

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Інститут післядипломної освіти

Кафедра інформаційних технологій

Кваліфікаційна робота бакалавра

на тему: Розробка Python плагіну візуалізації наслідків повені
для геоінформаційної системи

Виконав студент групи КН-5
спеціальності 122 «Комп'ютерні науки»
Тойчиєв Бабамурат

Керівник к.геогр.н., доцент
Кузніченко Світлана Дмитрівна

Консультант _____

Рецензент к.техн.н., доцент
Гнатовська Ганна Арнольдівна

Одеса 2023

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	5
ВСТУП	6
1 ПОРІВНЯЛЬНИЙ АНАЛІЗ СУЧАСНИХ ГІС ПАКЕТІВ.....	8
1.1 Програмний комплекс ArcGIS Desktop	8
1.2 Пакет ГІС MapInfo Professional	10
1.3 Пакет QGIS і інші відкриті настільні ГІС.....	12
2 ОПИС ПРОГРАМНИХ ЗАСОБІВ РОЗРОБКИ ІНСТРУМЕНТІВ.....	16
ГЕООБРОБКИ	16
2.1 Створення інструментів за допомогою ModelBuilder	17
2.2 Створення інструментів геообробки за допомогою Python.....	21
3 РОЗРОБКА ІНСТРУМЕНТУ ГІДРОЛОГІЧНОГО АНАЛІЗУ	28
ПРОСТОРОВИХ ДАНИХ	28
3.1 Розробка інструменту гідрологічного аналізу даних	30
3.2 Створення моделі розрахунку зон затоплення.....	36
3.2.1 Алгоритм функції заливки	40
3.2.2 Створення скрипту для розрахунку зон затоплення	43
ВИСНОВКИ.....	48
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	50
ДОДАТОК А Лістинг скриптів інструментів геообробки просторової інформації	52

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БГД	– база геоданіх
ГІС	– географічна інформаційна система
ДДЗ	– дані дистанційного зондування
ПБГД	– персональна база геоданих
ПЗ	– програмне забезпечення
СУБД	– система управління базами даних
ЦМР	– цифрова модель рельєфу
DEM	– Digital Elevation Model (цифрова модель рельєфу)
ESRI	– Environmental Systems Research Institute
GRASS	– Geographic Resources Analysis Support System
NASA	– National Aeronautics and Space Administration
OGC	– Open Geospatial Consortium
OSGeo	– Open Source Geospatial Foundation
SAGA	– System for Automated Geoscientific Analyses
SDTS	– Spatial Data Transfer Standard
SRTM	– Shuttle Radar Topography Mission
UTM	– Universal Transverse Mercator
WGS-84	– World Geodetic System, 1984

ВСТУП

Водні ресурси і водогосподарський комплекс визначають соціально-економічну стійкість і напрям розвитку країни. Від водогосподарської та екологічної безпеки, рівня водозабезпечення населення і соціальної сфери якісною питною водою, безперебійності і достатності водопостачання галузей економіки, стану водних об'єктів і водних ресурсів, достовірності прогнозування надзвичайних водогосподарських ситуацій, їх своєчасного запобігання та мінімізації шкоди залежить національна безпека держави. Сучасні умови потребують якісно нові інформаційні інструменти в оперативному управлінні, використанні і охороні водних ресурсів та меліоративних об'єктів України.

На території України розташована велика кількість водних об'єктів і для автоматизації вирішення низки завдань з управління водними ресурсами необхідно використовувати можливості ГІС-аналізу та просторового моделювання. Геоінформаційні системи (ГІС) загального призначення мають широкий набір інструментів геообробки просторової інформації. Але для виконання окремих вузькоспеціалізованих завдань аналізу (наприклад, гідрологічного), іноді є необхідним створення власних інструментів, які можуть бути виконані в середовищі ГІС.

Метою даної кваліфікаційної роботи є Розробка Python плагіну візуалізації наслідків повені для геоінформаційної системи ArcGIS 10.7.

Для досягнення поставленої мети в роботі необхідно вирішити наступні завдання:

- провести дослідження моделей та форматів для представлення просторової інформації;
- виконати порівняльний аналіз сучасних ГІС пакетів та їх функціональних можливостей для здійснення гідрологічного аналізу;
- провести огляд сучасного програмного забезпечення для створення інструментів геообробки просторової інформації;

- розробити алгоритми розрахунку та візуалізації зон затоплення в ГІС;
- реалізувати інструменти для проведення геообробки даних, зокрема для виконання гідрологічного аналізу та розрахунку зон затоплення територій.

Структура кваліфікаційної роботи складається з анотації, вступу, чотирьох розділів, висновків, переліку посилань на 15 найменувань, додатків. Повний обсяг роботи становить 56 сторінки, містить 18 рисунків і 4 таблиць.

1 ПОРІВНЯЛЬНИЙ АНАЛІЗ СУЧАСНИХ ГІС ПАКЕТІВ

1.1 Програмний комплекс ArcGIS Desktop

Серед універсальних векторних ГІС особливе місце займає лінійка ГІС компанії ESRI (США).

Платформа ArcGIS 10 є оптимальним рішенням для побудови корпоративної ГІС. Вона побудована на основі стандартів комп'ютерної галузі, включаючи об'єктну архітектуру COM, .NET, Java, XML, SOAP, що забезпечує підтримку загальноприйнятих стандартів, гнучкість пропонуємих рішень, широкі можливості взаємодії. Фундаментальна архітектура ArcGIS забезпечує її використання у багатьох прикладних сферах і на різних рівнях організації роботи користувачів ГІС [1,2].

ArcGIS є інтегрованим набором програмних продуктів ГІС для розробки повнофункціональної ГІС. ArcGIS дозволяє користувачам розгортати функціональність ГІС в тому місці де вона потрібна в настільному варіанті, серверному або у вигляді спеціально створеного додатки; для Web або для роботи в польових умовах (рис.1.1).

Настільна ГІС є основною платформою для ГІС професіоналів, які об'єднують, створюють і використовують географічну інформацію і знання [3,4].

Серверна ГІС дозволяє програмному забезпеченню ГІС бути централізованим на серверах додатках, надаючи ГІС можливості великому числу користувачів через мережі.

Мобільна ГІС дозволяє польовому персоналу компанії збирати, зберігати, оновлювати, обробляти, аналізувати та відображати географічну інформацію. [5-7].

ArcGIS for Desktop – настільна ГІС, сімейство програм з єдиним інтерфейсом і загальними принципами роботи, але таких, що розрізняються за доступною функціональністю.



Рисунок 1.1 – Структура взаємодії основних програмних продуктів фірми ESRI

У ArcGIS for Desktop є чотири основні застосунки, які можна використовувати для картографування і візуалізації [8]:

1) ArcMap – основний застосунок геоінформаційної системи ArcGIS для картографування, редагування, аналізу даних і управління ними. Він використовується для роботи з двовимірними об'єктами.

2) ArcGlobe – застосунок для візуалізації геоданих в тривимірному вигляді, призначений для роботи з великими наборами даних, які повинні відображатися з різним рівнем деталізації. ArcGlobe є опціональною частиною додаткового модуля ArcGIS 3D Analyst.

3) ArcScene – додаток для тривимірної візуалізації сцен і ділянок з можливістю навігації. ArcScene є частиною додаткового модуля ArcGIS 3D Analyst.

4) ArcCatalog – каталог, який використовується для організації та управління різними типами географічної інформації: базами геоданих, файлами растрів, документами карт, глобусами, 3D-сценами, наборами інстру-

ментів геообробки, моделями, скриптами Python, ГІС-сервісами, метаданими тощо. Інформація представляється у вигляді дерева і дозволяє вибрати ГІС-елемент, переглянути його властивості та отримати доступ до інструментів.

Середовище ArcGIS містить багатий набір вбудованих засобів програмування. Починаючи з випуску ArcInfo 8.1, розробники ГІС використовують COM-технологію розробки ArcObjects SDK for .NET для таких модулів ArcGIS, як ArcMap, ArcCatalog і ArcScene. ArcObjects є бібліотекою COM-компонентів, написаних на мові програмування C++. Оскільки ArcGIS повністю побудована на ArcObjects, то можна використовувати COM, щоб повністю налаштувати і розширити платформу ArcGIS-meaning. Розширення моделі даних ArcObjects можна зробити практично на усіх COM і .NET сумісних мовах програмування (Visual Basic, C #, Visual Basic.NET, Java і Python).

Починаючи з версії 9.0 у ArcGIS з'явився Python [9], який зараз є основною мовою програмування для написання скриптів геообробки в ArcGIS і область його застосування продовжує розширюватися. Функціональність геообробки доступна при використанні бібліотеки ArcPy, яка забезпечує доступ до додаткових функцій, класів і модулів, що дозволяють швидко створювати як прості, так і складні робочі процеси.

1.2 Пакет ГІС MapInfo Professional

Універсальна ГІС MapInfo Professional розроблена компанією MapInfo Corp. (США) і орієнтована на широке коло користувачів. Система має великий набір функцій для роботи з не топологічними просторовими даними [10].

Для збереження даних можна використовувати СУБД ORACLE і MS SQL Server. ГІС MapInfo Professional має розвинуті засоби візуалізації і редагування картографічних даних (рис. 1.2), засоби тематичного картування (діапазони, діаграми, розмірні символи, щільність точок, індивідуальні значення).

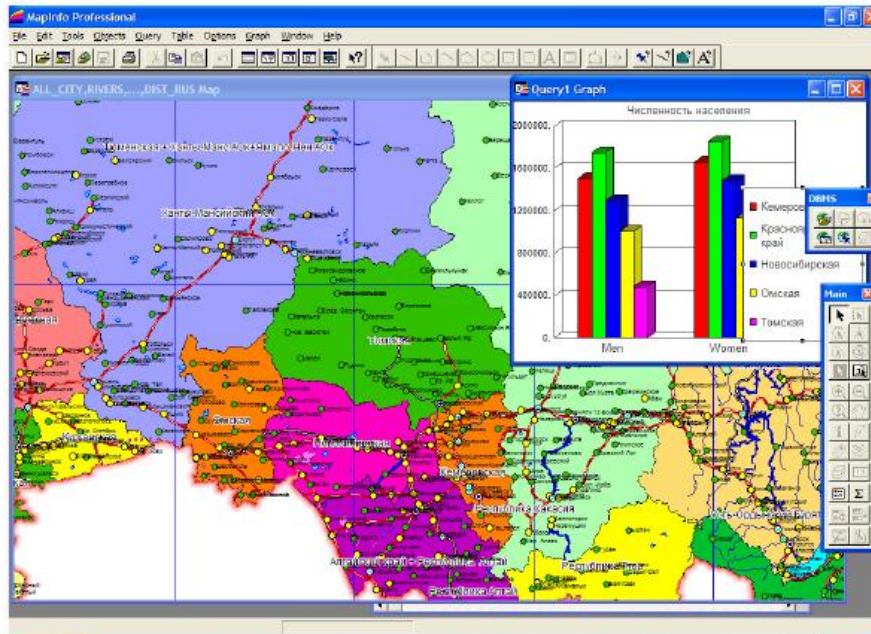


Рисунок 1.2 – Засоби візуалізації у ГІС MapInfo Professional

Для роботи з растровими зображеннями в системі передбачено два режими: з реєстрацією і без неї. Перший використовується для сумісництва координатно-прив'язаних растрів з векторними даними, а другий – тільки для їх перегляду.

MapInfo Professional підтримує велику кількість графічних форматів, у тому числі тих, що використовуються для збереження даних ДЗЗ. У системі є розвинуті засоби ділової графіки, що дозволяють формувати різні графіки (лінійні, стовбчасті, кругові, площадні, тривимірні). Всі графіки будуються за даними, що зберігаються у таблицях. Всі сформовані в ГІС данні (картографічні, табличні, ділова графіка і ін.) можуть бути виведені на друк. Для цього в системі є механізм створення макету друку.

Для розширення можливостей цієї системи використовується спеціальна мова програмування MapBasic. За допомогою цієї мови можна модифікувати інтерфейс користувача, додавати до ГІС нові функції, керувати картами. У мові також є можливість підключення динамічних бібліотек (DLL). Ком-

панія також пропанує додаткові модулі, що призначені для вирішення різних прикладних завдань [10].

1.3 Пакет QGIS і інші відкриті настільні ГІС

Визначення відкритого програмного забезпечення розроблено організацією Open Source Initiative і використовується для визначення відповідності ліцензії на програмне забезпечення стандартам відкритого ПЗ. Основні особливості відкритого ПЗ згідно з визначенням включають вільне поширення, доступний вихідний код, дозвіл на модифікацію цього вихідного коду [11].

Перелік FreeGIS.org на даний момент налічує приблизно 350 відкритих програмних пакетів ГІС різного типу. Відкриті ГІС створюються і підтримуються різними спільнотами та організаціями: комерційними компаніями, групами ентузіастів або дослідницькими організаціями. Найбільші проекти залучають досить велику кількість розробників і вкладення в розробку досить серйозні (табл. 1.1).

Таблиця 1.1 – Характеристики програмної бази та оцінки витрат деяких відкритих ГІС

Проект	Рядків коду, тис. (приріст)	Розробників, чол.	Витрати, чол./рік	Оцінка витрат, тис. \$USD
GRASS GIS	737 (42%)	62	200	11000
gvSIG	2162 (20%)	62	609	33495
Quantum GIS	440 (227%)	40	114	6270
GDAL	1035 (67%)	29	337	18535

Справжній бум розвитку відкритих ГІС і пов'язане з ним поява нових користувальницьких систем припадає на початок ХХІ століття. У цей період почалася розробка таких користувальницьких ГІС, як SAGA GIS (2001) в Ні-

меччині, gvSIG (2003) в Іспанії, міжнародний проект Quantum GIS (2002). У 2007 пропріетарний раніше пакет, призначений для ГІС-аналізу та завдань дистанційного зондування ILWIS (Integrated Land and Water Information System) офіційно став доступний під ліцензією GNU GPL, тим самим перейшовши до лав відкритого програмного забезпечення [11].

У 2006 році з метою підтримки і сприяння розробці проектів відкритих геопросторових технологій і баз даних з'явилася некомерційна організація Open Source Geospatial Foundation (скорочено OSGeo, www.osgeo.org). Крім підтримки відкритих проектів, під егідою OSGeo випускається журнал, ведеться розробка і поширення навчальних матеріалів, проводяться щорічні міжнародні конференції (FOSS4G), присвячені відкритому ПЗ ГІС.

Ще одна організація, яка відіграє важливу роль у розвитку відкритих ГІС – Open Geospatial Consortium (OGC, www.opengeospatial.org). Вона відповідальна за розробку стандартів взаємодії та обміну даними між різними ГІС-платформами. Крім університетів і адміністративних органів, членами OGC є в тому числі і розробники комерційних ГІС-платформ і баз даних. Наприклад стандарт, що описує інтерфейс доступу до географічних даних, що зберігаються в БД був реалізований як у відкритих базах даних (наприклад, PostgreSQL + PostGIS), так і в комерційних, таких як Oracle і IBM DB2.

Існуюче ПО ГІС можна умовно поділити на 3 класа: це веб ГІС, настільні ГІС і просторові бази даних. У табл. 1.2 представлені типові стеки відкритого ПЗ для настільних ГІС.

Розробка QGIS почалася в 2002 році групою ентузіастів. Метою розробки було створення простого у використанні і швидкого переглядача географічних даних для операційних систем сімейства Linux. Однак, із зростанням проекту з'явилася ідея використовувати QGIS як простий графічний інтерфейс для GRASS, отримуючи таким чином в своє розпорядження його аналітичні та інші функції.

Таблиця 1.2 – Інструментальні шари відкритих настільних ГІС платформ

Тип ПЗ	Представники	Група
Додатки	QGIS, GRASS, OSSIM, uDig, MapWindow GIS	Користувальницький інтерфейс
Середовище розробки	Eclipse, QT, OpenGL, SharpDevelop	
Високорівневі утилити	GeoTools, PostGIS, MapWindow GIS ActiveX	Збереження даних
Високорівневі скриптові мови програмування	Python, Perl, R	Обробка даних
Низькорівневі утилити	Shapelib, JTS/GEOS, GDAL/OGR, GMT	
Низькорівневі мови програмування	C, C++, Java, Fortran, C#, VB.NET	Системне ПЗ
Операйна система	Linux, Microsoft Windows	

На сьогоднішній момент група розробників QGIS вирішила початкові завдання і працює над розширенням функціональних можливостей, давно вийшли за рамки простого переглядача. За рахунок використання крос-платформного тулкіта QT, QGIS доступна для більшості сучасних платформ (Windows, Mac OS X, Linux) і поєднує в собі підтримку векторних і растрових даних, а також здатна працювати з даними, що надаються різними картографічними веб-серверами і багатьма поширеними просторовими базами даних. Функціональність QGIS може бути розвинена за допомогою створення модулів розширення на C++, або Python.

QGIS має одне з найбільш розвинених співтовариств в середовищі відкритих ГІС, при цьому кількість розробників постійно збільшується, чому

сприяє наявності хорошої документації по процесу розробки і зручна архітектура.

2 ОПИС ПРОГРАМНИХ ЗАСОБІВ РОЗРОБКИ ІНСТРУМЕНТІВ ГЕООБРОБКИ

Геообробка – це обробка географічної інформації, одна з основних функцій географічної інформаційної системи. Вона дає можливість створення нової інформації шляхом виконання операцій над існуючими даними. Будь-яка зміна або вилучення інформації при роботі з даними, є рішенням завдань геообробки. Це може бути, наприклад, завдання конвертації географічних даних в інший формат, або завдання, яке може полягати в послідовному виконанні декількох операцій, таких як вирізання, вибірка і подальший перетин наборів даних.

В ArcGIS є можливість виконувати завдання геообробки декількома способами, а саме [12]:

- запустити інструмент, скориставшись його діалоговим вікном;
- запустити інструменти з командного рядка;
- побудувати модель, послідовно використовуючи ряд інструментів геообробки, і запустити її;
- створити і запустити скрипт, який ініціює операції інструментів геообробки.

Кожен із системних інструментів, який встановлюється разом з ArcGIS, призначений для виконання однієї операції з географічними даними. Як правило, ці інструменти слід запускати послідовно, тобто результати роботи одного інструменту є вхідними даними для іншого інструмента. Перетворивши спеціалізовані моделі і скрипти у власні інструменти, з'являється можливість використовувати їх поряд з іншими системними інструментами при створенні послідовності завдань. Можна створити власну бібліотеку інструментів, яка будить вирішувати важливі для всього ГІС проекту індивідуальні завдання.

2.1 Створення інструментів за допомогою ModelBuilder

Конструктор моделей ArcGIS (Model Builder) – потужний засіб, що дозволяє, за допомогою побудови відповідних моделей більш ефективно управляти процесами обробки даних в ArcGIS. Елементом моделі може бути один з інструментів з ArcToolbox, скрипт на мові Python, VBScript та ін.

Модель в ModelBuilder – це відображення опису робочих процесів, які з'єднані один з одним, в послідовності інструментів геообробки, подаючи вихід одного інструменту в інший інструмент в якості входу. ModelBuilder можна також розглядати як візуальну мову програмування для побудови робочих потоків.

Основні переваги ModelBuilder [13]:

- ModelBuilder – це зручний в роботі додаток для створення і запуску робочих потоків, що містять послідовність інструментів;
- за допомогою ModelBuilder можна створювати власні інструменти. Інструменти, створені за допомогою ModelBuilder, можуть використовуватися в засобах підтримки скриптів Python і в інших моделях;
- ModelBuilder, поряд із засобами підтримки скриптів, надає можливість інтеграції ArcGIS з іншими додатками.

Вікно ModelBuilder складається з вікна відображення, в якому будується блок-схема моделі, головного меню і панелі інструментів, яку можна використовувати для роботи з елементами у блок-схемі моделі (рис. 2.1).

ModelBuilder має три набори параметрів, що використовуються в моделі:

1) Властивості моделі – ці властивості, які дозволяють задавати назву моделі, напис, опис, відносний шлях, властивості параметрів, змінні середовища моделі, довідку і кількість проходів.

2) Властивості діаграми – це властивості, що дозволяють змінювати розташування елементів на діаграмі, а також її колір і стиль.

3) Властивості відображення – це властивості, що дозволяють змінювати зовнішній вигляд і інші графічні властивості окремих елементів.

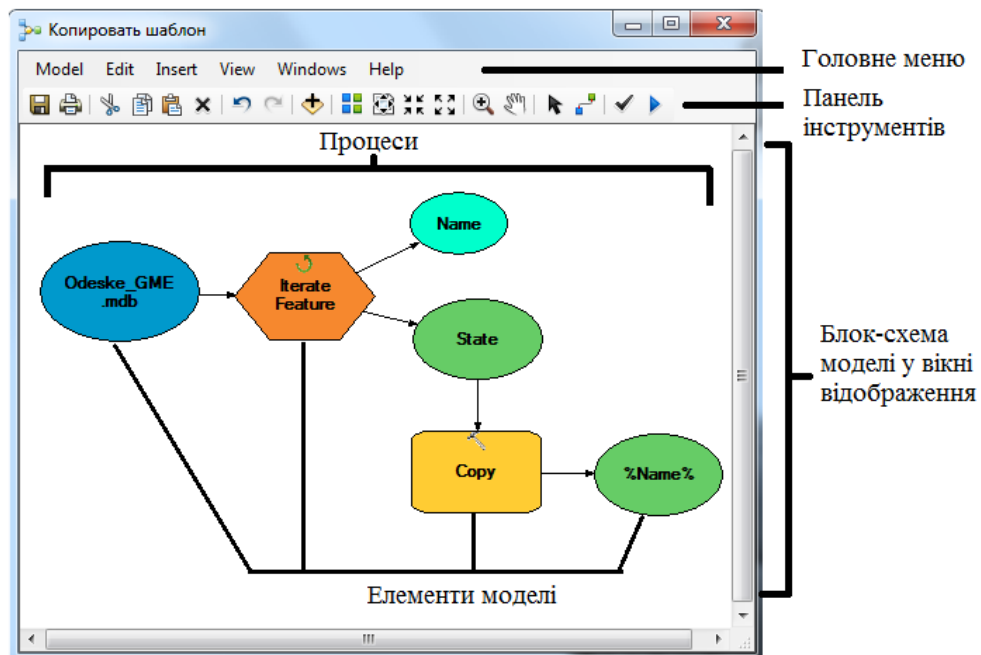


Рисунок 2.1 – Вигляд вікна ModelBuilder

Процес моделі складається з інструментів і всіх змінних, з'єднаних з ним (рис. 2.1). З'єднувачі показують послідовність виконання. У моделі часто буває кілька процесів, і вони можуть виконуватися по ланцюжку, так що вихідні дані одного процесу стають вхідними даними іншого. Кожен процес в моделі знаходиться в одному з чотирьох станів: «Не готовий до виконання (Not ready-to-run)», «Готовий до виконання (Ready-to-run)», «Виконується (Running)» або «Було виконано (Has-been-run)».

Моделі можна запускати з ModelBuilder або з діалогового вікна інструменту моделі, вікна Python, або в автономному скрипті. При запуску процесів з ModelBuilder їх виконання можна відстежувати в діалоговому вікні ходу геобробки.

ModelBuilder дозволяє створювати прості або складні моделі за допомогою ряду методів, які дозволяють ефективно управляти логічною послідо-

вністю виконання моделі, її даними, а також швидше здійснювати завдання, що повторюються. В групі інструментів Ітератори (Iterator) ModelBuilder містить дванадцять ітераторів, які повторюють процес або набір процесів для наборів вхідних значень (табл.2.1).

Таблиця 2.1 – Опис групи інструментів Ітератори (Iterator)

Ітератор	Опис
For	Виконує ітерації з початкового до кінцевого значення даної змінної. Він працює так само, як і оператор For в будь-якій мові програмування, виконуючи ітерацію задану кількість разів.
While	Виконується, поки умова є істиною.
Iterate Feature Selection	Виконує ітерації над об'єктами в класі просторових об'єктів.
Iterate Row Selection	Виконує ітерації для рядків в таблиці.
Iterate Field Values	Виконує ітерації для кожного значення в поле.
Iterate Multivalue	Виконує ітерації для списку значень
Iterate Datasets	Виконує ітерації для кожного набору даних в Робочій області (Workspace) або в Наборі класів об'єктів (Feature Dataset).
Iterate Feature Classes	Виконує ітерації для класів просторових об'єктів в робочій області або в наборі класів об'єктів.
Iterate Files	Виконує ітерації над файлами в папці.
Iterate Rasters	Виконує ітерації для растрів в Робочій області (Workspace) або в Каталозі растрів (Raster Catalog).
Iterate Tables	Виконує ітерації для таблиць в робочій області.
Iterate Workspaces	Виконує ітерації з робочими областями в папці.

Існує сім інструментів геообробки, які підтримують додаткові функції в ModelBuilder (табл.2.2). Ці інструменти можна використовувати в діалогових вікнах і скриптах.

Таблиця 2.2 – Опис інструментів тільки моделей (Model Only)

Інструменти	Опис
Calculate Value	Інструмент видає результуюче значення на основі зазначеного виразу Python
Collect Values	Інструмент розроблений для того, щоб збирати вихідні дані ітератора або конвертувати список декількох значень в один набір вихідних даних. Вихідні дані Collect Values можна використовувати як вхідні дані для таких інструментів як Злиття, Додати і Мозаїка
Get Field Value	Інструмент отримує значення першого рядка заданого поля з таблиці
Merge Branch	Інструмент об'єднує два або більше логічних гілок в один вихід
Parse Path	Інструмент отримує з набору вхідних даних його файл, шлях, тільки ім'я файлу і тільки розширення. Вихідні дані можуть використовуватися як вбудовані змінні в імені вихідних даних інших інструментів.
Select Data	Інструмент вибирає дані в батьківському елементі даних, наприклад, папці, базі геоданих, наборі класів об'єктів або покритті.
Stop	Виводить модель з циклу ітерації, якщо вхідний значення встановлено в True або False. Він функціонально схожий на ітератор While, але корисний для зупинки моделі в тому випадку, якщо в моделі є один ітератор While і не можна додати додаткові ітератори.

Інструменти моделі повністю інтегровані в середовище геообробки і працювати з ними можна так само, як з системними інструментами або інструментами скриптів. Як і всі інші інструменти геообробки, інструменти моделей можна запускати з діалогового вікна, через скрипти Python, а також додавати і запускати їх в іншій моделі. Скрипти Python і зовнішні програми, що запускаються за допомогою скрипта, можна інтегрувати в модель так само, як і системні інструменти, що додаються і запускаються в моделі [13].

2.2 Створення інструментів геообробки за допомогою Python

Мова Python є незалежною, міжплатформною, відкритою мовою програмування, швидкою, потужною і легкою в освоєнні. Вона широко використовується і підтримується [14].

Python з'явився в ArcGIS версії 9.0. З тих пір він використовувався в якості однієї з мов для написання скриптів, що містять процеси геообробки; область його застосування продовжує розширюватися. Кожен випуск розширював можливості Python і робив його використання все більш зручним.

ESRI остаточно впровадив Python в ArcGIS та розглядає цю мову в якості основного засобу, який задовольняє всі потреби користувачів. Перелічимо деякі переваги Python:

- легкий у вивченні, він ідеально підходить для початківців, залишаючись при цьому відмінним засобом для досвідчених користувачів;
- відмінно масштабований, він підходить як для великих проектів, так і для маленьких одноразових програм, відомих як скрипти;
- портативність і міжплатформність;
- вбудованість (написання скриптів в ArcGIS);
- стабільна і впевнена робота;
- велике співтовариство користувачів.

Python поширюється на всю систему ArcGIS, перетворюючись в мову аналізу, перетворення даних, автоматизації картографічних процесів, і дозволяє збільшити продуктивність цих робіт.

Нижче наведені основні терміни, пов'язані з геообробкою за допомогою Python [15]:

- ArcPy – бібліотека ArcPy забезпечує доступ з Python до всіх інструментів геообробки, включаючи додаткові модулі, а також пропонує велику кількість корисних функцій і класів для роботи з даними ГІС. Використовуючи Python і ArcPy, можна розробляти велику кількість зручних програм для роботи з географічними даними.

- Модулі ArcPy – являє собою файл Python, що містить функції і класи. ArcPy підтримується різними модулями, включаючи модуль доступу до даних (arcpy.da), модуль картографії (arcpy.mapping), додатковий модуль ArcGIS Spatial Analyst (arcpy.sa) і додатковий модуль ArcGIS Network Analyst (arcpy.na).

- Класи ArcPy – класи можна використовувати для створення об'єктів, що їх називають екземплярами. Такі класи ArcPy як SpatialReference і Extent часто використовуються як ярлики для завдання параметрів для інструментів геообробки, які інакше довелося б ставити у вигляді складних рядків.

- Функції ArcPy – частина програми, що виконує певне завдання. Функція може включатися в більш велику програму. У ArcPy всі функції геообробки представлені у вигляді функцій, проте не всі функції є інструментами геообробки. Крім інструментів в ArcPy є кілька функцій для поліпшення робочих процесів геообробки з використанням Python. Функції (часто звані методами) можуть використовуватися для створення списків певних наборів даних, вилучення властивостей набору даних, перевірки імені таблиці перед її додаванням в базу геоданих, а також для виконання багатьох інших корисних завдань геообробки.

- Автономний скрипт Python – це виконуваний файл з розширенням .py, який можна запустити з командного рядка, з інтегрованої середовища розробки Python (IDE) або двічі клацнувши файл .py в провіднику Windows.
- Інструмент-скрипт Python – це скрипт Python, який доданий в набір інструментів геообробки. Після додавання в якості інструменту-скрипта, він починає вести себе як будь-який інший інструмент геообробки – його можна відкрити і запустити з діалогового вікна, використовувати у вікні Python або ModelBuilder, а також викликати з інших скриптів та інструментів-скриптів.
- Вікно Python. За допомогою вікна Python можна легко і швидко скористатися Python безпосередньо з ArcGIS для інтерактивного запуску інструментів геообробки і функцій, а також скористатися перевагами, які дає використання модулів і бібліотек Python. Це вікно також дозволяє користувачам вивчати Python. Вікно Python може використовуватися для виконання окремих рядків коду на Python з висновком повідомлень про результат в цьому ж вікні. Це корисно для експериментів з синтаксисом і роботи з кодом невеликої довжини, а також дає можливість тестування завдань поза скрипта.
- Надбудова Python (add-in) – призначена для користувача функціональність, яка була реалізована на мові Python, наприклад, набір інструментів на панелі інструментів, включених в додаток ArcGIS for Desktop, призначена для забезпечення додаткових функціональних можливостей із супроводу призначених для користувача завдань. Для ясності процесу розробки надбудов Python слід завантажити і використовувати майстер надбудов Python для оголошення типу настройки. Майстер згенерує всі необхідні для роботи надбудови файли.
- Набори інструментів Python – набори інструментів геообробки, створені повністю в Python. Набір інструментів Python і інструменти, що містяться в ньому, виглядають і діють так само, як інструменти та набори інструментів, створені будь-яким іншим способом. Набір інструментів Python (.pyt) являє собою файл ASCII, що визначає набір інструментів і один або кілька інструментів.

ArcPy – це пакет, який заснований на успішному модулі `arcgisscripting`. Його метою є створення основи для успішного і продуктивного виконання аналізу географічних даних, конвертації даних, управління даними і автоматизації карти в Python.

Додатки та скрипти ArcGIS написані з використанням ArcPy дозволяють отримати доступ до численних модулів Python, розроблених користувачами ГІС і програмістами, що працюють в різних галузях. Ще одна перевага використання ArcPy в середовищі Python полягає в тому, що Python є універсальною мовою програмування, яка дозволяє працювати в режимі інтерпретації, що дає можливість швидко моделювати і перевіряти скрипти в інтерактивному середовищі, а також підтримує можливість написання великих додатків.

З технічної точки зору інструменти геообробки представляють собою функції, які можна викликати з `arcpy`, як і будь-яку іншу "рідну" функцію цього модуля. Проте, щоб уникнути плутанини між інструментами і відмінними від них функціями, такими як утиліта `ListFeatureClasses()`, існують чіткі відмінності:

- інструменти документуються інакше, ніж функції. У кожного інструменту є довідкова сторінка в довідковій системі ArcGIS Desktop. Функції документуються в документації ArcPy;

- інструменти на відміну від функцій повертають об'єкт `result`;

- інструменти створюють повідомлення, до яких можна звертатися за допомогою багатьох функцій, таких як `GetMessages()`. Функції не створюють повідомлень;

- інструменти ліцензуються на рівні продукту (ArcGIS for Desktop Basic, Standard або Advanced) або на рівні додаткових модулів ArcGIS Network Analyst, ArcGIS Spatial Analyst та ін. Функції не потребують ліцензування, вони встановлюються разом з ArcPy.

У наступному прикладі використовуються інструменти з наборів `Data Management` і `Conversion`. До вхідного класу об'єктів, який містить список ву-

лиць, додається поле, потім воно обчислюється, а клас об'єктів завантажується в базу геоданих ArcSDE.

```
>>> import arcpy
>>> Arcpy.AddField_management ("c:/data/Odesska.mdb/
streets", "LENGTH_MILES", "TEXT")
>>> Arcpy.CalculateField_management ("c:/data/Odesska.mdb/
streets", "LENGTH_MILES", "!shape.length@miles!", "PYTHON_9.3")
>>> Arcpy.FeatureClassToFeatureClass_conversion ("c:/data/
Odessa.mdb/streets", "Database Connections /MySDE.sde/
OdessaDataset", "streets")
```

При виконанні інструменту геообробки результат його роботи повертається у вигляді об'єкта `result`. Зазвичай це шлях до вихідного набору даних, який був створений або оновлений за допомогою інструменту. В інших випадках це можуть бути інші типи значень, такі як число або логічне значення. Якщо вихідні дані є багатозначним параметром, значення можуть бути повернуті як список в списку.

У наступних прикладах коду показано, як захоплюються повернуті значення і якими вони можуть бути.

Повертається шлях вихідного класу просторових об'єктів. Результат можна використовувати як вхідні дані для іншої функції.

```
>>> Result = arcpy.Buffer_analysis ("rivers", "riverBuf",
"50 METERS")
>>> Print result
C:\data\Odessa.mdb\riverBuf
>>> Arcpy.Clip_analysis("streets", result, "streets_50m")
Повертається кількість просторових об'єктів.
>>> Result = arcpy.GetCount_management("streets_50m")
>>> Print result.getOutput (0)
54
```

Повертається список індексів просторової сітки за замовчуванням для класу просторових об'єктів.

```

>>> Result = arcpy.CalculateDefaultGridIndex_management(
"streets_50m")
>>> For i in range (0, result.outputCount):
... Print result.getOutput (i)
...
560
200
0

```

Параметри середовища геообробки можна розглядати як додаткові параметри, які впливають на результат роботи інструменту. Вони відрізняються від звичайних параметрів інструментів тим, що вони задаються окремо від інструменту і використовуються інструментом під час його роботи. Такі параметри середовища, як область інтересу, система координат вихідного набору даних і розмір комірки нового набору растрових даних, можна задати заздалегідь, потім вони будуть використані під час роботи інструменту.

Параметри середовища доступні у вигляді властивостей класу env. Ці властивості можна використовувати для отримання поточних значень параметрів середовища і для їх завдання. Нижче наведені приклади використання значень параметрів середовища.

Задаються параметри середовища робочої області.

```

>>> Arcpy.env.workspace = "c:/data/Odesska.mdb"
>>> Arcpy.Buffer_analysis( "streets", "streetBuf", "500
METERS")

```

Задається індекс просторової сітки для значення, що повертається інструментом.

```

>>> Arcpy.env.spatialGrid1 =
arcpy.CalculateDefaultSpatialGridIndex_management("streets").
GetOutput (0)

```

Повертаються поточні настройки розміру комірки растра і перевіряється, чи використовується це значення в якості вихідного.

```

if arcpy.env.cellSize! = 30:
arcpy.env.cellSize = 30

```


Крім інструментів ArcPy містить велику кількість функцій для поліпшення підтримки процесу геообработки. Функції можна використовувати для створення списків певних наборів даних, отримання властивостей наборів даних, перевірки наявності даних, перевірки імен таблиць перед додаванням їх в базу геоданих або для виконання безлічі інших завдань, що вирішуються за допомогою скриптів.

У наступному прикладі коду показано отримання властивостей даних і перевірка додаткового модуля.

```
import arcpy
# Prints True
print arcpy.Exists("c: /data/Odesska.mdb/streets")
# Prints NAD_1983_StatePlane_Oregon_North_FIPS_3601_Feet
sr = arcpy.Describe("c: /data/Odesska.mdb/streets").
spatialReference
print sr.name
# Prints Available
print arcpy.CheckExtension ("spatial")
arcpy.CheckOutExtension ("spatial")
```

3 РОЗРОБКА ІНСТРУМЕНТУ ГІДРОЛОГІЧНОГО АНАЛІЗУ ПРОСТОРОВИХ ДАНИХ

Необхідно створити інструменти, які дозволять користувачам без навичок роботи з вбудованими засобами ArcGIS, виконувати дії геообробки даних. До інструментів, які необхідно створити згідно умовам технічного завдання відносяться:

- 1) Інструмент гідрологічної геообробки даних: побудова річної мережі та басейнів річок для вхідного растру ЦМР.
- 2) Інструмент розрахунку зон затоплення за вказаною висотою та координатами точки-джерела повені.

В якості вхідних даних для просторового моделювання будемо використовувати файли DEM – цифрові моделі рельєфу формату STRM.

SRTM (Shuttle Radar Topographic Mission) – глобальна радіолокаційна багаторазова топографічна зйомка, яка проведена для більшої частини території земної кулі. Основна база даних SRTM знімків за різні роки перебуває у вільному доступі на американському сервері usgs.gov у вигляді растрових файлів, представлених як квадратні області розміром $1 \times 1^\circ$ з просторовою роздільною здатністю 1 арксекунди (~ 30 м) або 3 арксекунди (~ 90 м). Це є достатнім для оціночних досліджень в масштабі області або адміністративного району.

Розглянемо докладніше основні етапи розробки зазначених вище інструментів геообробки.

Спочатку необхідно створити свій власний набір інструментів в ArcToolbox, який може зберігатися або в окремій папці на диску, або у базі геоданих (БГД). В цьому наборі інструментів і будуть зберігатися моделі нових інструментів і звідти вони будуть доступні для запуску користувачеві.

Для створення набору інструментів в окремій папці в панелі Catalog виваємо контекстне меню на назві поточної папки, в якій зберігається ГІС проект, і обираємо New – Toolbox (рис.3.1). А далі, таким же чином, в

Toolbox створюємо набір інструментів Toolset, в якому розміщуємо нові моделі Model (рис.3.2).

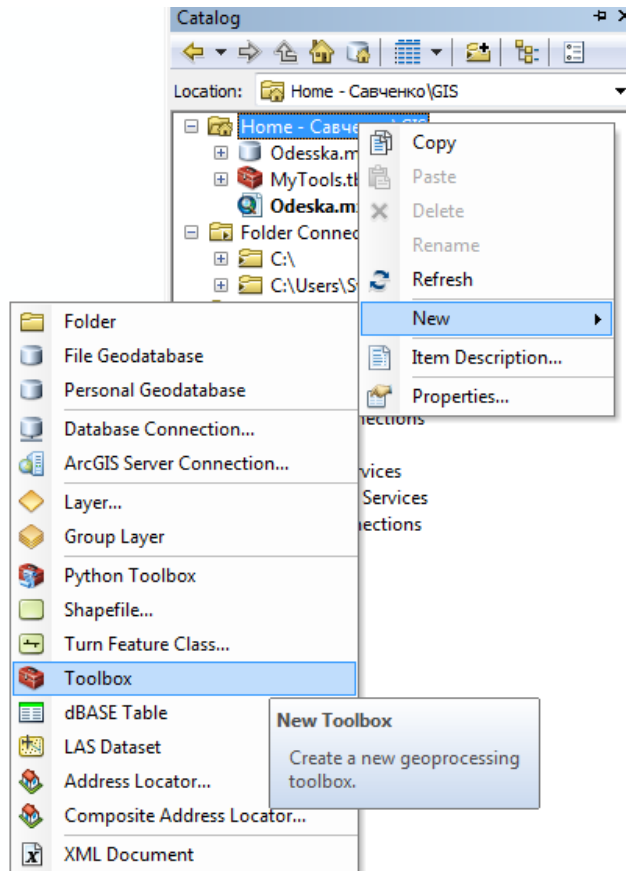


Рисунок 3.1 – Створення набору інструментів проекту

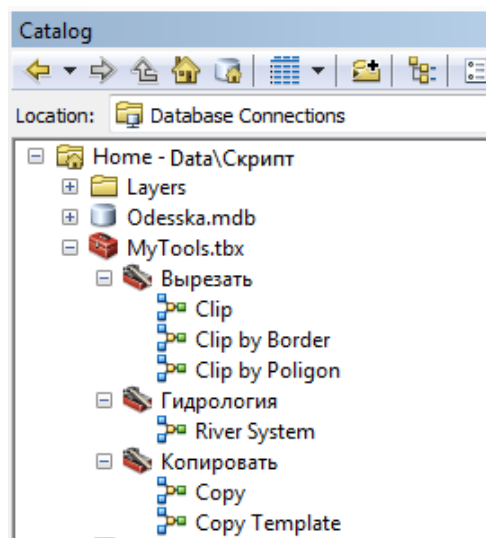


Рисунок 3.2 – Провідник набору власних інструментів геообробки

При створенні нової моделі слід також вказати які шляхи (абсолютні чи відносні) будуть в ній використовуватися. Якщо інструмент пропонується для використання на різних комп'ютерах, то бажано вказати відносні шляхи. Тому перше ніж розробляти модель у вікні ModelBuilder треба відкрити вкладку властивостей моделі, у якій вказуємо назву моделі і галочку використання відносних шляхів.

3.1 Розробка інструменту гідрологічного аналізу даних

Інструменти групи Гідрологія (Hydrology) використовуються для моделювання потоку води по поверхні. Інформація про форму земної поверхні може використовуватися в різних галузях, наприклад в регіональному управлінні водним господарством. Тому має важливе значення розуміння принципів руху води по поверхні, а також впливу змін в даній області на напрямок потоку.

Розглянемо розробку власного інструменту, який дозволить користувачам абстрагуватися від послідовного використання вбудованих інструментів з функціями гідрологічного аналізу для моделювання руху води по поверхні. Створимо один інструмент для отримання гідрологічної інформації з цифрової моделі рельєфу (ЦМР).

З початку наведемо аналіз вбудованих засобів гідрологічного аналізу, який пропонується пакетом ArcGIS. Інструменти з групи Гідрологія можуть застосовуватися окремо і використовуватися послідовно для побудови мережі водотоків або виділення вододілів. Огляд інструментів цієї групи наведений в табл. 3.1 [8].

Інструменти гідрологічного моделювання в наборі інструментів додаткового модуля ArcGIS Spatial Analyst надають методи для опису фізичних компонентів поверхні. Гідрологічні інструменти дозволяють ідентифікувати приймачів, визначати напрямок потоку, обчислювати сумарний стік, розмежовувати вододіли і створювати мережі водотоків.

Таблиця 3.1 – Огляд інструментів групи Гідрологія

Інструмент	Опис
Басейн (Basin)	Створює растр, що містить контури всіх дренажних басейнів
Заповнення (Fill)	Заповнює локальні зниження в растрі поверхні для видалення всіх невеликих помилок і неточностей, притаманних даними.
Сумарний стік (Flow Accumulation)	Створює растр потоку накопичення в кожену клітинку. Можна додатково застосувати фактор ваги.
Напрямок стоку (Flow Direction)	Створює растр напрямку стоку з кожної комірки по найближчій сусідній комірці вниз по схилу найбільшої крутизни
Довжина лінії стоку (Flow Length)	Обчислює відстань вгору або вниз за течією, або зважену відстань, уздовж потоку для кожної комірки
Локальне зниження (Sink)	Створює растр, що визначає локальні зниження або райони внутрішнього дренажу
Прив'язка точки гирла (Snap Pour Point)	Прив'язує точки гирлов до комірки з найбільшим сумарним стоком у межах заданої відстані
Ідентифікація водотоків (Stream Link)	Присвоює унікальні значення секціям растрової лінійної мережі, розташованим між зчленуваннями ліній
Порядок водотоку (Stream Order)	Привласнює число, яке визначає порядок, сегментам растра, що представляють сегменти лінійної мережі
Водотік в просторовий об'єкт (Stream to Feature)	Перетворює растр, який представляє лінійну мережу в векторні об'єкти, що представляють мережу ліній
Вододіл (Watershed)	Визначає водозбірну область, розташовану вище набору комірок растра.

На рис. 3.3 показаний приклад водної мережі, отриманий на основі моделі рельєфу.

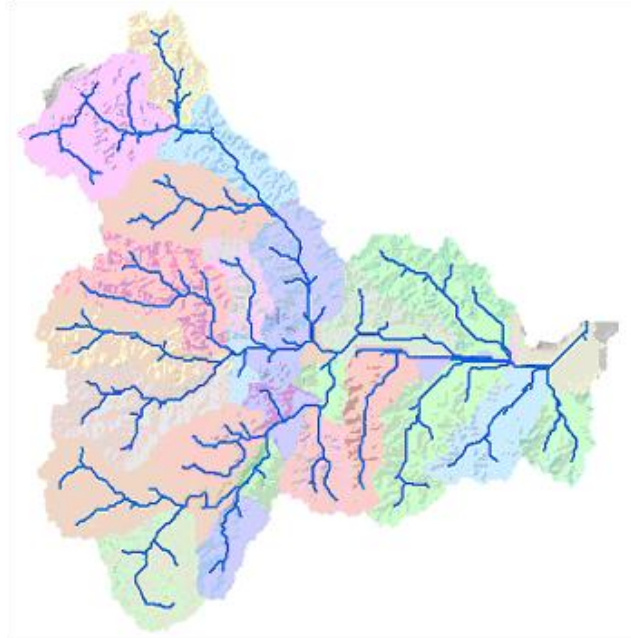


Рисунок 3.3 – Приклад мережі водотоків, отриманих на основі моделі рельєфу

Таким чином, використовуючи растр висоти або ЦМР в якості вхідних даних, можливо автоматично визначити дренажну систему і виявити кількісні характеристики системи.

При використанні ЦМР в якості вхідних даних інструменту Напрямок стоку (Flow Direction) визначається напрямок, в якому вода витікає з кожної комірки.

За допомогою інструменту Локальне зниження (Sink) визначаються локальні зниження в вихідній ЦМР. Локальне зниження – це комірка або набір просторово суміжних комірок, напрямку стоку з яких не може бути присвоєно жодного з восьми коректних значень напрямків в растрі напрямку стоку. Таке відбувається в тому випадку, якщо всі сусідні комірки вище, ніж комірка, що оброблюється, або коли дві комірки перетікають одна в одну, утво-

рюючи петлю з двох комірок. Тобто, це комірка з невизначеним напрямком стоку; навколо неї не існує комірок з меншою, ніж у цієї комірки, висотою. Локальні зниження, є проблематичними, тому що вода, яка потрапляє в них, не може виходити зовні. Якби локальні зниження були заповнені водою, в цих точках вода «витікала б» з поверхні. Точка гирла – це прикордонна комірка з найменшою висотою для водозбірної області локального зниження.

Щоб забезпечити належне дренажне картування, ці локальні зниження можна заповнити за допомогою інструменту Заповнення локальних знижень (Fill). Всі локальні зниження, які за глибиною менше, ніж обмеження по встановленому параметру z , нижче, ніж найнижча з прилеглих сусідніх комірок, і будуть заповнені до висоти своїх точок гирла. Інструмент Локальне зниження (Sink) можна використовувати замість інструменту Заповнення (Fill) – для знаходження числа знижень і визначення їх глибин. Знання глибини локальних знижень може допомогти у визначенні відповідного значення обмеження по z . Інструмент Fill може бути також використаний для видалення піків. Пік – це комірка, навколо якої немає комірок вище неї.

За допомогою інструменту Вододіл (Watershed) окреслюються вододіли для заданих розташувань.

Щоб створити мережу водотоків, слід використовувати інструмент Сумарний стік (Flow Accumulation) для обчислення растрі сумарного стоку в кожному комірку, що визначається як сума ваг для всіх комірок, які стікають в кожному комірку, розташовану вниз по схилу. Вихідний растр напрямку потоку, створений в попередньому кроці, використовується в якості вхідних даних. Комірки з невизначеним напрямком стоку будуть тільки накопичувати потік; вони не будуть робити внесок в стік в напрямку вниз по схилу. Сумарний стік ґрунтується на кількості комірок, з яких здійснюється стік в кожному конкретну комірку в вихідному растрі. Та комірка, яка обробляється в поточний момент часу, не враховується в сумі. Вихідні комірки з високим сумарним стоком – це ділянки концентрованого стоку; вони можуть бути використані для визначення русел водотоків. Вихідні комірки з сумарним стоком, рі-

вним нулю, – це локальні топографічні піки; вони можуть бути використані для виділення хребтів або ліній вододілу.

Далі визначається система мережі водотоків. Це завдання можна виконати за допомогою інструменту Умова (Con) або Алгебра карт (Map Algebra). До комірок застосовується порогове значення, що дозволяє вибрати комірки з високим сумарним стоком. Наприклад, для визначення мережі водотоків будуть використовуватися комірки, в які виконується стік з більш ніж 100 комірок. Для створення растра мережі водотоків з комірками з сумарним стоком більше або рівним 100 застосовують інструмент Умова (Con), який дозволяють привласнити таким коміркам значення, рівне одиниці, а іншим коміркам – значення фону (NoData – немає даних).

Щоб уявити порядок кожного сегмента в гідрологічній мережі, застосовують інструмент Порядок водотоків (Stream Order). Для впорядкування водотоків застосовуються дві технології, Шреве і Страхлера.

При використанні інструменту Довжина лінії стоку (Flow Length) можна визначити довжину шляху потоку вниз або вгору за течією з кожної комірки в межах даного вододілу. Це корисно для обчислення часу переміщення води через вододіл.

Розробимо модель для інструменту, який буде для довільно заданого растру ЦМР створювати мережу водотоків і вододілів. Вигляд цієї моделі наведений на рис. 3.4.

Модель послідовно зв'язує всі інструменти гідрологічного аналізу, які були зазначені вище і має лише один вхідний параметр – довільний растр ЦМР.

Інструмент носить назву River System і був протестований на ЦМР Кодимського району (рис. 3.5). Після виконання інструменту була отримані векторні шарі річкової мережі та басейнів водотоків району (рис. 4.621).

Розглянуті моделі були також створені як скрипти на мові програмування Python, лістинг яких приведений у додатку А.

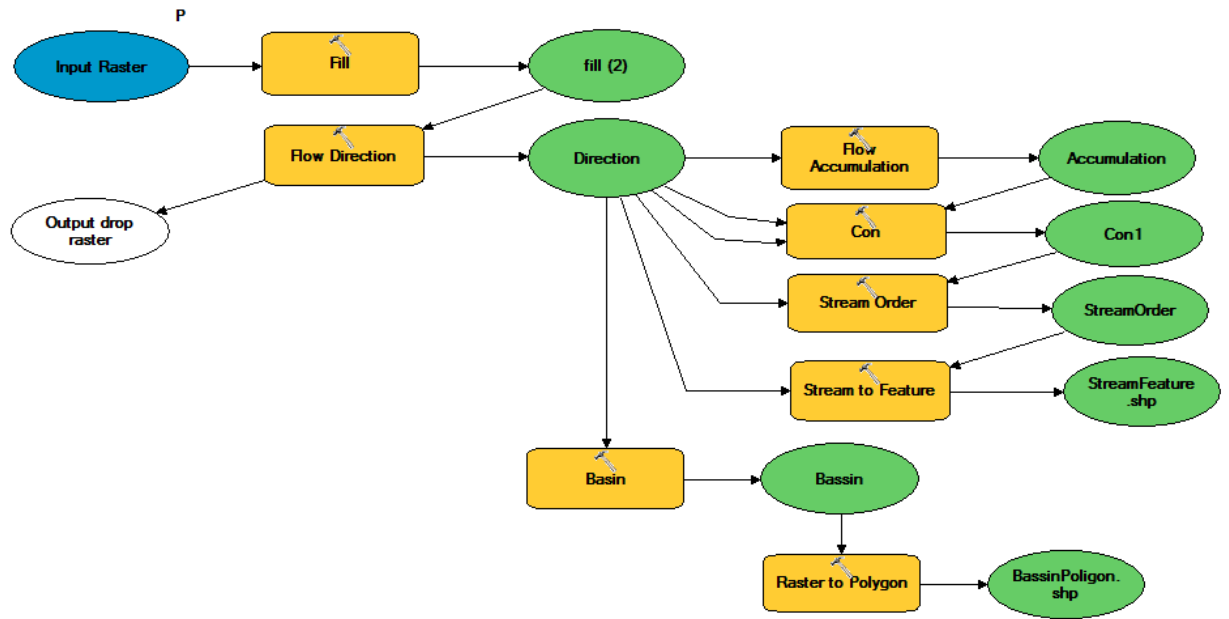


Рисунок 3.4 – Модель інструменту River System

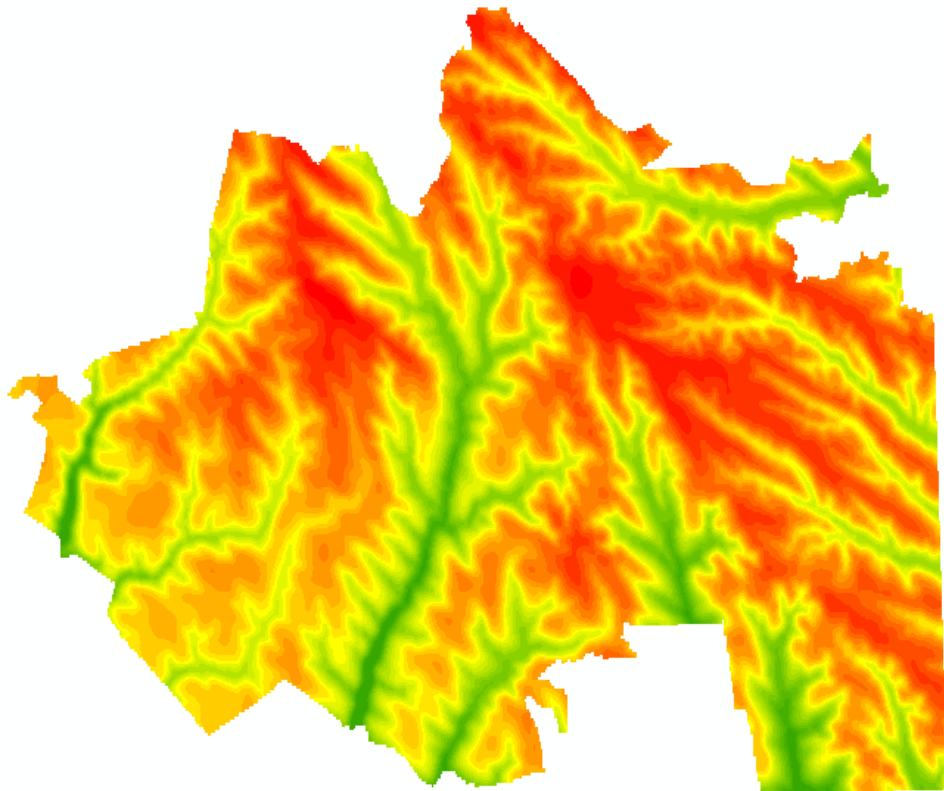


Рисунок 3.5 – ЦМР Кодимського району Одеської області

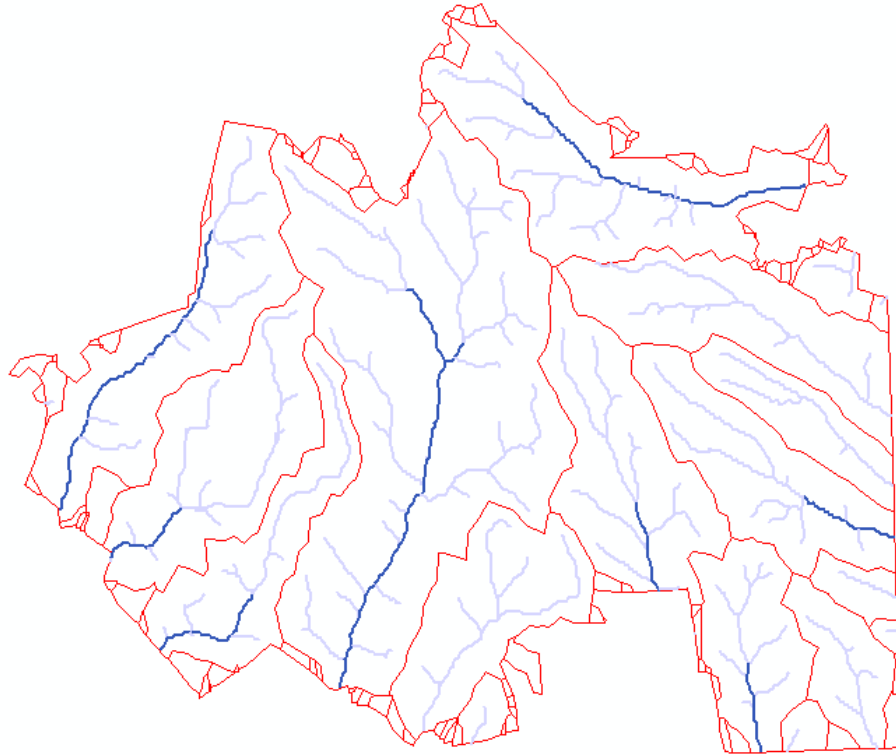


Рисунок 3.6 – Векторні шарі річкової мережі та басейнів водотоків
Кодимського району

3.2 Створення моделі розрахунку зон затоплення

Повінь є одним з найбільш поширених і руйнівних стихійних лих. Геопросторові моделі – це потужний інструмент для оцінки впливу повені та пом'якшення цього впливу до того, як воно станеться. Побудова складних моделі гідрологічних паводків це дуже важке та трудомістке завдання, які виконуються висококваліфікованими фахівцями. Ці моделі необхідні інженерам при створенні систем контролю паводків. Проте, більш прости моделі можуть швидко оцінити прогностичний вплив повені, що насувається, для виконання перших заходів та попередження потенційних жертв.

Щоб зрозуміти вплив повені в області, що досліджується, можна використовувати простий і зрозумілий інструмент, який називається моделлю розрахунку зон затоплення. Ця модель починається з однієї точки і заповнює

область максимальним об'ємом води, який може утримувати басейн на конкретній стадії повені. Зазвичай цей аналіз є найгіршим сценарієм. Сотні інших факторів впливають на підрахунок того, скільки води потрапить в басейн після паводку. Але вже за цією простою моделлю першого порядку можна отримати багато важливої інформації.

На рис.3.7 представлена ЦМР (DEM) з точкою джерела, що відображається у вигляді жовтої зірки. У реальному аналізі ця точка буде показником потоку, де існують дані про рівень води в річці.

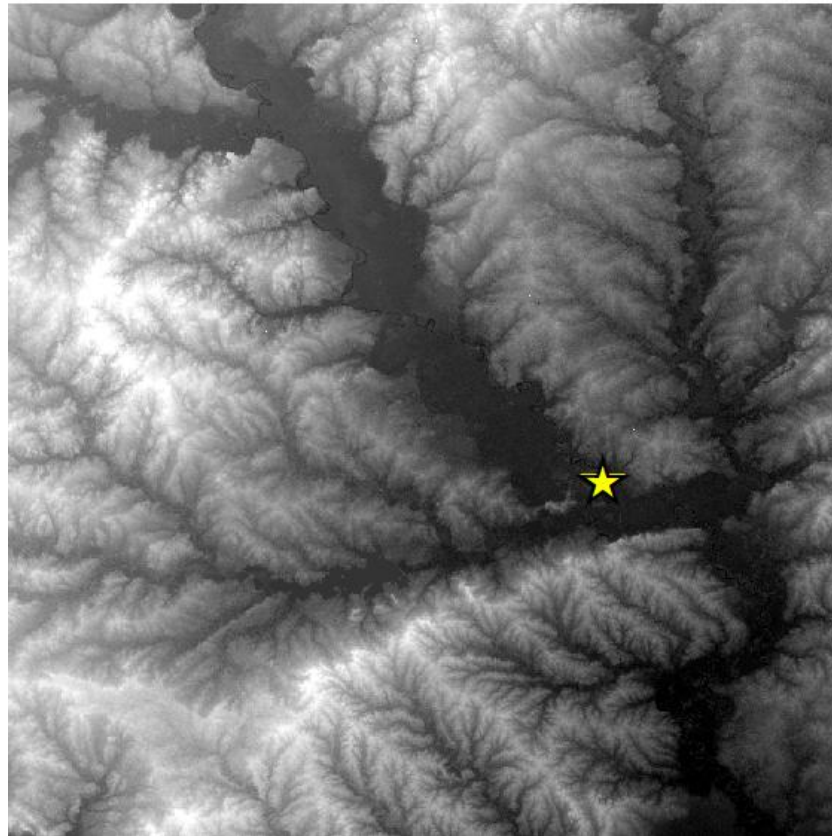


Рисунок 3.7 – Фрагмент цифрової моделі рельєфу з маркером джерела повені

Алгоритм, який будемо використовувати для розрахунку зон затоплення, називається алгоритмом заливки. Цей алгоритм добре відомий в області комп'ютерних наук і використовується в класичній комп'ютерній грі Сапер

для очищення порожніх квадратів на дошці, коли користувач клацає квадрат. Це також метод, який використовується в графічних програмах, таких як Adobe Photoshop, для заливки області суміжних пікселів одного кольору іншим кольором.

Є багато способів реалізувати цей алгоритм. Один з найбільш поширених способів – це рекурсивне сканування кожного пікселя зображення. Проблема з рекурсією полягає в тому, що треба обробляти пікселі більш ніж один раз, що створює непотрібний обсяг роботи. Використання ресурсів для рекурсивної заливки може легко привести до збою програми навіть на зображенні середнього розміру.

В скрипті, що розроблюється, використовується заливка, що заснована на чотирьох спрямованій черзі, яка може проходити комірку більш ніж один раз, але гарантує, його обробку тільки один раз. Черга містить тільки унікальні необроблені комірки, використовуючи вбудований в Python тип `set` (множина), який містить тільки унікальні значення. Будемо використовувати дві множини: `fill`, який складається з комірок, які нам потрібно залити, і `filled`, який містить оброблені комірки.

У скрипті на першому етапі виконуються наступні кроки:

- 1) Витягнути інформацію з заголовка ASCII DEM.
- 2) Відкрити DEM у вигляді `numpy` масиву.
- 3) Визначити початкову точку як рядок і стовпець в масиві.
- 4) Задати поріг значення висоти затоплення.
- 5) Відфільтрувати територію тільки до бажаного значення висоти і нижче.
- 6) Обробити відфільтрований масив.
- 7) Створіть масив `1, 0, 0` (тобто двійковий масив), де «затоплені» пікселі дорівнюють `1`.
- 8) Зберегти масив затоплення як файл ASCII DEM.

Щодо формату файлу ASCII, то він має наступний вигляд:

```
NCOLS xxx
```

```

NROWS xxx
XLLCORNER xxx
YLLCORNER xxx
CELLSIZE xxx
NODATA_VALUE xxx
row 1
row 2
.
.
row n

```

Нижче представлені визначення ключових слів:

NCOLS і **NROWS** – кількість стовпчиків і рядків в растрі, який заданий файлом ASCII.

XLLCORNER і **YLLCORNER** – координата лівого нижнього кута левої нижньої комірки.

CELLSIZE – це розмір комірки растру.

NODATA_VALUE – значення, яке використовується для представлення комірок із знанням NoData.

Нижче наведено приклад файлу ASCII:

```

NCOLS 480
NROWS 450
XLLCORNER 378922
YLLCORNER 4072345
CELLSIZE 30
NODATA_VALUE -32768
43 2 45 7 3 56 2 5 23 65 34 6 32 54 57 34
35 45 65 34 2 6 78 4 2 6 89 3 2 7 45 23 5 ...

```

Для створення файлу ASCII в ArcGIS може бути використана функція бібліотеки ArcPy – `arcpy.RasterToASCII_conversion()`.

3.2.1 Алгоритм функції заливки

Для обробки матриці ASCII будемо використовувати бібліотеку NumPy. Почнемо з визначення функції `floodFill()`, яка є основою моделі розрахунку зон затоплення.

Алгоритм заливки стартує із заданої комірки і починає перевірку сусідніх комірок на схожість (подібність). Фактором подібності є висота. Якщо сусідня комірка має ту же або меншу висоту, що і поточна комірка, то ця комірка позначається для перевірки її сусіда, поки не буде перевірена вся регулярна сітка (`grid`). NumPy не призначений для обходу масиву таким способом, але він як і раніше ефективний для обробки багатовимірних масивів в цілому. При проході через кожен комірку виконується перевірка її сусідів на півночі, півдні, сході і заході. Будь-яка з цих комірок, яка може бути залита, додається у множину `filled`, а її сусіди додаються у множину `fill`, яке перевіряється алгоритмом. Алгоритм, за яким працює функція заливки `floodFill()`, представлено на рис.3.8.

Як згадувалося раніше, якщо виникне спроба додати одне і те ж значення в множину двічі, вона просто ігнорує дублюючу запис і підтримує унікальний список. Використовуючи множини в масиві, перевірка комірки виконується тільки один раз, тому що множина `fill` містить унікальні комірки. Наступний код реалізує функцію заливки:

1) Імпортуємо потрібні бібліотеки:

```
import numpy as np
from linecache import getline
```

2) Створюємо функцію заливки `floodFill()`:

```
def floodFill(c, r, mask):
```

3) Створюємо множини для відстеження комірок, які вже охопили:

```
filled = set()
fill = set()
fill.add((c, r))
width = mask.shape[1]-1
```

```
height = mask.shape[0]-1
```

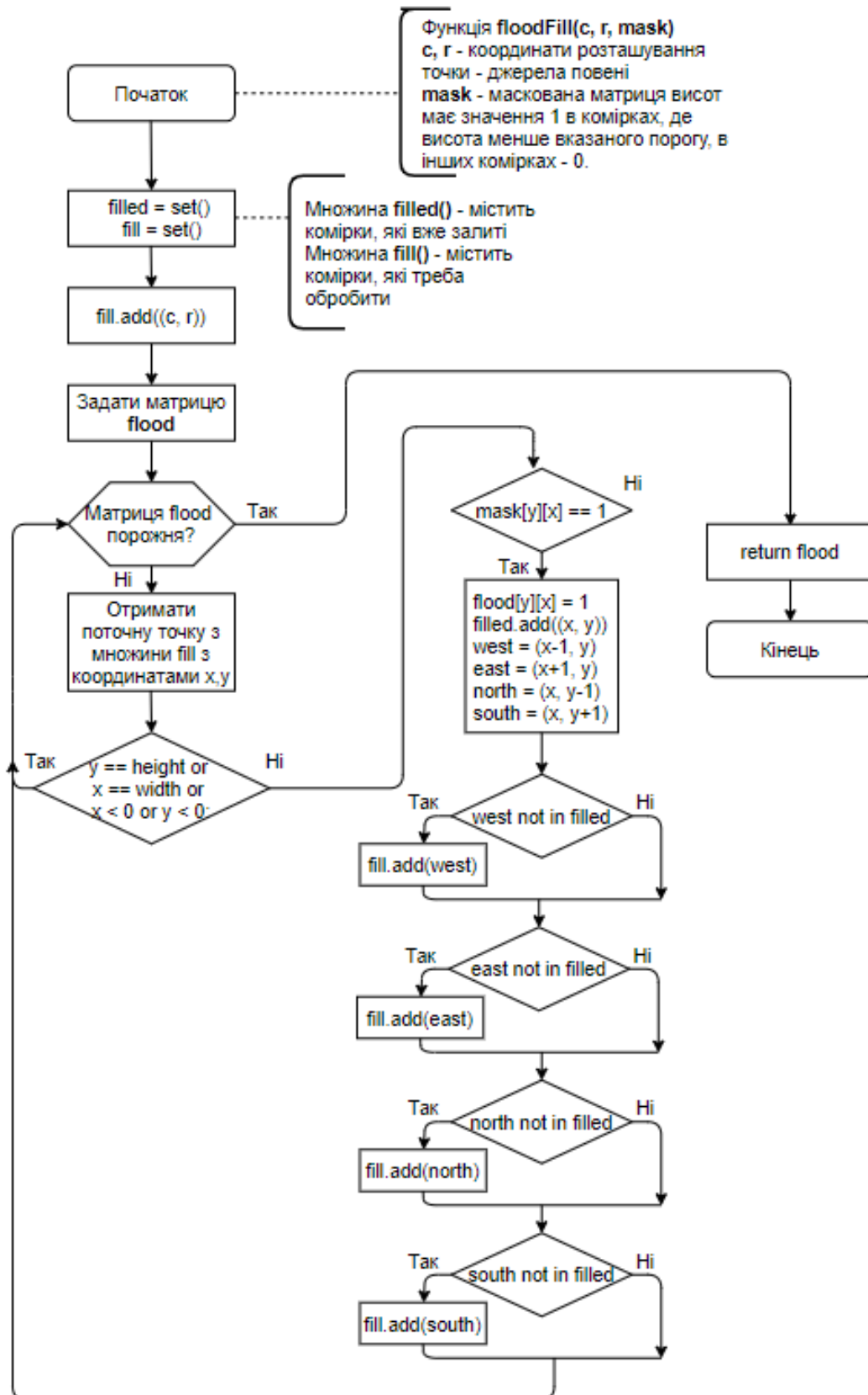


Рисунок 3.8 – Блок-схема алгоритму заливки

4) Створюємо масив затоплення flood:

```
flood = np.zeros_like(mask, dtype=np.int8)
```

5) Проходимо комірки і заливаємо їх, чи ні:

```
while fill:
    x, y = fill.pop()
```

6) Якщо територія вище паводкової води, пропустіть її:

```
if y == height or x == width or x < 0 or y < 0:
    continue
```

7) Якщо висота території дорівнює або менше паводкової води, запов-
ніть її:

```
if mask[y][x] == 1:
    flood[y][x] = 1
    filled.add((x, y))
```

8) Перевіряємо оточуючі сусідні комірки, щоб побачити, чи потрібно
їх заповнювати, коли комірки закінчаться, повертаємо з функції побудовану
матрицю затоплення:

```
west = (x-1, y)
east = (x+1, y)
north = (x, y-1)
south = (x, y+1)
if west not in filled:
    fill.add(west)
if east not in filled:
    fill.add(east)
if north not in filled:
    fill.add(north)
if south not in filled:
    fill.add(south)
return flood
```


3.2.2 Створення скрипту для розрахунку зон затоплення

Далі у кодї скрипта виконується завантаження даних про місцевість із файлу ASCII. Задаємо ім'я вихідного grid файлу та виконуємо алгоритм на отриманих даних. Початком алгоритму заливки є довільна точка (sx, sy) в нижній частині височини. У реальному застосуванні ці пункти, ймовірно, будуть відомим місцем, таким як датчик потоку або порушення дамби. На завершальному кроці зберігаємо вихідну сітку grid.

Необхідно виконати наступні дії:

- 1) Оголосити вихідні (source) та цільові (target) імена даних:

```
source = "terrain.asc"
target = "flood.asc"
```

- 2) Відкрити вхідні дані:

```
print("Opening image...")
img = np.loadtxt(source, skiprows=6)
print("Image opened")
```

- 3) Створити масив маску для всіх висот нижче 70 метрів:

```
wet = np.where(img < 70, 1, 0)
print("Image masked")
```

- 4) Розібрати геопросторову інформацію з заголовка:

```
hdr = [getline(source, i) for i in range(1, 7)]
values = [float(h.split(" ")[-1].strip()) for h in hdr]
cols, rows, lx, ly, cell, nd = values
xres = cell yres = cell * -1
```

- 5) Встановити початкову точку, яка розташована в руслі річки:

```
sx = 2582
sy = 2057
```

- 6) Запустити функцію floodFill:

```
print("Beginning flood fill")
fld = floodFill(sx, sy, wet)
print("Finished flood fill")
header = ""
```

```
for i in range(6):
    header += hdr[i]
```

7) Зберегти результат роботи моделі:

```
print("Saving grid")
with open(target, "wb") as f:
    f.write(bytes(header, 'UTF-8'))
    np.savetxt(f, fld, fmt="%1i")
print("Done!")
```

Після роботи скрипту отримаємо файл ASCII, в якому комірки растру, що будуть затоплені мають значення 1, а інші – 0. Приклад карти зон затоплення наведено на рис. 3.9.

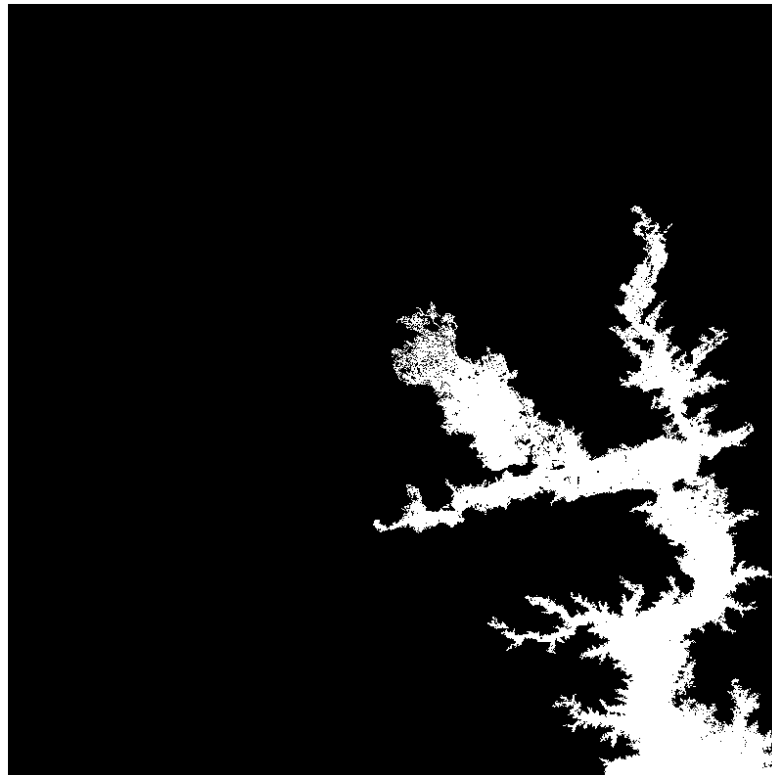


Рисунок 3.9 – Растр, що є результатом роботи скрипту розрахунку зон затоплення при значенні висоти повені 70 м

При суміщенні растрів рис.3.7 та рис.3.9 та класифікації матриці висот різними кольорами (растр затоплення, який охоплює всі області менше 70

метрів, забарвлений в синій колір), отримуємо карту, що наведена на рис. 3.10.

Зображення на рис. 3.11 практично ідентично рис. 3.9, за винятком відфільтрованої маски, з якої була отримана заливка. Це робиться шляхом створення файлу для масиву з ім'ям `wet` замість `fld`, щоб показати несуміжні області, які були включені як частина повені (відмічено зеленим колом). Ці області не пов'язані з вихідною точкою, тому вони навряд чи будуть досягнуті під час повені.

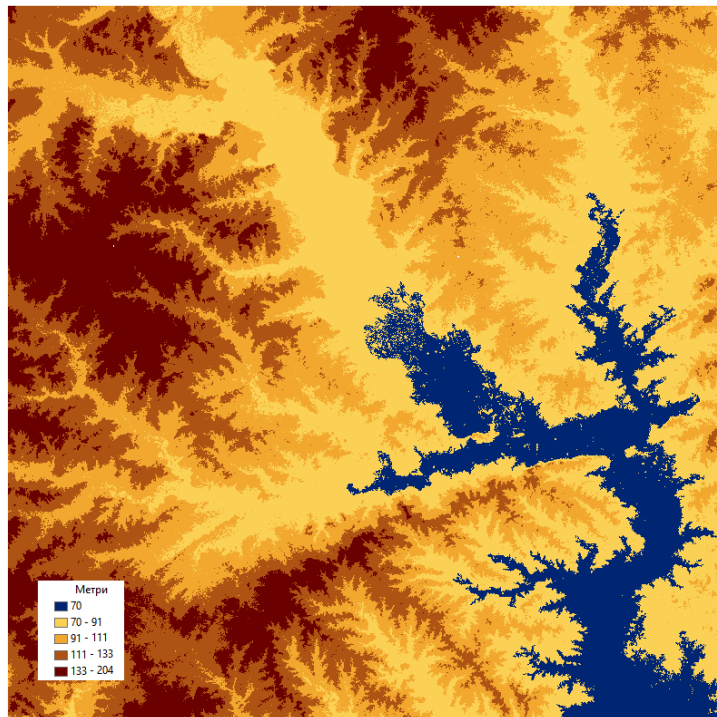


Рисунок 3.10 – Растр зони затоплення для значення висоти повені 70 м, побудований за допомогою функції заливки

Змінюючи значення висоти, можна будувати додаткові растри зон затоплення. На рис. 3.11 показано растр сумісний растр для висот повені 70 та 90 м. Тому є можливість робити більші чи менші кроки та показувати різні впливи як різні шари карти затоплення.

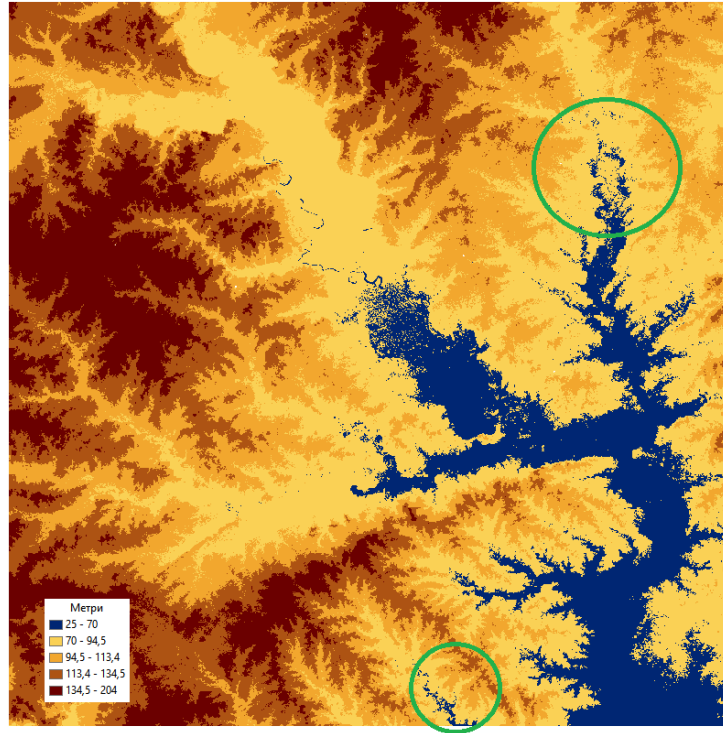


Рисунок 3.10 – Растр зони затоплення для значення висоти повені 70 м, побудований шляхом класифікації висот

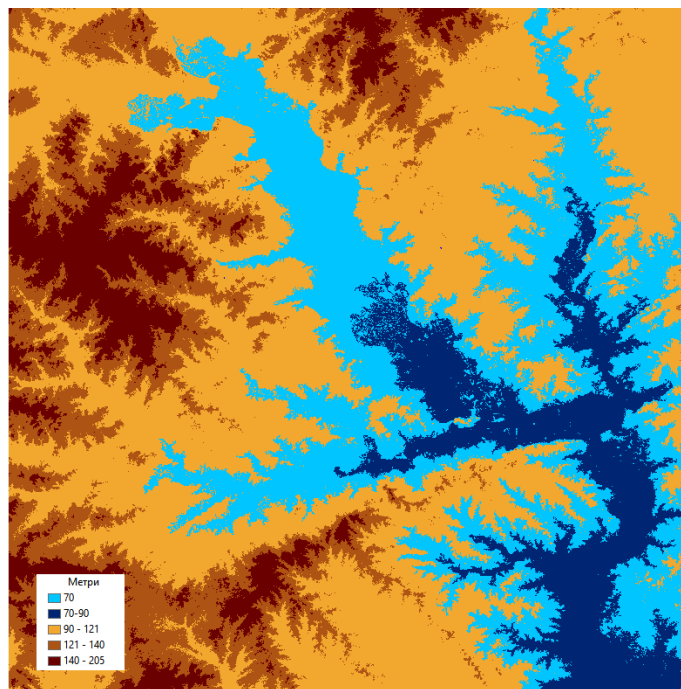


Рисунок 3.11 – Растр зон затоплення для значення висоти повені 70 м та 90 м., побудовані за допомогою функції заливки

Подальшим напрямком розвитку моделі може бути використання методу `polygonize()` GDAL маски потоку для побудови векторного полігону затоплення. Далі можна здійснити вибір будівель, що потребляють у полігон для визначення впливу повені на населення. Тобто на основі отриманої карти зон затоплення при умові доопрацювання моделі може бути побудована система підтримки прийняття рішень по управлінню ризиками повеней. За допомогою подібної системи можуть бути прийняті оперативні рішення по запобіганню великих втрат та жертв повені серед населення.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було проведено аналіз сучасних програмних засобів геообробки просторової інформації. Показано, що найбільш зручним засобом для роботи з векторними та растровими даними є створення власних скриптів Python, які можуть виконуватися автономно чи запускатися з середовища геоінформаційної системи.

В роботі було створено два власних інструменти геообробки:

1) Інструмент гідрологічної геообробки даних, який дозволяє побудувати розвинену річну мережу та басейни річок для території, що розглядається.

2) Інструмент розрахунку зон затоплення за вказаною висотою та координатами точки-джерела повені.

В якості вхідних даних для просторового моделювання були використані файли DEM – цифрові моделі рельєфу формату STRM. Для обробки растрів застосовувалися бібліотеки ArcPy, NumPy, GDAL. Додаткова обробка вихідних даних та карт, отриманих в результаті виконання скриптів, проводилася в пакеті ГІС ESRI ArcGIS 10.7. Скрипт розрахунку зон затоплення є автономним і може бути виконаний в будь-якому іншому ГІС-пакеті, який підтримує Python, наприклад в QGIS. Скрипт для проведення гідрологічного аналізу був створений за допомогою візуального редактору ModelBuilder з використанням бібліотеки ArcPy, та оснований на наборі гідрологічних функцій пакету Spatial Analyst.

Для розрахунку зон затоплення було використано рекурсивний алгоритм заливки, що використовується в графічних редакторах для заливки кольором обмежених областей зображення. І хоча для рішення завдання запропонований алгоритм може розрахувати лише наближені зони затоплення (гірший варіант), подібний інструмент може бути корисним для оцінки можливих ризиків повені.

Таким чином, запропоновані в роботі скрипти для гідрологічного аналізу та розрахунку зон затоплення можуть бути використані як окремі інструменти геообробки в ГІС та будуть корисними для фахівців-гідрологів для забезпечення більш гнучкого та оперативного управління водними ресурсами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Зацерковний В.І., Тішаєв І.В., Віршило І.В., Демидов В.К. Геоінформаційні системи в науках про Землю : монографія. Ніжин : НДУ ім. М. Гоголя, 2016. 510 с.
2. Зацерковний В.І. Геоінформаційні системи і системи дистанційного зондування Землі в задачах ефективного землекористування. Математичне моделювання в економіці 201. 40–48.
3. Павленко Л. А. Геоінформаційні системи: навчальний посібник / Л. А. Павленко. – Х. : Вид. ХНЕУ, 2013. – 260 с.
4. Світличний О.О., Плотницький С.В. Основи геоінформатики: Навчальний посібник. Суми: ВТД «Університетська книга», 2006. 295 с.
5. Свідзінська Д.В. Методи геоекологічних досліджень: геоінформаційний практикум на основі відкритої ГІС SAGA: навчальний посібник. К.: Логос, 2014. 402 с.
6. Шипулін В.Д. Основи ГІС-аналізу: навч. посібник. Харк. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. Х.:ХНУМГ, 2014. 330 с.
7. Офіційний сайт компанії ESRI Inc. URL: www.esri.com (дата звернення 21.05.2023)
8. Сайт справочної системи ArcGIS. URL: webhelp.esri.com/arcgisdesktop (дата звернення 21.05.2023)
9. Westra E. Python Geospatial Development. URL: <https://theswissbay.ch/pdf/Gentoomen%20Library/Programming/Python/Python%20Geospatial%20Development%20%282010%29.pdf> (дата звернення: 14.05.2023)
10. MapInfo Professional 9.0. URL: <http://base.dnsgb.com.ua/files/book/MapInfo-9.0.pdf> (дата звернення 21.05.2023)

11. Определение Open Source. URL: https://ru.wikipedia.org/wiki/Определение_Open_Source (дата звернення 21.05.2023)
12. GIS Algorithms and Programming. URL: <https://geography.osu.edu/courses/geog-5222> (дата звернення 21.05.2023)
13. David W. Allen Getting to Know ArcGIS ModelBuilder URL: <http://esripress.esri.com/bookresources/modelbuilder.pdf> (дата звернення 21.05.2023)
14. Офіційний сайт Python. URL: <http://www.python.org> (дата звернення 21.05.2023)
15. Tateosian L. Python for ArcGIS. Springer International Publishing Switzerland, 2015. 544 p.

ДОДАТОК А
ЛІСТИНГ СКРИПТІВ ІНСТРУМЕНТІВ ГЕООБРОБКИ
ПРОСТОРОВОЇ ІНФОРМАЦІЇ

```
-----  
# River System.py  
# Created on: 2023-03-21 18:14:42.00000  
# Usage: River System <Input_Raster>  
# Description:  
# -----  
# Import arcpy module  
import arcpy  
# Check out any necessary licenses  
arcpy.CheckOutExtension("spatial")  
# Script arguments  
Input_Raster = arcpy.GetParameterAsText(0)  
if Input_Raster == '#' or not Input_Raster:  
    Input_Raster = "C:\\Data" # provide a default value if  
unspecified  
# Local variables:  
fill__2_ = Input_Raster  
Direction = fill__2_  
Accumulation = Direction
```

```

Con1 = Accumulation

StreamOrder = Con1

StreamFeature_shp = StreamOrder

Bassin = Direction

BassinPoligon_shp = Bassin

Output_drop_raster = fill__2_

# Process: Fill
arcpy.gp.Fill_sa(Input_Raster, fill__2_, "")

# Process: Flow Direction
arcpy.gp.FlowDirection_sa(fill__2_, Direction, "NORMAL",
Output_drop_raster)

# Process: Flow Accumulation
arcpy.gp.FlowAccumulation_sa(Direction, Accumulation, "",
"FLOAT")

# Process: Con
arcpy.gp.Con_sa(Accumulation, Direction, Con1, Direction, "")

# Process: Stream Order
arcpy.gp.StreamOrder_sa(Con1, Direction, StreamOrder,
"STRAHLER")

# Process: Stream to Feature
arcpy.gp.StreamToFeature_sa(StreamOrder, Direction,
StreamFeature_shp, "SIMPLIFY")

# Process: Basin
arcpy.gp.Basin_sa(Direction, Bassin)

# Process: Raster to Polygon
arcpy.RasterToPolygon_conversion(Bassin, BassinPoligon_shp,
"SIMPLIFY", "VALUE")

```

```
import numpy as np

from linecache import getline

def floodFill(c, r, mask):

    # cells already filled

    filled = set()

    # cells to fill

    fill = set()

    fill.add((c, r))

    width = mask.shape[1]-1

    height = mask.shape[0]-1

    # Our output inundation array

    flood = np.zeros_like(mask, dtype=np.int8)

    # Loop through and modify the cells which

    # need to be checked.

    while fill:

        # Grab a cell

        x, y = fill.pop()

        if y == height or x == width or x < 0 or y < 0:

            # Don't fill

            continue

        if mask[y][x] == 1:

            # Do fill

            flood[y][x] = 1

            filled.add((x, y))
```

```
# Check neighbors for 1 values

west = (x-1, y)

east = (x+1, y)

north = (x, y-1)

south = (x, y+1)

if west not in filled:

    fill.add(west)

if east not in filled:

    fill.add(east)

if north not in filled:

    fill.add(north)

if south not in filled:

    fill.add(south)

return flood

source = "sarata11.asc"

target = "flood.asc"

print("Opening image...")

img = np.loadtxt(source, skiprows=6)

print("Image opened")

a = np.where(img < 50, 1, 0)

print("Image masked")

# Parse the headr using a loop and

# the built-in linecache module

hdr = [getline(source, i) for i in range(1, 7)]

values = [float(h.split(" ")[-1].strip().replace(',','.')) for

h in hdr]
```

```
cols, rows, lx, ly, cell, nd = values
xres = cell
yres = cell * -1

# Starting point for the
# flood inundation
sx = 2582
sy = 2057
print("Beginning flood fill")
fld = floodFill(sx, sy, a)
print("Finished Flood fill")
header = ""
for i in range(6):
    header += hdr[i]
print("Saving grid")
# Open the output file, add the hdr, save the array
with open(target, "wb") as f:
    f.write(bytes(header, 'UTF-8'))
    np.savetxt(f, fld, fmt="%1i")
print("Done!")
```