# Útjelző táblák automatikus grafikai tervezése: a SignpostPrinter program használata

(Molnár András J., 2015. október)

Program verzió: 1.0.

A SignpostPrinter JAVA alapú program arra ad lehetőséget, hogy előzetesen már tartalmilag megtervezett és excelbe bevitt útjelző táblatervekből automatikusan készítsünk nyomtatható táblagrafikákat PDF formátumban.

## Áttekintés - Az automatizált grafikai tervezés folyamata

Az alábbi folyamat lépéseit kell végrehajtani a helyes működtetéshez. Ezek mindegyikét és a különböző lehetőségeket részletesen kifejtjük lejjebb, külön fejezetekben:

1. A program telepítése (kibontása és bemásolása, 1x végzendő el).
2. Az alkalmazott piktogramok könyvtárának áttekintése és szükség szerinti bővítése.
3. A tervek összeállítása erre a célra készült excel sablonban:
   1. A sablon támogatja az adatok áthozását egyszerre egy vagy több útvonaltáblázat-táblatervező excelből [ld. külön dokumentáció] is, ugyanakkor annak használata nem kötelező
   2. Az adatokat kézzel is meg lehet adni ill. be lehet másolni az excelbe. Egyes adatok szövegként jelennek meg a táblán, mások a program számára jelentenek utasítást, pl. egy adott helyre egy piktogramot vagy elválasztó vonalat helyezzen be.
   3. Különböző méretű-paraméterű táblákat külön excelben adjunk meg és külön is futtassuk a generálást.
   4. A tervek összeállítása után, ha azokban excel-képletek is szerepelnek, le kell cserélni a képleteket az értékekre, mivel a program a képleteket nem értelmezi.
4. A tervezendő táblák paramétereinek megadása a megfelelő tábla-paraméterfájl kiválasztásával vagy elkészítésével (tábla méretek, betűméret, keret, színek stb).
5. A program futtatásához a bemeneti-kimeneti és a piktogramkönyvtár fájlnevek, valamint a PDF lapméret és hasonló paraméterek megadása egy fájlban. Ez lesz a konfigurációs fájl.
6. A program futtatása a konfigurációs fájllal és a generált eredmények visszacsatolása (szükség szerinti módosítások a bemeneti adatokon és újrafuttatás[ok], a program hibajelzései és a kimeneti eredmények függvényében).

Egyes, sajátos grafikájú ill. elrendezésű táblák grafikai terveit, melyek generálása túlmutat a program sablon-lehetőségein, külön, kézzel érdemes elkészíteni.

## A program kibontása, telepítése

A SignpostPrinter program maga nem igényel különösebb telepítést, a fájlokat kicsomagolva másoljuk be egy tetszőleges mappába. A gépen kell viszont legyen JAVA telepítve (legalább 1.8-as verzió ajánlott, a JRE csomag telepítendő, pl. innen ki lehet választani a gépünkhöz megfelelőt: <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>).

## Piktogramkönyvtár

Az alkalmazható piktogramok kódjait és a nekik megfelelő képfájl-neveket a apicto\_dict.txt szöveges szótárfájl tartalmazza (ennek abban a mappában a helye, ahonnan futtatjuk majd a SignpostPrinter programot). A szótárfájlt a Windows Jegyzettömbbel szerkeszthetjük. Minden sorában egy piktogram leírása szerepel: annak kódja majd utána szóközökkel vagy tabulátorokkal elválasztva a fájl elérése és neve, pl.:

[vasút] picto/vasut.pdf

A fenti sor azt jelenti, hogy a táblatervek szövegében szereplő „[vasút]” hivatkozásokat cserélje le a picto/vasut.pdf-ben lévő piktogrammal. A szótárfájl szabadon bővíthető vagy módosítható. A „picto/” előtag azt jelzi, hogy a vasut.pdf fájl a picto nevű almappában van (a program futtatási mappáján belül). Ide érdemes bemásolni az új piktogramokat, amiket hozzá szeretnénk adni, és mindegyikhez egy-egy új sort beírni a picto\_dict.txt fájlba, amivel a meglévőekhez hasonlóan megadjuk, hogy milyen […]-néven akarunk rájuk hivatkozni a táblatervekben (ez a piktogramnév mint szöveg mindenütt le lesz cserélve magára a piktogram képre, azaz be lesz szúrva a táblára a piktogramnév helyére).

A piktogramnév tartalmazhat kis és nagybetűket, számokat, speciális karaktereket, szóközöket is. A bezáró ] jel jelzi a név végét. A kis- és nagybetűk különbözőnek számítanak.

A program a vektoros grafikákat pdf formátumban tudja fogadni, míg a rasztereseket jpg vagy png formátumban. A képek mérete (szélesség x magasság) tetszőleges, azokat automatikusan méretezi majd a táblaméretekhez viszonyítva.

A # jellel kezdődő sorok megjegyzésnek számítanak, ezt és az üres sorokat lehet használni a szótár tagolásához, nagyobb egységek elnevezéséhez (pl. „##### turistajelek #####”, stb).

A piktogramokat bármilyen programmal szerkeszthetjük. Nem minden típusú PDF-et tud a program kezelni, mindig teszteljük a piktogramot mintatábla generálásával. Ha nem sikerül így vektorosként kezelni egy piktogramot, akkor az mindig járható, hogy nagyfelbontású raszteressé (JPG) konvertáljuk és úgy tesszük be. Ajánlott szerkesztő program pl. az InkScape, amely ingyenesen letölthető ( <https://inkscape.org/en/download/windows/> ).

Figyeljünk oda a piktogram átlátszóságára: többnyire nem téglalap alakúak a jelképek, de a képfájl mindig az, ezért a képnek egyes részeit átlátszóként kell megadnunk a szerkesztőben, különben a táblán „fehér sarkokkal” fog megjelenni.

## Táblatervek áthozása útvonaltáblázat-táblatervező excelből

Amennyiben az adatokat egy útvonaltáblázat-táblatervező excelből akarjuk áthozni, akkor járjunk el az alábbiak szerint, ha viszont kézzel akarjuk megadni, akkor ugorjunk a „Táblatervek bemeneti excel-sablonja” c. részre.

Válasszuk ki a forrásadatainkat tartalmazó útvonaltáblázat-táblatervező excelfájlt, és nyissuk meg az excelben.

Nyissuk meg a SignpostPrinter-hez tartozó üres bemeneti excelfájlt (ures\_tabla\_export\_urlap\_v1.1.xlsx) és mentsük el saját néven, ami utal a táblázásunkra (pl. Bakony\_tabla\_export\_urlap\_v1.1.xlsx).

Ez utóbbiban minden tábla egy excel sornak fog megfelelni. Ezért soronként ki kell töltenünk a B, C, E és F oszlopokat az egyes táblák kiválasztásához és adataik áthozásához:

* a B oszlopba a forrásadatokat tartalmazó (útvonaltáblázat-táblatervező excel) fájl nevét írjuk be, szögeletes zárójelbe téve, a végén a fájltípust(kiterjesztés) nem lefelejtve, pl.: [Bakony\_utvonaltabl\_reszl\_urlap\_v6.11.xls]
* a C oszlopba annak a tábla űrlapnak a sorszámát írjuk be, ahonnan az aktuális táblát áthozni akarjuk (pl. ha a fenti fájlban a „Tábla űrlap (2)” nevű lapon lévő táblákból akarunk választani, akkor írjunk ide 2-t),
* az E oszlopba írjuk be a tábla típusát ill. irányát: H – helymegjelölő, azaz nyíl nélküli, B – balra nyíl, J – jobbra nyíl,
* az F oszlopba pedig a tábla űrlapon belüli táblaazonosítót (pl. b1., a2. - ami a tábla mellett található, tipikusan a nyílrész fölötti cellában).

Ha ezek alapján az excel azonosítani tudja a táblát (fájl-űrlap-irány-azonosító szerint), akkor a tábla adatok megjelennek a tábla sorában a megfelelő pozíciókon. Ezt minden táblára külön kell megtenni, persze másolással lehet gyorsítani. Egy tervfájlba több forrásfájlból is hozhatunk adatokat, mivel minden táblára külön adjuk meg a fájlnevet is, sőt akár az áthozott adatokat lehet később bővíteni-kombinálni olyan táblatervekkel is, amelyekhez nem tartozik útvonaltáblázat-táblatervező excel, hanem csak itt adtuk meg közvetlenül, kézzel.

Bizonyos elemeket általában nem tudunk áthozni az útvonaltáblázat-táblatervező excelből, ezért kézzel kell itt megadnunk, vagy legalábbis finomhangolnunk - ilyenek pl. a sorok banner-formázásai (kiemelt színes csíkokként, a szín megadásával), egyes esetekben a lábléc szövegek és a sor-megjegyzések (kisbetűs kommentek), de ilyenek lehetnek maguk a célpont kiírások is: ami túl hosszú, pl. rövidíteni kell, vagy egyes részeket sorkommentbe tenni.

Érdemes azt tenni, hogy az adatok áthozása után mentsük el a fájlt így képletekkel, hogy az útvonaltáblázat-táblatervező esetleges későbbi módosításakor kaphassunk egy automatikusan számolt új táblázatot, viszont új néven (\_MOD név-végződéssel) mentsük el és abban dolgozzunk tovább úgy, hogy a képleteket átalakítjuk értékekké (ld. az erre vonatkozó fejezetet alább). Így szabadon tudunk továbblépni a táblatervek finomhangolásával (ld. következő fejezet), függetlenítve az eredeti értékektől, de szükség szerint visszanyúlva oda, ha később maga az útvonaltáblázat-táblatervezőben kell módosítani.

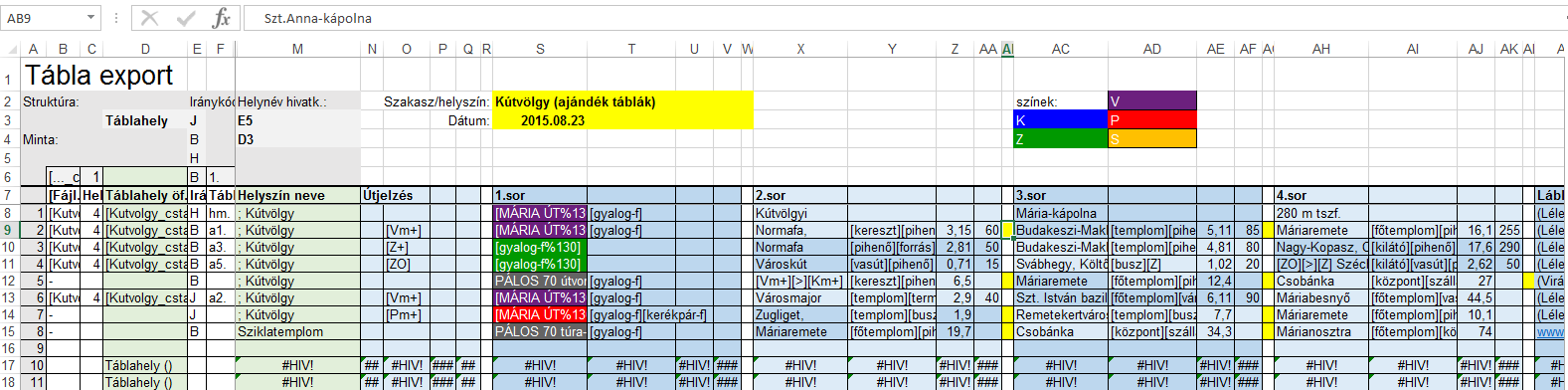
## Táblatervek bemeneti excel-sablonja

A táblatervező excelben az alábbiak szerint tudjuk megadni ill. finomhangolni a táblaterveinket. Ld. ehhez a minta kimenetet és a paraméterezésnél lévő mintatábla ábrát is (külön fejezetekben a futtatásról és paraméterezésről). Érdemes egy példafájlt venni alapul, amit új néven mentünk el és abban szabadon dolgozhatunk, a benne lévő adatok átírásával; vagy vegyük az üres sablont és mentsük el saját néven.

Az alábbiak referenciaként és magyarázatként tekinthetők.

Általános elvek

* Minden sor, aminek az A oszlopában 0-nál nagyobb egész szám van, vagy ’+1’ végű képlet, valamint az E oszlopban érvényes táblatípust adunk meg (H – helymegjelölő, azaz nyíl nélküli, B – balra nyíl, J – jobbra nyíl), alapból egy táblának számít és a táblára kerülő feliratok, piktogram-hivatkozások és adatok a sor megfelelő celláiban szerepelnek, ezeket olvassa ki a program és illeszti be a kimeneti táblasémába (ld. lent az egyes oszlopok leírását).
* Az excel beépített színeit használhatjuk saját jelölésekhez a tervben, ezeket a színeket nem veszi figyelembe a program. Ha úgy akarunk színt megadni, hogy figyelembe vegye a program, akkor mindig a „további színek”-ből válasszunk (csak bizonyos celláknál, mind a háttér mind a betűszínre vonatkozóan), vagy formátummásolóval másoljuk a „színek:” résznél előre e célból megformázott mintacellákról.
* A piktogramokat ne képként illesszük be a tervbe, hanem piktogramnévvel hivatkozzunk rájuk, szögletes zárójelek között, pl. [Vm+] mint a lila Mária út jelzés, [vasút] mint a vonat piktogram, stb. Minden alkalommal, ahol egy piktogramnév szerepel szögletes zárójelben, a program a kész táblán a nevet magára a piktogram képére cseréli. A nagy- és kisbetűk itt számítanak, nem mindegy hogyan írjuk!
* Piktogram-hivatkozások szerepelhetnek útjelzés mezőben, célpont nevében (pl. „Kutya-hegyi [S][K+] elágazás” – itt hivatkozhatunk útjelzésváltásokra is pl.: „[S][>][K+] Kutya-hegy”, ha ahhoz a S jelzésű útról a K+-re kell áttérni), sőt akár a km és idő szövegrészben is, de leginkább a célponthoz tartozó piktogram mezőben, ahol jellemzően egymás mellett többet is megadhatunk az ott található turisztikai jellemzőkből (pl. [vasút][központ][vár] stb). Nem szerepelhet viszont piktogram hivatkozás sorkommentben (kisbetűs megjegyzés közvetlenül valamely táblasor alatt) vagy lábléc szövegben.
* Minden piktogram automatikusan méreteződik abba a mezőbe ahová kerül. Ha ez a méret nem megfelelő, akkor lehet %-osan nagyíttatni vagy kicsinyíttetni, úgy hogy a piktogram hivatkozásban a záró szögletes zárójel előtt % jel után adjuk meg a kért méretezés % arányát, pl.: [MÁRIA ÚT%130] azt jelenti, hogy a „MÁRIA ÚT” nevű, azaz a [MÁRIA ÚT] piktogramot az automatikus méretezéshez képest 130%-ra növelje. Az eredményt ellenőrizni kell, a program nem ad hibaüzenetet, ha pl. egy nagyított piktogram „kilóg” és átfed mással.
* A táv km-eket megadhatjuk számmal vagy szöveggel, odaírhatjuk a végére a „km”-t vagy el is hagyhatjuk (utóbbi esetben a program kiegészíti).
* A menetidő adatot mindig percben adjuk meg, számként (célszerűen 5 percre kerekítve 15-20 perc fölött), a program automatikusan óó:pp (óra:perc) formátumúra alakítja.
* A többi mező kötelezően szöveges típusú, ha tehát valamely rendkívüli okból számot adnánk meg, akkor írjunk az elejére egy ’ jelet, s akkor az excel szövegként értelmezi (különben nem fogja elfogadni a program).
* Ha a sorkomment helyére egy | jelet teszünk, akkor sorkomment helyett az adott sor alá egy vízszintes útvonal-elválasztó vonal kerül; a táblaformátum paraméterfájlban megadott vastagság szerint (paraméterekről ld. külön fejezetet).
* Ha az utolsó sorban adunk meg sorkommentet, akkor az lehet hogy összeér a lábléc szöveggel, ezért ilyenkor érdemes a lábléc bal oldali szövegét elhagyni.
* Éljünk a másolás és a formátummásolás lehetőségével, valamint alkalmazhatunk képleteket is, ha végül azokat értékekre cseréljük.

Az egyes oszlopok (cellák) szerepe:

* Az A oszlopban 0-nál nagyobb egész számként, vagy ’+1’ végű képletként a tábla sorszáma szerepel (a generáláskor a program üzeneteiben ezek alapján azonosíthatjuk be a táblát).
* A B-C-D és F oszlopok csak az adatok áthozására szolgálnak (ld. külön fejezet), a program maga ezeket nem veszi figyelembe.
* Az M és F oszlopokban megadhatjuk a tábla helyének nevét és azon belüli sorszámát, de ezeket sem veszi figyelembe a program (a láblécben viszont hivatkozhatunk rájuk képlettel, amit később lecserélünk az értékre, a lecserlélést ld. külön fejezetben).
* Az E oszlopban a tábla típusát ill. irányát kell megadni: H a nyíl nélküli helymegjelölő táblákhoz, J a jobbra, B a balra mutató irányjelző táblákhoz. Bármi más szerepel itt, a program hibát jelez és kihagyja a sort.
* Az O-P-Q oszlopokban külön-külön legfejlebb 1-1 útjelzés-piktogramot adhatunk meg, melyek a nyílrészben (ha van) egymás alatt jelennek meg. Ide egyéb szöveget nem írhatunk és egy cellába több piktogram-hivatkozást sem tehetünk. Az esetek többségében csak egy útjelzés van, azt az O oszlopba írjuk be.
* Ezek után az S oszloptól kezdve, 5-5 oszlopban adhatók meg a táblasorok adatai, amik függőlegesen egymás alatt lesznek a táblákon (4 soros táblák esetén az oszlopok jelentése: S-W 1.sor, X-AB 2.sor, AC-AG 3.sor, AH-AL 4.sor adatai, rendre ugyanabban a struktúrában).
  + Az első cellában (S, X, AC, AH) a célpont neve, aminek a színét ha egyénire állítjuk (háttér és betű), akkor az egész sor kiemelt bannercsíkot kap. Ezt az út nevének kiemelésére használhatjuk a legtöbb esetben, többnyire a tábla 1. soránál, de elvben bármelyiknél.

1. ábra: Bemeneti táblaterv excel minta

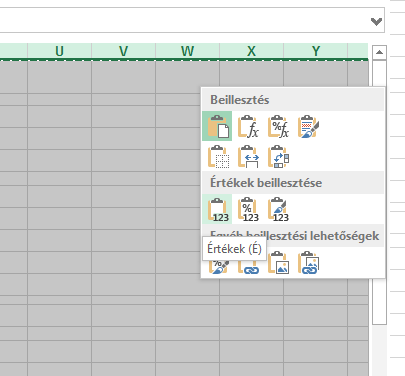
* + A második cellában (T, …) mindig a piktogramok amiket a célpont után akarunk megjeleníteni, tagolva (a feltüntetett célpont turisztikai jellemzői).
  + A harmadik cellában (U, …) a hozzá tartozó táv km-ben, javasoltan 1 tizedesre kerekítve. Szövegként is megadható, „km” utótaggal, de nem kötelező.
  + A negyedik cellában (V, …) pedig a menetidő szerepel, számként, percben, amit a program óó:pp formára hoz majd a generáláskor. Figyelem! Ha a km és a menetidő is elmarad, akkor a piktogramok becsúsznak a jobb szélre!
  + Az ötödik cellában pedig a sorkomment, amit kisbetűvel kell a célpont neve alá írni. Ha beljebb akarjuk kezdeni a feliratot (többnyire javasolt), akkor kezdjük szóközökkel. Mivel ez egy keskeny cella, s így nem látszik, hogy van-e benne szöveg, javasolt az ilyen kitöltött cellákat sárga háttérszínnel ellátni (az itt megadott szín nem fog megjelenni a táblákon). A sorkomment helyére | jelet is tehetünk, az a vízszintes elválasztó vonalat jeletni a sor alatt.
* A tábla lábléc bal és jobb oldali szövegét a AM, AN oszlopokban adhatjuk meg (négysoros táblák esetén).

## Excel képletek lecserélése értékekre

Ha az excelben képletek vannak (pl. egy szám valami másból számolódjon, vagy egy szöveg valahonnan máshonnan kerüljön át automatikusan), azokat nem értelmezi a program. Ez azért alattomos hibalehetőség, mert excelben nézve nem tűnik fel rögtön, hogy képletként van-e valami megadva: a cellákban általában nem a képleteket, hanem az általuk szolgáltatott eredmény adatot látjuk. Ezért minden esetben, ha elkészültünk a táblatervek szerkesztésével, tegyük meg azt a lépést, hogy lecseréljük az excel képleteket az értékekre.

Ezután viszont már a cellák függetlenednek a forrás értékektől, ha voltak ilyenek, szóval érdemes előtte elmenteni és új néven folytatni az átalakítást, mert lehet hogy egy későbbi módosításkor jó lenne ha a képletek automatikusan újra számolódnak. Tipikusan a \_MOD fájlnév-végződést érdemes adni az értékekre konvertált excelfájlnak, megőrizve az eredeti képletest is.

Az átalakítás menete: Jelöljük ki az összes cellát [Ctrl+A], majd másolás és beillesztés ugyanoda, de már nem képletként, hanem értékként [Ctrl+C, majd Ctrl+V, és itt a felugró kis vágólapos tool-t (Ctrl) lenyitva válasszuk ki az Értékek beillesztése közül az elsőt (123 - Értékek)]:



. ábra: Képletek cseréje értékekre Excelben

## Táblaparaméterek: a táblák kinézete

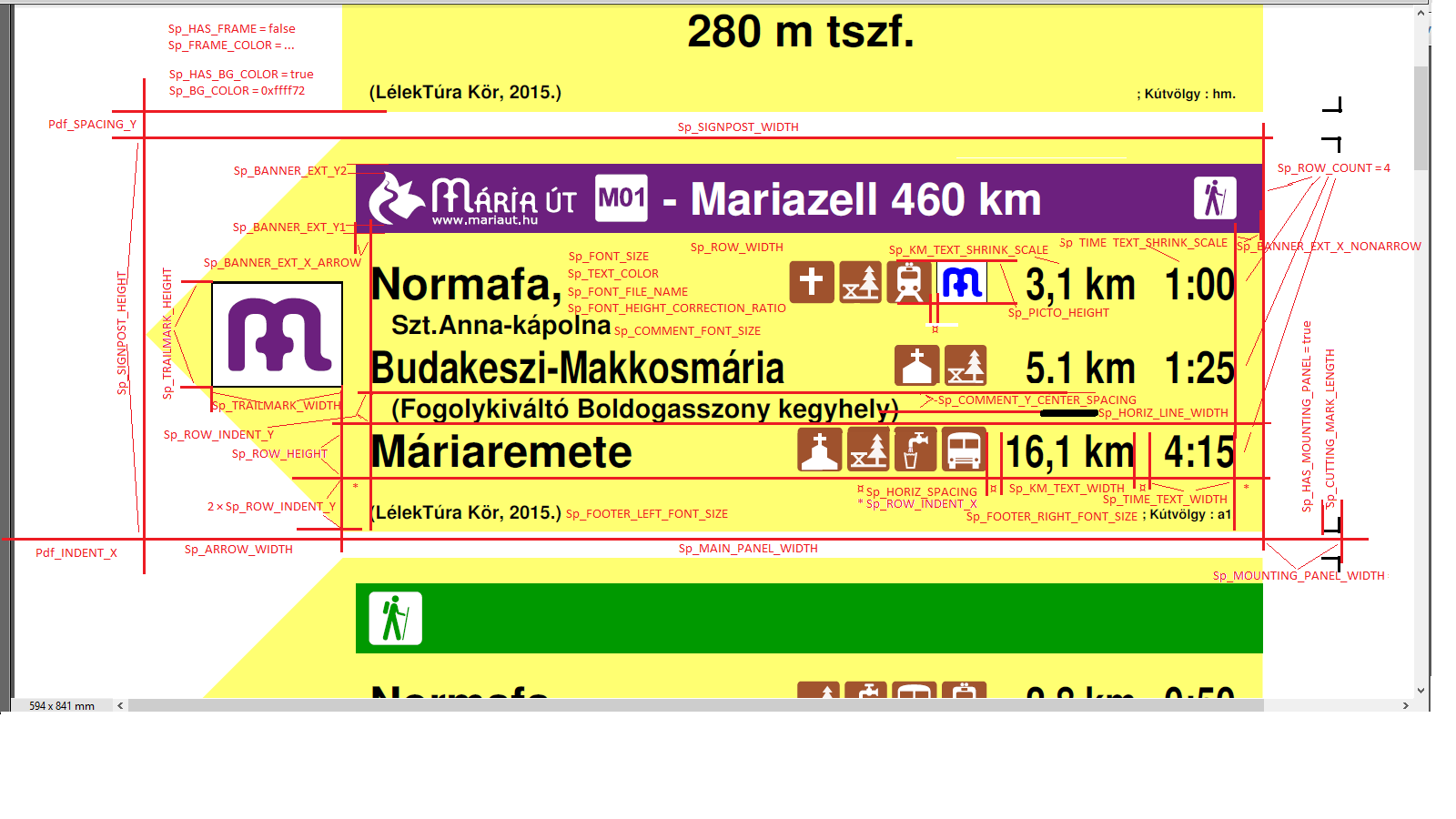
A táblák kinézetét leíró paramétereket külön szöveges fájlban adhatjuk meg, melyet szintén a Jegyzettömbbel szerkeszthetünk. A kiterjesztés (fájltípus) ajánlottan: „.properties”. A fájlnév javasoltan signpost-tal kezdődik. A programhoz eleve tartozik néhány ilyen táblaleíró fájl, ezek közül is választhatunk, de sajátot is tervezhetünk:

* signpost425x150TransparentBlackFrame.properties : 425x150, háttér nélkül (átlátszó), fekete keret, nincs margó
* signpost425x150YellowBlackFrame.properties : 425x150, sárga háttér, fekete keret, nincs margó
* signpost425x150YellowNoFrame.properties : 425x150, sárga háttér, keret nélkül, nincs margó
* signpost425x150YellowNoFrameMountingPanel.properties : 425x150, sárga háttér, keret nélkül, szerelési margóval és vágójellel

Ha saját táblaformátumot tervezünk, vegyük az egyiket mintának, másoljuk le, adjunk neki olyan nevet, amelyről azonosítani fogjuk tudni, és írjuk át a benne szereplő értékeket, az ott szereplő formátumban. Figyelem! Vannak paraméterek, amelyek egymással célszerűen összefüggnek, ezekre megjegyzések utalnak a mintafájlokban (pl. a tábla szélessége a nyílrész és a fő panel szélesség összege).

Az alábbi ábrán hozzávetőlegesen látható egy kész mintatáblán, hogy mely a táblaparaméter milyen méretet vagy egyéb tulajdonságot jelöl. A program rugalmassága eddig terjed; azaz ebbe a sémába illeszkedő grafikai megjelenést tudunk vele elérni:

* fő panel (main\_panel), melynek jobb vagy bal oldalán nyíl (arrow) lehet (ezek mérete megadható: signpost\_height, signpost\_width, …), benne 1-3 db turistajel lehet egymás alatt, lábléc jobb és bal oldali szöveg (footer) [hogy melyik táblán van nyíl és milyen irányba, azt a táblaterv excelben kell megadni];
* tetszőleges számú de fixen megadott db táblasor (signpost\_row, ajánlottan 2-5), mindegyik alatt lehet kiegészítő kisbetűs megjegyzés (comment).
  + A táblasorok bármelyike lehet kiemelt bannercsík,
  + a táblasorok mindegyikén négy mező: cél-név, piktogramok (picto\_text), táv (km\_text) és menetidő (time\_text). Ez utóbbi kettő egymás alá jobbra van igazítva, a cél neve maga balra. A program a hosszú cél-neveket automatikusan próbálja zsugorítani, a betűméret (font\_size) a magasságot jelenti, a táv-idő kiírás automatikusan zsugorított, az arány itt viszont paraméterként megadható (shrink\_scale) .
* Ezen kívül a margók és helyközök paraméterezhetők (pl. …horiz\_space, indent…), valamint
* egységesen a betű- és háttérszín (text\_color, bg\_color; has\_bg\_color: van-e háttér)
* a betűtípus (Sp\_FONT\_FILE\_NAME = fonts/FreeSansBold.ttf – ez a tipikus, de mást is használhatunk, csak másoljuk be a betűtípus fájlt a fonts almappába és adjuk meg itt)
* a keret (has\_frame, frame\_color),
* a vízszintes útvonal-elválasztó vonal vastagsága (ha alkalmazzák: horiz\_line\_width),
* továbbá lehet a tábláknak egy margó része (mounting\_panel), a felszereléshez, a nyíllal ellentétes oldalon.
* A nem nyíl táblák helymegjelölő táblának számítanak, azaz ezekre midig középre igazítva kerülnek fel a szövegek.



. ábra: A táblaparaméterek jelentései

## Konfigurációs fájl

A program futtatásához össze kell állítani, hogy miből mit készítsen, minek a segítségével, s az milyen legyen. Ezt egy másik paraméterfájlban, a konfigurációs fájlban adhatjuk meg, ami szintén szöveges és a Jegyzettömbbel szerkeszthető.

A konfigurációs fájl neve tipikusan „config”-gal kezdődik és a vége „.properties”. A programhoz találunk minta fájlokat, pl. config\_LeaderSajo.properties a sajómenti táblák előállítását írja le. A saját táblázásunkhoz célszerű egy ilyet lemásolni, átnevezni és átírni a magunk adataival.

A paraméterfájlban az alábbiakat kell megadnunk, tetszőleges sorrendben. A fájlok ha almappában vannak, írjuk be elé az almappa nevét egy ’/’ jellel elválasztva (pl.: ’signpost\_io/minta.xlsx’ a signpost\_io nevű mappában lévő minta.xlsx-re vonatkozik):

* A bemeneti excel fájl neve - ebből olvassuk az adatokat:

inputFileName = *…. excel fájl neve (végén xls vagy xlsx)*

* A kimeneti PDF fájl neve - ebbe nyomtatjuk a grafikákat:

outputFileName = *… pdf fájl neve*

* A tábla formátum paramétereket (méretek, színek, stb) tartalmazó fájl:

signpostPropFileName = *….properties fájl neve*

* A piktogramkönyvtár fájl - innen olvassuk a piktogramhivatkozásokat és a hozzájuk tartozó képfájlok nevét:  
  pictoDictFileName = *picto\_dict.txt (vagy saját könyvtárfájlt is megadhatunk helyette)*
* PDF oldalméret:   
  Pdf\_PAGE\_SIZE = *A0 vagy A1, ...*
* PDF vízszintes (bal) margó:

Pdf\_INDENT\_X = *50 vagy amit szeretnénk*

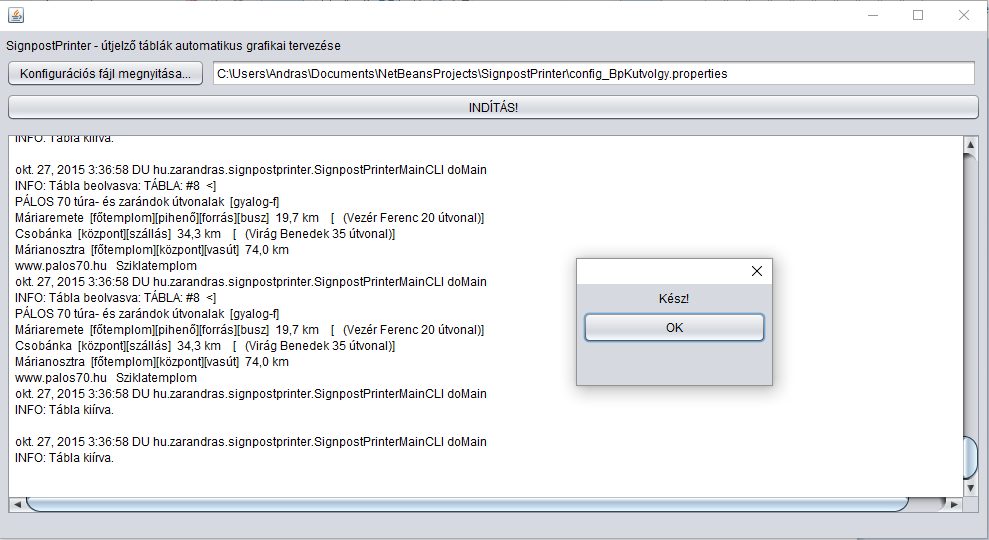
* Táblák közti függőleges térköz a PDF-ben:

Pdf\_SPACING\_Y = *10 vagy amit szeretnénk*

Ha mindez megvan, futtathatjuk a programot az alábbiak szerint.

## A SignpostPrinter program futtatása

A futtatáshoz duplán kattintsunk rá a Java (JAR) fájlra (SignpostPrinter-1.0-jar-with-dependencies.jar). Ekkor, ha a JAVA fel van telepítve a gépre, az alábbi ablak keret jelenik meg.



4. ábra: A SignpostPrinter program felülete

Itt kiválaszthatjuk az előző fejezet alapján megszerkesztett konfigurációs fájlt (beírhatjuk a nevét, vagy a „Konfigurációs fájl megnyitása” gombbal kiválaszthatjuk a fájlrendszerből).

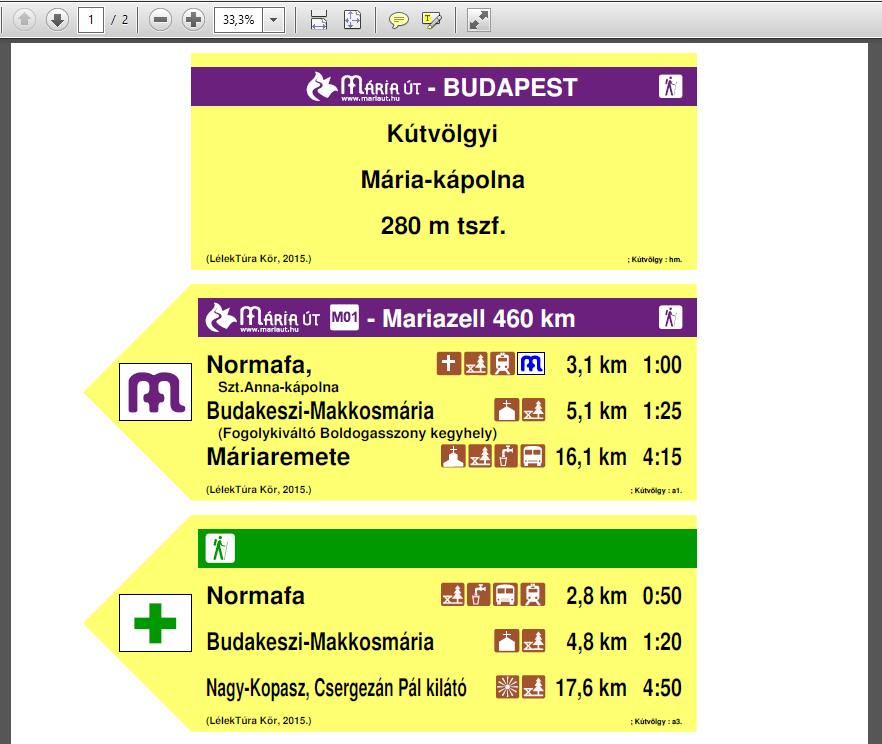
Ezután kattintsunk az INDÍTÁS! gombra és elindul a generálás. A program üzeneteit a nagy fehér mezőben láthatjuk futás közben, minden táblánál kapunk egy INFO üzenetet a tábla tartalmáról („Tábla: #...” – itt a tábla sorszáma szerepel ahogy az excel A oszlopobában megadtuk) és hogy „Tábla kiírva”. A hibajelzések is itt jelennek meg, ha valamelyik táblát pl. nem sikerült elkészíteni (ld. külön fejezetben).

A futás végén jelenik meg a Kész! feliratú kis ablak, vagy hibajelzéssel, ha programhiba miatt állt le. [Figyelem! Akkor is „Kész!” felirat jelenik meg ez az ablak, ha útközben valamelyik tábla hibás volt és azt kihagyta vagy félbehagyta, de ment tovább a generálás, ezért mindig érdemes áttekinteni a futás végén az üzeneteket, hogy volt-e közben generálási hiba.]

Fontos! Ha az eredmény PDF fájl már létezik a megadott néven, akkor újabb futtatás (felülíró generálás) előtt be kell csukni, különben a program nem tudja írni a PDF-et.

A készítendő PDF fájlt kérdezés nélkül felülírja a program, ha létezik. Ez azért van, mert az esetek többségében az eredmény többszöri finomítás után lesz végleges. Ha megvan, akkor érdemes emiatt elhelyezni vagy lemásolni a program könyvtárából (tipikusan a signpost\_io almappából).

A generálás eredmény PDF-je valahogy így kell kinézzen, a megadott tartalmi terv és paraméterek függvényében:



. ábra: Minta kimenet (PDF)

## A SignpostPrinter hibajelzései

A program hibakezelése exception alapú, azaz ha ilyet látunk a kimeneti üzenetekben, akkor valami hiba történt. Ettől még a program futhat tovább, ha pl. csak egy tábla terve hibás, a többi generálható lesz. Egy exception csak akkor állítja le a futást, ha a táblák generálását nem lehet elkezdeni vagy folytatni (pl. a piktogramkönyvtár hiányzik vagy hibás).

Tipikus hibajelzések:

* „A program futása hiba miatt leállt!”: olyan hiba történt, ami miatt nem folytatható a program futása (pl. az alábbiak közül valamelyik vagy más fatális probléma adódott).
  + „A piktogramkönyvtár fájl nem található vagy nem olvasható”: hiányzik, hibás, nincs megadva vagy nincs jó helyen a picto\_dict.txt (vagy helyette kiválasztott saját) piktogramkönyvtár leíró fájl. Ld. a Piktogramkönyvtár c. fejezetben. Ennek a program futtatási mappájában kell tipikusan lennie (ahol a jar fájl is van, amivel elindítjuk), hacsak nem adunk meg neki külön almappát.
  + „A megadott betűtípus nem tölthető be”: a táblaparaméterfájlban szereplő betűtípus állomány (Sp\_FONT\_FILE\_NAME) nem található vagy hibás. Ellenőrizzük, jól írtuk-e be a fájlnevet (tipikusan a „font/” mappa-előtaggal, ahogy a mintafájlokban van), s valóban ott van-e, megfelelő formátumú-e (alapvetően .ttf, azaz TrueType fontot használhatunk).
  + „Hiba a paraméterek beolvasása közben a … fájlból!”: a megadott táblaparaméter- vagy konfigurációs fájl nem található, nem olvasható vagy hibás, esetleg hiányos. Ha a fájlnév helyén a hibaüzenetben „(null)” szerepel, akkor nincs megadva táblaparaméter- vagy konfigurációs fájl. A konfigurációs fájlt a program indításakor kell kiválasztani, a táblaparaméter-fájl nevét viszont magában a konfig.fájlba kell beírni. Leggyakoribb eset, hogy elírjuk valamelyik fájl nevét, vagy nem a megfelelő helyen van (abban a mappában kell lennie, ahonnan a programot futtatjuk, azaz ahol a jar fájl van, amivel elindítjuk).
* „Tábla beolvasási/kiírási hiba!”: hiba van az útjelző táblával, ahol épp tart a program: nem olvasható be a terve (excel hiba), vagy nem írható ki (valamit rosszul adtunk meg az excelben, vagy a PDF fájl nem írható). A program futása nem szakad meg, de a hibás táblát kihagyja vagy félbehagyja. Ez a hibajelenség az alábbiak valamelyikével szokott együtt járni:
  + „Tábla formátum hiba:” az excelben hibásan adtunk meg valamit, erről bővebb információt is szokott adni a program. Ld. még a „Táblatervek bemeneti excel-sablonja” című fejezetet.
  + „x. excel sor kihagyva” (esetleg továbbá: „táblairány kód: …”): ha hibásan adunk meg valamit az excelben, akkor kihagyja a program. Ez a legtöbb esetben a táblairány kód hibás megadását jelenti, ami H, B, J lehet, attól függően, hogy nyíl nélküli, balra vagy jobbra mutató nyílról van szó.
  + „x. excel cella: A … nem szövegként van megadva!”: olyankor kapjuk ezt az üzenetet, ha szöveg helyett számot vagy képletet, vagy üres mezőt talál a program az excelben. Fontos, hogy minden mező a neki megfelelő formátumban legyen, a legtöbb esetben ez szöveg. Ha egy szöveg helyén mégis számot adunk meg, akkor az excel cellába írjuk be a szám elé a ’ jelet, s akkor szövegként lesz értelmezve. Szintén gyakori hiba, hogy képleteket hagytunk az excelben. Ilyenkor alakítsuk át azokat értékké (ld. fent az excelt tárgyaló fejezeteket).
  + „Az alábbi piktogram nem található”: a táblatervek olyan piktogramra hivatkoznak, ami nem szerepel a piktogramkönyvtár szótárfájlban. Azaz, egy olyan szögletes zárójeles piktogramnevet adtunk meg, amihez nem mondtuk meg, hogy melyik képfájl tartozik. Az is lehet, hogy elírtuk a nevet, a név pontosan úgy kell szerepeljen a táblaterv excelben, ahogy a piktogramkönyvtár szótárfájlban van.
  + „Hiba az alábbi piktogram betöltésekor”: a piktogram nevét jól írtuk, szerepel is a szótárban, de a hozzárendelt fájl nincs meg, nem jó helyen van vagy nem nyitható meg. Ellenőrizzük a szótárfájlban, hogy jól írtuk-e be a piktogram nevét (almappával, általában a „picto/” előtaggal és a típus utótaggal, pl. „.pdf”, „.jpg”, „.png”), valamint hogy a fájlt tényleg úgy hívják és ott van a helyén, bemásolva, megfelelő típusú (PDF vagy JPG, PNG).
  + „Hiba az alábbi tábla kiírásakor”: a PDF fájl valamiért nem írható (tovább). Lehet, hogy megtelt a lemez, vagy nincs tovább hozzáférés. Az is lehet, hogy a PDF fájl meg van nyitva egy nézőben, ilyenkor zárjuk be.

Ha ettől eltérő, ismeretlen hibajelzést kapunk és nem tudjuk értelmezni, lépjünk kapcsolatba a program készítőjével.

A program a hibás méretezéseket nem ellenőrzi, tehát ha pl. valami kilóg vagy elcsúszik, azt csak a kimeneti fájlból tudjuk megállapítani. Ilyenkor módosítsuk a táblaformátum paramétereket vagy a tábla tartalmi terveket (excel), a hiba jellegétől függően.