

# Modulfejlesztés QGIS környezetben monitoring adatok feldolgozásához

SZAKDOLGOZAT

KÉSZÍTETTE:

**Urbán Eszter Klára**  
Programtervező Informatikus hallgató

TÉMAVEZETŐ:

**Dr. Balla Dániel Zoltán**  
egyetemi adjunktus

Debrecen

2024

# Tartalomjegyzék

1. Bevezető.....	1
1.1. Téma fontossága .....	1
1.2. A térinformatikai feladatok automatizálásának fontossága.....	2
1.3. Célok .....	2
2. Elméleti háttér .....	4
2.1. A térinformatika .....	4
2.1.1. Alkalmazási terület .....	4
2.1.2. Térinformációs rendszerek .....	6
2.1.3. Adatmodellek .....	7
2.1.4. Adattípusok .....	8
2.2. Víztisztaság index .....	9
2.2.1. Felhasználási terület.....	10
2.2.2. Előnyök és hátrányok.....	10
2.3. Adatbázisok .....	11
2.3.1. Globális .....	11
2.3.2. Regionális.....	12
3. Módszerek és Algoritmusok .....	13
3.1. QGIS.....	13
3.2. Felhasznált adatok.....	13
3.3. Modul fejlesztés .....	14
3.4. Víztisztaság indexek.....	17
3.5. QGIS-ben való automatizálás .....	19
3.5.1. Kategorizálás WQS értékek alapján és interpolálás .....	20
4. Eredmények.....	22
4.1. A QGIS modul működésének bemutatása.....	22
4.2. Interpoláció és plusz modulok .....	32
4.2.1. Interpoláció.....	32
4.2.2. OSM.....	34
4.3. Teszteredmények és tematikus térképek .....	36
5. Összegzés .....	40
6. Irodalomjegyzék .....	42

# 1. Bevezető

Szakedolgozatom a talajvíz minták minőségének kiszámításához fejlesztett modullal foglalkozik az egyik legelterjedtebb térinformatikai programcsomag felhasználásával. A Quantum GIS egy térinformatikai rendszer térképek elemzésére és téradatokkal kapcsolatos műveletek elvégzésére. A térinformatika már a tanulmányaim során felkeltette az érdeklődésemet és amikor szakdolgozat témát kellett választani azonnal megragadta a figyelmemet ez az terület. A fejlesztett modult magam írtam meg Python nyelven és a fejlesztői felületet is magam készítettem el hozzá.

Dolgozatomban először is a témát szeretném jobban bemutatni, valamint, hogy milyen technológiákat használtam fel a fejlesztés alatt. A vizek minőségindexeit is ismertetem a későbbiekben. Ezután ismertetem magát a programkódot is és annak működését. Ezt követően az eredményül kapott térképek (tematikus pont és eloszlástérképek) is bemutatásra kerülnek.

## 1.1. Téma fontossága

A vizek minőségének ismerete globális szinten fontos az emberiségnek. Nemcsak az egészség megőrzése érdekében, de a környezetvédelem, valamint a fenntarthatóság is fontos szemponttá vált az utóbbi időkben.

- **Egészségügyi szempontból** a tiszta ivóvíz nélkülözhetetlen a földön élők számára. Az rossz minőségű víz, például szennyezett vagy fertőzött víz fogyasztása súlyos egészségügyi problémákat okozhat.
- **Környezetvédelmi szempontból** a vízminőség alapvető fontosságú az ökoszisztémák számára, beleértve a folyókat, tavakat és a tengeri környezetet egyaránt.
- **Fenntarthatóság szempontjából** a tiszta víz elengedhetetlen az élelmiszertermeléshez és az ipari folyamatokhoz. A fenntartható vízgazdálkodás segít megőrizni a vízkészleteket, ami alapvető fontosságú az emberiség és a környezet számára.
- **Gazdasági szempontból** az ipari folyamatok és a mezőgazdaság is jelentős mennyiségű vizet használ, és a vízminőség befolyásolja ezeknek az ágazatoknak a hatékonyságát és fenntarthatóságát.

- **Klímaváltozás és természeti katasztrófák**, mint például árvizek, jelentős hatással lehetnek a vízminőségre mivel megnőhet a vízszennyezés kockázata.

## 1.2. A térinformatikai feladatok automatizálásának fontossága

A térinformatika olyan terület, amely térbeli adatok gyűjtésével, tárolásával, elemzésével és megjelenítésével foglalkozik. Az automatizálásnak ezen a területen számos előnye van, amelyek hozzájárulnak a hatékonysághoz, a pontossághoz és a döntéshozatal javításához.

Az automatizálás lehetővé teszi az ismétlődő és rutinszerű feladatok gyors és pontos végrehajtását, csökkentve ezzel az emberi hibák és az időveszteség lehetőségét. Az automatizált rendszerek segítik az adatok egységesítését és integrációját, amely fontos a térbeli adatok összekapcsolásában és összehasonlításában. Az automatizált eszközök fejlett térbeli modellezési lehetőségeket kínálnak, amelyek segítik a valósághűbb elemzéseket.[1]

## 1.3. Célok

Célom a munkámmal, hogy a szakirodalomban legelterjedtebb vízminőség indexek kiszámítását megkönnyítsem és felgyorsítsam. A megfelelő adatokkal (indexérték, X és Y koordináta, pH, EC,  $\text{NH}_4^+$ ,  $\text{NO}_2^-$ ,  $\text{NO}_3^-$ ,  $\text{PO}_4^{3-}$ , COD,  $\text{Na}^+$  és az évszám amikor a mérés készült) gyors és egyszerű használat mellett pontos és könnyen értelmezhető végeredményt kapjunk vissza. Célom volt továbbá, ha az adatok különböző időpontokból állnak rendelkezésre és ezek fel vannak tüntetve minden évhez külön térképet készítsen a program. Mivel a QGIS lehetővé teszi saját modulok készítését így ez is ott készül Python nyelven. Bemenete egy excel fájl, illetve a kimeneti könyvtár, ahova a végeredményt fogja elkészíteni. A térképen megjelenő mérési eredmények, vízminőség szerint különböző színekkel lesznek ellátva (Kiváló vízminőség – világos zöld, Jó vízminőség – zöld, Rossz vízminőség – sárga, Nagyon rossz vízminőség – narancssárga, Bármilyen használatára alkalmatlan - piros). Ezek az eredmények a WQS (vízminőségi státusz) értékeket fogják ábrázolni. Ezen pontok alapján egy interpolációt is készít, ami az egész területet lefedi és megmutatja mekkora részen milyen minőségű a víz. A modulban több számolás is végbemegy és eredményei az attribútum táblában megtekinthetők. Ilyen eredmény lesz a WQI (Water Quality Index), ami egy általános vízminőség index. A WQS (Water Quality Status), WQS Rank a WQI értéke

alapján meghatározza milyen kategóriába esik az adott minta, szöveges és 1-től 5-ig terjedő numerikus értékkel. A Cd (Contamination degree) a talajvízben található káros anyagok együttes hatását írja le. Cds, Cds Rank, valamint a lehetséges használat pedig ezt egyszerűsíti szöveges és 1-től 5-ig terjedő numerikus értékkel.

## 2. Elméleti háttér

### 2.1. A térinformatika

A legáltalánosabb megfogalmazásban a térbeli adatok informatikai eszközökkel történő gyűjtésével, az adatok feldolgozásával kapott térbeli információk előállításával, ezek rendszerezésével és tárolásával, valamint a tér adatok és információk közzétételével foglalkozó tudományterületek összességét jelenti.

A geoinformatika az interdiszciplináris megközelítés révén összekapcsolja a földtudományokat, elősegítve ezen területek szoros együttműködését és új szintézisek létrehozását. A térinformatika alapvető elemei az adatgyűjtés, adattárolás, adatkezelés, adatelemzés és a megjelenítési folyamatok.[2]

A térinformatika számos tudományág és iparág területén alkalmazható, mint például a földtudományok, városi tervezés, közlekedési tervezés, környezetvédelem és mezőgazdaság. A térinformatikai rendszerek segítségével lehetőség van térképek, adatbázisok és egyéb eszközök használatával komplex térbeli problémák kezelésére és megértésére. A térinformatika a modern technológia, például a számítógépek, a műholdas képalkotás és a térbeli adatbázisok fejlődésével egyre fontosabb szerepet játszik a különböző tudományágakban és iparágakban.

#### 2.1.1. Alkalmazási terület

A tevékenységek jellege szerinti elkülönítésében három főbb alkalmazási csoportot különböztethetünk meg.

Az első csoportba tartoznak a hosszabb idő óta fejlesztett technológiák, amelyek a térinformatikával a közös eljárásaik és adatszolgáltatásuk révén vannak kapcsolatban. Ilyen a földmérés és más mérnöki tevékenységek, ahol a technológiai fejlődéssel hétköznapivá vált a digitális mérőállomás, a lézerszkennerek, a GPS és a hozzá tartozó GNSS hálózat használata. Ez a folyamat jellemzi a térképészet témakörét is; itt a térképkészítés, az automatikus térképezés fejlesztése emelhető ki. A távérzékelés a felvételező érzékelők fejlesztésének köszönhetően lett progresszív irányvonal.

Második kategória a menedzsment és a döntéshozatal csoportja, mely a legtöbb alkalmazást tudhatja magáénak. Legtipikusabb témakörei a természeti erőforrások

nyilvántartása és adatkezelése, a tulajdon ellenőrzése és az adózás miatt fontos földhivatali-nyilvántartás, illetve a műszaki GIS egyik legnagyobb alkalmazóihoz tartozó közműnyilvántartás.

Harmadik csoport a tudományos és kutatási tevékenység, melynek hazai színterei az egyetemek, laboratóriumok és más különböző intézmények. Hazánkban elsősorban a műszaki térinformatikához és a természeti erőforrásokhoz kötődő problémák, a földrajzi folyamatok tudományos feldolgozása jellemző.[3]

A térinformatika gyakorlati alkalmazása kiemelkedő fontossággal bír városi tervezés és fejlesztés területén. Ezen tudományág lehetővé teszi a városi területek alapos elemzését, infrastruktúra tervezését és a városi környezet hatékonyabb kezelését. Térinformatikai eszközök segítségével lehetőség nyílik a földhasználat hatékony tervezésére és rendszerezésére, kiterjedve a mezőgazdasági, erdészeti, ipari és más területekre is.[4]

Az ökológiai rendszerek elemzése és természeti erőforrások monitorozása terén is nélkülözhetetlen a térinformatika alkalmazása. A környezetvédelmi tervezésben a térinformatika segítségével lehetőség nyílik a fenntartható tervezésre és a környezet védelmére.

A katasztrófavédelem területén a térinformatika rendkívül hasznos, hiszen lehetőséget teremt a gyors és hatékony döntéshozatalra sürgősségi helyzetekben, segítve a katasztrófavédelmi tervek kialakítását.

A közlekedési hálózatok tervezése, forgalomkezelés és logisztikai folyamatok optimalizálása terén is kiemelkedő szerepet játszik a térinformatika. GPS-alapú útvonaltervezés segítségével könnyedén optimalizálhatók a közlekedési folyamatok és javítható a forgalom irányítása.[5]

A telekommunikációs hálózatok tervezése és optimalizálása terén a térinformatika elengedhetetlen eszköz, beleértve a mobiltelefonhálózatok lefedettségének tervezését és optimalizálását is.

Az egészségügyi szolgáltatások tervezésében és optimalizálásában, például kórházak elhelyezkedésének vagy járványok terjedésének elemzésével a térinformatika hozzájárul a hatékonyabb egészségügyi infrastruktúra kialakításához.

A bányászati területek feltérképezése, a kőolaj- és gázkitermelés tervezése, valamint az alternatív energiaforrások térbeli elemzése terén is kiemelkedő szerepet játszik a térinformatika, elősegítve a fenntartható és hatékony erőforrás-használatot.[6]

### 2.1.2. Térinformációs rendszerek

Első rövidebben bemutatásra kerülő szoftver az ESRI ArcGIS. A legtöbb térinformatikával foglalkozó szakember kapcsolatba kerül vele és használja a mindennapokban. Az ESRI az 1990-es évek elején az ArcView 1.0, 3.0 verzióival sikeresen hozta át a geoinformatikát az elavultabb UNIX-os környezetből a modernebb Windows-os környezetbe. Így ez egérrel, grafikus felületen keresztül kezelhetőbb, felhasználóbarátabb lett. Az ArcView 8.0-s változata több modulból áll, (ArcCatalog, ArcMap, ArcToolbox) és az egyes modulok is úgynevezett plugin-okkal bővíthetők, testreszabhatóak.

A program eredetileg inkább vektoros alapú volt, de mára raszteres feldolgozásra is alkalmas. Térképeinkhez ma már többféle típusú adatbázis köthető, habár korábban úgynevezett shapefile-okat hozott létre a rendszer, mely valójában három állományból állt, és saját adatbázisból dolgozott.

A GRASS (Geographic Resources Analysis Support System) egy UNIX/LINUX alapú, raszteres-vektoros kombinált GIS alkalmazás. A fejlesztése még 1982-ben kezdődött, az Amerikai Egyesült Államok hadseregének kutató-fejlesztő részlegén. A szabadszoftverré válás még szélesebb körben terjesztette el a programot. Létezik már Windows alatt emulátor segítségével használható verziója is, de sokkal több speciális tudást és informatikai szakértelmet kíván meg a felhasználoktól, mint a korábban említett rendszerek, azonban képességeiben azoktól semmiben sem marad el.

Quantum GIS (QGIS) egy felhasználóbarát, nyílt forráskódú geoinformatikai szoftver, amely GNU General Public License alatt érhető el. A fejlesztése hivatalosan 2002-ben kezdődött, eredetileg egyszerűbb GIS böngészőprogramnak szánták, később fejlődött tovább, mostanra Linuxra, Windowsra és Mac OSX operációs rendszerre is kiadták. Jelenleg alkalmas más programokkal létrehozott adatrendszerek böngészésére, önálló GIS létrehozására, lekérdezések készítésére, valamint fejlett vektoros analízáló eszközkészlettel rendelkezik.[7]

Ha a programban nem találunk megfelelő plugin-t a feladatunk végrehajtására Python vagy C++ nyelven tudunk saját modulokat is írni hozzá. Emiatt a funkció miatt esett a választásom erre a rendszerre amikor a dolgozatomat elkezdtem.

A felületen egyszerre több réteget is meg lehet jeleníteni, valamint a számításaink eredményeit is. Képesek vagyunk pontokat, vonalakat elhelyezni a térképeken. Nagy előnye, hogy online térképeket is meg lehet jeleníteni rajta, ilyen például a OSM (Open street Map) amit a későbbikében én is használni fogok.



Többféle raszteres állománytípust is tud kezelni (többek közt: rgb, gif, jpg, tiff, bmp, asc). Képes adatbázis kapcsolatot teremteni Oracle Spatial, PostgreSQL és MySQL adatbázisokkal. Amennyiben saját magunk definiálunk vektor réteget, shape formátumban történik az adatok tárolása.[7]

### 2.1.3. Adatmodellek

A térinformatikai adatmodellek a térbeli adatok leírására, tárolására, és kezelésére szolgálnak. Ezek az adatmodellek segítenek strukturált kereteket kialakítani a térbeli információkhoz, és meghatározzák azok kapcsolatát és szerveződését.

- **A vektoros modellek** lényege, hogy az ábrázolandó területet és a rajta lévő objektumokat pontok és a köztük lévő egyenesek együtteseként fogja fel. Ezen az elven az sem változtat, hogy a legtöbb rendszer alkalmas szabályos ívek generálására is a pontok között, mivel az ívek is elképzelhetők differenciálisan kicsiny egyenes darabokból (húrokból) alkotott poligonokként.[8] Gyakran használt térbeli információk tárolására, például települések, utak, vagy területek reprezentálására.
- **A raszteres adatmodell** alapvető, kisebb részekre oszthatatlan elemére egy angol eredetű szót használunk: ez a pixel, amely az angol "picture element" szavakból ered. Magát a modellt úgy kell elképzelnünk, hogy a kiválasztott területet egy óriási négyzethálóra fedjük. Egy tereptárgy több kiségyzethálóból (pixelből) állhat, ezeket a pixeleket színek kódok különböztetik a környezetén levő pixelektől. Ezt az adatmodellt elsősorban a légi és műholdfelvételek digitális formátumának tekinthetjük.[9]
- **A térbeli modell** esetén térbeli információkat tárolnak, és lehetőség van térbeli lekérdezések és analízisek készítésére. Olyan alkalmazásokban használják, ahol a térbeli és attribútum adatokat egyesítve kell kezelni, például városi tervezés vagy földhasználati tervezés.
- **Az objektum-relációs adatmodell** olyan relációs adatbázist alkalmaz, amely a vektoros adatmodelleket és objektumorientált elemeket egyesíti. Térbeli objektumok és azok attribútumai közötti kapcsolatokat leíró alkalmazásokban használják.
- **A szemantikus adatmodell** a térbeli objektumok közötti szemantikus kapcsolatokat is leírja, és lehetővé teszi azok jelentésének mélyebb megértését.

Komplex térinformatikai alkalmazásokban, ahol a térbeli objektumok közötti kapcsolatok fontosak a döntéshozatalban.

Ezek az adatmodellek különböző alkalmazási területeken használhatók, és segítik a térbeli információk hatékony kezelését és értelmezését a térinformatikában. Az alkalmazások céljától és az adatok komplexitásától függően választhatók ki a legmegfelelőbb adatmodell típusai.

#### **2.1.4. Adattípusok**

Az shp (shapefile) fájl tartalmazza az objektumokat leíró geometriai adatokat. Az shx egy index állomány, az shp fájl adataihoz. Az adatbázist és a geometriai adatokat tároló shp állományt köti össze. A dbf pedig egy adatbázis állomány. A térképi objektumhoz tartozó tematikus információkat, az attribútum adatokat tartalmazza. A cpk (Character Set Encoding) fájl definiálja a karakterkódolást, amelyet a shapefile attribútumtáblájában használnak. A prj (Projection) fájl tartalmazza a térbeli referencia információkat, beleértve a térképi projekciót és a koordináta-rendszert. Ezek a beállítások határozzák meg, hogy a térbeli adatok milyen módon vannak vetítve a síkra, és hogyan rendezik a koordinátákat.

A térinformatikai szoftverekben egyéb a térképen megjelenő kiegészítéseket is létre lehet hozni. Ilyen a pont, ami egyetlen térbeli pontot reprezentál, amelynek koordinátái adott rendszerben vannak megadva. A vonal egy egyenes vagy görbült vonalat reprezentál, amely lehetővé teszi a két pont közötti kapcsolat ábrázolását. A poligon zárt alakzatot reprezentál, amelynek körvonalát vonalak alkotják. A poligonoknak van területük és körvonaluk. A térbeli kiterjedés az adott térbeli régiót határozza meg, általában egy téglalap vagy doboz formájában, és koordinátákkal van meghatározva.[10]

A térinformatikai raszter egy olyan adattípus, amely a térről való információkat rács formájában tárolja. Minden rács cella tartalmazhat egy értéket, például magasságot vagy hőmérsékletet. A Domborzati modellek (Digital Elevation Model - DEM): A földfelszín domborzatát reprezentáló raszter adattípus. Minden rács cella a tereptárgy magasságát tartalmazza. A térinformatikai térkép olyan adattípus, amely a térbeli objektumokat és azok attribútumait kombinálja egy grafikus reprezentációval.

- A fentebb említett raszteres állománytípusok mint például a GIF (Graphics Interchange Format) képformátum, amely 256 szín mélységet támogat és lehetővé teszi az animációkat is. Leggyakrabban egyszerűbb grafikus elemek, ikonok, vagy alacsony színfelbontású képek tárolására használható.

- A JPEG (Joint Photographic Experts Group) tömörített képformátum, amely lehetővé teszi a magas minőségű fotók tárolását. Elterjedt digitális fényképek és más színben gazdag képek tárolására.
- A TIFF (Tagged Image File Format) rugalmas és sokoldalú képformátum, amely támogatja a tömörítést és magas színbontást. Jól használható nyomdai minőségű képek és képmanipulációs feladatokhoz.
- A BMP (Bitmap) egyszerű, nem tömörített képformátum, ami a pixel értékeit közvetlenül tárolja. Windows rendszereken gyakran használják, viszont nagy méretük és a tömörítés hiánya miatt kevésbé hatékonyak.
- Az ASC (ASCII) szöveges fájlformátum, amely ASCII karaktereket használ a magassági értékek tárolására. Földi magassági adatok tárolására alkalmas, például digitális domborzati modellekben.

## 2.2. Vízminőség index

Brown R. M. és munkatársai (1970) által kidolgozott súlyozott átlagolással készült vízminőség indexét (Water Quality Index, WQI) az elmúlt évtizedekben számos alkalommal módosították. A WQI-t alapul véve dolgozták ki többek között a US National Sanitation Foundation Water Quality Index-et (NSFWQI), a Canadian Water Quality Index-et (Canadian Council of Ministers of the Environment (CCME)). A felszín alatti vizek jó és gyenge minőség közötti kategorizálására számos módszertani megközelítés született. Mivel a vízminőségi állapot számos fizikai, kémiai és biológiai paraméterrel írható le, a nagy mennyiségű adat jelentősen megnehezíti az értékelést és az összehasonlítást. A probléma megoldása érdekében Horton (1965) javasolta az első vízminőség index bevezetését 10 fontos vízkémiai paraméter alapján. A különféle szennyezettségi indexek legfontosabb előnye a kémiai, fizikai és biológiai paraméterek egyetlen számmá történő összevonása, amely a vízminőségre vonatkozó információkat közérthetővé teszi a nyilvánosság és a politikai döntéshozók számára is.[11]

A WQS (Water Quality Status) WQI értéke alapján mondja meg milyen kategóriába esik az adott víz, szövegesen megfogalmazva a WQS Rank pedig ezt írja le 1-től 5-ig terjedő skálával. A Cd (Contamination degree) a talajvízben található káros anyagok szennyezésének

együttes mértékét írja le. Cds, Cds Rank ugyanúgy, mint az előző esetben a Cd-t egyszerűsítik szöveges és skálás formába.

### **2.2.1. Felhasználási terület**

A vízminőség indexek segítségével figyelemmel kísérhető a víztestek minősége az idő múlásával. Ez lehetővé teszi a víztestek változásainak követését és az esetleges problémák korai azonosítását.

Alkalmazhatók a vízbázisok ésszerű és fenntartható kezelésére. A vízminőség-monitorizált eredményeinek felhasználásával meghatározhatók a megfelelő vízhasználati intézkedések és gyakorlatok.

Alkalmazhatók a vízi ökoszisztémák állapotának felmérésére és védelmére. Emellett az állatok és növények, valamint az emberi életminőség szempontjából fontos ökológiai folyamatok elemzésére használhatók. Segítenek értékelni, hogy a víz milyen mértékben felel meg az emberi fogyasztásra vonatkozó egészségügyi szabványoknak. Ez közegészségügyi szempontból elengedhetetlen.[12]

Az ipari vállalkozások számára fontos, hogy a víz, amit használnak vagy kibocsátanak, megfeleljen a környezeti előírásoknak. Vízminőség indexek alkalmazásával monitorozhatják a folyamatok hatását a vízminőségre.[13]

Az építési projektek és infrastruktúra fejlesztések előtt fontos megérteni a vízminőség állapotát. A vízminőség indexek segíthetnek az érzékeny területek és védett ökoszisztémák azonosításában.

Hasznosak lehetnek a vízkatasztrófák, például áradások vagy szennyezések előrejelzésében és kezelésében.

### **2.2.2. Előnyök és hátrányok**

Előny lehet, hogy összetett képet nyújtanak a víztest állapotáról, mivel többféle paramétert vesznek figyelembe, például kémiai, biológiai és fizikai jellemzőket. A rendszeres vízminőség-monitorizált és az indexek használata segíthet az esetleges problémák, szennyeződések vagy változások korai felismerésében, így lehetővé téve az időbeni intézkedéseket. Használata segíthet a döntéshozóknak, például kormányzati szerveknek vagy vízgazdálkodási hatóságoknak, hogy hatékonyabb és célzottabb intézkedéseket hozzanak a vízvédelem, vízgazdálkodás vagy katasztrófa-előrejelzés terén. Hozzájárulhatnak a

vízfenntarthatósághoz, mivel segítenek az ésszerű vízhasználati politikák és gyakorlatok kidolgozásában.[14]

A hátrányuk pedig, hogy az indexek összetettek lehetnek, és azok kiszámolása és értelmezése szakértelmet igényel. Ez néha bonyolulttá teheti azok használatát. Hatékonysága nagymértékben függ az elérhető adatok minőségétől és mennyiségétől. Hiányos vagy hiányzó adatok esetén az értékelés pontossága csökkenhet. Egyetlen index nem feltétlenül képes minden környezeti jelenséget vagy problémát lefedni. Az eltérő környezeti körülmények miatt szükség lehet különböző indexek alkalmazására különböző területeken. Az adatok gyűjtése és az indexek kiszámolása költséges lehet, különösen, ha magas felbontású adatokra vagy folyamatos monitorozásra van szükség. Az indexek értékelésekor az emberi döntéseket és értékeléseket is figyelembe kell venni, ami néha szubjektivitáshoz vezethet.[14]

## **2.3. Adatbázisok**

### **2.3.1. Globális**

A vízvédelem és vízgazdálkodás területein számos térinformatikai adatbázis érhető el, amelyek segítségével modellezhetők, elemezhetők és vizsgálhatók a vízforrásokkal kapcsolatos adatok. Ilyen például a USGS Waters Data az Egyesült Államok Földtani Szolgálatával által üzemeltetett Waters adatbázis számos hidrológiai adatot tartalmaz, beleértve vízállásokat, vízhozamokat és vízminőségi adatokat is. A térképes eszközök lehetővé teszik a felhasználók számára, hogy térinformatikai elemzéseket végezzenek az adatokon.

A GRDC (Global Runoff Data Centre) globális vízhozam adatokat gyűjt és terjeszt. Az adatok a világ különböző pontjairól származnak és tervezési, modellezési és elemzési célokra használhatók.

A World Resources Institute (WRI) által üzemeltetett Aqueduct platform térinformatikai eszközöket biztosít a vízhiány, vízminőség és egyéb vízügyi kockázatok elemzéséhez. Ez magában foglalja a térképek, adatok és modulok széles skáláját.

A Copernicus Land Monitoring Service - Water Component az európai Copernicus programnak a vízre vonatkozó eleme térinformatikai adatokat kínál a talajnedvesség, felszíni víz és hőmérséklet elemzéséhez.

A NASA által biztosított Earthdata platform olyan eszközöket és adatokat kínál, amelyek alkalmazhatók a víz, a talajnedvesség és más vízügyi tényezők térbeli elemzéséhez.[15]

Az Európai Drought Observatory (EDO) olyan térinformatikai eszközöket és adatokat kínál, amelyek segítenek a szárazságok térbeli elemzésében és monitorozásában Európában.

Ezeket az adatbázisokat folyamatosan frissítik az újabb mérések alapján, így az adott projekt céljait és eszközeit ennek megfelelően kell megválasztani.

### **2.3.2. Regionális**

Magyarországra vonatkozóan is léteznek olyan adatbázisok és térinformatikai adatforrások, amelyek segítenek az elemzésében és monitorozásában.

Az Országos Vízügyi Főigazgatóság (OVF) által üzemeltetett HidroWeb rendszer magyarországi hidrometeorológiai adatokat és térbeli információkat kínál (például: vízállások, vízhozamok, és más hidrológiai adatok).[16]

A Geoinformatikai Rendszer a Környezetvédelemért (GRaK) is általuk üzemeltetett rendszer, amely térinformatikai eszközöket és adatokat nyújt a környezeti adatok kezeléséhez, ideértve a vízforrásokat is.

A KSH térinformatikai adatokat is rendelkezésre tud bocsátani a vízügyi és környezeti területeken Magyarországon.

Az FÖMI (Földmérési és Távérzékelési Intézet) által üzemeltetett rendszerek térinformatikai adatokat szolgáltatnak, ideértve a talajnedvességet, talajtípusokat, és egyéb környezeti tényezőket.

Ezek a rendszerek folyamatosan frissülnek, és hasznosak lehetnek a vizekkel kapcsolatos projektekhez, a környezeti monitorozáshoz és a területfejlesztéshez.

## 3. Módszerek és Algoritmusok

### 3.1. QGIS

A QGIS (Quantum GIS) nyílt forráskódú földrajzi információs rendszerként működik, ami lehetővé teszi a felhasználók számára a térbeli információk elemzését és szerkesztését, valamint grafikus térképek összeállítását és exportálását. A QGIS támogatja a raszter-, vektor- és hálórétegeket. A vektor adatokat pont-, vonal- vagy sokszögjellemzőként tárolják.

Támogatja a shapefile-okat, a geoadatbázisokat, a dxf, a MapInfo, a PostGIS és más iparági szabványokat. Webes szolgáltatások, beleértve a Web Map Service-t és Web Feature Service-t, szintén támogatottak az adatok külső forrásokból történő felhasználásához. Ilyen például a projektben is felhasznált OSM térkép.

A QGIS-t az Open Source Geospatial Foundation (OSGeo) fejleszti és támogatja, és számos fejlesztő és felhasználó közössége van a világ minden részén.[17]

Integrálódik más nyílt forráskódú GIS csomagokkal, ideértve a PostGIS-t, a GRASS GIS-t és a MapServer-t. A fejlesztőknek lehetőségük van kiegészíteni a QGIS rendszerét saját modulokkal, amelyeket vagy Python vagy C++ nyelveken lehet elkészíteni. Ezek segítik, megkönnyítik a térképek feldolgozását.[18]

### 3.2. Felhasznált adatok

A mintavételi terület Báránd, az Alföld keleti részén, a Nagy-Sárréten, a Sebes-Körös hordalékkúpjának nyugati lábánál helyezkedik el, lakosságszáma 2611 fő [KSH 2019]. Balla D. és munkatársai korábbi kutatásai szerint az elmúlt évtizedekben a vizsgált településen 1200 háztartásában éves szinten átlagosan 120.000 m<sup>3</sup> elhasznált vízből keletkező szennyvíz 30-40%-a szivároghatott a talajba. Ez a jelentős szennyezőanyag kibocsátás pedig a település talajvízkészletének erőteljes elszennyeződéséhez és a vízminőség jelentős romlásához vezetett, amit rendszeres évenkénti mintavétellel folyamatosan vizsgálnak a potenciálisan szennyezett területek azonosításához.[11]

A vizsgálatba 2013-ban 40, 2022-ben pedig 34 ázott kutat vontak be. A kutakból végzett mintavétel során a talajvízkutak felső 1 méteres vízrétegét mintázták meg. Ezen adatokat számomra Balla Dániel tanár úr biztosította a teszteléshez.

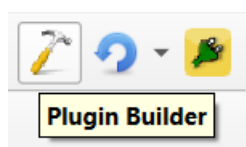
Az excel tábla tartalmazza a kutak egyedi azonosítóját (ID) valamint a kutak helyét is X, Y koordinátával, ami WGS 84-ben van megadva. Ami egy olyan szabvány, amit a kartográfiában, geodéziában és műholdas navigációban használnak. Az adatbázis tartalmazza a mérési adatokat is, amelyekre szükség lesz a számolások során. Ilyen értékek a pH, az EC, az  $\text{NH}_4^+$ , az  $\text{NO}_2^-$ , az  $\text{NO}_3^-$ , a  $\text{PO}_4^{3-}$ , a COD és az  $\text{Na}^+$ . Valamit tartalmazza az adott évet amikor a mérés megtörtént. Ezeknek az adatoknak ilyen sorrendben kell az excel táblában megjelenni a számolások elvégzéséhez. Valamint az is fontos, hogy a koordináták WGS 84 (hosszúság és szélesség) formában legyenek megadva (1. ábra).

ID	X_WGS	Y_WGS	pH	EC	NH4	NO2	NO3	PO4	COD	Na	Date
1	21,22342	47,30114	7,96	2550	0,61	0,10	54,19	0,13	3,40	257,60	2013
2	21,22570	47,30175	7,49	3850,00	0,14	0,02	9,64	0,10	1,57	270,59	2022
2	21,22570	47,30175	8,28	2710	0,43	0,85	103,65	0,23	2,40	249,55	2013
3	21,22544	47,29985	7,80	5241,00	0,75	0,10	33,85	0,13	1,98	28,64	2022
3	21,22544	47,29985	8,16	2120	0,35	1,28	94,46	1,31	4,70	270,10	2013

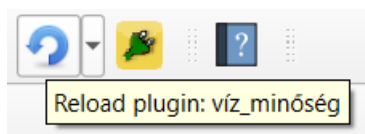
1. ábra: Az első három adat a tesztelő adatbázisból

### 3.3. Modul fejlesztés

Ahhoz, hogy QGIS-ben saját modult készítsünk, először is a szoftveren belül le kell töltenünk a Plugin Builder kiegészítő modult. Ha ezt megtettük egy gyorsgomb fog megjelenni a modul eszköztárában (2. ábra). Ennek segítségével tudjuk létrehozni és beállítani a modulunk specifikációit majd pedig legenerálni a szükséges fájlokat. A modulfejlesztés megkönnyítésére szükségünk lesz egy másik modulra is, amit szintén ezen a módon tudunk letölteni, ez a Plugin Reloader. Ez is hozzáad egy gyorsgombot az eszköztárhoz, ahol be tudjuk állítani, hogy melyik modult szeretnénk lefrissíteni a megnyomásával. Jelen esetben a saját plugin-om van beállítva a *víz\_minőség* (3. ábra).



2. ábra: Plugin Builder gyorsgomb



3. ábra: Plugin Reloader gyorsgomb



Első lépésben meg kell adnunk a modulunk nevét, ezután az osztályunk nevét, amit majd a fejlesztés alatt használni szeretnénk. Továbbá opcionálisan egy leírással is kiegészíthetjük a modult, ahol pontosan leírhatjuk annak működését. A modul verzió száma alapértelmezetten 0.1 lesz, amit át lehet írni szükség szerint, valamint a QGIS verziója is automatikusan beállításra kerül az aktuálisan használt verzióra (jelen esetben 3.0). Emellett meg kell adnunk a készítő nevét és e-mail címét is. Következő lépésben még bővebb leírást lehet adni a készülő plugin-hoz.

Ezután azt kell megválasztani, hogy milyen formátumú plugin-t szeretnénk létrehozni. Három féle lehetőség elérhető jelenleg: a **tool buttom with dialog**, a **tool buttom with dock widget** és a **processing provider**.

Jelen esetben a **tool buttom with dialog** lett használva mivel szükségem lesz a UI készítésnél gombokra és szöveges felületre is és ez a funkció támogatja ezt. Meg kell megadni, hogy a készülő projekt melyik fülön (modul, adatbázis, vektor, raszter vagy web) illetve milyen címmel jelenjen meg. A **tool buttom with dock widget** is hasonló viszont itt van egy alap kezelő felület, aminek a helyét tudjuk változtatni, de kötött a funkcionalitása. A **processing provider** pedig feldolgozási és elemzési funkciókat lát el.

Következő lépésben a check boks-ok mindegyikét bepipálva hagytam (4. ábra), de a legfontosabb a makefile és a pb\_tool. A makefile egy automatikusan generált fájl, ami összekötést biztosít a plugin fájlok között. A pb\_tool pedig lehetővé teszi, hogy interaktív elemeket rakjunk a felhasználói felületre.

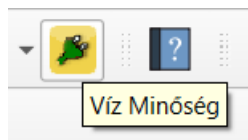


4. ábra: Modul készítésénél szükséges elemek

Ezek után a publikációs beállítások kerülnek előtérbe. Itt meg lehet adni saját weboldalt, a plugin nyelvét, valamint legyen-e jelezve, hogy ez egy kísérleti célú plugin.

Alapértelmezett esetben a QGIS saját könyvtárába helyezi a modulokat (..\QGIS\QGIS3\profiles\default\python\plugins), viszont lehetőségünk van saját cél mappát kiválasztani.

A QGIS újraindítása után a modulok között meg tudjuk találni és letölteni azt. Amint ezt megtettük megjelenik egy plusz gomb a modul eszköztárban, ami a saját plugin-unkat fogja elindítani (5. ábra). Jelenleg nincs hozzá kötve funkcionalitás, ha elindítjuk az alapértelmezett UI jelenik meg egy Ok és Cancel gombal.[19]



5. ábra: Víz Minőség saját készítésű modul gyorsgombja

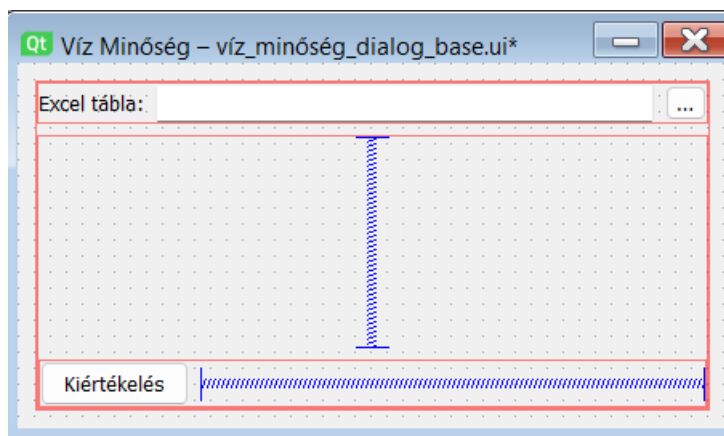
A UI elkészítéséhez szükségünk lesz a QGIS-el együtt letöltött Qt Designer szoftverre. Ezt megnyitva az előbb generált fájlok közül a víz\_minőség\_dialog\_base.ui nevűt kell kiválasztani. Ennek segítségével fogjuk tudni létrehozni a saját felületünket. Az alapértelmezett felület egy gombtömböt tartalmaz, ami az Ok és Cancel gombokat tartalmazza.

A projektben elkészített modulhoz szükségem volt egy fájl bekérő felületre, amit a QgsFileWidget elnevezésű QGIS custom widget-el oldottam meg. Az elem segítségével lehetőségünk van megadni a fájl útvonalát, amelyet későbbiekben a kódban fel tudunk dolgozni.

A számolások és kiértékelések elvégzésére egy QPushButton lett használva, amit, ha megnyomunk először egy mappát vár, ahova az elkészült térképeket fogja legenerálni. Ha nem adunk meg semmilyen fájlt és úgy nyomjuk meg a kiértékelést, akkor egy hiba üzenet jelenik meg.

A gombok és widget-ek beállítása fontos, hiszen később a kódban használni szeretnénk őket, ezért érdemes átállítani ezek nevét a könnyebb hivatkozás érdekében.

A UI úgy lett megszerkesztve, hogy reszponzív legyen és minden azon elhelyezett gomb és szöveg a helyén maradjon akármilyen méretváltozás is következik be. Ezeket Vertical és Horizontal Spacer-ek segítségével lett elkészítve. Az „Excel tábla:” felirat pedig egy Text szövegdobozban lett elhelyezve (6. ábra).[20] [21]



6. ábra: A modul UI felülete

Ezek után a modulnak a kódjának a megírása a következő lépés. Ezt a víz\_minőség.py szerkesztésével tudjuk megtenni. Itt kell lekódolni minden olyan funkciót, amit a modultól elvárunk ilyenek a számolásokat, a gombok és egyéb a felhasználói felületen megjelenő widget-ek funkciója.[22]

### 3.4. Vízminőség indexek

Projektben azokat a vízminőség indexeket használtam fel, amelyek széles körben ismertek és alkalmazottak.

A vízminőség indexek meghatározásához az alábbi vízkémiai paramétereket használtam fel: pH érték, az EC ( $\mu\text{S}/\text{cm}$ ), az  $\text{NH}_4^+$  ( $\text{mg L}^{-1}$ ), az  $\text{NO}_2^-$  ( $\text{mg L}^{-1}$ ), az  $\text{NO}_3^-$  ( $\text{mg L}^{-1}$ ), a  $\text{PO}_4^{3-}$  ( $\text{mg L}^{-1}$ ), a COD ( $\text{mg L}^{-1}$ ) és az  $\text{Na}^+$  ( $\text{mg L}^{-1}$ ) értékek. Ezek alapján tudjuk meghatározni egy mintának a vízminőségét. Ezen kémiai paraméterek az érvényben lévő szabványok értelmében ezen határértékekkel rendelkeznek: pH = 8.5, EC = 2500,  $\text{NH}_4$  = 0.5,  $\text{NO}_2$  = 0.5,  $\text{NO}_3$  = 50,  $\text{PO}_4$  = 0.5, COD = 4.5, Na = 200. Ezen értékek szükségesek a vízminőség indexek számításához.[23]

Először is a WQI (Súlyozott Vízminőség Index) került kiszámításra, ami meghatározza a 0-100<-ig terjedő skálán a vizek milyenségét 5 kategóriára bontva. Ennek a képlete:

$$WQI = \sum Q_n W_n / \sum W_n$$

ahol  $Q_n$  az  $n$ . vízminőségi paraméter minőségi besorolása,  $W_n$  az  $n$ . vízminőségi paraméter egységnyi súlya. A  $Q_n$  értékét az alábbi egyenlet alapján számítjuk ki:

$$Q_n = 100 \left[ \frac{(V_n - V_i)}{(V_s - V_i)} \right]$$

ahol  $V_n$  az  $n$ -edik paraméter tényleges értéke,  $V_i$  az adott paraméter ideális értéke [ $V_i=0$ , kivéve: pH ( $V_i=7$ )], a  $V_s$  az  $n$ . vízminőségi paraméter szabványos megengedett határértéke. Az egységnyi súly ( $W_n$ ) a következő képlettel számolhatjuk ki:

$$W_n = \frac{k}{V_s}$$

ahol  $k$  az arányosság állandója, amelyet a következő egyenlet alapján számolhatunk ki:

$$k = \left[ 1 / \sum \frac{1}{V_s} = 1, 2, \dots, n \right]$$

A WQI értékek alapján állapítható meg a WQS (Vízminőség Állapot) ami szöveges formában írja le a vizek minőségét, valamint a WQS Rank, ami pedig egy 1-től 5-ig terjedő skálát használ ehhez. Ez a következőképpen alakul (1. táblázat):[24]

1. táblázat: WQI, WQS, WQS Rank kategóriák és a lehetséges vízfelhasználás táblázat

WQI	WQS	WQS Rank	Lehetséges használat
0-25	Kiváló vízminőség	1	Lakossági, öntözés, ipari
26-50	Jó vízminőség	2	Mezőgazdasági, ipari
51-75	Rossz vízminőség	3	Mezőgazdasági, ipari
76-100	Nagyon rossz vízminőség	4	Mezőgazdasági
100 <	Bármilyen használatra alkalmatlan	5	Felhasználás előtt tisztító kezelés szükséges

Fontosnak tartottam még a Cd (Contamination Degree), szennyezettségi index meghatározását is. Az index a talajvízre káros paraméterek együttes hatását tükrözi, oly módon, hogy minden határérték feletti paramétert figyelembe vesz. Az index ebből következően a szennyező faktorok összegének tekinthető. Itt a végeredmény egy 0-tól 3-ig terjedő skálán helyezkedik el, amit 5 részre bontunk. Ennek kiszámítása a következőképpen történik:

$$C_d = \sum_{i=1}^n C_{fi} \text{ ahol: } C_{fi} = \frac{C_{Ai}}{C_{Ni}} - 1$$

$C_{fi}$  = szennyezettségi faktor i-edik komponensre;  $C_{Ai}$  = i-edik komponens analitikai értéke;  $C_{Ni}$  = i-edik komponens szennyezettségi határértéke.

Ezek alapján megállapítható a  $C_d$ s, ami szintén egy 1-től 5-ig terjedő skálán mutatja a víz szennyezettségét és a szennyezettségi fok pedig egy szöveges leírása az előzőkének. A lehetséges használat is megmondható ezen számítások elvégzésével. Ezt pedig a következő tábla szemlélteti (2. táblázat): [25]

2. táblázat: Szennyezettségi fok táblázat

Cd	Cd Rank	Szennyezettség foka	Lehetséges használat
0	1	nem szennyezett	Lakossági, öntözés, ipari
<1	2	alacsony	Mezőgazdasági, ipari
1-2	3	közepes	Mezőgazdasági, ipari
2-3	4	szennyezett	Mezőgazdasági
>3	5	erősen szennyezett	Felhasználás előtt tisztító kezelés szükséges

### 3.5. QGIS-ben való automatizálás

A projektem célja nem csak az volt, hogy hasznos számolásokat tudjunk elvégezni, hanem hogy ezt minél egyszerűbben meg lehessen oldani. Ezért is választottam a QGIS-t, mint szoftver ehhez a feladathoz. Az, hogy saját modult lehet létrehozni, lényegesen megkönnyítette a munkámat. Törekedtem az egyszerű felhasználói felület megteremtésére is. Valamint automatizálni tudtam a végeredmények térképen való megjelenést is. Az elkészült térképek nagyban befolyásolja a kapott adatok feldolgozási sebességét, valamint a látványon is nagyméreteken javítanak. Így az adott vízminőség indexek kiszámításához csak a megfelelő excel táblába rendezett adatai szükségesek és máris megvan az eredmény egy térképre vetítve.

### 3.5.1. Kategorizálás WQS értékek alapján és interpolálás

A cél az volt, hogy a modul létrehozzon minden, a fájlban szereplő évhez egy shp fájlt és azt automatikusan meg is jelenítse. Ezeken a térképeken a WQS értékek alapján történik az automatikus kategorizálás a megfelelő jelölő színnel és a hozzá tartozó minősítéssel (Bármilyen használatra alkalmatlan: Piros, Nagyon rossz víz minőség: Narancssárga, Rossz víz minőség: Sárga, Jó víz minőség: Sötét zöld, Kiváló víz minőség: Világos zöld). A kiszámolt vízminőség indexek és státuszok mindegyike (WQI, WQS, WQS Rank, Cd, Cds, Cd Rank, lehetséges használat) megjelenik az attribútum táblákban és ezek onnan az ID-k segítségével könnyen azonosíthatók, valamint, ha szeretnénk természetesen manuális beállításokkal ezekre is tudunk szűrni a térképi megjelenítéshez.

A kutak helyének az azonosításához pontokat használtam, amihez szükség van a X és Y koordinátákra az adatok közül. A pontok nagysága alapértelmezetten 4, ami így jól látható és a színe mindig a WQS értéknek megfelelően fog beállítására kerülni. Ez mind a shp fájlban tárolódik el és annak megjelenítésével rajzolódik ki (7. ábra).

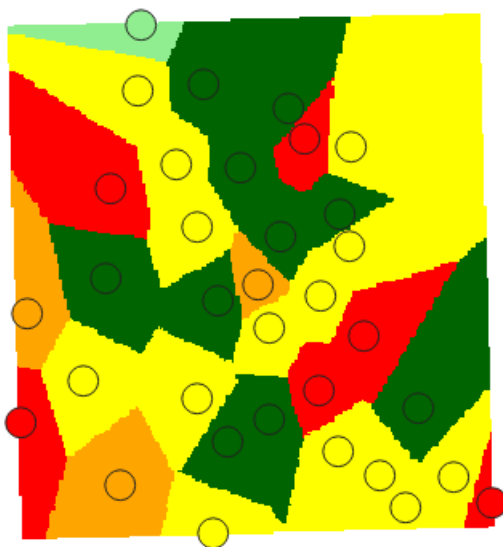


7. ábra: Példa a két külön évhez generált WQS érték automatizálásról

Az interpoláció a térbeli adatok közötti hiányos pontok közötti értékek becslését jelenti. Ezáltal az interpoláció segít kiegészíteni a hiányzó adatokat, például a magassági pontok közötti magasságértékek becslésében vagy más térbeli jellemzők interpolálásában. Ebben az esetben a kiszámolt WQS értékeket becsli meg a kutak között és köti őket össze. Ez lehetővé teszi a felhasználók számára, hogy részletes vizuális elemzéseket végezzenek térbeli adatokon alapuló döntéshozatalhoz.[26]

Az interpoláció automatizálásához a modulkészítés során nem minden interpolációs eszköz használata volt megengedett, így végül a legközelebbi szomszédos rács módszerrel (Grid - Inverse Distance to a Power) készült el az interpolálás. Az interpolációs módszer egy négyzet alakú területet fog kialakítani a pontokból és azon belül határozza meg az értékeket. Ezen felül a sikeres interpolációhoz szükség van egy shp fájlra, ami jelen esetben az előbb generált monitorizált adatokat tartalmazó fájl lesz. Továbbá meg kell adni az interpoláció koordinátáját, amelynek pedig a WQS Rank értékek lesznek. A többi beállítás mind alapértelmezetten marad, ezt a későbbiekben a kód bemutatásánál részletesen kifejtem.

Az interpoláció eredményei egy raszter fájlban (.tif) kerülnek tárolásra. Itt minden évhez egy raszter szeretnénk generálni, valamint a WQS Rank értékek alapján stilizálni az összekötött területeket. Ezek a program indítása után azonnal megjelenítésre kerülnek a térképen a pontokkal egyidejűleg (8. ábra).[27]



8. ábra: 2022-es évi adatokhoz tartozó interpoláció

## 4. Eredmények

### 4.1. A QGIS modul működésének bemutatása

Azt, hogy a modul milyen műveletek elvégzésére legyen képes a víz\_minőség\_dialog.py fájlban tudjuk lekódolni.

Első megnyitáskor az alap generált kódrészlet fog látszódni melyet feladatunk kiegészíteni a saját kódrészünkkel. Itt láthatjuk a szöveg kódolását, illetve egy automatikusan generált dokumentációt.

```
1. # -*- coding: utf-8 -*-
2. """
3. /*****
4. VizMinosegDialog
5.         A QGIS plugin
6. Ez a modul adott kutak vizének a minőséget írja le.
7. Generated by Plugin Builder: http://g-sherman.github.io/Qgis-Plugin-Builder/
8.         -----
9.     begin           : 2024-01-18
10.    git sha          : $Format:%H$
11.    copyright        : (C) 2024 by Urbán Eszter Klára
12.    email            : urbaneszter2003@gmail.com
13. *****/
14.
15. /*****
16. *
17. * This program is free software; you can redistribute it and/or modify
18. * it under the terms of the GNU General Public License as published by
19. * the Free Software Foundation; either version 2 of the License, or
20. * (at your option) any later version.
21. *
22. *****/
23. """
```

Az alap generálás része egy kódrészlet, ami a UI betöltéséért felelős. Itt hívja meg a víz\_minőség\_dialog\_base.ui-t, ami a felhasználói felületet indítja amit előzőleg megszerkesztettünk.

```
1. # This loads your .ui file so that PyQt can populate your plugin with the elements from Qt Designer
2. FORM_CLASS, _ = uic.loadUiType(
3.     os.path.join(os.path.dirname(__file__), "víz_minőség_dialog_base.ui")
4. )
```



Az automatikus generálás része továbbá még egy osztály is, amiben majd a saját függvényünket is meg kell írni. Itt található egy alap függvény, ami a felhasználói felületet hívja meg és használja, ez az alábbi ábrán látható kódrészlet 10. sorában jól látható. Itt tudjuk beállítani továbbá a gombjaink funkcióit is. A 12. sorban definiáljuk, hogy kattintásra induljon el a program nevezetű függvény, ami a vízminőségi mutatók számításáért felelős.

```
1. class VizMinosegDialog(QtWidgets.QDialog, FORM_CLASS):
2.     def __init__(self, parent=None):
3.         """Constructor."""
4.         super(VizMinosegDialog, self).__init__(parent)
5.         # Set up the user interface from Designer through FORM_CLASS.
6.         # After self.setupUi() you can access any designer object by doing
7.         # self.<objectname>, and you can use autoconnect slots - see
8.         # http://qt-project.org/doc/qt-4.8/designer-using-a-ui-file.html
9.         # #widgets-and-dialogs-with-auto-connect
10.        self.setupUi(self)
11.
12.        self.kiertekes.clicked.connect(self.program)
```

Ahhoz, hogy a modul minden része megfelelően működjön több importálásra is szükség volt. Ilyen például az 1. sorban az os modul, amely a fájlok, könyvtárak kezelését teszi lehetővé. A 2. sor az interpolációs rasztert létrehozásához szükséges. A 3. sorban lévő import az shp fájl tartalmát tölti fel a megfelelő adatokkal. A 4. sorban lévő import pedig betölti a kiválasztott excel fájlt a használathoz. Az 5. sortól a 15. sorig bezárva a qgis.core importál minden olyan kiegészítést, ami elősegíti a shp fájl és az interpoláció elkészítését és megjelenítését. A 16. sorban a UI-hoz szükséges Widgeteket importáljuk. A 17. sorban a színezéshez szükséges import található. A 18. sorban pedig a kész fájlok elkészítési helyének a megadásához szükséges import található.

Az importok között is van olyan, ami alapértelmezetten a generálás után benne volt a projektben, ilyen az 1. sorban szereplő os és a 16. sorban szereplő importok, mivel ezek felelősek az alapértelmezett UI elindításáért.

```
1. import os
2. import subprocess
3. import geopandas as gpd
4. from openpyxl import load_workbook
5. from qgis.core import (
6.     QgsCategorizedSymbolRenderer,
7.     QgsColorRampShader,
8.     QgsProject,
9.     QgsRasterLayer,
10.    QgsRasterShader,
11.    QgsRendererCategory,
```

```

12. QgsSingleBandPseudoColorRenderer,
13. QgsSymbol,
14. QgsVectorLayer,
15. )
16. from qgis.PyQt import QtWidgets, uic
17. from qgis.PyQt.QtGui import QColor
18. from qgis.PyQt.QtWidgets import QFileDialog

```

A számolások elvégzéséhez szükség van a vízkémia összetevők MSZ szabvány szerinti határértékére és az ideális értékére ( $V_i/phV_i$ ) amelyek konstans változók így ezek a VizMinosegDialog osztályon kívül kerültek felvételre.

```

1. pH_hatar = 8.5
2. EC_hatar = 2500
3. NH4_hatar = 0.5
4. NO2_hatar = 0.5
5. NO3_hatar = 50
6. PO4_hatar = 0.5
7. COD_hatar = 4.5
8. Na_hatar = 200
9. Vi = 0
10. pH_Vi = 7

```

Hasonlóan konstans értékek a kategorizálás és az interpoláció elvégzéséhez szükséges színek meghatározása is, így ezek is a VizMinosegDialog osztályon kívül kerültek felvételre. A wqs\_szinek a kategorizálásnál megfelelő szöveget a megfelelő színre, míg a wqs\_rank\_szinek pedig az interpolációnál a számok alapján végzik el ugyanezt a színkódolást.

```

1. wqs_szinek = {
2.     "Bármilyen használatra alkalmatlan": QColor("#FF0000"),
3.     "Nagyon rossz víz minőség": QColor("#FFA500"),
4.     "Rossz vízminőség": QColor("#FFFF00"),
5.     "Jó vízminőség": QColor("#006400"),
6.     "Kiváló vízminőség": QColor("#90EE90"),
7. }
8.
9. wqs_rank_szinek = {
10.    "5": QColor("#FF0000"),
11.    "4": QColor("#FFA500"),
12.    "3": QColor("#FFFF00"),
13.    "2": QColor("#006400"),
14.    "1": QColor("#90EE90"),
15. }

```

A vízminőség indexértékek meghatározása, a fájlok létrehozása és a térképi nézetben történő megjelenítését a program nevű függvényben történik meg (1. sor).

Az első feladat a fájl beolvasása volt. Ennek eredményét a 2. sorban a `Bekero_excel_fajl_utvonal` változóban tárolom el. A 4. sortól a 6. sorig azt kezeljük le, hogy ha nem adunk meg fájlt akkor egy hibaüzenet jelenjen meg. A 8. sorban a munkafuzet eltárolja a betöltött excel táblát a megadott útvonalról. A 9. sorban a munkalap pedig az előbb betöltött munkafuzetet fogjuk aktiválni, hogy tudjunk vele dolgozni.

```
1. def program(self):
2.     Bekero_excel_fajl_utvonal = self.excel_bekero.filePath()
3.
4.     if not Bekero_excel_fajl_utvonal:
5.         QtWidgets.QMessageBox.critical(self, "Hiba", "Adj meg egy excel fájlt.")
6.         return
7.
8.     munkafuzet = load_workbook(filename=Bekero_excel_fajl_utvonal)
9.     munkalap = munkafuzet.active
```

Következő lépésben a betöltött és aktivált munkalapból nyerjük ki az adatokat. Az első 12 sorban minden adatnak létrehozunk egy listát, ahol azok értékeit tároljuk majd. A 14. sorban egy for ciklus segítségével bejárjuk a beolvasott fájl oszlopait. A 15. sorban az aktuális oszlop számát adja meg az `oszlop_index` változóban. A 16. sorban egy üres listát hozunk létre az aktuális oszlopok tárolására. A 18. sorban egy belső for ciklust indítunk el a sorokon való iteráláshoz az adott oszlopban. A `min_row=2` paraméter azt mondja meg hogy az excelben a 2. sortól kezdje az adatok felvételét a `values_only=True` pedig csak a cella értékeket kezeli nem pedig a cellát magát. A 19. sorban pedig hozzáadjuk az oszlop listához ezeket az eredményeket. A 21. sortól pedig azt vizsgáljuk, hogy melyik oszlophoz tartozik az adott érték és eltároljuk a megfelelő helyen.

```
1.     id = []
2.     x_wgs = []
3.     y_wgs = []
4.     pH = []
5.     EC = []
6.     NH4 = []
7.     NO2 = []
8.     NO3 = []
9.     PO4 = []
10.    COD = []
11.    Na = []
12.    Date = []
13.
14.    for betu in range(ord("A"), ord("L") + 1):
15.        oszlop_index = betu - ord("A") + 1
16.        oszlop = []
17.
```

```

18.     for sor in munkalap.iter_rows(min_row=2, values_only=True):
19.         oszlop.append(sor[oszlop_index - 1])
20.
21.     if chr(betu) == "A":
22.         id = oszlop
23.     if chr(betu) == "B":
24.         x_wgs = oszlop
25.     if chr(betu) == "C":
26.         y_wgs = oszlop
27.     if chr(betu) == "D":
28.         pH = oszlop
29.     if chr(betu) == "E":
30.         EC = oszlop
31.     if chr(betu) == "F":
32.         NH4 = oszlop
33.     if chr(betu) == "G":
34.         NO2 = oszlop
35.     if chr(betu) == "H":
36.         NO3 = oszlop
37.     if chr(betu) == "I":
38.         PO4 = oszlop
39.     if chr(betu) == "J":
40.         COD = oszlop
41.     if chr(betu) == "K":
42.         Na = oszlop
43.     if chr(betu) == "L":
44.         Date = oszlop

```

A WQI kiszámolása minden kémiai paraméterhez tartozó  $k$  értékkel kezdődik. Ezt úgy kapjuk, ha 1-et elosztjuk a határértékkel (1.-8. sor). A 10. sorban az összes  $k$  értékét összeadva egy  $k\_sum$  értéket határozzuk meg. A 12. sorban a  $k$  értékét határozzuk meg, amit úgy kapunk meg hogy 1-gyel elosztjuk a  $k\_sum$  értékét.

Ezek után minden kémiai paraméternek kiszámoljuk a  $W_n$  értékét, amit úgy kapunk meg hogy a  $k$  eredményét elosztjuk az adott határértékkel (14.-21. sor). A  $Q_n$  úgy számolandó, hogy az adott kémiai összetevő értékét kivonjuk az ideális értékből. Ennek az eredményét elosztjuk a határértékből kivont ideális értékkel. Az eddig kiszámolt értéket pedig megszorozzuk 100-zal. Az adott értéket, az adott kémiai paraméter értékein végig iterálva kapjuk meg (23.-30. sor).

A  $Q_n W_n$  a  $W_n$  és a  $Q_n$  értékeket szorozza össze úgy, hogy a  $Q_n$  értéknek mindig az adott  $i$ -edik elemét veszi figyelembe, amit egy for ciklussal nyer ki az adott kiszámolt  $Q_n$  értékek közül és ezen megy végig, amíg a végére nem ér a  $Q_n$  listának (32.-39. sor).

A WQI érték pedig a  $Q_n W_n$  értékek  $i$ -edik elemének az összeadását jelenti, amíg az értékek végére nem ér (41.-51. sor).

```

1. k_ph = 1 / pH_hatar
2. k_EC = 1 / EC_hatar
3. k_NH4 = 1 / NH4_hatar
4. k_NO2 = 1 / NO2_hatar
5. k_NO3 = 1 / NO3_hatar
6. k_PO4 = 1 / PO4_hatar
7. k_COD = 1 / COD_hatar
8. k_Na = 1 / Na_hatar
9.
10. k_sum = k_ph + k_EC + k_NH4 + k_NO2 + k_NO3 + k_PO4 + k_COD + k_Na
11.
12. k = 1 / k_sum
13.
14. Wn_pH = k / pH_hatar
15. Wn_EC = k / EC_hatar
16. Wn_NH4 = k / NH4_hatar
17. Wn_NO2 = k / NO2_hatar
18. Wn_NO3 = k / NO3_hatar
19. Wn_PO4 = k / PO4_hatar
20. Wn_COD = k / COD_hatar
21. Wn_Na = k / Na_hatar
22.
23. Qn_pH = [100 * ((pH_ertek - pH_Vi) / (pH_hatar - pH_Vi)) for pH_ertek in pH]
24. Qn_EC = [100 * ((EC_ertek - Vi) / (EC_hatar - Vi)) for EC_ertek in EC]
25. Qn_NH4 = [100 * ((NH4_ertek - Vi) / (NH4_hatar - Vi)) for NH4_ertek in NH4]
26. Qn_NO2 = [100 * ((NO2_ertek - Vi) / (NO2_hatar - Vi)) for NO2_ertek in NO2]
27. Qn_NO3 = [100 * ((NO3_ertek - Vi) / (NO3_hatar - Vi)) for NO3_ertek in NO3]
28. Qn_PO4 = [100 * ((PO4_ertek - Vi) / (PO4_hatar - Vi)) for PO4_ertek in PO4]
29. Qn_COD = [100 * ((COD_ertek - Vi) / (COD_hatar - Vi)) for COD_ertek in COD]
30. Qn_Na = [100 * ((Na_ertek - Vi) / (Na_hatar - Vi)) for Na_ertek in Na]
31.
32. Qn_Wn_pH = [Wn_pH * Qn_pH[i] for i in range(len(Qn_pH))]
33. Qn_Wn_EC = [Wn_EC * Qn_EC[i] for i in range(len(Qn_EC))]
34. Qn_Wn_NH4 = [Wn_NH4 * Qn_NH4[i] for i in range(len(Qn_NH4))]
35. Qn_Wn_NO2 = [Wn_NO2 * Qn_NO2[i] for i in range(len(Qn_NO2))]
36. Qn_Wn_NO3 = [Wn_NO3 * Qn_NO3[i] for i in range(len(Qn_NO3))]
37. Qn_Wn_PO4 = [Wn_PO4 * Qn_PO4[i] for i in range(len(Qn_PO4))]
38. Qn_Wn_COD = [Wn_COD * Qn_COD[i] for i in range(len(Qn_COD))]
39. Qn_Wn_Na = [Wn_Na * Qn_Na[i] for i in range(len(Qn_Na))]
40.
41. wqi = [
42.     Qn_Wn_pH[i]
43.     + Qn_Wn_EC[i]
44.     + Qn_Wn_NH4[i]
45.     + Qn_Wn_NO2[i]
46.     + Qn_Wn_NO3[i]
47.     + Qn_Wn_PO4[i]
48.     + Qn_Wn_COD[i]
49.     + Qn_Wn_Na[i]
50.     for i in range(len(Qn_Wn_pH))
51. ]

```

A WQS és WQS Rank a WQI alapján határozható meg. Az első két sorban létrehozunk az új értékeknek egy listát. A 3. sorban lévő for ciklus végig iterál a WQI értékein és ha az adott intervallumba esik akkor a WQS-hez és a WQS Rank-hez a megfelelő értéket adjuk hozzá.

```

1.   wqs = []
2.   wqs_rank = []
3.   for wqi_ertekek in wqi:
4.       if wqi_ertekek > 100:
5.           wqs.append("Bármilyen használatra alkalmatlan")
6.           wqs_rank.append(5)
7.       elif wqi_ertekek > 76:
8.           wqs.append("Nagyon rossz víz minőség")
9.           wqs_rank.append(4)
10.      elif wqi_ertekek > 51:
11.          wqs.append("Rossz vízminőség")
12.          wqs_rank.append(3)
13.      elif wqi_ertekek > 26:
14.          wqs.append("Jó vízminőség")
15.          wqs_rank.append(2)
16.      else:
17.          wqs.append("Kiváló vízminőség")
18.          wqs_rank.append(1)

```

A Cd vízminőség index kiszámításához először a  $C_{fi}$  értéket kell meghatározni. Ezt úgy tesszük meg hogy az adott kémiai paraméter értékét elosztjuk a határértékével és ezt kivonjuk 1-ből. Az adott értéket, amit a számolásban használunk úgy kapjuk meg hogy egy for ciklussal az adott kémiai paraméter értékein iterálunk végig (1.-8. sor).

Létrehozunk a Cd értékek tárolására is egy listát a 10. sorban. A Cd érték minden egyes paraméterből származó  $C_{fi}$  értéke, amennyiben az nem negatív. Ha a  $C_{fi}$  negatív, akkor azt a Cd értéket nem vesszük figyelembe. Ezt szintén egy for ciklus segítségével állapítjuk meg (11.-40. sor).

```

1.   Cfi_pH = [(pH_ertekek / pH_hatar) - 1 for pH_ertekek in pH]
2.   Cfi_EC = [(EC_ertekek / EC_hatar) - 1 for EC_ertekek in EC]
3.   Cfi_NH4 = [(NH4_ertekek / NH4_hatar) - 1 for NH4_ertekek in NH4]
4.   Cfi_NO2 = [(NO2_ertekek / NO2_hatar) - 1 for NO2_ertekek in NO2]
5.   Cfi_NO3 = [(NO3_ertekek / NO3_hatar) - 1 for NO3_ertekek in NO3]
6.   Cfi_PO4 = [(PO4_ertekek / PO4_hatar) - 1 for PO4_ertekek in PO4]
7.   Cfi_COD = [(COD_ertekek / COD_hatar) - 1 for COD_ertekek in COD]
8.   Cfi_Na = [(Na_ertekek / Na_hatar) - 1 for Na_ertekek in Na]
9.
10.  cd = []
11.  for i in range(len(pH)):
12.      ph_Cfi = Cfi_pH[i]
13.      EC_Cfi = Cfi_EC[i]

```

```

14.     NH4_Cfi = Cfi_NH4[i]
15.     NO2_Cfi = Cfi_NO2[i]
16.     NO3_Cfi = Cfi_NO3[i]
17.     PO4_Cfi = Cfi_PO4[i]
18.     COD_Cfi = Cfi_COD[i]
19.     Na_Cfi = Cfi_Na[i]
20.
21.     if ph_Cfi >= 0:
22.         CD = ph_Cfi
23.     else:
24.         CD = 0
25.     if EC_Cfi >= 0:
26.         CD = CD + EC_Cfi
27.     if NH4_Cfi >= 0:
28.         CD = CD + NH4_Cfi
29.     if NO2_Cfi >= 0:
30.         CD = CD + NO2_Cfi
31.     if NO3_Cfi >= 0:
32.         CD = CD + NO3_Cfi
33.     if PO4_Cfi >= 0:
34.         CD = CD + PO4_Cfi
35.     if COD_Cfi >= 0:
36.         CD = CD + COD_Cfi
37.     if Na_Cfi >= 0:
38.         CD = CD + Na_Cfi
39.
40.     cd.append(CD)

```

A Cds, Cds Rank és a lehetséges felhasználás értékeit a Cd eredményei alapján tudjuk meghatározni. Ezeknek az értékeknek is készítünk egy üres listát, amit majd az eredményeivel töltünk fel. A 4. sorban a for ciklus segítségével minden Cd végeredményein végig iterálva a megfelelő eredményt adjuk hozzá az újonnan létrehozott értékekhez (4.-26. sor).

```

1.     cds = []
2.     cds_rank = []
3.     lehetséges_hasznalat = []
4.     for cd_ertek in cd:
5.         if cd_ertek == 0:
6.             cds.append("Nem szennyezett")
7.             cds_rank.append(1)
8.             lehetséges_hasznalat.append("Lakossági, mezőgazdasági és ipari")
9.         elif cd_ertek <= 1:
10.            cds.append("Alacsony")
11.            cds_rank.append(2)
12.            lehetséges_hasznalat.append("Mezőgazdasági és ipari")
13.        elif cd_ertek < 3 and cd_ertek >= 1:
14.            cds.append("Közepes")
15.            cds_rank.append(3)
16.            lehetséges_hasznalat.append("Mezőgazdasági és ipari")
17.        elif cd_ertek <= 6 and cd_ertek >= 3:

```

```

18.         cds.append("Magas")
19.         cds_rank.append(4)
20.         lehetséges_hasznalat.append("Mezőgazdasági")
21.     elif cd_ertekek > 6:
22.         cds.append("Nagyon magas")
23.         cds_rank.append(5)
24.         lehetséges_hasznalat.append(
25.             "Felhasználás előtt tisztító kezelés szükséges"
26.         )

```

Ahhoz, hogy a kiszámolt adatokat egy shp fájlba tudjuk gyűjteni, először is szükség van egy mappára, ahova azokat létre tudja hozni. Ezt a Kiiró\_mappa\_utvonal tárolja el és a UI felületen tudjuk megadni hova is szeretnénk a fájlokat generáltatni (1.-3. sor).

A gdf egy GeoDataFrame-t tárol el, ami majd a földrajzi adatok kezelését segíti elő. Ez létrehozza az összes olyan oszlopot, amit az eredmények között látni szeretnénk vagy szükséges a pontok kirajzolásához, vagy az interpoláció elkészítéséhez (5.-18. sor). Tartalmaz még egy geometry oszlopot, ami a pontok helyét tartalmazza a térképen az X és Y koordinátákat alapul véve (19. sor). A crs a koordináta rendszert fixálja le, ami jelen esetben az EPSG:4326, ami a WGS84-nek felel meg. Az excel fájlban ilyen formátumban kell megadni a koordinátákat, a megfelelő megjelenítés érdekében (20. sor).

```

1.     Kiiró_mappa_utvonal = QFileDialog.getExistingDirectory(
2.         self, "Fájlok létrehozási helye"
3.     )
4.
5.     gdf = gpd.GeoDataFrame(
6.         {
7.             "Id": id,
8.             "x_wgs": x_wgs,
9.             "y_wgs": y_wgs,
10.            "Date": Date,
11.            "WQI": wqi,
12.            "WQS": wqs,
13.            "WQS_Rank": wqs_rank,
14.            "CD": cd,
15.            "CDS": cds,
16.            "CDS_Rank": cds_rank,
17.            "Használat": lehetséges_hasznalat,
18.        },
19.        geometry=gpd.points_from_xy(x_wgs, y_wgs),
20.        crs="EPSG:4326",
21.    )

```

Az 1. sorban a for ciklus végigmegy a létrehozott dataframe évszámain és az azonosakat egyben fogja kezelni ennek eredményeképpen. A 2. sorban a dataframe-ben a Date értékeit



átírjuk „ev” névre, hogy tudjunk rá hivatkozni a fájl és a réteg címadásában. A 3. sorban a dataframe-et megtisztítjuk azokatól az adatoktól, amelyek csak segítenek a kategorizálásban és nem szolgálnak szükséges információval az attribútumtáblában, ilyen például a Date és a koordináták.

Az 5. sorban létrehozuk a kimeneti fájlokat a cél könyvtárba (amit a UI felületen tudunk megadni), évszámnak megfelelő névvel ellátva. A 6. sorban pedig ebbe a shp fájlba elmenti az elkészített dataframe-t.

A 8. sor egy QgsVectorLayer objektumot hoz létre, amit reteg\_kategorizalt néven tárolunk el és az adott évszám feltüntetjük a nevében. Ez az előbb elkészített shp fájlt alakítja át megjeleníthető réteggé.

A 10. sorban egy listát alakítunk ki a kategóriák tárolásához. A for ciklus felhasználja a wqs\_szinok szótárt, amiben a wqs\_kategoria a kulcs a szín pedig a hozzárendelt színkód (11. sor). Ahogy ezen végig iterál, minden WQS értékhez hozzárendel egy pontot (12. sor). Ez a pont 4 egység alapértelmezett nagyságot kap és a megfelelő színt a hozzárendelés alapján (13.-14. sor).

15. sorban egy QgsRendererCategory objektum jön létre, ami kategorizálja a megjelenítés során a wqs\_kategoria kulcsértékeit, az előzőekben definiált pontokat és a kulcs értéket string formátumban. Ezeket az eredményeket hozzáadjuk a kategoria listához a 16. sorban.

18. sorban a wqs\_pontok QgsCategorizedSymbolRenderer osztályt tárolnak a WQS értékek és a kategoria alapján létrehozva. Ez lehetővé teszi a kategorizált pontok megjelenítését. A 19. sorban a wqs\_pontok-at állítja be a réteg szimbólumainak kategorizálására. A következő sorban pedig hozzáadjuk az előbb kategorizált réteget a térképhez, ami megjeleníti az évekre bontott kategorizált pontokat.

```
1.     for ev in gdf["Date"].unique():
2.         ev_gdf = gdf[gdf["Date"] == ev]
3.         ev_gdf_rovid = ev_gdf.drop(["Date", "x_wgs", "y_wgs"], axis=1)
4.
5.         shpfile = os.path.join(Kiir_o_mappa_utvonal, f"{ev}.shp")
6.         ev_gdf_rovid.to_file(shpfile)
7.
8.         reteg_kategorizalt = QgsVectorLayer(shpfile, f"{ev}", "ogr")
9.
10.        kategoriak = []
11.        for wqs_kategoria, szin in wqs_szinok.items():
12.            pont = QgsSymbol.defaultSymbol(reteg_kategorizalt.geometryType())
13.            pont.setColor(QColor(szin))
14.            pont.setSize(4)
```

```

15.         kategoria = QgsRendererCategory(wqs_kategoria, pont, str(wqs_kategoria))
16.         kategoriak.append(kategoria)
17.
18.         wqs_pontok = QgsCategorizedSymbolRenderer("WQS", kategoriak)
19.         reteg_kategorizalt.setRenderer(wqs_pontok)
20.         QgsProject.instance().addMapLayer(reteg_kategorizalt)

```

## 4.2. Interpoláció és plusz modulok

### 4.2.1. Interpoláció

Ahhoz, hogy interpolálni tudjunk, szükségünk van tif fájlra. Ezt az 1. sorban a `raszter_file` tárolja el. A `Kiirro_mappa_utvonal` megadja, hogy hová hozza létre a raszteres fájlt a mintavételi időpontok szerint évekre bontva.

A 3. sortól a raszter határozza meg, hogy milyen interpolációt szeretnénk végrehajtani. A `gdal_grid` pontszerű adatokból készít raszter adatokat. A `-l` kapcsoló a bemeneti shp fájl nevét fogja meghatározni, ami a `f'{ev}'` nevű shp fájl lesz. A `-zfield` meghatározása szükséges az interpoláció elvégzéséhez, ebben az esetben ez a `WQS_Rank` értékei lesznek. Ezen értékek alapján tölti majd ki a pontok közti területet. Az `-a` és a `nearest:radius1=0.0:radius2=0.0:angle=0.0:nodata=0.0` kapcsoló részletesen meghatározza a raszterizálás paramétereit. A `nearest` a közelítő módot jelzi, valamint meghatározza a keresőtávolságok értékeit (`radius1` és `radius2`), a szöget (`angle`) és a hiányzó értékeket (`nodata`). A `-ot` meghatározza a kimeneti adattípus raszter formátumát, ami `Float32` (lebegőpontos számok). A `-of` a kimeneti fájl formátumot adja meg ami jelen esetben `GeoTIFF`. A `shpfile` a bementi fájl elérését írja le a `raszter_file` pedig a kimeneti fájl útvonalát.

A 19. sorban az előbb létrehozott rasztert futtatja le. A 21. sorban a `raszter_reteg` fogja eltárolni a létrehozott raszter rétegeket, amelyeknek a neve tartalmazza az évszámot is.

```

1.         raszter_file = os.path.join(Kiirro_mappa_utvonal, f'{ev}_raszter.tif')
2.
3.         raszter = [
4.             "gdal_grid",
5.             "-l",
6.             f'{ev}',
7.             "-zfield",
8.             "WQS_Rank",
9.             "-a",
10.            "nearest:radius1=0.0:radius2=0.0:angle=0.0:nodata=0.0",
11.            "-ot",
12.            "Float32",
13.            "-of",

```

```

14.         "GTiff",
15.         shpfile,
16.         raszter_file,
17.     ]
18.
19.     subprocess.run(raszter)
20.
21.     raszter_reteg = QgsRasterLayer(raszter_file, f"{ev}_raszter")

```

A színek változó egy listát hoz létre, ami először a kulcs értékeket string-é alakítja át, mivel ezek int formátumban vannak megadva. A for ciklusban a szótár kulcsain megyünk végig, amelyeket visszaalakítunk int formába és ezeket a számokat rendezzük. Ezek alapján jön létre egy map, amiben a számokhoz tartozó színek rendezve vannak.

A 6. sorban szin\_ramp néven létrehozunk egy QgsColorRampShader objektumot. A 7. sorban beállítjuk a színek diszkrét megjelenését, ami azt jelenti, hogy szakaszosan jelenjenek meg, ne pedig folytonosan változzanak. A 8.-13. sorig a színek és a Rankok megfelelő összekötését készíti el a QgsColorRampShader.ColorRampItem osztály segítségével. Az enumerate függvény felsorolja az értékeket és a megfelelő szint hozzáátársítja.

A 15. sorban létrehoz egy új QgsRasterShader objektumot, amelyet a raszter\_shader változóhoz rendelünk. A következő sorban beállítja a rács színátmenetét (szin\_ramp) a rács shader funkciójaként. Ez azt jelenti, hogy a raszter\_shader objektum használni fogja a korábban definiált színátmenetet a színezéshez. A 18.-20. sorig tartó renderer létrehoz egy QgsSingleBandPseudoColorRenderer objektumot, amely a rács rétegen egyetlen sávval dolgozik és alkalmazza rá a korábban definiált raszter\_shader-t. A raszter\_reteg a réteg adatszolgáltatóját jelöli, amelyből a renderer származtatja az adatokat.

A 22. sorban a raszter\_reteg a renderer alapján állítja be az interpoláció minden részletét. A 24. sorban pedig hozzáadjuk az előbb létrehozott réteget a térképhez, megjelenítve az évekre bontott interpolált eloszlás térképet.

```

1.     szinek = [
2.         wqs_rank_szinek[str(rank)]
3.         for rank in sorted(map(int, wqs_rank_szinek.keys()))
4.     ]
5.
6.     szin_ramp = QgsColorRampShader()
7.     szin_ramp.setColorRampType(QgsColorRampShader.Discrete)
8.     szin_ramp.setColorRampItemList(
9.         [
10.            QgsColorRampShader.ColorRampItem(rank, color)
11.            for rank, color in enumerate(szinek, 1)
12.        ]
13.    )

```

```
14.  
15.     raszter_shader = QgsRasterShader()  
16.     raszter_shader.setRasterShaderFunction(szin_ramp)  
17.  
18.     renderer = QgsSingleBandPseudoColorRenderer(  
19.         raszter_reteg.dataProvider(), 1, raszter_shader  
20.     )  
21.  
22.     raszter_reteg.setRenderer(renderer)  
23.  
24.     QgsProject.instance().addMapLayer(raszter_reteg)
```

#### 4.2.2. OSM

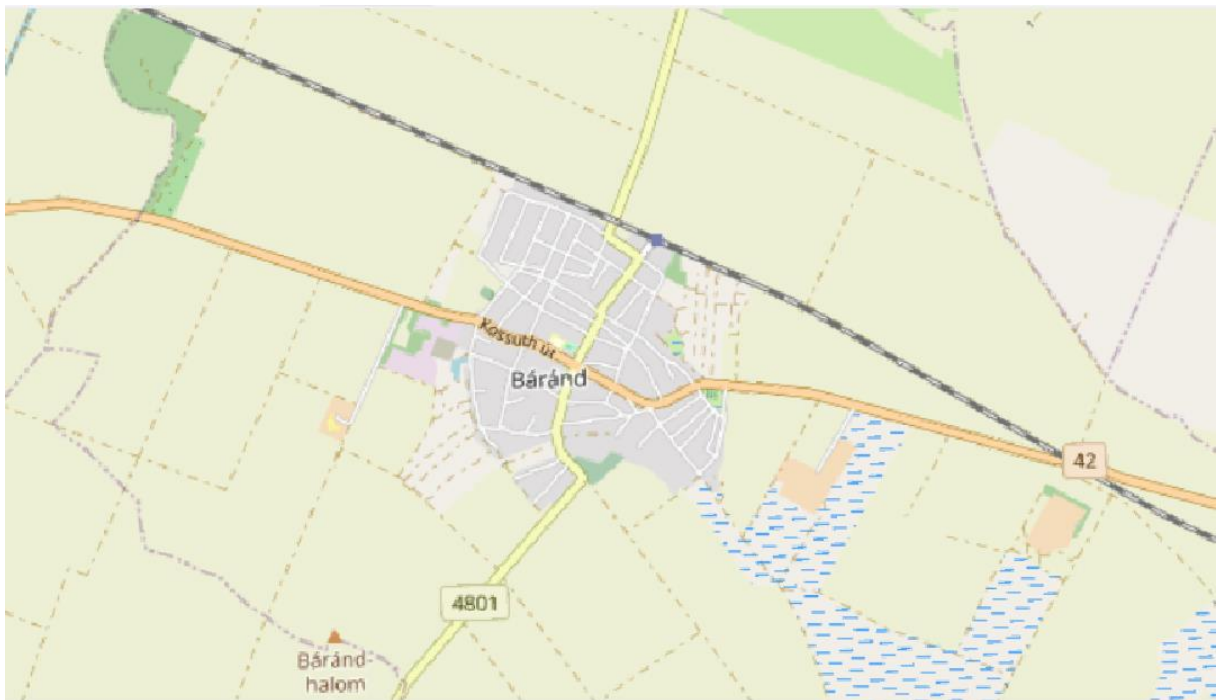
Az OSM (OpenStreetMap) egy világtérkép, amelyet bárki használhat és szerkeszthet, kiegészítheti a saját tudásával, felrajzolhatja a környékét, vagy frissítheti az általa fontosnak ítélt részeket és szabad licenc alatt elérhető.[28]

Ahhoz, hogy használni tudjuk ezt a térképet QGIS felhasználói felületén, annyit kell tennünk, hogy a bal oldalon található sávban a Böngésző részt nyitjuk meg. Azon belül pedig az XYZ Tiles-ra jobb egérgomb kattintással az Új kapcsolat-ra nyomva. Itt meg kell adni milyen címet szeretnénk adni a rétegnek (jelen esetben az OpenStreetMap) és meg kell adni egy URL-t, amit használathoz hív meg (<https://tile.openstreetmap.org/{z}/{x}/{y}.png>).

Ennek az elvégzését követően az XYZ Tiles lenyithatóvá válik és ott lesz az OpenStreetMap nevezetű réteg gomb. Ezt aktiválva megjelenik egy részlete a világtérképnek a QGIS térképi nézetében, amelybe tetszőlegesen tudunk navigálni, kicsinyíteni és nagyítani az adott földrajzi területre (9. ábra). Ez tökéletes alapot biztosít a tájékozódásra a vizsgált területről a számítási eredményeink betöltését követően (10. ábra).



9. ábra: OSM térkép részlet



10. ábra: Vizsgált terület az OSM térképen

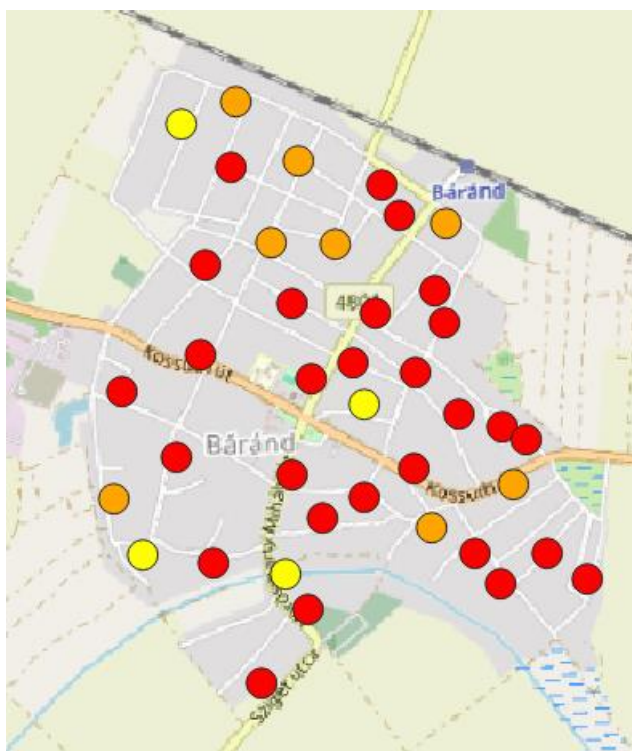
### 4.3. Teszteredmények és tematikus térképek

A monitoring kutak a 2013-as évben nagyon rossz vízminőséggel rendelkeztek, viszont a mérések alapján a 2022-es évre ez elkezdett javulni. A 3. táblázat szemlélteti a 2-es azonosítójú monitoring kút vízminőségi adatait a két különböző mintavételi évben.

3. táblázat: Egy kút értékei két különböző évben

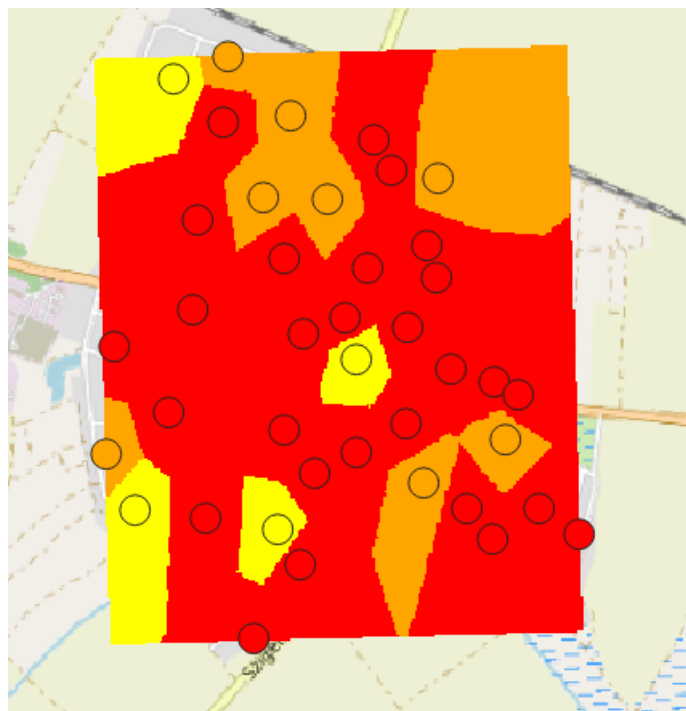
ID	Date	WQI	WQS	WQS Rank	Cd	Cds	Cds Rank	Lehetséges használat
2	2022	18,273	Kiváló vízminőség	1	0,892	Alacsony	2	Mezőgazdasági és ipari
2	2013	98,959	Nagyon rossz víz minőség	4	2,102	Közepes	3	Mezőgazdasági és ipari

A monitoring kutak vízminőségi eredményeinek vizuális interpretációja alapján kijelenthető, hogy a 2013-as mérések során a monitoring kutak döntő része rossz vízminőséggel jellemezhető (11. 12. ábra).



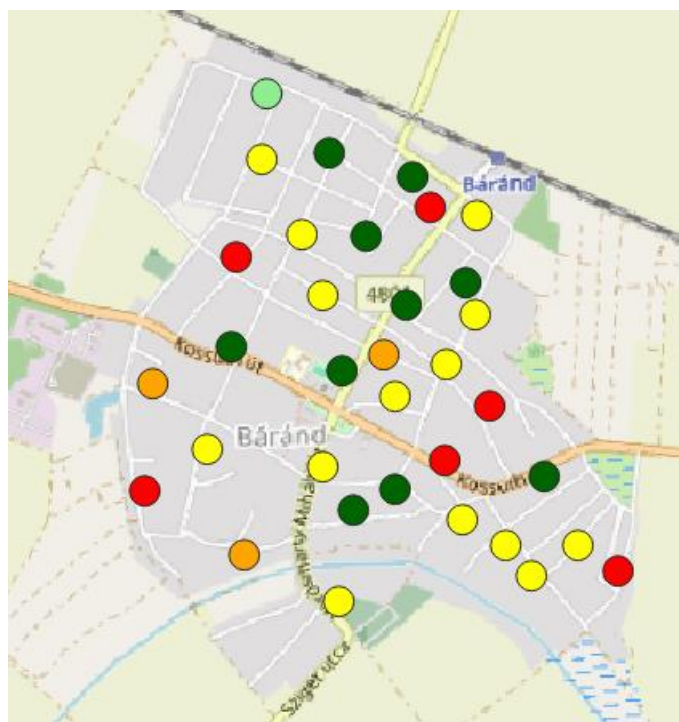
11. ábra: 2013-es évi mérések kategorizált eredményei



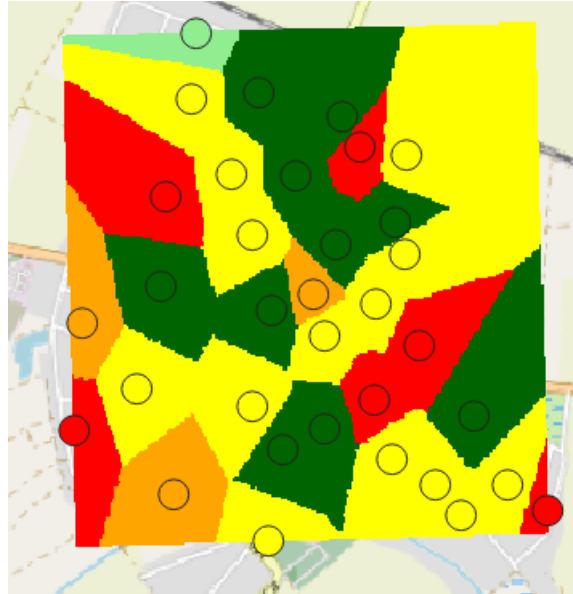


12. ábra: 2013-es évi mérésekből készített interpoláció eredményei

A 2022-es mintavétel után a monitoring kutak vizeinek minősége jelentősen javult. Néhány helyen a kitermelt víz még mindig nem alkalmas semmilyen használatra, de a legtöbb helyen már a jó, esetleg a kiváló jelző kategória is megjelent (13. 14. ábra).

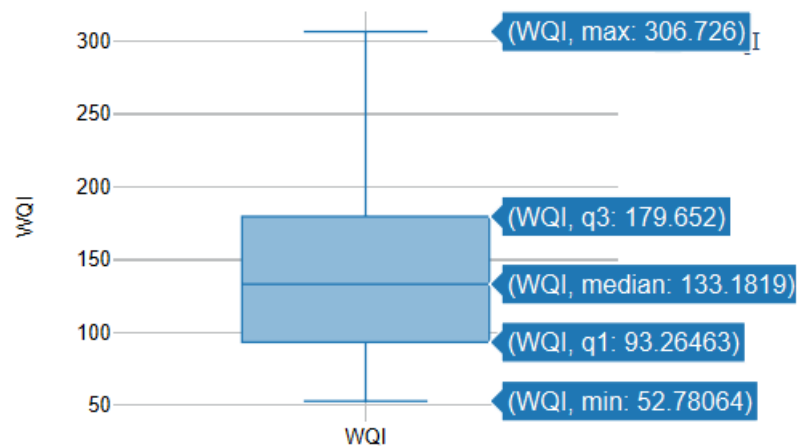


13. ábra: 2022-es évi mérések kategorizált eredményei



14. ábra: 2022-es évi mérésekből készített interpoláció eredményei

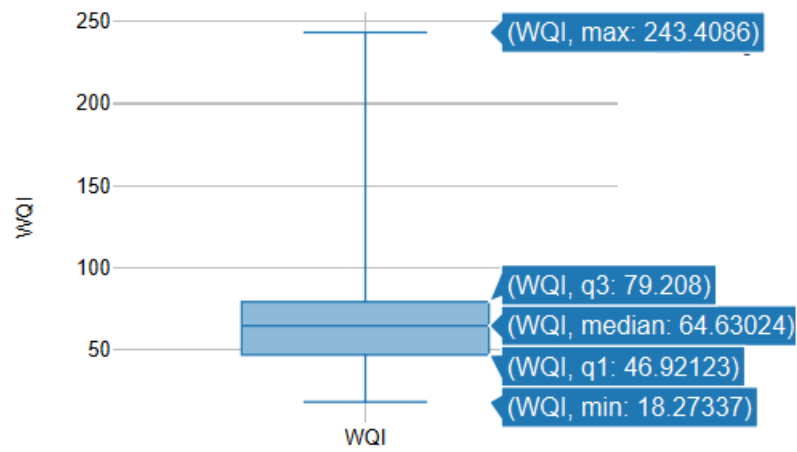
A 2013-as eredményhez készített boxplot a WQI értékek alapján a következő ábra mutatja be (15. ábra). Az ilyen megjelenítésen könnyen leolvasható a maximum és a minimum értéke is, valamint a medián is. A q1 és q3 értékek a kvartiliseket jelentik.



15. ábra: WQI boxplot a 2013-as mérések eredményéről

A WQI értékek alapján a 2022-es eredményekhez is készítettem boxplotot (16. ábra). A boxplotok készítéséhez a DataPlotly letölthető modult használtam. [29]





16. ábra: WQI boxplot a 2022-as mérések eredményéről

## 5. Összegzés

A szakdolgozatom legfőbb célja az volt, hogy felhívjam a figyelmet a térinformatikai fejlesztések mennyire megkönnyíthetik a környezeti kutatások adatainak feldolgozását. Az olyan számolások automatizálásával, amelyek nagyon gyakoriak és sokszor használtak, valamint azok azonnali kiértékelésével az ilyen folyamatok elvégzése felgyorsítható.

A dolgozatom elméleti részben ismertetésre került a térinformatika, mint multidiszciplináris tudományterület és a hozzá tartozó különböző felbontású rendszerek és specifikációi. Bemutattam a különböző térbeli adatokat és adatmodelleket is, amelyeket a térinformatikai szoftverek és rendszerek használnak. A felhasznált térbeli információs rendszer, a QGIS szintén bemutatásra került. Végül a modulfejlesztés lépéseit ismertettem.

Az elméleti részben továbbá a felhasznált vízminőség indexeket fejtettem ki, illetve, hogy ezek milyen szerepet töltenek be a monitorizált adatok feldolgozásában. Ismertettem a vízminőség indexek alkalmazási területeit és milyen jelentőséggel is bírnak. Bemutattam a modulban kiszámításra kerülő vízminőség indexek képleteit és működésüket. Valamint, hogy milyen globális vagy regionális adatbázisok találhatók a vizek minőségének ellenőrzésére.

A gyakorlati projekt munka során egy Python alapú modult készítettem QGIS szoftverkörnyezetben, ami két vízminőség index kiszámítására és térképen történő vizualizációjára alkalmas. Az adatok több mint 10 éves monitorizált adatokon alapuló kutatás méréseinek eredményei, amit Balla Dániel tanár úr biztosított számomra. A vizsgált kutak Bárádon találhatók, amelyet egy az Alföld keleti részén elterülő település.

A modul futtatásának eredményei automatikusan megjelennek a térképes nézetben, az egyes monitorizált pontok vízminőségi kategóriái alapján stilizálva egy OSM alaptérképpel együtt. A vízminőségi státuszok numerikus értékei alapján a modul interpolációt is végez a betáplált monitorizált pontok alapján.

A számítási és vizualizációs folyamatok eredménye így igazán látványos és figyelemfelkeltő a felhasználók számára. A projekt feladatorientált megközelítéséből adódóan legfőbb felhasználási célja a későbbi monitorizált adatok automatikus feldolgozása és kiértékelése.

Jövőbeli fejlesztési terveim között szerepel az UI finomítása, átdolgozása és a modul bővítése többféle vízminőségi mutató meghatározáshoz.

Az egyetem alatt szerzett programozási ismereteimet ebben a projektben tudtam megmutatni, felhasználni először teljesen önállóan. A projekt rengeteg komplex kihívást állított eléem, rengeteg új dolgot tanultam a fejlesztése alatt, illetve nagyon örültem, hogy olyan munkát hoztam létre, ami nem csak látványos, mert térképekkel és azokok kiegészítésével foglalkozik, de hasznos is lehet.

## Köszönetnyilvánítás

Szeretném megköszönni Balla Dániel tanár úrnak, hogy a Térinformatikai ismeretek tantárgy alatt felkeltette a figyelmem a téma iránt, valamint a dolgozat elkészítésének lehetőségét, és a sok támogatást a hónapok alatt.

Hálás vagyok minden olyan oktatónak, aki részt vett a tanulmányaimban, mivel nélkülük ez a szakdolgozat nem valósulhatott volna meg.

Nagyon köszönöm családtagjaimnak, akik minden lehetőséget megteremtettek számomra, hogy az egyetemen minél jobban teljesítsek. Hálásan köszönöm kitartó türelmüket és a mindennapi feladatokban nyújtott nem szűnő segítségüket!

## 6. Irodalomjegyzék

- [1] Kiss Kinga: Térinformatikai adatfeldolgozási és -elemzési eljárások automatizálása összehasonlító tájökológiai vizsgálatokhoz GRASS GIS segítségével. Pécsi Tudományegyetem Természettudományi Kar Földtudományok. 2014.  
[https://real.mtak.hu/144507/1/kiss\\_20142.pdf](https://real.mtak.hu/144507/1/kiss_20142.pdf) (Használva: 2024. 02. 03.)
- [2] Herczeg Ádám, Turai Endre: Geoinformatika. 2. A térinformatika. Miskolci Egyetem Földtudományi Kar, 2011.  
[https://dtk.tankonyvtar.hu/xmlui/bitstream/handle/123456789/8826/MFGFT6008T\\_02.pdf?sequence=2&isAllowed=y](https://dtk.tankonyvtar.hu/xmlui/bitstream/handle/123456789/8826/MFGFT6008T_02.pdf?sequence=2&isAllowed=y) (Használva: 2024. 02. 03.)
- [3] Dr. Habil. Mucsi László, Dr. Habil. Nagyváradi László, Dr. Kovács Ferenc, Dr. Szatmári József: Geoinformatika alapjai 13. fejezet. Szegedi Tudományi Egyetem Természeti Földrajzi és Geoinformatikai Tanszék, 2010. <http://earth.geo.u-szeged.hu/~laci/ab-Geoinfo-tananyag/ch13.html#id486326> (Használva: 2024. 02. 03.)
- [4] Herczeg Ádám, Turai Endre: Geoinformatika. 11. A terület- és településfejlesztés geoinformatikai vonatkozású rendszerei. Miskolci Egyetem Földtudományi Kar, 2011.  
[https://dtk.tankonyvtar.hu/xmlui/bitstream/handle/123456789/8826/MFGFT6008T\\_11.pdf?sequence=11&isAllowed=y](https://dtk.tankonyvtar.hu/xmlui/bitstream/handle/123456789/8826/MFGFT6008T_11.pdf?sequence=11&isAllowed=y) (Használva: 2024. 02. 03.)
- [5] Herczeg Ádám, Turai Endre: Geoinformatika. 9. A geoinformációk jelentősége a földtani kutatásban. Miskolci Egyetem Földtudományi Kar, 2011.  
[https://dtk.tankonyvtar.hu/xmlui/bitstream/handle/123456789/8826/MFGFT6008T\\_09.pdf?sequence=9&isAllowed=y](https://dtk.tankonyvtar.hu/xmlui/bitstream/handle/123456789/8826/MFGFT6008T_09.pdf?sequence=9&isAllowed=y) (Használva: 2024. 02. 03.)
- [6] Herczeg Ádám, Turai Endre: Geoinformatika. 10. A geoinformációk jelentősége a bányászati termelésben és a környezetgazdálkodásban. Miskolci Egyetem Földtudományi Kar, 2011.  
[https://dtk.tankonyvtar.hu/xmlui/bitstream/handle/123456789/8826/MFGFT6008T\\_10.pdf?sequence=10&isAllowed=y](https://dtk.tankonyvtar.hu/xmlui/bitstream/handle/123456789/8826/MFGFT6008T_10.pdf?sequence=10&isAllowed=y) (Használva: 2024. 02. 03.)
- [7] Herczeg Ádám, Turai Endre: Geoinformatika. 6. A nyitott térinformatikai rendszerek. Miskolci Egyetem Földtudományi Kar, 2011.  
[https://dtk.tankonyvtar.hu/xmlui/bitstream/handle/123456789/8826/MFGFT6008T\\_06.pdf?sequence=6&isAllowed=y](https://dtk.tankonyvtar.hu/xmlui/bitstream/handle/123456789/8826/MFGFT6008T_06.pdf?sequence=6&isAllowed=y) (Használva: 2024. 02. 05.)

- [8] Budapesti Műszaki és Gazdaságtudományi Egyetem (BME) Automatizálási és Alkalmazott Informatikai Tanszék  
[http://www.agt.bme.hu/tutor\\_h/terinfo/t21.htm#vektor](http://www.agt.bme.hu/tutor_h/terinfo/t21.htm#vektor) (Használva: 2024. 02. 05.)
- [9] Budapesti Eötvös Loránd Tudományegyetem (ELTE) Látható Földtudományi Tanszék, 2005. <http://lazarus.elte.hu/hun/dolgozo/jesus/terinfo/raszter.htm> (Használva: 2024. 02. 05.)
- [10] dr. Tóth Géza: Térinformatika a gyakorlatban közgazdászoknak 2014. <https://gtk.uni-miskolc.hu/files/6405/Terinfo.pdf> (Használva: 2024. 02. 12.)
- [11] Balla Dániel, Mester Tamás: Assessment of water quality in the rural environment of the Hungarian Great Plain using analytic web tool, A talajvíz minőségének vizsgálata egy analitikus webes alkalmazás segítségével (Letöltve: 2023. 06. 07.)
- [12] Sály P., Erős T.: Vízfolyások ökológiai állapotminősítése halakkal: minősítési indexek kidolgozása. 2016. [https://real.mtak.hu/50323/1/Saly.Eros\\_Pisces.Hungarici\\_2016\\_u.pdf](https://real.mtak.hu/50323/1/Saly.Eros_Pisces.Hungarici_2016_u.pdf) (Használva: 2024. 02. 12.)
- [13] Csósz László: Lakossági, ipari vízfelhasználás és a vízfelhasználást veszélyeztető káresemények, különös tekintettel az ipari eredetű vízszennyezésekre. 2016. <https://folyoirat.ludovika.hu/index.php/mkk/article/view/2146/1417> (Használva: 2024. 02. 12.)
- [14] Dr. Clement Adrienne, Dr. Szilágyi Ferenc: Környezeti monitoring (Felszíni és felszín alatti víz mennyiségi és minőségi monitoring). 2011. [https://witch.mik.pte.hu/oktatas/Tanszeki\\_anyagok/Kornyezetmernok\\_Tanszek/Tant%E1rgyak/K%E6rnyezetv%E9delmi%20m%E9r%E9stechnika,%20monitoring/aj%E1nlott%20irodalom/egy%E9b/monitoring\\_eloadasok\\_oravazlata\\_bme\\_j%E3.pdf](https://witch.mik.pte.hu/oktatas/Tanszeki_anyagok/Kornyezetmernok_Tanszek/Tant%E1rgyak/K%E6rnyezetv%E9delmi%20m%E9r%E9stechnika,%20monitoring/aj%E1nlott%20irodalom/egy%E9b/monitoring_eloadasok_oravazlata_bme_j%E3.pdf) (Használva: 2024. 02. 12.)
- [15] NASA Earth Data hivatalos oldala: <https://www.earthdata.nasa.gov/learn/pathfinders/agricultural-and-water-resources-data-pathfinder> (Használva: 2024. 02. 12.)
- [16] OVF (Országos Vízügyi Főigazgatóság) hivatalos oldala: <https://www.ovf.hu/carousel/udvozoljuk-az-orszagos-vizugyi-foigazgatosag-honlapjan> (Használva: 2024. 02. 12.)
- [17] OSGeo hivatalos oldala: <https://www.osgeo.org/about/> (Használva: 2024. 02. 18.)

- [18] James Gray: Getting Started With Quantum GIS. 2008.  
<https://www.linuxjournal.com/content/getting-started-quantum-gis> (Használva: 2024. 02. 18.)
- [19] Rob van Putten: QGIS tutorial - Plugin development - part 1 2021.  
<https://www.youtube.com/watch?v=GccxvQ1ypbc> (Használva: 2023. 06. 18.)
- [20] Rob van Putten: QGIS tutorial - Plugin development - part 2 2021.  
<https://www.youtube.com/watch?v=TXujpzI4Yvs&t=73s> (Használva: 2023. 06. 18.)
- [21] Rob van Putten: QGIS tutorial - Plugin development - part 2 2022.  
<https://www.youtube.com/watch?v=KQ8mOpVQ62w&t=258s> (Használva: 2023. 06. 18.)
- [22] Joel Lawhead: QGIS Python Programming Cookbook. 2015. (Letöltve: 2023. 06. 18.)
- [23] Tirkey Poonam, Bhattacharya Tanushree, Chakraborty Sukalyan: Water quality indices- important tools for water quality assessment: A review. 2015.  
<https://airccse.com/ijac/papers/1115ijac02.pdf> (Használva: 2024. 02. 24.)
- [24] Nirdeśh Kumar Ravi, Pawan Kumar Jha, Kriti Varma, Piyush Tripathi, Sandeep Kumar Gautam, Kirpa Ram, Manish Kumar, Vijay Tripathi: Application of water quality index (WQI) and statistical techniques to assess water quality for drinking, irrigation, and industrial purposes of the Ghaghara River, India. 2023.  
<https://www.sciencedirect.com/science/article/pii/S2772809923000266> (Használva: 2024. 02. 24.)
- [25] Balla Dániel, Kiss Emőke, Mester Tamás: Vízminőség adatok értékelése és megjelenítési lehetőségei egy WebGIS alkalmazás segítségével. (Letöltve: 2024. 02. 24.)
- [26] Márkus Béla: Térinformatika 11. Interpoláció és domborzatmodellezés. 2010.  
[https://dtk.tankonyvtar.hu/bitstream/handle/123456789/7808/0027\\_TEI11.pdf?sequence=1&isAllowed=y](https://dtk.tankonyvtar.hu/bitstream/handle/123456789/7808/0027_TEI11.pdf?sequence=1&isAllowed=y) (Használva: 2024. 02. 27.)
- [27] Hans van der Kwast: Spatial interpolation in QGIS. 2020.  
<https://www.youtube.com/watch?v=By-K5C7jDQc> (Használva: 2023. 08. 30.)
- [28] OSM hivatalos oldala: <https://www.openstreetmap.hu/> (Használva: 2024. 03. 02.)
- [29] Gis at UP: QGIS Plots and graphs. 2021.  
<https://www.youtube.com/watch?v=73SdnHvVrvA> (Használva: 2024. 03. 05.)