

Eötvös Loránd Tudományegyetem  
Informatikai Kar  
Térképtudományi és Geoinformatikai Tanszék

# TEMATIKUS TÉRKÉPEK GENERÁLÁSA WEBES FELÜLETEN KERESZTÜL

**Szabó Tímea**

térképész mesterszakos hallgató

Témavezető:

**Dr. Gede Mátyás**  
adjunktus



Budapest, 2013

# Tartalomjegyzék

<b>Tartalomjegyzék</b>	<b>2</b>
<b>1. Bevezetés</b>	<b>4</b>
<b>2. A tematikus kartográfia</b>	<b>5</b>
2.1. A tematikus ábrázolási módszerek . . . . .	5
2.1.1. Jelmódszer . . . . .	5
2.1.2. Felületi módszer (alaprajzos ábrázolás) . . . . .	6
2.1.3. Pontmódszer . . . . .	6
2.1.4. Kartogrammódszer . . . . .	7
2.1.5. Diagrammódszer . . . . .	7
2.1.6. Izovonalmódszer . . . . .	7
2.1.7. Mozgásvonalak módszere . . . . .	8
2.2. Virtuális térképek tematikus ábrázolási módszerei . . . . .	8
<b>3. Példák ingyenesen elérhető webes tematikus térképgenerálókra</b>	<b>9</b>
3.1. CartoDB . . . . .	9
3.2. Google Fusion Tables . . . . .	10
3.3. StepMap . . . . .	10
<b>4. Felhasznált technológiák</b>	<b>12</b>
4.1. Fájltípusok . . . . .	12
4.1.1. Bemeneti fájlok . . . . .	12
4.1.2. Kimeneti fájlok . . . . .	13
4.2. PHP . . . . .	15
4.3. A MapServer és a PHP MapScript ismertetése . . . . .	15
4.3.1. A MapServer . . . . .	15
4.3.2. A Mapfile . . . . .	16
4.3.3. A PHP MapScript . . . . .	17

<b>5. A weboldal bemutatása, működése</b>	<b>18</b>
5.1. A weboldal szerkezete . . . . .	18
5.2. A weboldal működése . . . . .	19
5.3. A weboldal elkészítése . . . . .	22
<b>6. Összefoglalás</b>	<b>30</b>
<b>7. Köszönetnyilvánítás</b>	<b>31</b>
<b>Irodalomjegyzék</b>	<b>32</b>
<b>Nyilatkozat</b>	<b>36</b>

# 1. fejezet

## Bevezetés

Dolgozatomban olyan lehetőségeket mutatok be, melyekkel webes felületen keresztül vektoros állomány és adattábla behívásával, a kívánt tematikus ábrázolási módszer kiválasztásával térkép jeleníthető meg. Az így létrehozott térkép különböző formátumokban menthető. E megoldások előnye, hogy egyszerűen, gyorsan, szoftverek telepítése nélkül hozhatunk létre tematikus térképeket.

Munkám során tanulmányoztam azokat a mai internetes alkalmazásokat, melyekkel tematikus térkép generálható. Ezek többsége a különböző térképszolgáltató óriáscégek (Google, Yahoo, OSM, Nokia) térképét használja háttérként, és a felhasználó által feltöltött fájlok (főként KML, XLS, CSV) segítségével adja a tematikus tartalmat. Dolgozatom végén ezekből mutatok be néhány jelentősebb weboldalt példaként.

Az előzetes keresés alapján azt állapítottam meg, hogy szükséges lenne egy olyan webes térinformatikai alkalmazás, mely tematikus térképek generálására szolgál. Ezért hoztam létre egy MapServer alapú weboldalt, melyre a felhasználó maga tölthet fel tetszőleges adatokat tartalmazó shapefile-t, és azt GIS (Geographic Information System) szoftverhez hasonlóan kategorizálva, stilizálva jelenítheti meg, készítheti el térképét. A weboldalon elérhető térkép megjelenítéséhez a MapServer egy leíró szövegfájlt használ. Ezt normál esetben statikusan, kézzel kellene megírni, de a honlapot használva ez a folyamat interaktívva tehető. A fájl a térkép elkészítése után letölthető, ezáltal elősegítheti a MapServer oktatását is. A weboldal elkészítésével azt kívánom megmutatni, hogy egy ilyen tematikus térképet saját anyagból is készíthetünk, nem kell feltétlenül térképszolgáltatót használni.

E dolgozatban először példákat hozok a manapság elterjedt módszerekről, majd a tematikus térképek elkészítésének módját tárgyalom, végül az ehhez készített honlapot mutatom be.



## 2. fejezet

# A tematikus kartográfia

### 2.1. A tematikus ábrázolási módszerek

A 19. század első évtizedeiben a természettudományok fejlődése és a statisztika kialakulása maga után vonta az új tematikus módszerek megjelenését. Erre 1800 és 1860 között került sor, amikor is kialakult napjaink kartográfiai ábrázolási módszereinek többsége. E korszak előtt a tematikus térképek a geofizikai térképeket leszámítva főként minőségi információt hordoztak, de ezt követően inkább mennyiségit a statisztikai adatok ábrázolása miatt [KLINGHAMMER, 2010].

A tematikus adatokat a háttértérkép (alaptérkép) helyezi el a térben. Ez a térkép egy leegyszerűsített általános térkép, mely lehet tartalmi (csak a témához kapcsolódó elemeket jeleníti meg) vagy formai (halványítás, egyszínűsítés) jellegű, illetve alkalmazható mindkettő együtt.

A tematikus kartográfia hagyományosan hét csoportot különít el, melyek közül csak négy (a kartogrammódszer, a diagrammódszer, a pontmódszer és a mozgásvonalak módszere) kapcsolódik kizárólag a tematikus térképekhez. A másik három (a jelmódszer, a felületi módszer és az izovalas módszer) megjelenik mind az általános, mind a tematikus térképeken [BARTOS, 2007].

#### 2.1.1. Jelmódszer

Ezzel a módszerrel a tárgyakat a nagy méretarány miatt helyzethűen, de alaprajztól eltérően ábrázoljuk. A jelmódszer tömör képírás, tehát az alaprajzos ábrázolásnál és annak magyarázó megírásánál sokkal gyorsabban olvasható, a vizuális hatás erősebb. A jel többletinformációja lehet minőségi és mennyiségi. A mennyiségi adatok függvényében a jelek nagysága változhat, ez történhet folyamatosan (lineáris vagy exponenciális függvény szerint), fokozatosan (lépcsőzetes növekedés) vagy értékegyiséggel (itt a jel nem növekszik, hanem sokszorozódik). A jelmódszer vonatkozhat

pontszerű vagy vonalas (2.1. ábra) diszkrétumokra.



2.1. ábra. Jelmódszerrel készített térkép

### 2.1.2. Felületi módszer (alaprajzos ábrázolás)

E módszert alkalmazva ábrázolhatjuk a tárgyak vagy jelenségek alakját, méretét (elterjedési terület), minőségét. Ahogyan a 2.2. ábrán láthatjuk, ezeket a zárt objektumokat homogén színkitöltéssel és határainak kijelölésével különböztük el egymástól. A határ kijelölése nem mindig egyértelmű, ekkor a területábrázolás vázlatos. Ilyenkor a határábrázolás történhet fogazással, vagy csak a terület jelölésével az arra vonatkozó megírás elhelyezésével.



2.2. ábra. Felületi módszerrel készített térkép

### 2.1.3. Pontmódszer

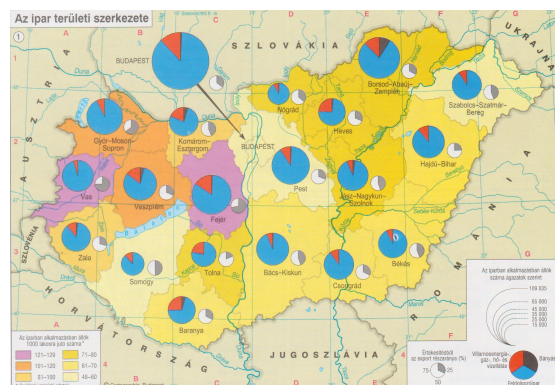
Az elemek szóródását, eloszlását mutatja be pontok segítségével. Egy pont több elem összevonásából keletkezik. Ritkán alkalmazzák, mert elkészítése elég munkaigényes, a térinformatikai szoftverek ritkán támogatják, de akkor sem teljeskörűen.

## 2.1.4. Kartogrammódszer

Felületre vonatkozó mennyiségi adatokat ábrázol: a felületkartogram (2.3. ábra) általában relatív adatot értékfokozatos (színfokozatos) felülettel, a jelkartogram általában abszolút adatot értékfokozatos jellel.

## 2.1.5. Diagrammódszer

A diagram a statisztikai értékeket jellegzetességük alapján tovább részletezi, így az összetevők arányát vagy azok időbeni változását mutatja be. A diagram vonatkozhat pontra (pontdiagram) vagy felületre (kartodiagram). Alakjuk szerint lehet kördiagram, négyzetdiagram, oszlopdiaagram, korfa, szélirány-gyakoriság, éghajlat-diagram. A 2.3. ábra egy kartodiagram ábrázolást szemléltet, melyen a diagram kördiagram.



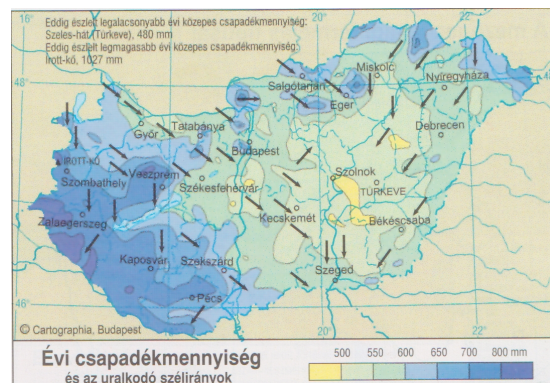
2.3. ábra. Kartogram- és diagrammódszerrel készített térkép

## 2.1.6. Izovonalmódszer

A kontinuumok ábrázolási módszere: az azonos értékű pontokat izovonallal kötjük össze, a szélsőséges értékű pontokat számszerűen jelöljük. A szakirodalom több mint száz izovonaltípust különít el: izopihsza (magasság; ez adja a szintvonalat), izobát (mélység), izobár (légnomás), izohalina (sótartalom) stb.. Két izovonal közti felületet eltérő színekkel jelölhetjük a szemléletesség érdekében, ha ezek az izovonalak szintvonalakat jelöltek, akkor így rétegszínezéses domborzatábrázolást hozunk létre. A 2.4. ábrán az évi csapadékmennyiség két izohiéta közti rétegszínezéssel jelenik meg, melynek köszönhetően könnyebben azonosítható a csapadékosabb terület.

### 2.1.7. Mozgásvonalak módszere

Dinamikus jelenségek bemutatására alkalmas, az ábrázolás nyíllal történik. A nyíl vastagsága a mennyiséget, színe a minőséget, iránya pedig a jelenség irányát fejezi ki. A digitális technológia már megengedi, hogy ezeket a dinamikus jelenségeket animációval jelenítsük meg, mely a felhasználó szemszögéből lehet statikus vagy interaktív. A 2.4. ábra ezzel a módszerrel mutatja az uralkodó szélirányt.



2.4. ábra. Izovonal- és mozgásvonal módszerrel készített térkép

## 2.2. Virtuális térképek tematikus ábrázolási módszerei

Az általam létrehozott alkalmazásban a felhasználó egy interaktív kezelőfelületen virtuális térképet hozhat létre, melyet saját adathordozójára menthet. A számítógép képernyőjén megjelenő térképeknek három típusa van: statikus, interaktív vagy animált [DENT, 2009]. A dolgozatomban statikus térképről beszélek, amelynek kimeneti fájl típusát a felhasználó választja meg, így az lehet raszter vagy vektor állomány.

A munkám során felhasznált alkalmazás a fent említett hét módszer közül csak négyet tud kezelni, ezeket sem teljeskörűen. A három kimaradt módszer: a pontmódszer, az izovonalmódszer és a mozgásvonalak módszere. Ezeket sajnos nem támogatja a MapServer. A többi módszer tekintetében a határok kijelölése csak folytonos vonallal lehetséges, a diagrammódszer esetében pedig csak kör- és oszlopdigrammos megoldás lehetséges.

## 3. fejezet

# Példák ingyenesen elérhető webes tematikus térképgenerálókra

### 3.1. CartoDB

A CartoDB [DE LA TORRE és ÁLVAREZ, 2012] olyan fejlesztés, melynek segítségével a valós idő is figyelembe vehető bármilyen tematikus térkép tekintetében, hiszen azonnal megjeleníti a változtatásokat. Ezen a weboldalon egy feltöltött ZIP, KML, CSV, GPX vagy XLS fájl tematikus adatait a benne lévő geoadatok alapján tudjuk megjeleníteni különböző általunk kiválasztható háttértérképen. Ezek az alaptérképek különböző nyilvános szolgáltatóktól érhetőek el, mint például a Nokia (Day), a MapBox (Streets, Terrain), GoogleMap (Satellite, Hybrid, Gray Roadmap, Dark), de megadhatunk saját vagy más kiszolgáltót is. Változtathatunk az adattáblán, csatolhatunk hozzá más táblát, adhatunk és törölhetünk sort vagy oszlopot, SQL lekérdezéseket hajthatunk végre. A táblaadatok alapján végzett szűréssel csak a kívánt adatokat jeleníthetjük meg a térképen. A KML fájl tartalmazhat pontokat, vonalakat vagy felületet - mindegyik típust meg tudja jeleníteni, leginkább egyszerűen, vizualizációs eszközök használata nélkül. Ha ennél bonyolultabbat szeretnénk, mint például egy népsűrűség-térképet, ahhoz egy olyan táblát kell előállítani, mely rendelkezik a kellő adatokkal. A stílus meghatározásánál választhatunk kartogrammodszert, de itt a kategória határértékeit nem tudjuk beállítani, csak a kategóriák számát. Ahhoz, hogy ez is teljesülhessen szükséges némi CSS ismeret, mellyel meghatározhatjuk a kategóriák értékeit és stílusát. Ám ezzel a művelettel az alkalmazás kikerül a felhasználóbarátként jellemezhető webes térképgenerálók közül.

## 3.2. Google Fusion Tables

A Google Fusion Tables [GOOGLE, 2013] a Google saját fejlesztése, tehát saját háttér-térképet, a Google Maps térképes, illetve műholdas megjelenítését használja, melyre a CartoDB-hez hasonlóan geoadatokkal rendelkező pontokat, vonalakat vagy felületeket képes felszerelni. Ezeket az adatokat a feltöltött CSV, TSV, TXT vagy KML fájlokból olvassa ki. A tematikus tartalom stílusát korlátozottan megadhatjuk a »Tools/Change map style...« menüjében, ahol kiválaszthatjuk a pontok, vonalak és felületek ábrázolásának módját. Ez történhet egyszerűen, csak egy színnel vagy sorolhatjuk kategóriákba mezőnként, határértékeket meghatározva, illetve két érték között folyamatos színátmenettel. Ahol határértékekkel különíthetjük el a csoportokat egymástól, ott bármilyen színt megadhatunk, a folyamatos színátmenet két szín között lehetséges, itt nem határozhatjuk meg a kategóriák értéktartományát, csak a számukat. Névrajz nem jelenik meg a térképen. Szűrést is végezhetünk a térképen a tábla adatai alapján.

## 3.3. StepMap

A StepMap [STEPMAP, 2013] egy német cég által készített weboldal, melynek háttér-térképét az OpenStreetMap biztosítja. Ez a térkép feldolgozza az egész világot a legkisebb településekig, részletessége a nagyítás szintjétől függ. Az alaptérképet egy kissé átdolgozták, bár még így is nagyon hasonlít a forrástérképre, névanyagként a GeoNames adatbázisát használják, a piktogramokat pedig egy külső céggel készítették el. A honlapon külön lehetőségünk van várostérképet készíteni, illetve kisebb méretarányban régió-ország-kontinens-térképet.

Ezen a felületen négy lépésből áll egy térkép generálása.

- 1. lépés:** Első lépésben kiválasztjuk a térkép kivágatát, ezt a világtérkép adott részére való nagyításával tehetjük meg vagy egy legördülő listából az adott országot tudjuk kiválasztani, amelybe még tovább nagyítható. Választhatunk a térképre domborzatárnyékolást vagy hipszometriás domborzatábrázolást, amelyhez a későbbiekben nem társul jelmagyarázat. Itt lehet kiválasztani az adott ország felületi színét, hogy egy áttekintő térképen azzal a kitöltéssel jelenjen meg. Végül pipálással megadható, hogy ábrázoljuk-e az országhatárokat, a megyéket vagy régiókat, tavakat, partokat, folyókat, autópályákat, mértéklécet és a kis áttekintőtérképet.
- 2. lépés:** Második lépésben a térképre pontokat vehetünk fel településre vagy földrajzi koordinátára való keresés után, illetve kézzel adott helyre szúrva. Kiválaszthatjuk, milyen betűtípussal legyen írva a választott hely neve, illetve milyen

legyen a helységjele. A kivágaton található országok és fővárosok neveit automatikusan kiírathatjuk a beállított stílusokkal. Az országok neveit felület helyett pontra vonatkoztatja. Két pont közé húzhatunk vonalat, amelynek stílusát is itt kell megadni. A piktogramok témák szerint vannak elkülönítve egy legördülő listában. Innen lehet kiválasztani a kívánt szimbólumot és elhelyezni a térképen, illetve mondhatjuk azt, hogy sajátot akarunk feltölteni számítógépünkről.

- 3. lépés:** A harmadik lépésben leírást, egy weboldal linkjét, illetve multimédiás fájlokat adhatunk a második lépésben megadott helységekhez.
- 4. lépés:** A negyedik lépésben a térképnek címet, leírást és kulcsszavakat adunk, kiválasztjuk a kontinenst, országot, és azon belül a régiót (Magyarország esetében csak a Balatont és Magyarországot választhatjuk), amelynek kivágatát ábrázoljuk, illetve azt hogy nyilvánossá tesszük-e vagy szerzőként csak magunk láthassuk.

Az így létrehozott térképet beágyazhatjuk saját weboldalunkba, kinyomtathatjuk vagy JPG formátumban menthetjük. A honlapba ágyazás többféle térképi szélességgel történhet, de a kép maximálisan 640 pixel széles lehet, melyhez az alkalmazás kódot is generál. A JPG fájl 640x640 pixeles felbontásig ingyen letölthető, annál nagyobbért már fizetni kell érte. A legnagyobb elérhető felbontású kép 2560x2560 pixel, mely 5,99 \$-ért megvásárolható.

## 4. fejezet

# Felhasznált technológiák

Ebben a fejezetben bemutatom az általam készített webes alkalmazás technikai megoldásait, készítésével kapcsolatos nehézségeket. Láthatjuk, hogy a fentiekben ismertetett internetes felületen működő tematikus térképgenerálók függenek a háttértérképet szolgáltató szervertől, ezért én olyan megoldást választottam, amely képes kizárólag saját adatokból dolgozni. Az én megvalósításommal a felhasználó úgy készíthet web-térképet, hogy feltölti az általam létrehozott internetes felületre a saját shapefile-ját, amelynek rendelkeznie kell a számára szükséges adatokkal, majd a rétegek létrehozásával és azok beállítása után a MapServer generálja a térképet. Bármilyen felbontásban, az alábbiakban ismertetett kimeneti fájlformátumokban és ingyenesen készíthető így térkép. Az előző módszerekhez hasonlóan, a MapServer sem kezeli kartográfiaiag helyesen a névrajz megjelenítését, ezért nálam nincs is beépítve ez a funkció, ezt sajnos utólag kell megoldania a felhasználónak.

### 4.1. Fájl típusok

#### 4.1.1. Bemeneti fájlok

##### ESRI Shapefile

Ezt a tárolási formátumot az ESRI cég fejlesztette ki 1990-ben. A tárolási mód leírását a cég nyilvánosságra hozta, és ma a legtöbb térinformatikai szoftver támogatja. A shapefile valójában legalább három állományt (*.shp*, *.shx* és *.dbf*) jelent, melyeknek elnevezése azonos, mivel a feldolgozó rendszerek csak ebből tudják megállapítani a kapcsolatot. Az elnevezés a korábbi MS DOS operációs rendszer szabályait követi, ami 8 karakteres nevet és 3 karakteres kiterjesztést jelent. Ezen kívül további azonos nevű, de más kiterjesztésű állományok is tartozhatnak a shapefile-hoz, amelyek karakterkódolási, indexelési és egyéb metaadatokat tartalmaznak.



A három kötelező állomány közül az *.shp* kiterjesztésű tartalmazza az alakzatok geometriai leírását. A fájl egy fix hosszúságú (100 bájtos) fejléccel, és alakzatonként egy változó hosszúságú rekordot tartalmaz.

Az *.shx* kiterjesztésű állomány egy fix hosszúságú fejléccel és alakzatonként egy fix hosszúságú rekordot tartalmaz. Ez utóbbi rekordok az alakzatok *.shp* állományon belüli kezdetét és hosszát tárolják.

Végül a *.dbf* kiterjesztésű állomány az alakzatokhoz tartozó egyéb szöveges, numerikus és dátum típusú információkat tárolja dBase formátumban.

A shapefile-okban tárolt alakzatok lehetnek két-, három- vagy négydimenziósak, amelynél a negyedik dimenzió méretinformációkat tárol. Az alakzatok nem tartalmazhatnak körívet, és egy shapefile-ban csak egyfajta geometriai típus lehet: pont, szakaszsorozat vagy sokszög. Ha többfajta alakzatot szeretnénk tárolni, akkor ezt több shapefile-ban tehetjük meg [ESRI, 1998].

#### **4.1.2. Kimeneti fájlok**

##### **JPEG**

A JPEG (Joint Photographic Experts Group) képek tárolására alkalmas fájlformátum. Ez a formátum a képen lévő információt veszteségesen tömöríti. Bár a tömörítés információvesztéssel jár, akár 10-100-szor kisebb fájl méret mellett is élvezhető a tömörített kép. Elsősorban fényképek, rajzok tárolására való [HAMILTON, 1992].

##### **PNG**

A PNG (Portable Network Graphics) képek tárolására, veszteségmentes tömörítésére alkalmas fájlformátum. Lehetővé teszi a képek átlátszóságát, a fényességüket - elméletben - függetleníteni tudja a megjelenítéstől, vagyis a színek ugyanúgy néznek ki nyomtatásban, és eltérő képességű kijelzőkön [W3C, 2003].

##### **GIF**

A GIF (Graphics Interchange Format) képek tárolására, veszteségmentes tömörítésére alkalmas fájlformátum. Mivel legfeljebb 8 bites színmélységű (256 színű) képeket tud kezelni, ezért elsősorban rajzok, grafikonok és egyéb hirtelen színátmenetű ábrák tárolására való [COMPUSERVE, 1990].

##### **GeoTIFF**

A GeoTIFF egy nem védjegyzett TIFF formátum. A cél az volt a GeoTIFF létrehozásánál, hogy a raszteres képeket helyesen pozícionálva és skálázva automatikusan

be lehessen olvasni tetszőleges térinformatikai rendszerbe. GeoTIFF a földrajzi információkat úgy tárolja, hogy az a felhasználó számára szinte láthatatlan marad, megkönnyítve ezzel a képi adatok szállíthatóságát, felhasználását [RITTER és RUTH, 2000].

## **KML, KMZ**

A KML (Keyhole Markup Language) olyan XML alapú jelölőnyelv, amelyben csak geodetikus koordinátarendszerben megadott objektumokat tudunk tárolni. Az objektumok koordinátáit hosszúság, szélesség, magasság alakban tudjuk megadni a WGS84-es ellipszoidot használó koordinátarendszerben. A jelölőnyelvet a Keyhole Inc. készítette, amelyet a Google 2004-ben felvásárolt. A Google javaslatára az OGC (Open Geospatial Consortium) felvette a szabványok közé ezt a formátumot, amelyet például a Google Earth program is használ. Egy KML dokumentum nem csak a geometriai objektum tulajdonságait, hanem annak megjelenítésével kapcsolatos információkat is tárolja. Ilyen információ például, hogy honnan, milyen nézőpontból látjuk az objektumot. A KML nem támogatja a köríveket, és nem adható meg koordinátarendszer, mivel az rögzített. A KMZ a KML fájl tömörített változata [GOOGLE, 2012].

## **SVG**

Az SVG (Scalable Vector Graphics) leírőnyelvet a World Wide Web Consortium (W3C) egyik munkacsoportja 1998-ban kezdte el kidolgozni. A cél egy olyan szabvány kifejlesztése volt, mely XML-t alkalmazva képes az ábrák leírására. Egy SVG dokumentum geometrikus alakzatokból álló ábrát tartalmaz szöveges formában. Ezek a grafikák később megjeleníthetők egy weboldalon vagy egy erre alkalmas program (pl. IrfanView) segítségével. A grafikában szereplő elemek általában görbék, síkidomok vagy szöveges objektumok, azonban lehetőség van arra is, hogy nem vektoros képet építsünk be a rajzba. Mivel szövegesen van leírva a kép tartalma, a keresőprogramok könnyebben tudnak információhoz jutni az ábrával kapcsolatban, mintha hagyományos, raszteres ábrázolást használnánk. A vektorgrafika másik nagy előnye a szabad átméretezhetőség a minőség romlása nélkül. Az SVG erre is rugalmas lehetőséget biztosít a felhasználó számára [W3C, 2005].

## **PDF**

A PDF (Portable Document Format) az Adobe Systems által kifejlesztett, dokumentumok tárolására alkalmas fájlformátum. A PDF alkalmas szöveget, ábrát és képeket tartalmazó dokumentum leírására eszközfüggetlen és felbontásfüggetlen formában. A

dokumentum lehet egyoldalas, és több ezer oldalas is, egyszerű vagy összetett: sokféle betűtípussal, ábrával, színnel és képpel, továbbá tartalmazhat interaktív beviteli mezőket, nyomógombokat stb., lehetőséget teremtve arra, hogy üres űrlapokat hozunk létre, amit a felhasználók később elektronikus formában is ki tudnak tölteni. A PDF nyílt szabvány, emiatt bárki jogdíjmentesen írhat alkalmazásokat, amelyek PDF formátumú dokumentumokat olvasnak vagy írnak [ADOBE SYSTEMS, 2013].

## 4.2. PHP

A PHP [OLSON, 2013] általános szerver oldali szkript nyelv, melyet dinamikus weblapok megalkotására hoztak létre. Az első szkript nyelvek egyike, amely külső fájl használata helyett HTML oldalba ágyazható. A kódot a webszerver PHP processzor modulja értelmezi, ezzel dinamikus weboldalakat hozva létre. Rasmus Lerdorf 1995-ben indította útjára. Ma a The PHP Group tartja fenn és fejleszti. A PHP szabad szoftver, de licensze nem csereszabatos a GNU licensszel, mivel megkötéseket tartalmaz a PHP név használatára.

A weboldal egy része statikus HTML5 [W3C, 2013] szabvány szerint megírt HTML kód, míg a dinamikus részt kis mértékben a JavaScript [JAVASCRIPT, 2003], de főként a PHP szkriptnyelven írt programkód állítja elő, melyek esztétikus megjelenését egy CSS [W3C, 2011] stílusfájl garantálja. Ezeket a kódokat a böngésző nem kapja meg, hanem a PHP értelmezi még szerver oldalon.

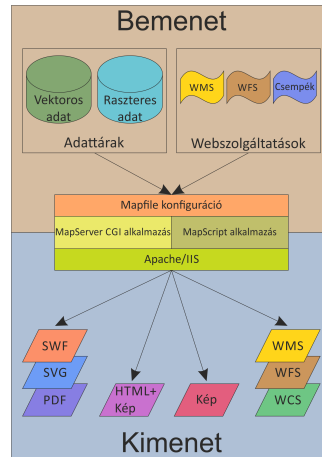
## 4.3. A MapServer és a PHP MapScript ismertetése

### 4.3.1. A MapServer

A MapServer [UNIVERSITY OF MINNESOTA, 2013] népszerű az UMN (University of Minnesota) által kifejlesztett nyílt forráskódú webes térinformatikai programcsomag. Segítségével interneten jeleníthetünk meg térképeket. Képes megjeleníteni és lekérdezéseket végrehajtani számos raszter, vektor vagy adatbázis formátumban, amint a 4.1. ábrán is látszik. Támogatja a legnépszerűbb szkriptnyelveket és fejlesztői környezetet (PHP, Python, Perl, Ruby, Java, .NET). Valós idejű vetületváltás és magas minőségű képgenerálás is jellemzi. A MapServer különböző nyílt forrású könyvtárak is használ:

- *shapelib*: ESRI shape állományok olvasása, írása,
- *gd*: raszteres kép létrehozása vektoros adatokból,

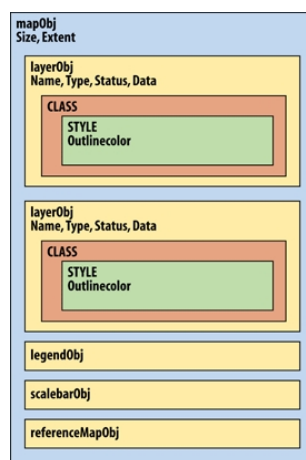
- *libtiff*, *libgeotiff*, *libjpeg*, *libpng*: kép formátumok írása, olvasása,
- *freetype*: true type fontok kezelése,
- *proj4*: vetületek,
- *postgis*: relációs adatbázis geometria elemek tárolásához [SIKI, 2003].



4.1. ábra. A MapServer felépítése

### 4.3.2. A Mapfile

A Mapfile *.map* kiterjesztésű egyszerű szövegfájl, amely leírja, hogy a különböző adatforrások tartalmát hogyan ábrázoljuk a térképen. A Mapfile egymásba ágyazott objektumokból áll, melyek mindegyike az objektum típusát jelző kulcsszóval kezdődik (pl.: MAP, LAYER), és az END kulcsszóval végződik.



4.2. ábra. A MapScript felépítése

### 4.3.3. A PHP MapScript

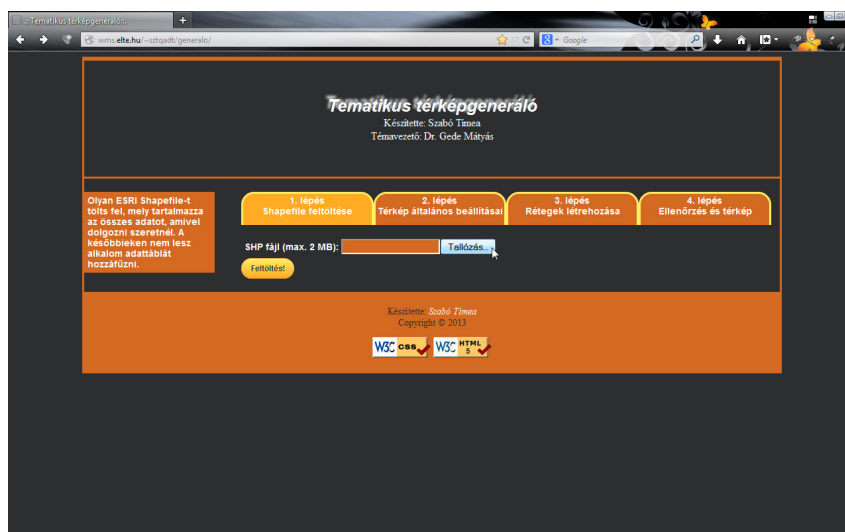
A PHP MapScript [UNIVERSITY OF MINNESOTA, 2013] olyan modul, mely lehetővé teszi, hogy a PHP szkriptnyelv használatával dinamikusan létrehozhassuk vagy módosíthassuk térképeinket MapServeren keresztül. A MapScript objektumszerkezete, hasonlóan a Mapfile-hoz, hierarchikus felépítésű. Ezt mutatja a 4.2. ábra, melynek alkalmazásomban használt objektumait a későbbiekben részletesen tárgyalom.

## 5. fejezet

# A weboldal bemutatása, működése

### 5.1. A weboldal szerkezete

A térkép generálásának folyamata, ahogyan a 5.1. ábrán láthatjuk, összesen 4 lépésből tevődik össze:



5.1. ábra. 1. lépés: A shapefile feltöltése

#### 1. lépés: A shapefile feltöltése

Ebben a részben tölthetjük fel az ESRI Shapefile állományt, melynek már minden szükséges adatot tartalmaznia kell, mivel a későbbiekben nem lesz mód adattáblák csatolására.

#### 2. lépés: A térkép általános beállításai

Itt adjuk meg a térkép címét, vetületét, a kép felbontásának minőségét és a kimeneti fájl típusát.

### 3. lépés: A rétegek létrehozása

Ebben a lépésben adjuk meg a réteg nevét, a hozzá tartozó shapefile-t, illetve annak típusát. Ezt követően el kell döntenünk, hogy minek alapján szeretnénk ábrázolni (jellel/felülettel minőségi/mennyiségi adatok alapján vagy diagrammal) az adatokat, melyik mezőből szűrjük ezeket, hány kategóriát különböztessünk meg, milyen attribútumok alapján és ezeknek milyen legyen a stílusa.

### 4. lépés: Ellenőrzés és térkép

Itt adhatunk a térképünkhöz még egy réteget, beállíthatjuk, hogy szükséges-e számunkra jelmagyarázat vagy fokhálózat, új térképét készíthetünk, illetve megjeleníthetjük a kész térképünket.

## 5.2. A weboldal működése

Hogyan készíthetünk egy szép, saját térképet ezen a weboldalon? Nyissuk meg böngészőnkben a [wms.elte.hu/~sztqadt/generalo/](http://wms.elte.hu/~sztqadt/generalo/) linken található oldalt.

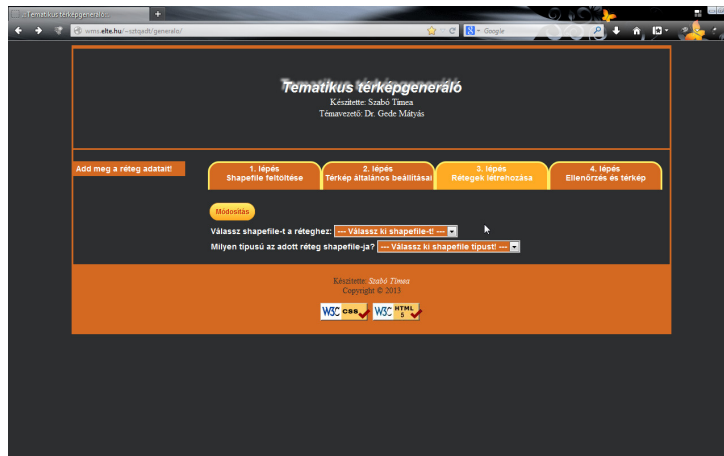
Először is birtokolnunk kell egy ESRI shape állományt, melyet össze kell kapcsolni az ábrázolni kívánt adatokkal. Ezt kell feltölteni az 1. lépésben [5.1. ábra]. Itt nem csak az egy állományt alkotó fájlokat tölthetjük fel, hanem egyszerre akár többet is.

A 2. lépésben [5.2. ábra] a térkép általános adatait kell megadni. Ide kell beírni a térkép címét, a készítendő kép méretét pixelben és a fájl típusát, mely lehet raszter (PNG, JPEG, GIF, GeoTIFF) vagy vektor (SVG, KML, KMZ, PDF). Ki kell választani a térkép vetületét, de ha az nincs benne a legördülő menüben, beírható a vetületi rendszer EPSG (European Petroleum Survey Group) kódja is, amit akár meg is kereshetünk a weboldalba ágyazott hivatkozás alatt.



5.2. ábra. 2. lépés: A térkép általános beállításai

A 3. lépés beállításai már nagyon függenek a tervezett térkép ábrázolási módszertől. Itt hozunk létre rétegeket a térképhez. Meg kell említenem, hogy a rétegek a létrehozás sorrendjéből alulról felfele helyeződnek el, tehát az újabb réteg elta-  
karja a korábbi. Amint a 5.3. ábra mutatja, ki kell választanunk a réteghez tartozó shapefile-t, amit az 1. lépésben töltöttünk fel, és meg kell adni ennek típusát (pont, vonal, felület), mely arra utal, hogy milyen objektumokat tartalmaz a fájl.



5.3. ábra. 3. lépés: A shapefile nevének és típusának kiválasztása

Ekkor felugrik a megjelenítés módjának kiválasztását segítő legördülő menü, melynek tartalma előző válaszuktól függ, azaz a shapefile típusától. Pont típus esetén a pontot ábrázolhatjuk jellel, melyet minőségi vagy mennyiségi adatok alapján határozzunk meg, vagy diagrammal. Vonaltípusú fájl esetén minőségi vagy mennyiségi adatok alapján megadott jellel ábrázolhatjuk a térképi objektumot. Felülettípus esetén választhatunk mennyiségi vagy minőségi adatok alapján történő kategorizálást. A mennyiségi adatokat a felületet homogén színekkel vagy diagrammal jelenítjük meg, míg minőségi adatoknál felületszínezést alkalmazhatunk.

A módszer kiválasztása után minden esetben a shapefile egy mezőjét kell megadni, mellyel a kategorizálást kívánjuk meghatározni, illetve a kategóriák számát is. A diagram esetén a kiválasztott mező a diagram méretét befolyásolja, ez adja az egészet, a 100%-ot. Ebben az esetben el kell döntenünk azt is, hogy kör- vagy oszlopdiagramot szeretnénk használni vagy a felület mennyiségi adatait homogén színekkel megjeleníteni. Az adatok csoportosításánál meg lehet adni az alapértelmezett kategórianevek helyett a sajátot és a szűrés feltételét. Ha minőség alapján osztályozunk, akkor csak ki kell választanunk a legördülő menüből az adott kategóriába tartozó elemek minőségét. Mennyiségi adatok osztályozásánál meg kell adni az adott kategória felső határértékét, azaz egy számértéket. Az adatok diagrammal való ábrázolásához kategóriánként ki kell választanunk a legördülő menüből a diagraemelemekhez tartozó



mezőket, mivel ezekben lévő számadatok határozzák meg az elem méretét, arányát.

A következő beállítások a kategóriák stílusát határozzák meg. Ezek többnyire csak a kitöltés és a körvonal színét kérik be [5.4. ábra], kivéve a diagramnál és a jeleknél, ahol ezeken felül más adatokat is be kell vinni. A diagramnál megkülönböztettem kör- és oszlopdigramot, melyeknél más-más adatot kértem be. A kördiagramnál be kell írni a kör maximális és minimális méretét pixelben, illetve az ahhoz tartozó adatokat az összértéket tartalmazó mező alapján, viszont az oszlopdigramnál a 100%-os oszlop magasságát és az oszlopok együttes szélességét kérik megadni [5.5. ábra]. Jellel történő ábrázolásnál ki kell választani a szimbólumot és meg kell adni annak méretét és vonalvastagságát is.



5.4. ábra. 3. lépés: A kategóriák stílusa általános esetben



5.5. ábra. 3. lépés: A kategóriák stílusa oszlopdigram esetén

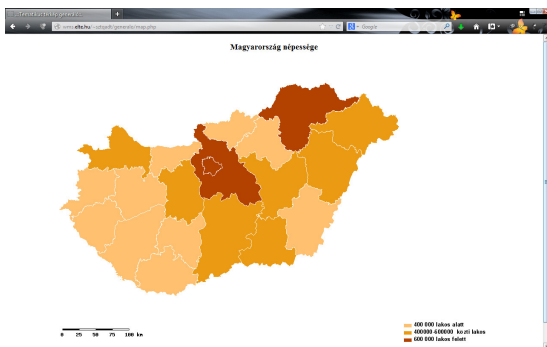
Végül a 4. lépésben [5.6. ábra] adhatunk újabb réteget a térképhez, pipálással választhatunk jelmagyarázatot vagy fokhálózatot hozzá, vagy új térkép szerkesztésébe kezdhetünk elfeledtetve az alkalmazással eddigi beállításainkat. A »Kész« gombra kattintva megtekinthetjük a térképet.



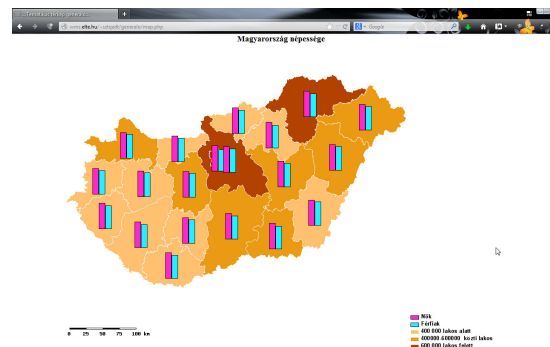
5.6. ábra. 4. lépés: Ellenőrzés és térkép

Ha raszterként jelenítjük meg a kimeneti fájlt, akkor az a böngészőben fog meg-

jelenni, onnan menthetjük saját gépünkre a jobb klikk »Kép mentése más néven...« paranccsal, vektorként automatikusan felugrik egy mentési ablak, melynek segítségével menthetjük vagy megnyithatjuk a fájlt. A böngészőbe való megjelenéskor fent láthatjuk a térkép címét, alatta magát a térképet, illetve ha kértünk, akkor jelmagyarázatot is. Az oldal alján látható a »Mapfile megtekintése« link, mely kattintásra behozza a MapScript által generált Mapfile tartalmát. Ezt kimásolva egy *.map* kiterjesztésű szövegfájlba és rendelkezve az itt generált térkép minden forrásadatával, CGI módszerrel újból létrehozható a térkép. Így használható ez az alkalmazás a MapServer oktatásában is.



5.7. ábra. Kartogrammódszerrel készített térkép



5.8. ábra. Kartogram- és diagrammódszerrel készített térkép

### 5.3. A weboldal elkészítése

Az általam létrehozott weboldal biztosítja a térkép előállításához szükséges dinamikus felületet. Ez az oldal HTML, PHP, JavaScript és CSS használatával készült. Fő pillére azonban a PHP (*index.php*), amely továbbítja a webes felületen megadott adatokat egy *map.php* nevű fájlba, mely a PHP MapScript parancsait tartalmazza. Ez a fájl a szerveren létrehozza a MapServerhez szükséges Mapfile-t, amelynek térképét és jelmagyarázatát megjeleníti a weboldalon. Az így létrejött kép elmenthető saját használatra.

#### A PHP MapScript szerkezete

A PHP kód `<?php` nyitó karaktersora után a `session_start()` függvény került meghívásra. Ennek segítségével a felhasználó számára egy egyedi azonosítót és `$_SESSION` nevű változót hoz létre, melybe a későbbiekben kifejtett a felhasználó által megadott adatok kerülnek.

A `colorRGB()` függvény az *index.php*-ban kiválasztott hexadecimális színt alakítja át RGB (Red Green Blue) színmodellé, mivel a MapServer csak ilyen formában képes felismerni a színt.

```
1 function colorRGB($hexasrt)
2 {
3     $r = hexdec(substr($hexasrt, 0, 2));
4     $g = hexdec(substr($hexasrt, 2, 2));
5     $b = hexdec(substr($hexasrt, 4, 2));
6     return array('r' => $r, 'g' => $g, 'b' => $b);
7 }
```

A `$_SESSION['formatum']` változóban tároltam a fájlformátumok adatait. Az *index.php*-ban kiválasztott formátumhoz tartozó adatokat a térkép objektum (Map Object) `outputformat` tagobjektumába ez alapján tudtam behívni. Ugyanígy történt a vetületeknél is a `$_SESSION['vetulet']` változó esetében, mely a listában szereplő vetületek EPSG-kódját tartalmazza.

### A térkép osztály (Map Object, mapObj)

Az `ms_newMapObj("alap.map")` konstruktorfüggvénnyel definiálom a térkép osztályt (MapObj), mely a munkakönyvtárban található *alap.map* nevű Mapfile-t használja, hogy ebből kiindulva létrehozza a MapServernek a térkép előállításához szükséges információkat. Ez a kiinduló fájl a MAP és az END indító és záró parancsszavakat kivéve akár üres is lehet. Az én esetemben ez a fájl tartalmazza a betűtípusok listájának (*fonts.list*) elérési útvonalát, illetve a jelmagyarázat (legend) formai megjelenését:

```
1 MAP
2   FONTSET "adat/font/fonts.list"
3
4   LEGEND
5     LABEL
6       TYPE truetype
7       FONT arial-bold
8       SIZE 10
9       ENCODING "UTF-8"
10      OFFSET 0 -5
11    END
12  END
13 END
```

A térkép osztály további tulajdonságait különböző tagfüggvényekkel adtam meg. A `scalebar` tagobjektummal a térképhez készülő mértékléchez használt mértékegységet, és annak fekete kontúrját definiáltam. Az előbb említett `$_SESSION['formatum']`

tartalma alapján az `outputformat` tagobjektumot használva adhatók meg a kimeneti fájlhoz tartozó beállítások. Ugyancsak itt adtam meg a `setExtent()` függvénnyel, hogy a térkép milyen területet ábrázoljon. A későbbiekben részletezem, hogy az ehhez szükséges vetületi koordinátákat hogyan nyertem ki a `shapefile`-ből. A térképet a felhasználó kedvére méretezheti, ennek beállításait a `setSize()` függvény kezeli. Az egyedi azonosító nevével a MapScript által meghatározott Mapfile-t a `save()` metódussal elmentettem, melyet a felhasználó tanulmányozhat.

```
1 $map = ms_newMapObj("alap.map");
2 $map->set("name", $_SESSION['cim']);
3
4 $map->scalebar->set("units", MS_KILOMETERS);
5 $map->scalebar->outlinecolor->setRGB(0,0,0);
6
7 $map->setProjection("init=epsg:".$EPSG, MS_TRUE);
8
9 $map->outputformat->set('name', $outputformat_name);
10 $map->outputformat->set('driver', $outputformat_driver);
11 $map->outputformat->set('mimetype', $outputformat_mimetype);
12 $map->outputformat->set('imagemode', $outputformat_imagemode);
13 $map->outputformat->set('extension', $outputformat_extension);
14 $map->outputformat->set("transparent", MS_ON);
15 $map->selectoutputformat($outputformat_select);
16 .
17 .
18 .
19 $map->setExtent(min($minxa), min($minya), max($maxxa), max($maxya));
20 $map->setSize($_SESSION['kep']['kepa'], $_SESSION['kep']['kep']);
21 $map->save("/home/webgis2013/sztqadt/public_html/generalo/adat/" .
    session_id() . ".map");
```

## A szimbólum osztály (Symbol Object, symbolObj)

A következőkben néhány előre definiált szimbólum leírását láthatjuk, melyet a weboldalon vonalas jelként vagy piktogramként alkalmazhatunk. Példaként szaggatott vonalat, pontot, illetve egy hotel jel leírását mutatom be, melyeket a szimbólum osztályban (`symbolObj`) hoztam létre. A térképészetben gyakran használt sraff szimbólum ilyen módon nem hozható létre, mivel a MapScript nem ismeri a szimbólum osztály `HATCH` típusát.

```
1 $szaggatott = ms_newsymbolobj($map, "szaggatott");
2 $SzaggatottSymbol = $map->getsymbolobjectbyid($szaggatott);
3 $SzaggatottSymbol->set("type", MS_SYMBOL_ELLIPSE);
```

```

4  $SzaggatottSymbol->set("filled", MS_TRUE);
5  $SzaggatottSymbol->set("sizex", 1);
6  $SzaggatottSymbol->set("sizey", 1);
7  $SzaggatottSymbol->set("inmapfile", MS_TRUE);
8  $aPoints[0] = 1;
9  $aPoints[1] = 1;
10 $SzaggatottSymbol->setpoints($aPoints);
11 $aPattern[0] = 8;
12 $aPattern[1] = 8;
13 $SzaggatottSymbol->setpattern($aPattern);
14
15 $pont = ms_newsymbolobj($map, "pont");
16 $PontSymbol = $map->getsymbolobjectbyid($pont);
17 $PontSymbol->set("type", MS_SYMBOL_ELLIPSE);
18 $PontSymbol->set("filled", MS_TRUE);
19 $PontSymbol->set("sizex", 1);
20 $PontSymbol->set("sizey", 1);
21 $PontSymbol->set("inmapfile", MS_TRUE);
22 $aPoints[0] = 1;
23 $aPoints[1] = 1;
24 $PontSymbol->setpoints($aPoints);
25
26 $hotel = ms_newsymbolobj($map, "hotel");
27 $HotelSymbol = $map->getsymbolobjectbyid($hotel);
28 $HotelSymbol->set("type", MS_SYMBOL_PIXMAP);
29 $HotelSymbol->set("filled", MS_TRUE);
30 $HotelSymbol->setimagepath("/home/webgis2013/sztqadt/public_html/
    generalo/adat/szimbolum/hotel.gif");
31 $HotelSymbol->set("inmapfile", MS_TRUE);

```

## A réteg osztály (Layer Object, layerObj)

A térképet alkotó rétegeket ebben az osztályban (LayerObj) kezeltem. Egy for ciklussal az összes térképi réteg közül a `getExtent()` függvény segítségével az `$extent` tömbbe gyűjtöttem ki a minimum és maximum vetületi koordinátákat, melyekkel a felhasználó megkérdezése nélkül jeleníthettem meg a térképen ábrázolni kívánt területet, így sikerült valóban felhasználóbarát felületet létrehozni. Ugyanezzel a for ciklussal végighaladva az összes rétegen, a felhasználó által kiválasztott ábrázolási módszer alapján meghatároztam a réteg típusát, mely a MapScript tulajdonságai alapján lehet pont, vonal, diagram vagy felület. A rétegeknek egy `layer+"szám"` típusú nevet adtam, mellyel létrehoztam egy ilyen nevű változót. Ezekkel a változókkal különíthetem el a rétegeket, melyeknek tulajdonságként meg kell adni a hozzá tartozó shapefile elérési útvonalát, illetve a réteg típusát. A felhasználói felületen választhatjuk az

adatok diagrammódszerrel való megjelenítésénél a kör- vagy oszlopdiagramot. Ezeknek különböző beállításai vannak, melyeket a \$reteg['diagramtípus'] tartalmazza. A kördiagramnál meghatározható, hogy melyik mező alapján szeretnénk méretezni a köröket, megadva a minimális és maximális méretét és az ezen adatokhoz tartozó értékeket. Az oszlopdiagramnál az összes egymás mellett álló oszlop szélességét és az oszlop maximális magasságát kell megadni.

```

1 $extent=Array();
2 for($i=0; $i<=$_SESSION['retegszam']; $i++)
3 {
4     $reteg=$_SESSION['reteg'][$i];
5     if ($reteg['shape']=="pont" and ($reteg['modszerek']=="jelmodszerek1" or
        $reteg['modszerek']=="jelmodszerek2"))
6         $retegtipus=MS_LAYER_POINT;
7     if ($reteg['modszerek']=="diagrammodszerek")
8         $retegtipus=MS_LAYER_CHART;
9     if ($reteg['shape']=="vonal")
10        $retegtipus=MS_LAYER_LINE;
11    if ($reteg['shape']=="felulet" and $reteg['modszerek']!="
        diagrammodszerek")
12        $retegtipus=MS_LAYER_POLYGON;
13    $layername = "layer" . $i;
14    $$layername = ms_newLayerObj($map);
15    $$layername->set("status", MS_DEFAULT);
16    $$layername->set("type", $retegtipus);
17    $$layername->set("data", "/home/webgis2013/sztqadt/public_html/
        generalo/adat/SHP/" . $reteg['shape_file']);
18    $extent[]=$$layername->getExtent();
19    if ($reteg['modszerek']=="diagrammodszerek")
20        {
21            switch($reteg['diagramtípus'])
22                {
23                case "kordiagram":
24                    $diagramtípus="CHART_TYPE=pie";
25                    $diagrambeallitas="CHART_SIZE_RANGE={$reteg['mezo']} {
                        $reteg['min_meret']} {$reteg['max_meret']} {$reteg['
                        min_ertek']} {$reteg['max_ertek']}";
26                    break;
27                case "oszlopdiagram":
28                    $diagramtípus="CHART_TYPE=bar";
29                    $diagrambeallitas="CHART_SIZE={$reteg['szelesseg']} {
                        $reteg['max_magassag']}";
30                    break;
31                }

```

```

32     $$layername->setprocessing($diagramtipus);
33     $$layername->setprocessing($diagrambeallitas);
34     }
35 .
36 .
37 .
38     }
39 $minxa=Array();
40 $minya=Array();
41 $maxxa=Array();
42 $maxya=Array();
43 foreach($extent as $ext)
44     {
45         $minxa[]=$ext->minx;
46         $minya[]=$ext->miny;
47         $maxxa[]=$ext->maxx;
48         $maxya[]=$ext->maxy;
49     }

```

### Az osztály nevű osztály (Class Object, classObj)

Ebben az osztályban a térképi elemek kategorizálására szolgáló változók lettek átadva az *index.php*-ből. Itt adtam meg, hogy ha mennyiségi adatokat szeretnénk osztályokba sorolni, akkor azok kerüljenek be egy kategóriába, melyek az adatokat tartalmazó mezőben kisebb vagy egyenlő értékűek a felhasználó által megadott felső határértékkel. Az utolsó kategória felső határértékét a MapServer alapértelmezetten nem kéri, ekkor a be nem sorolt elemeket helyezi ebbe az osztályba. Emiatt, ha egyszerűsített alaptérképet szeretnénk a ponttal, vonallal vagy diagrammal megjelenített tematikus tartalomhoz, beleütközünk abba a problémába, hogy kettőnél kevesebb osztályt kartogrammszerűen nem definiálhatunk, így két kategória meghatározása után mindkettőnek ugyanazt a kitöltést adjuk. E megoldás, ha a térképen nem is, de a jelmagyarázatban látszani fog. Feltételek sokaságának bevezetésével ki lehet küszöbölni mindezt. Itt adtam át a kategóriák neveit is, melyeket a felhasználó beállíthat az alapértelmezett név átírásával. Mennyiségi adatok alapján való osztályozásnál a ciklus az utolsó kategória nevét nem állítja be, ezért azt külön végeztem el.

```

1     foreach($reteg['kat'] as $key => $val)
2     {
3         $oclass='class'.$key;
4         $$oclass=ms_newClassObj($$layername);
5         $$oclass->set('name', $reteg['katnev'][$key]);

```

```

6     if (($reteg['modszerek']=='kartogrammodszerek') or ($reteg['modszerek']
7         =='jelmodszerek2'))
8         $$oclass->setExpression("(".$reteg['mezo'].")<=" . $val['veg
9             ].");
10        if (($reteg['modszerek']=='feluletimodszerek') or ($reteg['modszerek']
11            =='jelmodszerek1'))
12            {
13                $$layername->set("classitem", $reteg['mezo']);
14                $$oclass->setExpression($val['minoseg']);
15            }
16        if (($reteg['modszerek']=='kartogrammodszerek') or ($reteg['modszerek']
17            =='jelmodszerek2'))
18            {
19                $oclass='class'.$reteg['kategoriak'];
20                $$oclass=ms_newClassObj($$layername);
21                $$oclass->set('name', $reteg['katnev'][$reteg['kategoriak']]);
22            }

```

## A stílus osztály (Style Object, styleObj)

Ez az osztály a kategóriák stílusait tartalmazza. A szimbólumokat kivéve, stílus megadásaként csak a kitöltéshez és körvonalhoz tartozó színt kértem. Az általam definiált colorRGB() függvénnyel alakítom át a felhasználó által a webes felületen kiválasztott hexadecimális színt RGB színmodellé, mert a MapServer csak ilyen formátumba képes felismerni azt. A honlap felületén a JSColor [ODVÁRKO, 2012] nevű JavaScriptben írt színválasztót használtam. Szimbólumoknál a méretet és a vonalvastagságot is bekéri az alkalmazás.

```

1     foreach($reteg['stilus'] as $key => $val)
2         {
3             $name = 'style'.$key;
4             $class = 'class'.$key;
5             $$name=ms_newStyleObj($$class);
6             if (isset($val['color']))
7                 {
8                     $color = colorRGB($val['color']);
9                     $$name->color->setRGB($color['r'], $color['g'], $color['b'])
10                        ;
11                }
12            if (isset($val['outlinecolor']))
13                {
14                    $outlinecolor = colorRGB($val['outlinecolor']);

```



```

14         $$name->outlinecolor->setRGB($outlinecolor['r'],
           $outlinecolor['g'], $outlinecolor['b']);
15     }
16     if ($reteg['modszere']=="diagrammodszere")
17         $$name->setbinding('SIZE', $reteg['kat'][$key]['ertek']);
18     if (isset($reteg['stilus'][$key]['szimbolum']))
19         $$name->set("symbolname", $reteg['stilus'][$key]['szimbolum'])
           ;
20     if (isset($reteg['stilus'][$key]['szimbolummeret']))
21         $$name->set('size', $reteg['stilus'][$key]['szimbolummeret']);
22     if (isset($reteg['stilus'][$key]['vonaltastagsag']))
23         $$name->set("width", $reteg['stilus'][$key]['vonaltastagsag'])
           ;
24     }

```

## A térkép fokhálózata

Egy új réteg hozzáadásával lehet létrehozni a térkép fokhálózatát. Ezt az *index.php* oldalon pipálással lehet hozzárendelni a térképhez. A rétegen belül létrehoztam egy osztályt, melynek a stílus objektumában definiáltam a fokhálózat vonalainak színét. A réteg tulajdonságait statikusan állítottam be, így a térképen minimum 2, maximum 30 vonalból álló fokhálózat jelenik meg.

```

1 if($_SESSION['fokhalozat']=='1')
2 {
3     $gridlayer = ms_newLayerObj($map);
4     $gridlayer->set("name", "GRID");
5     $gridlayer->set("status", MS_DEFAULT);
6     $gridlayer->set("type", MS_LAYER_LINE);
7     $gridlayer->connectiontype = MS_GRATICULE;
8     $fokhalo=ms_newClassObj($gridlayer);
9     $fokhalo->set("name", "Fokhalozat");
10    $halo=ms_newStyleObj($fokhalo);
11    $halo->color->setRGB(0,0,255);
12    ms_newgridobj($gridlayer);
13    $gridlayer->grid->set("minsubdivide", 5);
14    $gridlayer->grid->set("maxsubdivide", 15);
15    $gridlayer->grid->set("minarcs", 2);
16    $gridlayer->grid->set("maxarcs", 30);
17    $gridlayer->grid->set("labelformat", "DDMMSS");
18 }

```

## 6. fejezet

# Összefoglalás

Dolgozatomban átfogó képet adtam a mai webes tematikus térképgenerálás állásáról, hibáiról. Ezekből okulva és más módszert használva készítettem egy saját internetes alkalmazást, mellyel szintén versenyképes térképet lehet generálni. Itt arra törekedtem, hogy a többi alkalmazástól eltérően az ESRI Shapefile-t használjam az elterjedt KML, CSV vagy TXT szüvegfájlok helyett.

Az általam készített alkalmazás jól használható az oktatásban, azt szórakoztatóvá is teheti. A térkép elkészítésével megjelenő Mapfile tanulmányozása is sokat segít a MapServer elsajátításában.

Ezt a módszert természetesen számtalan módon lehetne továbbfejleszteni, hogy még egyszerűbb legyen a kezelőfelülete, még átláthatóbb legyen. Meg lehetne oldani, hogy a rétegek sorrendje utólag is módosítható legyen, illetve sűrű funkciókat beépítve a kevésbé hozzáértők is könnyebben kezelhetnék az alkalmazást. További bemeneti fájlformátumok támogatása lehetséges. Külső térképszolgáltató használatával megoldható lenne a háttértérkép biztosítása a tematikus tartalomhoz.

Mint láthattuk, ez is kiválóan megfelelő módszer arra, hogy ingyen készíthessünk térképet, akár szoftver telepítése nélkül is. Térképet szerkeszthetünk bárhol, ahol internetkapcsolattal rendelkezünk, és bármiről, legyen az számítógép, táblagép vagy okostelefon, mivel nincs szükségünk semmilyen kompatibilitásra.

## 7. fejezet

# Köszönetnyilvánítás

Köszönettel tartozom elsősorban témavezetőmnek, Dr. Gede Mátyásnak a fent említett módszerbe való bevezetésért, illetve diplomamunkám elkészítéséhez nyújtott támogatásáért. Továbbá köszönöm Kádár Ivánnak a weboldallal kapcsolatos hasznos ötleteket. Végül, de nem utolsó sorban, köszönetet mondok családomnak a tanulmányaim során nyújtott szerető támogatásért.

# Irodalomjegyzék

- [ADOBE SYSTEMS, 2013] ADOBE SYSTEMS, Incorporated (2013): About Adobe PDF, [www.adobe.com/products/acrobat/adobepdf.html](http://www.adobe.com/products/acrobat/adobepdf.html) (utolsó elérés: 2013. május 30.)
- [BARTOS, 2007] BARTOS-ELEKES, Zsombor (2007): Bevezetés a térképészetbe, Presa Universitară Clujeană, Cluj-Napoca, pp. 137-160., ISBN: 978-973-610-619-4.
- [COMPU SERVE, 1990] COMPU SERVE, Incorporated (1990): Graphics Interchange Format Version 89a. Programming Reference, Columbus, Ohio, [www.w3.org/Graphics/GIF/spec-gif89a.txt](http://www.w3.org/Graphics/GIF/spec-gif89a.txt) (utolsó elérés: 2013. május 30.)
- [ESRI, 1998] ESRI, Incorporated (1998): ESRI Shapefile Technical Description. An ESRI White Paper, [www.esri.com/library/whitepapers/pdfs/shapefile.pdf](http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf) (utolsó elérés: 2013. május 30.)
- [GOOGLE, 2012] GOOGLE (2012): KML Tutorial, [https://developers.google.com/kml/documentation/kml\\_tut](https://developers.google.com/kml/documentation/kml_tut) (utolsó elérés: 2013. május 30.)
- [GOOGLE, 2013] GOOGLE (2013): Fusion Tables, [www.google.com/fusiontables](http://www.google.com/fusiontables) (utolsó elérés: 2013. május 30.)
- [HAMILTON, 1992] HAMILTON, Eric (1992): JPEG File Interchange Format Version 1.02, C-Cube Microsystems, [www.jpeg.org/public/jfif.pdf](http://www.jpeg.org/public/jfif.pdf) (utolsó elérés: 2013. május 30.)
- [DENT, 2009] DENT, Borden D. (2009): Introduction to virtual and web mapping, In: DENT, Borden D. (szerk.): Cartography: Thematic Map Design, 6. kiadás, McGraw-Hill Companies, Inc., New York, pp. 280-297., ISBN: 978-0-07-294382-5.
- [JAVASCRIPT, 2003] JAVASCRIPT (2003): JavaScript Reference, [www.java2s.com/Code/JavaScriptReference/CatalogJavaScriptReference.htm](http://www.java2s.com/Code/JavaScriptReference/CatalogJavaScriptReference.htm) (utolsó elérés: 2013. május 30.)

- [KLINGHAMMER, 2010] KLINGHAMMER, István (2010), szerk.: Térképészet és geoinformatika I., ELTE Eötvös Kiadó, Budapest, p. 193., ISBN: 978-963-312-027-9.
- [ODVÁRKO, 2012] ODVÁRKO, Jan (2012): JSColor, <http://jscolor.com/> (utolsó elérés: 2013. május 30.)
- [OLSON, 2013] OLSON, Philip (2013): PHP Manual, <http://php.net/manual/en/index.php> (utolsó elérés: 2013. május 30.)
- [RITTER és RUTH, 2000] RITTER, Niles, & RUTH, Mike (2000): GeoTIFF Format Specification. GeoTIFF Revision 1.0, [www.remotesensing.org/geotiff/spec/contents.html](http://www.remotesensing.org/geotiff/spec/contents.html) (utolsó elérés: 2013. május 30.)
- [SIKI, 2003] SIKI, Zoltán (2003): Térképek internetes publikálása - Lehetőségek, megoldások, [www.agt.bme.hu/maps/im.html](http://www.agt.bme.hu/maps/im.html) (utolsó elérés: 2013. május 30.)
- [STEPMAP, 2013] STEPMAP (2013): StepMap. Design your map, [www.stepmap.com](http://www.stepmap.com) (utolsó elérés: 2013. május 30.)
- [DE LA TORRE és ÁLVAREZ, 2012] DE LA TORRE, Javier, & ÁLVAREZ, Sergio (2012): What is CartoDB?, <http://cartodb.com/media> (utolsó elérés: 2013. május 30.)
- [UNIVERSITY OF MINNESOTA, 2013] UNIVERSITY OF MINNESOTA (2013): MapServer 6.2.1 Documentation, [www.mapserver.org/documentation.html](http://www.mapserver.org/documentation.html) (utolsó elérés: 2013. május 30.)
- [UNIVERSITY OF MINNESOTA, 2013] UNIVERSITY OF MINNESOTA (2013): PHP MapScript, [www.mapserver.org/mapscript/php/](http://www.mapserver.org/mapscript/php/) (utolsó elérés: 2013. május 30.)
- [W3C, 2003] WORLD WIDE WEB CONSORTIUM (W3C) ajánlása (2003): Portable Network Graphics (PNG) Specification (Second Edition), [www.w3.org/TR/PNG/](http://www.w3.org/TR/PNG/) (utolsó elérés: 2013. május 30.)
- [W3C, 2005] WORLD WIDE WEB CONSORTIUM (W3C) ajánlása (2005): Scalable Vector Graphics (SVG) Full 1.2 Specification, [www.w3.org/TR/SVG12/](http://www.w3.org/TR/SVG12/) (utolsó elérés: 2013. május 30.)
- [W3C, 2011] WORLD WIDE WEB CONSORTIUM (W3C) ajánlása (2011): Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification, [www.w3.org/TR/CSS2/](http://www.w3.org/TR/CSS2/) (utolsó elérés: 2013. május 30.)
- [W3C, 2013] WORLD WIDE WEB CONSORTIUM (W3C) ajánlása (2013): HTML5 Reference, <http://dev.w3.org/html5/html-author/> (utolsó elérés: 2013. május 30.)

# Ábrák jegyzéke

2.1. ábra: Jelmódszerrel készített térkép

CARTOGRAPHIA KFT. (2001): A közúthálózat forgalma. In: Középiskolai földrajzi atlasz, Cartographia Kft., Budapest, p. 25., ISBN: 963-352-553-5.

2.2. ábra: Felületi módszerrel készített térkép

CARTOGRAPHIA Kft. (2001): Közigazgatás (1990-1998). In: Középiskolai földrajzi atlasz, Cartographia Kft., Budapest, p. 43., ISBN: 963-352-553-5.

2.3. ábra: Kartogram- és diagrammódszerrel készített térkép

CARTOGRAPHIA KFT. (2001): Az ipar területi szerkezete. In: Középiskolai földrajzi atlasz, Cartographia Kft., Budapest, p. 20., ISBN: 963-352-553-5.

2.4. ábra: Izovonal- és mozgásvonal módszerrel készített térkép

CARTOGRAPHIA KFT. (2001): Évi csapadékmennyiség és az uralkodó szélirányok. In: Középiskolai földrajzi atlasz, Cartographia Kft., Budapest, p. 17., ISBN: 963-352-553-5.

4.1. ábra: A MapServer felépítése

[http://mapserver.org/\\_images/architecture.png](http://mapserver.org/_images/architecture.png), fordította: Szabó Tímea, 2013 (utolsó elérés: 2013. május 30.)

4.2. ábra: A MapScript felépítése

The MapScript API's hierarchical object structure. In: MITCHELL, Tyler (2005): Web Mapping Illustrated: Using Open Source GIS Toolkits, O'Reilly Media, Inc., Sebastopol, p. 278., 14-1. ábra, ISBN: 978-0-596-00865-9.

5.1. ábra: 1. lépés: A shapefile feltöltése

Szabó Tímea, 2013

5.2. ábra: 2. lépés: A térkép általános beállításai

Szabó Tímea, 2013

5.3. ábra: 3. lépés: A shapefile nevének és típusának kiválasztása

Szabó Tímea, 2013

5.4. ábra: 3. lépés: Kategóriák stílusa általános esetben

Szabó Tímea, 2013

5.5. ábra: 3. lépés: Kategóriák stílusa oszlopdiagram esetén

Szabó Tímea, 2013

5.6. ábra: 4. lépés: Ellenőrzés és térkép

Szabó Tímea, 2013

5.7. ábra: Kartogrammódszerrel készített térkép

Szabó Tímea, 2013

5.8. ábra: Kartogram- és diagrammódszerrel készített térkép

Szabó Tímea, 2013

## CD-melléklet tartalma

### 1. A *Honlap* mappa tartalma:

- *alap.map*
- *hotel.gif*
- *index.php*
- *kordiagram.png*
- *map.php*
- *oszlopdiaqram.png*
- *pont.jpg*
- *stilus.css*
- *szaggatott.jpg*
- *tab\_aktiv.png*
- *tab\_aktualis.png*
- *tab\_hatralevo.png*
- *w3c-valid-html5.png*
- *adat mappa*
- *jscolor mappa*

### 2. A *Diplomamunka* mappa tartalma:

- *Diplomamunka\_Szabo\_Timea\_2013.pdf*

# NYILATKOZAT

Alulírott **Szabó Tímea** (ETR azonosító: **SZTQADT.ELTE**) nyilatkozom, hogy jelen dolgozatom teljes egészében saját, önálló szellemi termékem. A dolgozatot sem részben, sem egészében semmilyen más felsőfokú oktatási vagy egyéb intézménybe nem nyújtottam be. A diplomamunkámban felhasznált, szerzői joggal védett anyagokra vonatkozó engedély a mellékletben megtalálható.

A témavezető által benyújtásra elfogadott diplomamunka PDF formátumban való elektronikus publikálásához a tanszéki honlapon.

HOZZÁJÁRULOK

NEM JÁRULOK HOZZÁ

Budapest, 2013. június 7.

.....

*a hallgató aláírása*