inventor alkalmazás egésze, rendszere, blokkjai nagyon hasonlóak a Scratch alkalmazás kezelőfelületéhez, tudásához, játékosságához. Jól átlátható felülete, gyors programozást és látványos alkalmazások elkészítését teszik lehetővé, a GitHub-on fellelhető számtalan mennyiségű beépülő modul pedig jól, sok helyen használhatóvá teszi. Ennek fényében a végén talán nem is olyan szembeötlő a különbség a „megálmodott”, a fotón látott számológép és az általam létrehozott Calculator között.



*15. ábra: Az eredmény*

FELHASZNÁLT IRODALOM & KÉPEK

* Google Android [Android - Secure & Reliable Mobile Operating System](https://www.android.com/)
* MIT App Inventor alkalmazás [MIT App Inventor](http://ai2.appinventor.mit.edu/?locale=hu#5239951315566592)
* Scratch alkalmazás [Scratch - Imagine, Program, Share (mit.edu)](https://scratch.mit.edu/)
* Főcím kép https://images.app.goo.gl/jofd4MA7zJebe3vt6
* 1. kép https://images.app.goo.gl/nLqgHrT5YMwTzcAe8