

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ  
Факультет інформаційних технологій

УДК 004.9:631.2  
«ПОГОДЖЕНО» «ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»  
Декан факультету Завідувач кафедри комп'ютерних наук

інформаційних технологій  
Глазунова О.Г., д.п.н., професор

Голуб Б.Л., к.т.н., доцент

2021 р. 2021 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА  
на тему: Програмне забезпечення системи моніторингу утримання  
сільськогосподарських об'єктів

Спеціальність: 121 Інженерія програмного забезпечення  
(код і назва)

Освітня програма Програмне забезпечення інформаційних систем

Орієнтація освітньої програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

доц., к.т.н.  
(науковий ступінь та вчене звання)

(підпис)

Голуб Б.Л.  
(ПБ)

Керівник магістерської кваліфікаційної роботи

доц., к.т.н.  
(науковий ступінь та вчене звання)

(підпис)

Басараб Р.М.  
(ПБ)

Виконав

Власенко О.В.  
(ПБ студента)

КИЇВ - 2021

# НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

ЗАТВЕРДЖУЮ

Завідувач кафедри \_\_\_\_\_

науковий ступінь, вчене звання) (підпис) (ПІБ)  
" " 20 року

## ЗАВДАННЯ

### ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Власенко Олександр Вячеславович  
(прізвище, ім'я, по-батькові)  
Спеціальність 121 «Інженерія програмного забезпечення»  
(код і назва)

Освітня програма Програмне забезпечення інформаційних систем  
(назва)

Орієнтація освітньої програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Тема магістерської кваліфікаційної роботи «Програмне забезпечення системи моніторингу утримання сільськогосподарських об'єктів» затверджена наказом ректора НУБіП України від «29» жовтня 2020р. №1636 «С»

Термін подання завершеної роботи на кафедру 2021 листопада 30  
рік місяць число

Вихідні дані до магістерської кваліфікаційної роботи: дані було отримано з загальнодоступних ресурсів.

Перелік питань, що підлягають дослідженню:  
1. Системний аналіз предметної області  
2. Моделювання системи  
3. Розробка системи

Дата видачі завдання «12» листопада 2020 р.

Керівник магістерської кваліфікаційної роботи Басараб Р.М.  
(підпис) (прізвище та ініціали)  
Завдання прийняв до виконання Власенко О.В.  
(підпис) (прізвище та ініціали студента)

ВСТУП	4
1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	6
1.1 Поняття дистанційного зондування землі	6
1.2 Електромагнітне випромінювання	9
1.3 Загальна схема ДЗЗ	13
1.4 Системи ДЗЗ: аерокосмічне та наземне	15
1.5 Ресурси для отримання інформації ДЗЗ	18
1.6 Обробка даних ДЗЗ	22
1.7 Проблема обробки великої кількості даних ДЗЗ	33
2 МОДЕЛЮВАННЯ СИСТЕМИ	38
2.1 Загально про UML	38
2.2 Діаграма прецедентів	43
2.3 Діаграма послідовності	45
3 РОЗРОБКА СИСТЕМИ	46
3.1 Технічне завдання	46
3.2 Використані технології	47
3.3 Алгоритм роботи	59
3.4 Тестування ПЗ	60
4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ	63
4.1 Реалізація програмного коду	63
4.2 Результат використання багатопотокового режиму в обробці супутникового зображення	67
4.3 GIL - Global Interpreter Lock	67
ВИСНОВКИ	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	71

# ВСТУП

# НУБІП України

Актуальність дослідження.

Способи дистанційного вивчення навколишнього середовища – це досить широка сфера науки, яка переживає бурний розвиток. Новітні способи дослідження об'єктів на відстані, на самперед за допомогою космічних апаратів або дронів, зробили великий крок в наданні інформації для досліджень та майбутніх розробок в науках дослідження Земної поверхні.

В даний період часу значну частину даних дистанційного зондування отримується за допомогою штучних супутників Землі. Їхня спроможність до масштабного огляду земної поверхні з космосу, можливість реєструвати сигнали в декількох спектральних діапазонах і швидкість переміщення дають змогу отримувати величезний об'єм даних.

Типовою рисою способів ДЗЗ є охоплення величезної території. За допомогою поєднання наземних досліджень та інформації дистанційного зондування ми спроможні скоротити час виконання робіт і знизити їх вартість за рахунок виділення ключових ділянок.

Обробка даних дистанційного зондування і їхня інтерпретація цілком пов'язана з обробкою цифрових зображень, так як аерокосмічні знімки надаються у вигляді растрових зображень.

Дані дистанційного зондування Землі можуть бути використані не тільки в наукових але і у виробничих цілях, так як стали доступними будь-якому користувачеві. ДЗЗ є одним з основних джерел актуальних і оперативних даних для геоінформаційних систем (ГІС). Основні області застосування даних ДЗЗ з космосу – вивчення стану навколишнього середовища, землекористування, вивчення рослинних угруповань, оцінка врожаю сільськогосподарських культур, оцінка наслідків стихійних лих і тому подібні.

Об'єкт дослідження - супутникові зображення поверхні Землі.

Предмет дослідження - методи, алгоритми та процеси обробки супутникових зображень поверхні Землі.

Мета дослідження – перевірка доцільності обробки супутникових знімків у багатопотоковому режимі.

Зміст поставлених завдань:

1. Проаналізувати предметну область
2. Сформулювати вимоги до програмного забезпечення
3. Сформулювати отримані результати і висновки ?

Методи дослідження – для проведення аналізу даних та підтвердження гіпотез з приводу обробки даних дистанційного зондування Землі у багатопотоковому режимі було використано мову програмування Python та ряд додаткових бібліотек.

Апробація результатів дослідження:

Власенко О.В.: Програмне забезпечення системи моніторингу утримання сільськогосподарських об'єктів. IV ВСЕУКРАЇНСЬКА НАУКОВО-ПРАКТИЧНА ІНТЕРНЕТ-КОНФЕРЕНЦІЯ СТУДЕНТІВ ТА АСПІРАНТІВ “ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ”

Режим доступу: <https://www.goob.su/5MC5>

Структура магістерської роботи. Робота складається зі вступу, основної частини, до якої входять 3 розділи, що включають – підрозділів, загальні висновки, список використаних джерел та додатки. Загальний обсяг роботи – сторінки, список використаних джерел становить – найменувань і займає – сторінок.

## 1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Поняття дистанційного зондування землі

Дистанційне зондування Землі – це засіб отримати інформацію про об'єкт з відстані без прямого контакту з об'єктом дослідження. Методи неконтактного одержання інформації можна віднести до способів ДЗЗ, наприклад сейморозвідка, гравірознавдя і тому подібні.

«Під дистанційним зондуванням Землі (Remote Sensing) розуміється спостереження і вимір енергетичних і поляризаційних характеристик випромінювання об'єктів у різних діапазонах електромагнітного (ЕМ) спектра з метою визначення місцеположення, виду, властивостей і тимчасової мінливості об'єктів навколишнього середовища без безпосереднього контакту з ним вимірювального приладу».[28]

Для вивчення, визначення стану або картографування поверхні використовують способи одержання інформації, тобто зображення земної поверхні в певних ділянках електромагнітного спектру з космічних або авіаційних апаратів (дронів). Тобто, ДЗЗ – це отримані за допомогою космічних або авіаційних апаратів фотознімки земної поверхні.

ДЗЗ використовується у багатьох сферах, не тільки у дослідженнях Землі. Наприклад, у військовій сфері дані ДЗЗ застосовують для розвідки. У невійськовій сфері більшість сфер застосування відноситься до категорії дослідження навколишнього середовища:

- **Атмосфера:** опади, температура, розподіл і тип хмар, концентрації газів і т. д.
- **Земна поверхня:** температура, топографія, вологість ґрунту, антропогенні навантаження, тип і стан рослинності.
- **Океан:** температура, топографія, колір водної поверхні (планктону) і т. д.

• **Кріосфера:** розподіл, стан і динамічні зрушення снігу, айсбергів, морського льоду, льодовиків.

Збір інформації за допомогою способу наземного дослідження є одним з давніших способів. Але він вимагає значних витрат як економічних так і часових, в залежності від досліджуваної території. Навіть якщо ми підемо на значні витрати, це не гарантує нам синхронності при наземних дослідженнях, тобто, одночасного нагляду за всіма ділянками.

Способи ДЗЗ позбавлені таких недоліків.

Накопичення даних про чимали області земної поверхні або атмосфери за малий проміжок часу, одержуючи майже моментальний фотознімок є однією з важливіших характеристик ДЗЗ. Менше ніж за півгодини формується фотознімок

майже чверті земної поверхні метеорологічним супутником Meteosat. Але справжню мить ДЗЗ з аспектом швидкості одержання даних також набувають в ситуаціях коли наземне дослідження є повільним, дорогим, політично незручним.

Потребу в космічних знімках в сучасних умовах обумовлена наступними характеристиками:

• **Об'єктивність** – кожен знімок з космічного апарату являє собою документ, який відтворює стан місцевості на момент зйомки. Так як зйомкою займаються різні компанії-оператори то підробити знімок це безглузда ідея, бо зміну даних буде легко виявити.

• **Актуальність** – отримати дані ДЗЗ можливо на будь-яку дату.  
• **Масштабність** – можливість зйомки великої за площею території з високою деталізацією.

• **Екстериторіальність** – для здійснення зйомки не потрібен дозвіл, так як ділянки зйомки не прив'язані до державних і територіальних кордонів.

• **Доступність** – є відкритими дані ДЗЗ з просторовим розширенням 2м і нижче.

У великій мірі сучасна служба сповіщення погодних умов заснована на спостереженнях з космічних апаратів. Слід виділити, що чим більше територія держави, тим ефективніше застосування дистанційних методів.

Ця діяльність здійснюється на основі поваги принципу повного і постійного суверенітету всіх держав і народів над своїми багатствами і природними ресурсами, з належним урахуванням визнаних за міжнародним правом інтересів інших держав і організацій, що перебувають під їх юрисдикцією. Подібна діяльність повинна здійснюватися таким чином, щоб не завдавати шкоди законним правам та інтересам

держави яку зондуєть.

Міжнародному співробітництву в сфері дистанційного зондування Землі, охоплюючи технічну допомогу і координацію сприяють ООН та відповідні установи ООН у сфері дистанційного зондування.

Відкритістю даних дистанційного зондування Землі регулюється політикою «відкритого неба». CEOS (Committee on Earth Observation Satellites) – це основний міжнародний орган координації політики в області дистанційного зондування Землі.

НУБІП України

НУБІП України

НУБІП України



## 1.2 Електромагнітне випромінювання

Світло - це електромагнітна радіація (EMR) у певному діапазоні електромагнітного спектра (рис. 1.1). Якщо зобразити світло в схематичному вигляді - то воно виглядатиме як графік електричного і магнітного поля, які коливаються перпендикулярно одне одному. Електромагнітний спектр - це класифікація енергії за довжиною хвилі (від космічної короткохвильової до радіо довгохвильової), що рухається зі швидкістю світла.

### ЕЛЕКТРОМАГНІТНІ ХВИЛІ

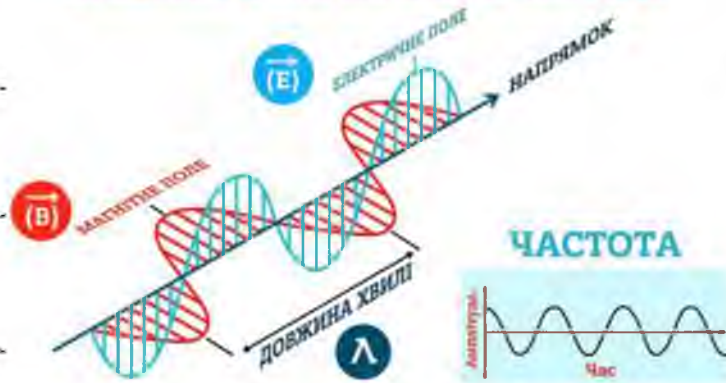


Рис. 1.1 Зображення електромагнітних хвиль

Біле світло складається з простих кольорових променів, з яких найбільше заломлення має фіолетовий, а найменше — червоний. Кожен колір спектра є монохроматичним, тобто однокольоровим, і кожному відповідає своя довжина і частота хвилі. Саме вони створюють електромагнітний спектр.

Довжина хвилі — відстань між двома сусідніми гребнями, або западинами, хвиль. За своєю суттю це відстань, яку проходить хвиля за проміжок часу, який дорівнює одному коливанию. Довжини хвиль дуже різноманітні (від нанометрів до десятків кілометрів).

Частота хвилі — це кількість повних коливань, які здійснює хвиля за 1 секунду. Тобто, чим коротші хвилі, тим більша в них частота.

На рис. 1.1 зображено електромагнітні хвилі.

Саме через зміну своїх довжин і частот хвилі набувають специфічних властивостей та діляться на гамма-, бета-, альфа-, рентгенівське випромінювання, ультрафіолетове, видимі промені, інфрачервоне та радіохвилі (рис. 1.2).



Рис. 1.2 Розподіл електромагнітних хвиль

Для отримання даних ДЗЗ використовують такі діапазони електромагнітного випромінювання, як ультрафіолетовий, видимий, інфрачервоний, мікрохвильовий і радіодіапазон.

На рис. 1.3 зображено довжини електромагнітних хвиль.

НУБІП України





Тепловий діапазон (2,5 мкм — 1 мм) надає інформацію про теплове поле. Було встановлено, що в різних типах рослинності, в насадженнях різної щільності, складу й віку порід, у приземному шарі, на рівні поверхні і в ґрунті перепад температур досягає кількох градусів. Тобто застосування теплового знімання надає додаткову інформацію про ліси, умови місць їх зростання тощо.

Теплова зйомка може використовуватися для виявлення хворих, ушкоджених і сухостійних дерев, оскільки за температурною яскравістю вони різко відрізняються від здорової рослинності.

Мікрохвильовий, СВЧ (1 мм — 1 м) — діапазон дає інформацію про топографічні характеристики територій та акваторій, запаси вологи в ґрунті й листках рослин, про впливи на рослини промислових викидів. Радіодіапазон (1 м — > 10 км) надає інформацію про підстиляючу поверхню. Для радарного знімання характерні глибокі тіні, які використовуються для виявлення об'єктів зі значними перепадами висот.

Радіодіапазон надає можливість аналізувати рельєф території, виявляти небезпечні природні процеси, такі як селі, зсуви тощо. Радіолокаційне знімання може проводитися за будь-яких погодних умов і в будь-який час доби.

Проходячи через нечіткий бар'єр між вакуумом та атмосферою, сонячне випромінювання частково розсіюється атмосферними газами й аерозольними домішками, частково поглинається і переходить у теплоту, що нагріває атмосферу.

Нерозсіяна і непоглинута в атмосфері пряма сонячна радіація досягає земної поверхні, частково відбивається нею, а більша частина поглинається і нагріває її.

«Сонячне випромінювання з найбільшою дозволеною швидкістю долає мільйони кілометрів, щоб дістатися Землі. Але потрапляючи в газовий щит Землі — атмосферу, велика кількість радіації, серед якої багато шкідливої, поглинається. Водяна пара, кисень і озон, вуглекислий газ знижують передачу у певних довжинах хвиль, створюючи смуги поглинання в атмосфері. Частини

електромагнітного спектра, для яких атмосфера прозора, називаються атмосферними вікнами. Наприклад, одне з них знаходиться у діапазоні 300–750 nm і пропускає ультрафіолет та видиме світло [59]».

### Поглинання

Більша кількість шкідливої іонізуючої радіації поглинається атмосферою, створюючи для людини комфортні умови життя на планеті. Приблизно 70% усього поглинання сонячної радіації в атмосфері відбувається завдяки воді.

### Розсіювання

Розсіювання світла – процес перенаправлення сонячної енергії через взаємодію з молекулами та частками, що знаходяться у атмосфері. Виділяють кілька типів розсіювання – це залежить від відносного розміру частинок, що беруть у ньому участь, у порівнянні з довжиною сонячної радіації.

Частинк електромагнітного спектра, яку бачить людина, називають видиме світло (400–700 nm). У лабораторних умовах людина може бачити в інфрачервоному діапазоні до 1050 nm, а молоді люди та діти сприймають довжини хвилі в ультрафіолеті приблизно до 310–313 nm.

## 1.3 Загальна схема ДЗЗ

Наш світ одночасно переживає масові цифрові перетворення. Автоматизовується та вимірюємо все, що рухається та змінюється, за допомогою супутників і безпілотників. З технічним розвитком засобів зйомки, збільшення періодичності, оперативності та доступності матеріалів дистанційного зондування Землі дають змогу вирішувати завдання, котрі раніше можливо було виконати на дорогому професійному обладнанні.

Завдяки поєднанню наземного дослідження з інформацією дистанційного зондування, використанню сучасної обчислювальної техніки і різноманітних

математичних моделей можна отримати комплексне вивчення проблеми за допомогою дистанційного дослідження.

В залежності від того, які інформаційні потоки (отримують з різних за фізичним змістом знімальних систем) використовуємо, систему ДЗЗ можна

поділити на:

- знімальні системи повітряного базування;
- знімальні системи космічного базування;
- лідарні та лазерні системи повітряного базування;
- радарні та радіолокаційні системи космічного базування.

Система дистанційного зондування Землі, котра проводить вимірювання природного випромінювання, працює за схемою пасивного ДЗЗ. Тобто ця система

проводить зйомку лише в часи коли природне випромінювання доступне:

- вдень - у видимому діапазоні;
- вдень і вночі - в тепловому інфрачервоному та мікрохвильовому

Існує друга схема - активне ДЗЗ. Прилад, що розміщений на супутнику, має джерело випромінювання, яке направлено на об'єкт дослідження, після чого відбите

випромінювання реєструється супутником.

Переваги:

- зйомка не залежить від часу доби,
- можливий контроль над потоком випромінювання, який йде від

супутника;

можна використовувати довжини хвиль, котрі занадто слабо представлені в спектрах природних випромінювачів.

«Прикладом активного ДЗЗ є лідар (англ. Light Identification Detection and Ranging (LIDAR) - «виявлення, ідентифікація і визначення дальності за допомогою світла»). Ця технологія використовується для отримання та обробки інформації про віддалені об'єкти і працює через явища поглинання і розсіювання



світла в оптично прозорих середовищах. Цей метод зйомки вимірює відстань до цілі (об'єкт, який ми досліджуємо) шляхом підсвічування цілі імпульсним лазерним світлом і вимірюванням відображених імпульсів датчиком. Різниця часу повернення сигналу лазера і довжини хвиль можуть бути використані для створення цифрових 3D-моделей цілі [57]».

«Лідар зазвичай використовується для створення карт високої роздільної здатності. Дана технологія застосовується в різних галузях, наприклад в геодезії і картографії, археології, географії, геології, лісовому господарстві, метеорології тощо, а також для керування та навігації деяких автономних автомобілів [57]».

На Рис. 1.4 Зображено пасивне та активне ДЗЗ.

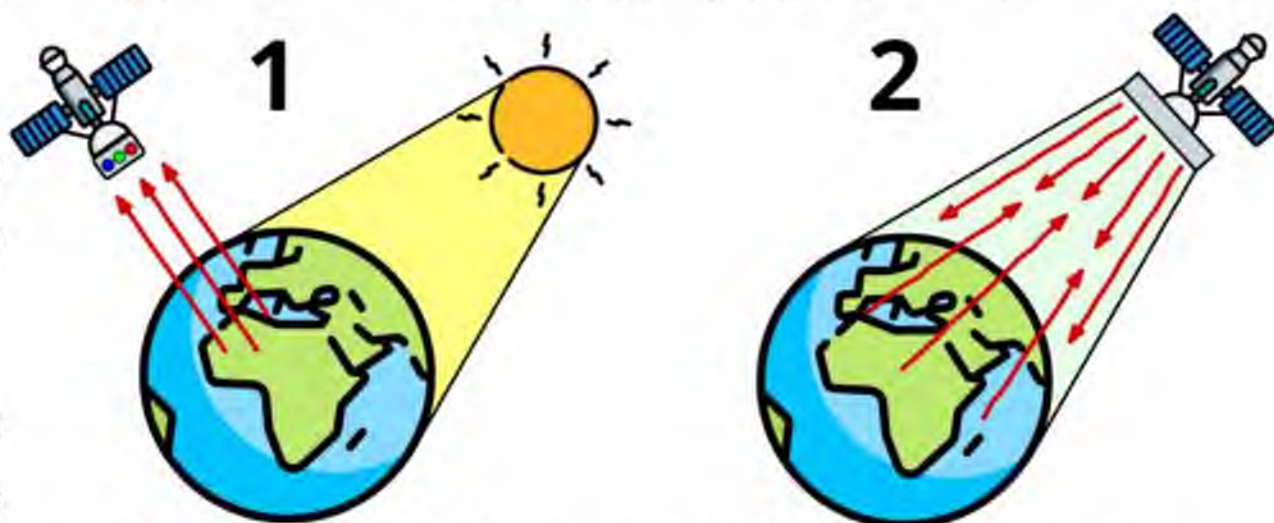


Рис. 1.4 Пасивне та активне ДЗЗ

#### 1.4 Системи ДЗЗ: аерокосмічне та наземне

ДЗЗ поділяється на наземне та аерокосмічне. Аерокосмічне ДЗЗ (англ. remote sensing) — здебільшого мається на увазі, коли носій базується в космічному просторі. Ефективність системи ДЗЗ великою мірою залежить від ефективності наземної інфраструктури керування супутником, прийому, обробки

та розповсюдження даних. Супутникова інформація для потреб користувачів приймається на розподілену мережу приймальних пунктів (рис. 1.5).

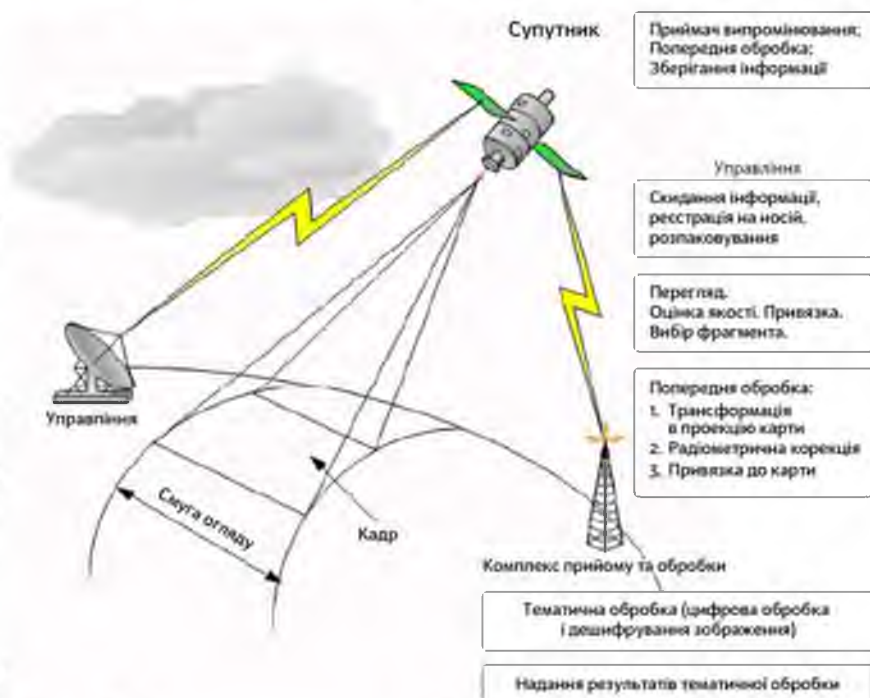


Рис. 1.5 Схема роботи ДЗЗ

### Авіаційні носії

Для здійснення дистанційного зондування Землі, апаратура для зйомки встановлюється на носій, який піднімається на потрібну висоту, переміщується відносно земної поверхні і забезпечує орієнтування в просторі.

Носії, що застосовуються для систем дистанційного зондування Землі поділяються на дві основні групи (рис. 1.6):

- авіаційні – літаки, безпілотники та інші.
- космічні – штучні супутники Землі, орбітальні станції та інші.



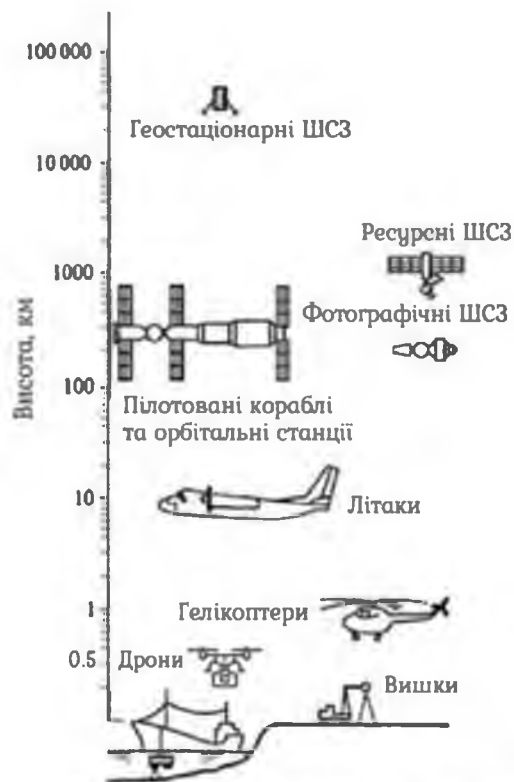


Рис. 1.6 Види носіїв

Для аерознімань використовують або спеціально пристосовані літаки, або гелікоптери. Залежно від напрямку оптичної осі знімального апарата розрізняють планове і перспективне аерознімання. При плановому (вертикальному) аерозніманні вісь знімального апарата приводить у стрімке положення, при якому знімок горизонтальний. Але через коливання літака в процесі польоту апарат нахилється. У порівнянні з плановим, перспективний знімок охоплює велику площу, а зображення утворюється в більш звичному для людини ракурсі. Технічні засоби постійно вдосконалюються, особливо це стосується такого параметра, як точність визначення кутів нахилу камери.

«Дрони — це мобільний інструмент з високою деталізацією даних. Оскільки висота польоту дрона зазвичай знаходиться в рамках від 100 до 300 метрів над поверхнею землі, то можна стримати знімки з роздільною здатністю в сантиметрах на піксель. Дрони дозволяють збирати величезну кількість інформації

в найкоротші терміни. Особливістю дронів є можливість використання спектральних камер, які дозволяють отримувати фотографії в ближньому інфрачервоному спектрі. На підставі таких знімків відбувається розрахунок NDVI-індексів [4]».

### Космічні носії

Космічний моніторинг впевнено увійшов у повсякденне життя суспільства. Вчені підраховали, що більше 18 000 рукотворних об'єктів потрапили на земну орбіту на 2018 рік. Всі вони знімають поверхню Землі та подають інформацію для подальшої обробки і вирішення конкретних завдань.

Технічний прогрес в галузі космічного знімання пов'язаний з:

- підвищенням просторової роздільної здатності знімальних систем;
- розширенням спектрального діапазону зареєстрованих зображень

та отриманням багатоспектральних знімків.

Класифікація космічних апаратів є завданням досить трудомістким, оскільки кожен апарат унікальний, а коло завдань, що розв'язують космічні апарати, постійно розширюється. Найбільш необхідними на сьогоднішній день є супутники зв'язку, навігаційні, ДЗЗ й наукові. Супутники військового призначення й супутники розвідники становлять окремий клас.

Щорічно запускаються в експлуатацію 20–25 нових систем. Піонером в напрямі створення систем надвисокої роздільної здатності є США, передусім фірми GeoEye та DigitalGlobe. Практично всі країни світу є учасниками спільних космічних міжнародних проєктів. США, Росія, Франція, Україна, Індія, Китай, Ізраїль, Японія, Німеччина та Італія взагалі мають власні системи космічного базування.

## 1.5 Ресурси для отримання інформації ДЗЗ

### EarthExplorer

Серед своїх конкурентів USGS (Служба геологічної зйомки США) володіє найстарішою колекцією безкоштовних різноманітних ГІС-даних. Доступ до них можна отримати за допомогою сервісу EarthExplorer (<https://earthexplorer.usgs.gov>), який працює через Google Maps (Рис. 1.7).

У сервісі є набори даних з відкритим визідним кодом, які були отримані в результаті співпраці з Індійською організацією космічних досліджень (ISRO) і Європейським космічним агентством (ESA). Це дані з супутників Resourcesat-1, 2 і Sentinel-2.

Також є багато даних з комерційних супутників з високою роздільною здатністю, таких як IKONOS-2, OrbView-3 і історичних даних SPOT.

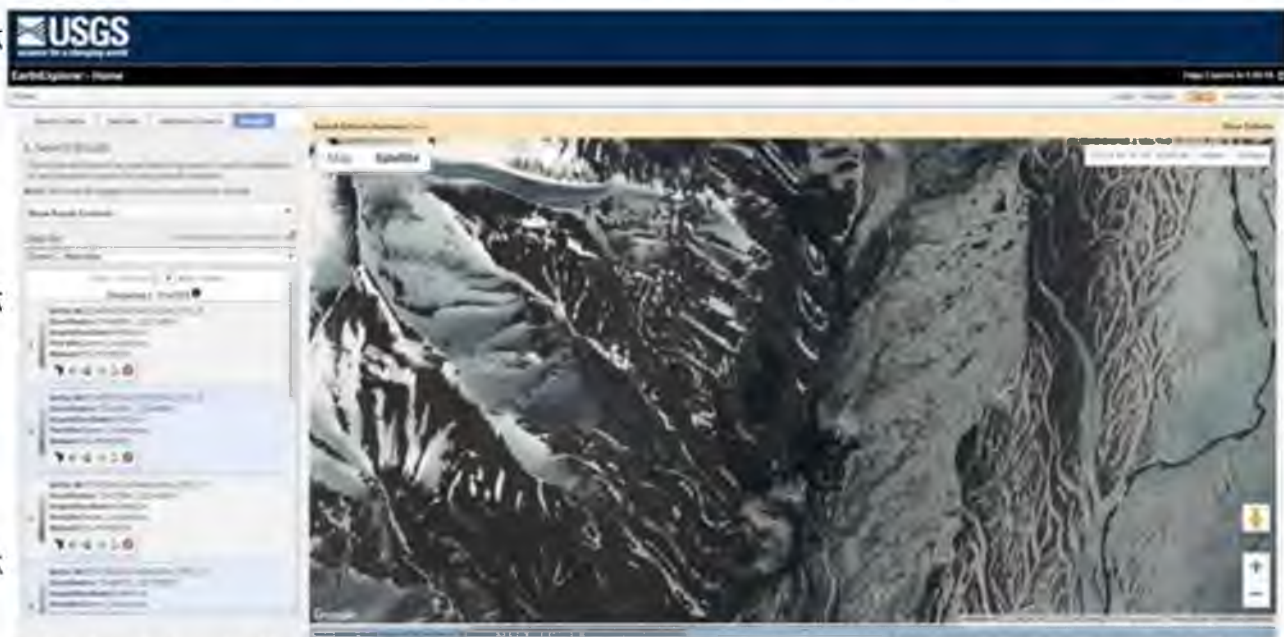


Рис. 1.7 Інтерфейс EarthExplorer

Пошук. Сервіс EarthExplorer дозволяє фільтрувати результати пошуку за датою, відсотку хмарності і будь-якій кількості сенсорів. Можна переглядати кожен тайм окремо або всі разом в залежності від уподобань.

Цікаво, що в EarthExplorer можна скористатися пошуком по функціям. Це дозволяє переглядати довгий список всіх можливих функцій і зустрічати дуже цікаві об'єкти, такі як стародавня дорога або кладовища. Все це на додаток до стандартних параметрах області інтересів, таких як адреса, завантаження файлів і

багато іншого.

**Аналитика.** Не доступна. Можлива тільки через стороннє програмне забезпечення.

**Експорт.** Завантаження знімків з USGS доступна через додаток масового завантаження. Також можна завантажити безліч продуктів з даними, включаючи рівні Level-1, 2, 3, знімки в природних кольорах, тепловізійні знімки і багато іншого в залежності від сенсора.

EarthExplorer є найбільш багатим джерелом безкоштовних супутникових знімків. Сервіс відрізняє велика кількість функцій. Інтерфейс не найзручніший для користувача, і потрібно певний навик, щоб звикнути до цього додатка.

### **LandViewer**

LandViewer (<https://eos.com/landviewer>) - це сучасний джерело супутникових даних і аналітики на основі AI. Сервіс представлений EOS, одним з ключових офіційних дистрибуторів супутникових знімків високого дозволу (Рис. 1.8).

Дані LandViewer охоплює величезну кількість загальнодоступних бібліотек. Це знімки із супутників CBERS-4, Sentinel-1, 2, MODIS / NAIP, Landsat-7, 8, а також Landsat-4, 5 для історичних знімків. Серед наборів знімків SPOT -5-7, Pleiades-1, Kompsat-2, 3, 3A, SuperView-1. Максимальна просторову роздільну здатність досягає 40 см на піксель.

З урахуванням усього вищесказаного ви можете отримати повний список супутникових даних і придбати великі плани зображень надвисокої роздільної здатності.



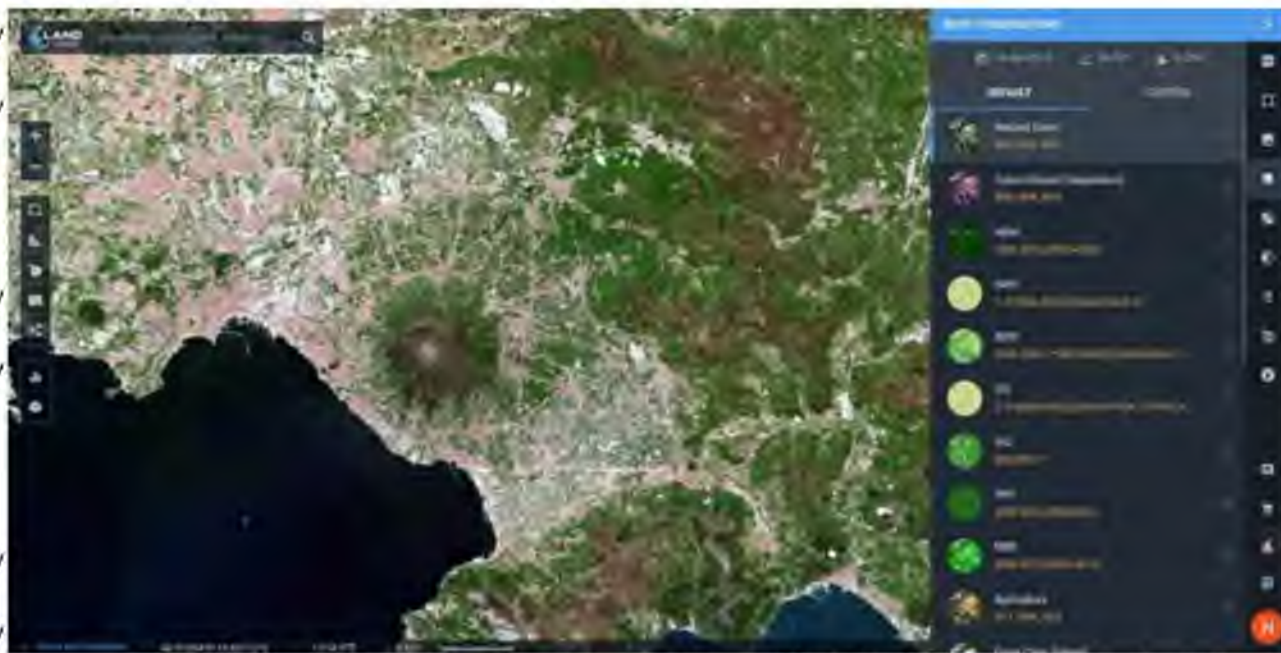


Рис. 1.8 Інтерфейс LandViewer

Пошук. Знімки легко знайти, задавши зацікавлену вас область за допомогою величезної кількості опцій і завантаження файлу. Вибирається тип сенсора і таймфрейм. Відсоток хмарного покриття, висота над рівнем моря і відсоток покриття AOI доступні в якості умов пошуку фільтра.

Крім того, ви можете зберегти свою область інтересів і підписатися на нові сцени, щоб легко отримати те, що вам знадобиться в майбутньому.

Аналітика. LandViewer відразу включає аналітичні можливості візуалізації вибраної фотографії, що робить сервіс унікальним серед багатьох інших платформ.

Пропонується більше 20 стандартних комбінацій і індексів, в тому числі: NDVI, NBR, SAVI. Можна створити свій власний користувальницький індекс, кластеризації, аналіз часових рядів і використовувати безліч інших інструментів для адаптації знімків до заданим потребам.

Експорт. У сховищі LandViewer є багато всього. Можна завантажити будь-який супутниковий знімок з широким спектром каналів або в природних кольорах. Він може бути повним або обрізаним в форматі JPEG, KMZ або GeoTIFF.

LandViewer – універсальне джерело супутникових знімків для ГІС-фахівців. Йому вдасться об'єднати величезний список аналітичних функцій в акуратний пакет для досліджень.

### 1.6 Обробка даних ДЗЗ

Дані дистанційного зондування – дані аерокосмічного або авіаційного зондування (Рис. 1.9). Дані які були отримані будь-якими дистанційними способами та містять інформацію про поверхню Землі, досліджуваних об'єктів, розташованих на ній чи надрах Землі.

Обробка даних ДЗЗ – процес корекції, перетворення та покращення, дешифрування, візуалізації над аерокосмічними або авіаційними знімками.

Етапи обробки даних космічних знімків:

- Попередня обробка
- Тематична обробка

Попередня обробка мультиспектральних даних

Попередня обробка – це процес виконуваний над аерокосмічними або авіаційними знімками задля їх корекції та поліпшення.

Слід пам'ятати, деякі методи поліпшення зображень (фільтрація, зміна контрасту) мають на увазі зміну спектральних характеристик знімка, тому після їх

застосування не можна застосовувати методи тематичної обробки, засновані на аналізі значень спектральної яскравості пікселів (класифікації, арифметичні перетворення каналів та ін.).

Види попередньої обробки:

- Геометрична корекція супутникових зображень
- Радіометричне калібрування знімків
- Радіометрична корекція впливу атмосфери
- відновлення пропущених пікселів

# НУБІП УКРАЇНИ

• **Контрастування**

Геометрична корекція включає:

- усунення на зображенні геометричних спотворень
- географічну прив'язку

# НУБІП УКРАЇНИ

**Причина геометричного спотворення знімка.**

**Кривизна поверхні Землі**

# НУБІП УКРАЇНИ

# НУБІП УКРАЇНИ

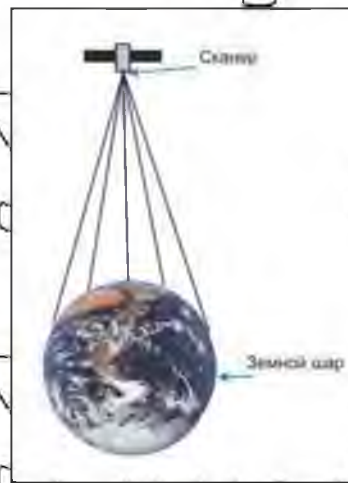


Рис. 1.9 Зображення зйомки з супутника

Так як точки сканованої місцевості не лежать на одній площині і нагляд ведеться не в надрі, а під кутом до Землі, то результатом буде спотворення знімку, спричинені кривизною Земної поверхні. Викривлення форми та розмірів об'єктів підвищується при віддаленні від центральної лінії.

# НУБІП УКРАЇНИ

**Спотворення форми об'єктів.** Квадрат перетвориться на прямокутник, пряма лінія на криву і тому подібно.

**Спотворення масштабу.** Знімки, що виконані за допомогою оптико-механічного сканера (MODIS, AVHRR, ETM та MSS – Landsat, Aster (TIR)) – масштаб стає дрібнішим при віддаленні від центральної лінії знімка. Тобто, якщо взяти два пікселі знімка однакового розміру: один із центральної області знімка, а другий з бічної, то піксель із бічної області міститиме велику площу Землі.

# НУБІП УКРАЇНИ



## Нерівності рельєфу

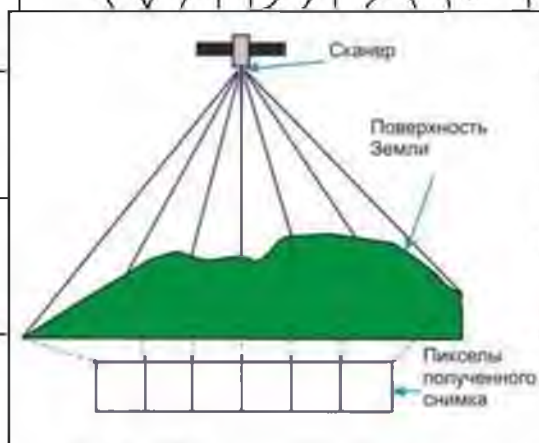


Рис. 1.10. Зображення нерівності рельєфу

Як і кривизна поверхні Землі, так і нерівність рельєфу викликає те саме спотворення, так як форми рельєфу складніші, ніж форма Землі, то і позбутися цих спотворень складніше (рис. 1.10).

Цей тип викривлення враховують тільки для гірських областей, оскільки знімки роблять в великій висоти, вплив форм рельєфу незначний.

## Обертання Землі

Так як сканування здійснюється не миттєво, як наприклад фотозйомка, тому під час сканування сцени обертання Землі впливає на зміну умов зйомки

## Рух космічного апарату у процесі формування зображення

Висота та форма орбіти супутника має вплив на властивості та якість знімку. Наприклад, кругова орбіта забезпечує однакову висоту зйомки земної поверхні, а отже, для однієї і тієї ж апаратури - однакове охоплення та роздільна здатність знімків.



# НУБІП України

## Радіометричне калібрування знімків.

Насамперед отримані з супутників знімки приходять як сирі значення яскравості DN (Digital Number). Адекватно порівнювати дані з іншими знімками в

# НУБІП України

даному форматі не варто.

Завданням радіометричного калібрування є процес приведення цих значень до фізичних одиниць.

# НУБІП України

## Радіометрична корекція впливу атмосфери.

**Поглинання та розсіювання.** Електромагнітні хвилі проходячи через атмосферу стикаються з озonom, водяною паром, вуглекислим газом, киснем та іншими перешкодами, через це вони поглинаються та розсіюються.

# НУБІП України

**Хмарність.** Основною задавою при зйомці з супутника є наявність хмар в оптичному діапазоні. Основними причинами спотворення яскравості пікселів на знімках є поглинання, розсіювання і хмарність.

# НУБІП України

# НУБІП України

## Контрастування

Контраст зображення – це різниця значення яскравості між максимальним та мінімальним значення (Рис. 1.11).

# НУБІП України

Слабкий контраст – найпоширеніший дефект зображень.

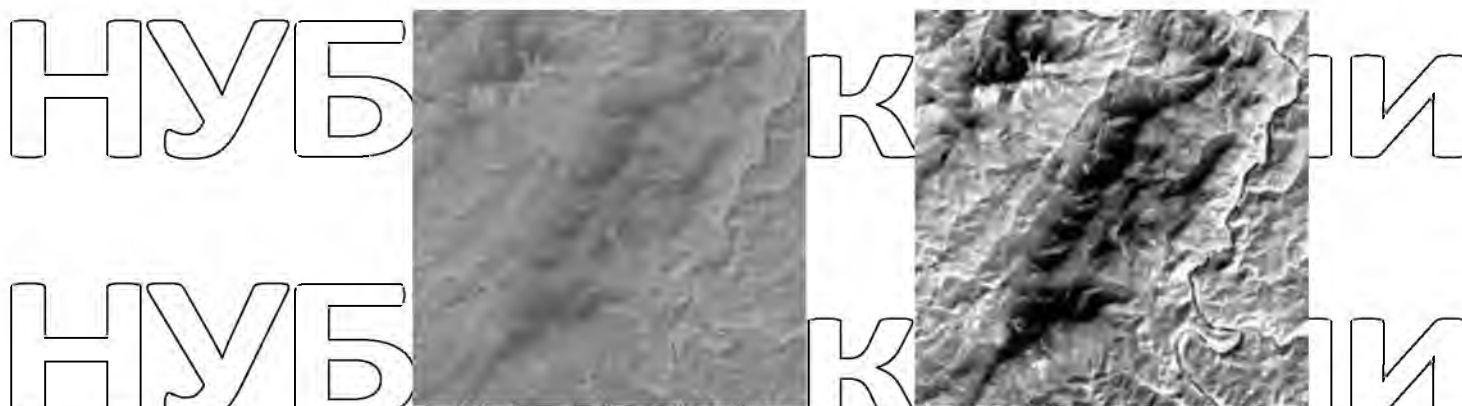


Рис. 1.11 Контрастування

Існує кілька методів підвищення розмаїття шляхом цифрової обробки.

Методи підвищення контрасту:

- Лінійне розтягування гистограми (Рис. 1.12). Веім значенням яскравості надаються нові значення з метою охопити весь можливий інтервал зміни яскравості (0, 255) (Рис. 1.13).

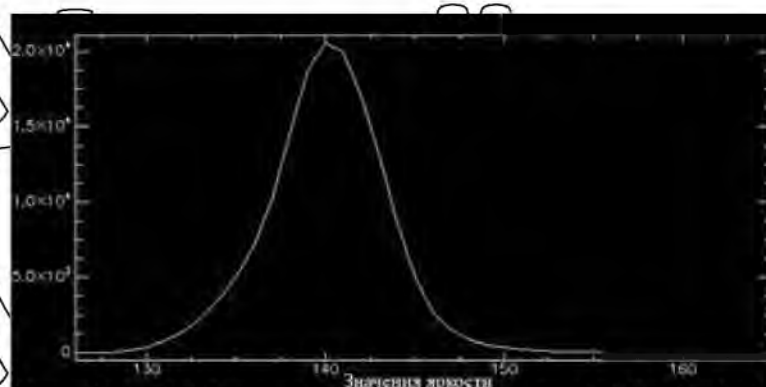
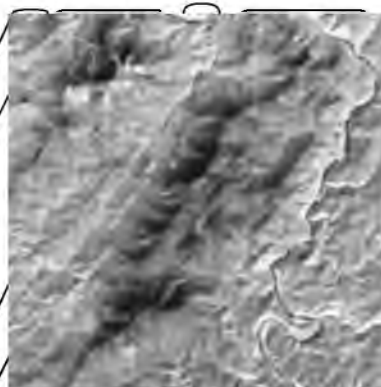


Рис. 1.12 Лінійне розтягування гистограми

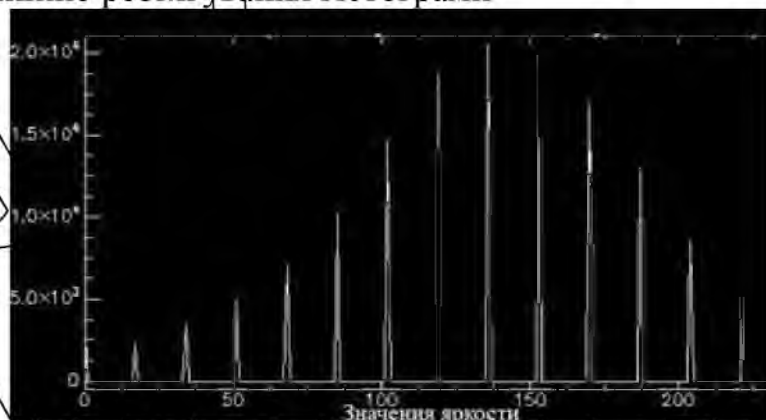
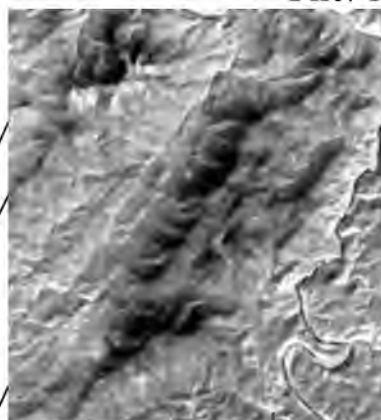


Рис. 1.13 Знімок після лінійного розтягування гистограми (діапазон значень яскравості від 0 до 255, середнє = 136.7)

• Нормалізація гистограми. На весь можливий інтервал зміни яскравості розтягується не вся гистограма, а її найінтенсивніша ділянка (Рис. 1.14).

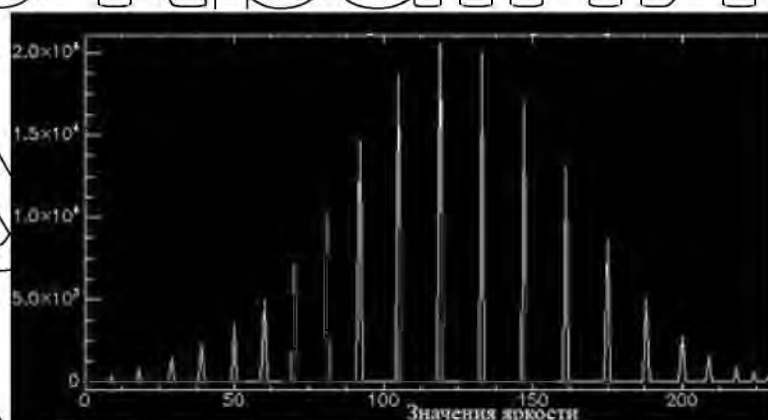
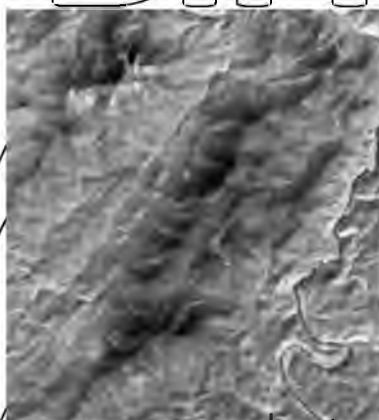


Рис. 1.14 Знімок після нормалізації гистограми (діапазон значень яскравості від 0 до 255, середнє = 121.5)

• Вирівнювання гистограми (лінеаризація, еквалізація) (Рис. 1.15). У процесі вирівнювання відбувається зміна значень яскравості пікселів таким чином, щоб для кожного рівня яскравості була однакова або близька кількість пікселів.

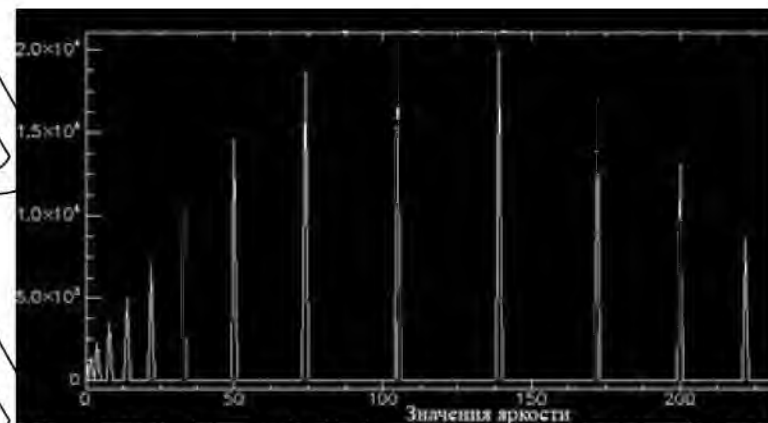
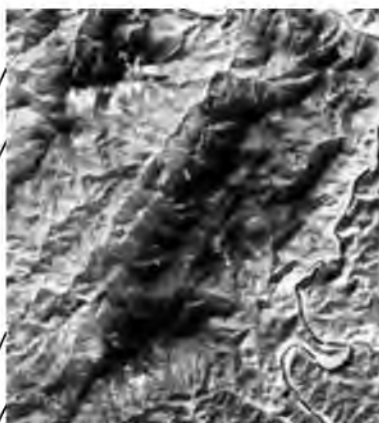


Рис. 1.15 Знімок після вирівнювання гистограми (діапазон значень яскравості від 0 до 255, середня = 115.9)

### Тематична обробка мультиспектральних знімків

Тематична обробка космічних знімків – це процес дешифрування чи розпізнавання об'єктів та явищ на космічних знімках.

Способи тематичної обробки:

- Колірні перетворення
- Індексні зображення
- Аналіз основних компонентів
- Метод спектральний поділ
- Класифікації

### Колірні перетворення.

Якщо вивести на екран дисплея зображення одного каналу багатоспектрального знімка, воно буде пофарбоване в сірі тони (напівтонове зображення).

Щоб отримати кольорове зображення, потрібно скласти разом три канали багатоспектрального знімка (Рис. 1.16). Один із яких буде червоний (R), інший зелений (G), третій синій (B).

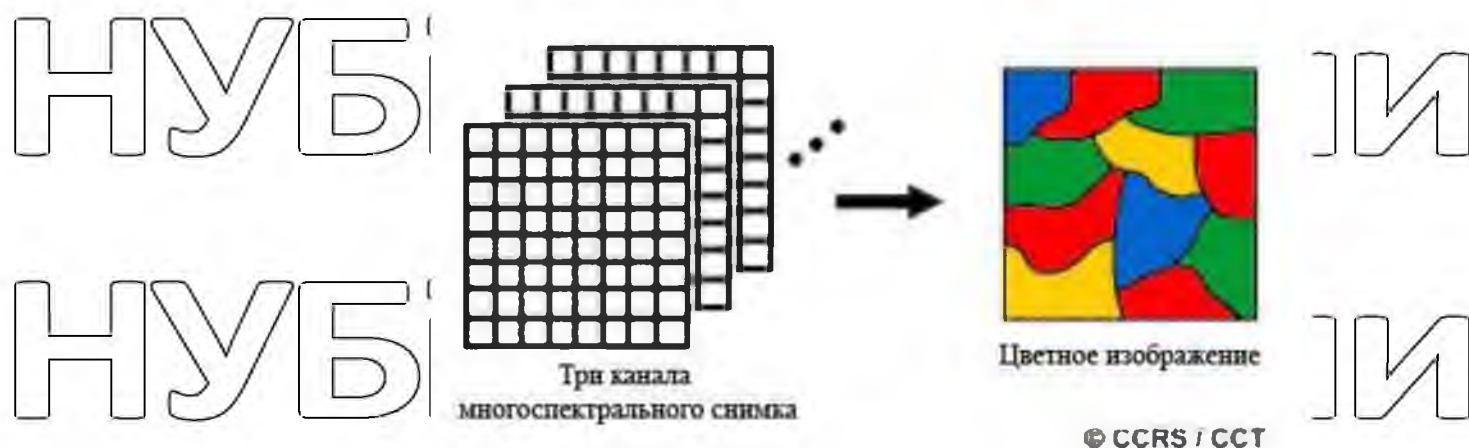


Рис. 1.16 Зображення кольорового перетворення

НУБІП УКРАЇНИ

**Можна фарбувати зображення в натуральні та помилкові кольори.** Якщо канали R, G, B зображення підставлені R, G, B канали знімка, то зображення представлено натуральних кольорах. Якщо у каналах R, G, B зображення

НУБІП УКРАЇНИ

представлено інші канали знімка, то зображення представлено у помилкових кольорах (Рис. 1.17)

**Механізм вибору кольору зображення.** Якщо канал зображення R

підставити канал знімка, у якому значення яскравості пікселів високі, а канали G і

НУБІП УКРАЇНИ

B канали з низькими значеннями яскравості, зображення буде пофарбовано переважно в червоні тони тощо (Рис. 1.18).

НУБІП УКРАЇНИ

НУБІП УКРАЇНИ



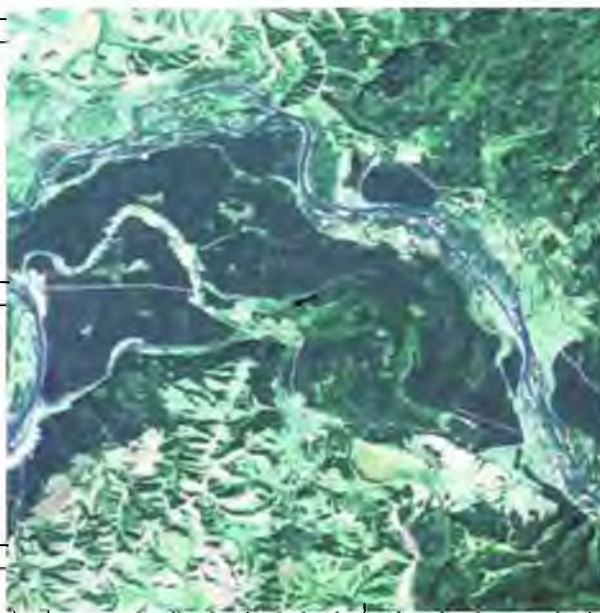


Рис. 1.17 Приклад забарвлення знімка у натуральні кольори

Таблиця 1

Канали зображення	Спектральні канали знімка	Колір води на зображенні	Колір рослинності на зображенні	Колір ґрунту на зображенні
R G B	Червоний Зелений Синій	Синій	Зелений	Коричнево-сірий

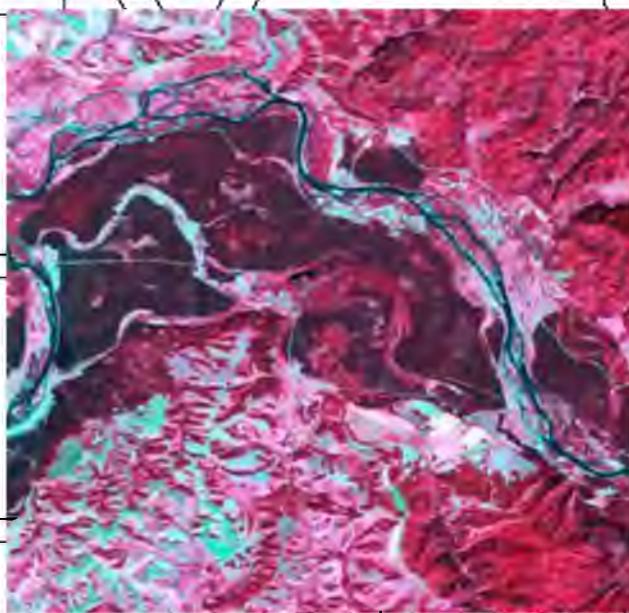


Рис. 1.18 Приклад забарвлення зображення в помилкові кольори

Таблиця 2

Канали зображення	Спектральні канали знімка	Колір води на зображенні	Колір рослинності на зображенні	Колір ґрунту на зображенні
R G B	Близький інфрачервоний червоний Зелений	Чорний	Червоний	Сіро-блакитний

Сутність об'єктів доцільно визначати на зображеннях у натуральних кольорах, а розділяти та оконтурювати об'єкти зручніше на зображеннях у фальшивих кольорах. Правильно підібрана шкала дозволяє оператору візуально виділяти об'єкти, що мало відрізняються за яскравістю на напівтоновому зображенні.

### Індексні зображення

Для отримання індексного зображення значення яскравості кожного пікселя обчислюється шляхом застосування арифметичних операцій над значеннями яскравості цього пікселя з різних каналів знімка.

Що таке зображення? При вивченні об'єктів за багатозональними знімками часто важливі не абсолютні значення, а характерні співвідношення між значеннями яскравості об'єкта в спектральних зонах. На таких зображеннях більш яскраво і контрастно виділяються об'єкти, що шукаються в порівнянні з вихідним знімком

(Рис. 1.19).

Приклади обчислення індексних зображень можна побачити в Таблиці 3.

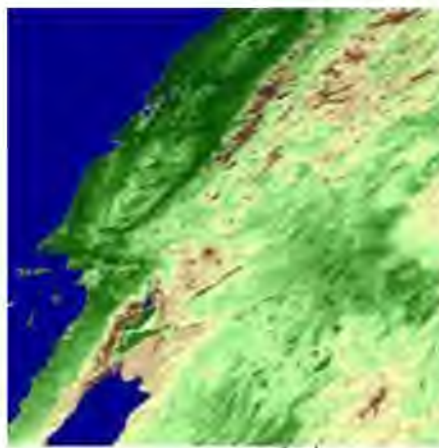


Рис. 1.19 Результат застосування вегетаційного індексу

Таблиця 3

Назва індексу	Формула	Застосування		
Індекс вмісту оксиду заліза	Відношення значень яскравості у червоному (К) каналі до значень яскравості у синьому каналі (С). $K/C$	Для виявлення вмісту оксидів заліза		
Індекс вмісту глинистих мінералів	Відношення значень яскравості в межах середнього інфрачервоного каналу $СИК1/СИК2$ СИК1 це діапазон від 1.55 до 1.75 мкм, СИК2 це діапазон від 2.08 до 2.35 мкм.	Для виявлення вмісту глинистих мінералів		
Індекс вмісту залізистих мінералів	Відношення значення яскравості в середньому інфрачервоному каналі (СИК1 див вище) до значення яскравості в ближньому інфрачервоному каналі (БІК). $СИК1/БІК$	Для виявлення вмісту залізистих мінералів		
Індекс червоноцвстості (R1)	Заснований на відмінності відбивної здатності червонокольорових мінералів у червоному (К) та зеленому (З) діапазонах $R1 = K - 3K + 3$	Для виявлення вмісту оксиду заліза у ґрунті		
Нормалізований диференціальний вегетаційний індекс (NDVI)	Хлорофіл листя рослин відображає випромінювання в ближньому інфрачервоному (БІК) діапазоні електромагнітного спектру і поглинає в червоному (К). Відношення значень	Показує наявність та стан рослинності. Значення NDVI варіюють від -1 до 1		
		<table border="1"> <tr> <td>Тип об'єкта</td> <td>Значення NDVI</td> </tr> <tr> <td>Густа рослинність</td> <td>0.7</td> </tr> </table>	Тип об'єкта	Значення NDVI
Тип об'єкта	Значення NDVI			
Густа рослинність	0.7			



	яскравості у цих двох каналах дозволяє чітко відокремлювати та аналізувати рослини від інших природних об'єктів. $NDVI = \frac{BIK - K}{BIK + K}$	<table border="1"> <tr> <td>Розряджена рослинність</td> <td>0.5</td> </tr> <tr> <td>Відкритий ґрунт</td> <td>0.025</td> </tr> <tr> <td>Хмари</td> <td>0</td> </tr> <tr> <td>Сніг та лід</td> <td>-0.05</td> </tr> <tr> <td>Вода</td> <td>-0.25</td> </tr> <tr> <td>Штучні матеріали (бетон, асфальт)</td> <td>-0.5</td> </tr> </table>	Розряджена рослинність	0.5	Відкритий ґрунт	0.025	Хмари	0	Сніг та лід	-0.05	Вода	-0.25	Штучні матеріали (бетон, асфальт)	-0.5
Розряджена рослинність	0.5													
Відкритий ґрунт	0.025													
Хмари	0													
Сніг та лід	-0.05													
Вода	-0.25													
Штучні матеріали (бетон, асфальт)	-0.5													
Нормалізований диференціальний індекс снігу (NDSI)	NDSI - це відносна величина, що характеризується відмінністю відбивної здатності снігу в червоному (K) і короткохвильовому інфрачервоному (KK) діапазонах. $NDSI = \frac{KK - K}{KK + K}$	Використовується виділення території, покритих снігом. Для снігу $NDSI > 0.4$ .												
Водний індекс (WI)	$WI = 0.90\text{мкм} / 0.97\text{мкм}$	Застосовується для визначення вмісту води в рослинності за гіперспектральними знімками.												

### 1.7 Проблема обробки великої кількості даних ДЗЗ

На прикладі сільського господарства ми розглянемо проблему обробки великої кількості даних дистанційного зондування Землі.

Сьогодні агроном – це менеджер та технолог виробництва сільгосппродукції, а також бригадир робітників. Його робота відбувається безпосередньо на відкритому повітрі в полях або в теплицях, розплідниках.

На агрономів покладається велика відповідальність за врожай, тому ці фахівці повинні мати і професіоналізм та низку особистісних якостей для управління колективом.

#### Обов'язки

Агроном має широкий перелік обов'язків. До них належать такі:

- підбір сільгоспкультур, ґридатних для вирощування у цій місцевості та вибір конкретних сортів;

НУБІП УКРАЇНИ

- визначення термінів посіву та збирання врожаю;
- вибір методів боротьби зі шкідниками сільськогосподарських культур та контроль за їх виконанням;

- цілорічний планування всіх видів сільськогосподарських робіт;

НУБІП УКРАЇНИ

- обхід закріпленої території та спостереження за зростанням сільськогосподарських культур;
- культивування ґрунтів, вивчення їх стану та правильний підбір добрив;

- розробка та здійснення заходів щодо рекультивації земель;

НУБІП УКРАЇНИ

- відстеження погодних умов (опадів, вологість, температура повітря) та коригування прийнятих рішень відповідно до їх змін;
- забезпечення безпеки вирощеного та зібраного врожаю;

- заготівля насінневого матеріалу до наступного сезону та забезпечення

НУБІП УКРАЇНИ

- комплексу заходів для збереження посівного фонду;
- розстановка підлеглих робітників на необхідних місцях та контроль за виконанням ними поставлених завдань;

- селекція нових рослин, окремих сортів;

НУБІП УКРАЇНИ

- аналіз досягнутих результатів (кількість урожаю) та визначення причин «провального» сезону у разі невиконання плану.

По суті, агроном – це сполучна ланка між робітниками та керівником підприємства у сільському господарстві.

НУБІП УКРАЇНИ

Досить велика кількість обов'язків для однієї людини. Але з використанням дистанційного зондування Землі та різноманітних методів обробки супутникового зображення а також зображення з дронів ми спроможні об'єднати життя агроному.

Дані дистанційного зондування Землі будуть в нагоді при:

НУБІП УКРАЇНИ

- Обході території, адже вона може бути занадто великою і через це агроном не в змозі у повній мірі дослідити зростання сільськогосподарських культур.

Не спроможність агронома виявити заражені ділянки в тяжко доступних місцях поля, адже досить тяжко, якщо не неможливо знайти заражену культуру в полі, що постилається на десятки кілометрів.

- Проаналізувати результати вирощування минулих

сільськогосподарських культур, вдалих або провальних і на основі цих даних зробити коригування у підборі сільськогосподарських культур для цієї чи іншої ділянки, а також спрогнозувати їх зростання та можливі проблеми.

Якщо розглядати більш детально, то моніторинг стану посівів за допомогою супутника — це аналітика, заснована на високоточних супутникових знімках. Виглядає це так — супутник робить знімки, програма, або аналітик, підбирає знімки з необхідною ділянкою та проводить аналіз.

Плюси та мінуси використання супутників

Плюси:

- базовий моніторинг;
- автоматизована обробка даних;
- можливість отримання інформації безкоштовно;
- можливість розрахунку NDVI індексу;
- можливість одержання історичної аналітики (історичні знімки);
- аналітика як окремих полів, так і певних культур,

Мінуси:

- погодні умови (у разі хмарності знімків не буде);
- час одержання знімків (від 5 до 8 днів);
- точність даних (при використанні дронів дані точніше);

Якщо принцип використання супутників зрозумілий майже всі, навіть тим, хто в це питання особливо не вникав, то застосування дронів, для моніторингу стану посівів, є досить інноваційним способом і викликає багато питань.

Раніше дрони застосовувалися виключно у військових сферах, з їх допомогою військові знищували повітряні цілі та збирали розвіддані. Сьогодні дрони доступні кожному, хто забажає їх придбати і тому їм знайшли застосування в різних сферах

життя. Вони широко використовуються для моніторингу погодних умов, бізнесу,

пов'язаного з фото та відео зйомкою, ведення пошуково-рятувальних операцій, гасіння пожеж, особистих цілей, звичайно ж у сільському господарстві і навіть для доставки товарів.

Як і говорилося раніше, застосування дронів для моніторингу стану посівів є

інноваційним способом для українського фермера і тому поступається у частоті використання супутниковому моніторингу. А дуже даремно, адже дрон це інструмент для точного землеробства.

Застосування дрона значно полегшує отримання необхідної інформації про

стан посівів. Це пов'язано з його мобільністю і можливістю самостійно регулювати висоту польоту від 10 до 200 (у деяких випадках навіть від 3-х метрів) над поверхнею землі, що в свою чергу дає можливість отримання знімків з розширення в сантиметрах на піксель, а не метрах як при використанні супутників.

Доречно буде уточнити, що дрони застосовуються не тільки для моніторингу стану посівів, їх також використовую для аналітики (NDVI, NDRI, RVI і т. д.) та внесення трихограми. Просто, залежно від поставленої мети, підбирається БПЛА з певними технічними можливостями (дальність та стабільність польоту, вагопідйомність, наявність спеціальної камери тощо).

Плюси та мінуси використання:

Плюси:

- рядки отримання інформації;
- можливість самостійного управління;
- можливість зйомки у хмарну погоду;

# НУБІП України

- отримання знімків із розширенням у сантиметрах;

Мінуси:

- погодні умови;
- вартість покупки дрону (чим більше функцій, тим дорожче);

# НУБІП України

- наявність так званих «no fly zone» (доляти біля та над стратегічними та об'єктами держ. призначення заборонені).

Не важливо який із способу моніторингу Ви для себе виберете, адже у будь-якому випадку вони спрямовані на збереження культурних рослин та збільшення врожайності, все залежить тільки від того, яку саме Ви хочете отримати інформацію.

# НУБІП України

Отримавши дані дистанційного зондування ми повинні провести над ними проаналізувати, завдяки наявності спеціальних спектральних камер можна розрахувати різноманітні вегетаційні індекси (NDVI, EVI, GNDVI, CVI і тому подібні). І як ми розуміємо інколи цих даних може бути велика кількість, адже ми може досліджуємо зображення не за один день, а за певний період, наприклад 2 – 3 роки, щоб на основі минулих даних проаналізувати майбутню врожайність, стан ґрунтів. Тому часу на обробку такої кількості даних може бути затрачено чимало. Постає питання «як можливо підвищити швидкість обробки?».

# НУБІП України

Ми могли б використати багатопотоковість. Розберемо, що це означає. Всі програми написані без використання багатопотоковості виконуються в одному потоці, тобто всі наші задачі виконуються послідовно одна за одною з можливим затримками (операціями I/O) (рис. 1.20). І ми розуміємо, що послідовне виконання і затримки збільшують час виконання. Використовуючи багатопотоковість ми могли б виконувати під завдання в різних потоках(рис. 1.21 і), що в теорії пришвидшить виконання.

# НУБІП України

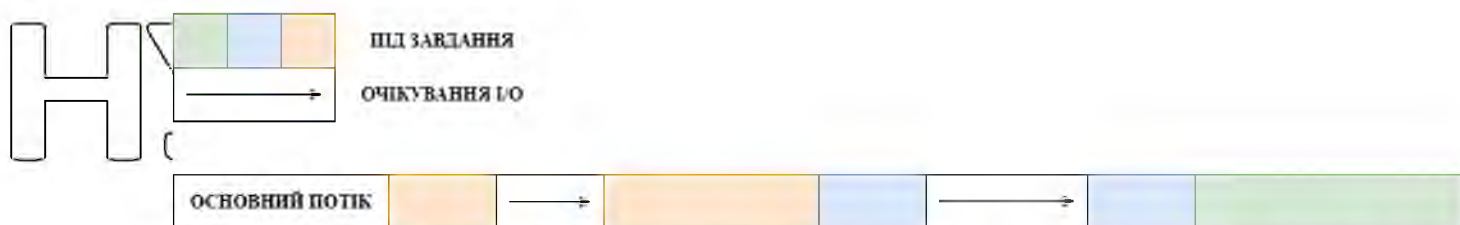


Рис. 1.20 Схеми виконання програми в однопоточковому режимі



Рис. 1.21 Схеми виконання програми в багатопотоковому режимі

## 2 МОДЕЛЮВАННЯ СИСТЕМИ

### 2.1 Загально про UML

Багато програмістів, зіткнувшись зі складним завданням, нехтують етапом проектування, посилаючись на те, що проектування — це втрата часу, і в даному випадку воно мені глвки заважатиме.

Найчастіше це твердження виявляється вірним, якщо завдання справді невелика і кваліфікації програміста достатня для визначення найбільш оптимального рішення.

Програмісти, які не використовують UML, діляться на кілька груп:

- почну писати код, а в процесі зрозумію, що та як;
- починаю форуми, хабр, medium, stack overflow, книгу, записи на стінах, знаки на полі;

- питаю у колег, може, хтось знає, як вирішити подібне завдання;

- почну малювати квадратики та схематично показувати, яке бачення завдання сформувалося у мене у свідомості.

Але при вирішенні складніших завдань завчасне планування та моделювання значно спрощують програмування. Крім того, вносити зміни до діаграм класів легше, ніж у вихідний код.

Можна провести аналогію із будівництвом будинку. Коли хтось хоче збудувати будинок, він не просто б'є молотком і розпочинає роботу. Йому потрібно мати план — план проектування, щоб він міг аналізувати та модифікувати свою

систему. Якщо ви вже почали описувати на папері ваше завдання, то це вже величезний плюс.

### Що таке UML

UML – уніфікована мова моделювання (Unified Modeling Language) – це система позначень, яку можна використовувати для об'єктно-орієнтованого аналізу та проектування. Його можна використовувати для візуалізації, специфікації, конструювання та документування програмних систем.



На рис. 2.1 зображено логотип мови UML.



Рис. 2.1 Логотип UML

Простіше кажучи, якщо подивитися картинки в пошукових системах, то стане зрозуміло, що UML – це щось про схеми, стрілочки та квадратики.

Важливе, що UML перекладається як Unified Modeling Language. Головне слово Unified. Тобто наші картинки зрозуміємо не лише ми, а й інші, які знають UML. Виходить, це така міжнародна мова малювання схем.

Плюси та мінуси UML проектування:

Плюси:

- можливість подивитися завдання з різних точок зору;
- іншим програмістам легше зрозуміти суть завдання та спосіб її

реалізації:

- діаграми порівняно прості для читання після досить швидкого ознайомлення з синтаксисом.

Мінуси:

- трата часу;
- необхідність знання різних діаграм та їх нотаций.



Діаграми даної нотації розрізняються за типами і описують різні аспекти розробки. Розрізняють 2 основні типи UML-діаграм: структурні та поведінкові.

Структурні діаграми відображають елементи, з яких складається система.

Поведінкові моделі описують процеси, що протікають в системі.

У кожній мові моделювання існує свій словник. Словник UML складається з наступних елементів:

- Сутності – це абстракції, які є основними елементами моделей;
- Відносини - це зв'язки між сутностями;
- Діаграми - це відображення взаємодії сутностей і відносин.

**Сутності.**

Сутності в UML можуть бути 4 типи:

- Структурні сутності є іменами існуючими моделі, її статичні частини.

Це класи, компоненти, інтерфейси і т.д., які відповідають фізичним елементам системи.

- Поведінкові сутності є дієсловами моделі, описують її поведінку в часі і просторі. Їх існує всього 2 типи: взаємодія (обмін повідомленнями) і автомобіль

(послідовність станів).

- Анотаційні сутності є пояснювальними частинами моделі. Це примітки та коментарі до елементів системи.

- Групуючі сутності - організуючі частини моделі. Вони організують елементи системи в групи.

**Відносини**

Існує 4 типи відносин в UML (Рис. 2.2 Види відносин):

- Залежність - це відношення між двома сутностями, при якому зміна однієї (незалежної) сутності призводить до зміни другої сутності. Графічно це зображається пунктирною стрілкою.

Узагальнення - це відношення "спеціалізації-узагальнення", де спеціалізований об'єкт може бути підставлений замість узагальненого.

- Асоціація - ставлення, що описує семантичний зв'язок між об'єктами.

Графічно зображується у вигляді суцільної стрілки, яка може містити короткість або імена ролей. (Агрегація - різновид асоціації і відображає відношення частини до цілого).

- Композиція - різновид агрегації, де взаємозв'язок частини з цілим ще більш сильна.

- Реалізація - відношення між класифікаторами, при якому один визначає зобов'язання, а другий здійснює його виконання.

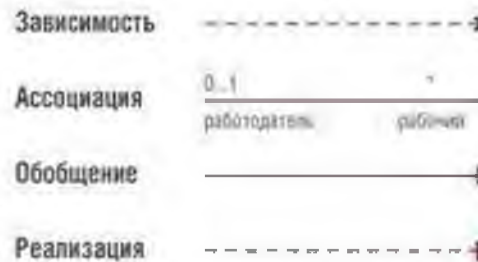


Рис. 2.2 Види відносин

в UML існує 10 типів діаграм:

- Об'єктів (object);
- Класів (class);
- Взаємодія (interaction);
- Варіантів використання (прецедентів, use-case);
- Послідовності (sequence);

НУБІП України

- Станів (statechart);
- Кооперацій (Colaboration);
- Компонентів (component);

- Розгортання (deployment);

НУБІП України

- Діяльності (activity)

## 2.2 Діаграма прецедентів

В уніфікованій мові моделювання (UML) діаграма прецедентів може узагальнити деталі користувачів вашої системи (також відомих як акторів) та їх взаємодію з системою. Щоб створити діаграму прецедентів, ви будете використовувати набір спеціалізованих символів і сполучників. Діаграма

прецедентів може допомогти вашій команді обговорити та представити:

НУБІП України

- Сценарії, в яких ваша система або програма взаємодіють з людьми, організаціями або зовнішніми системами
- Цілі, яких ваша система або програма допомагає досягти цим суб'єктам (відомим як актори).

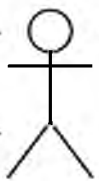
НУБІП України

- Сфера дії вашої системи

**Актори:** користувачі, які взаємодіють із системою. Актором може бути людина, організація або стороння система, яка взаємодіє з вашим додатком або системою. Вони повинні бути зовнішніми об'єктами, які виробляють або споживають дані (Рис. 2.3).

НУБІП України

# НУБІП України



Actor

Рис. 2.3 Зображення актора

# НУБІП України

**Система:** конкретна послідовність дій і взаємодій між акторами та системою. Систему також можна назвати сценарієм.

**Цілі:** Кінцевий результат більшості випадків використання. Успішна діаграма

повинна описувати дії та варіанти, які використовуються для досягнення мети.

# НУБІП України

**Прецеденти:** овали горизонтальної форми, які представляють різні варіанти використання, які може використовувати користувач (Рис. 2.4).

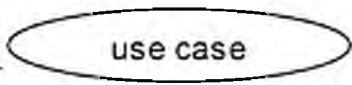


Рис. 2.4 Зображення прецедента

# НУБІП України

**Актори:** фігурки, які представляють людей, які фактично використовують варіанти використання (рис).

**Асоціації:** межа між акторами та варіантами використання. У складних діаграмах важливо знати, які дійові особи пов'язані з якими варіантами використання.

Діаграму прецедентів можна побачити на рис. 2.5 Діаграма прецедентів

# НУБІП України

# НУБІП України



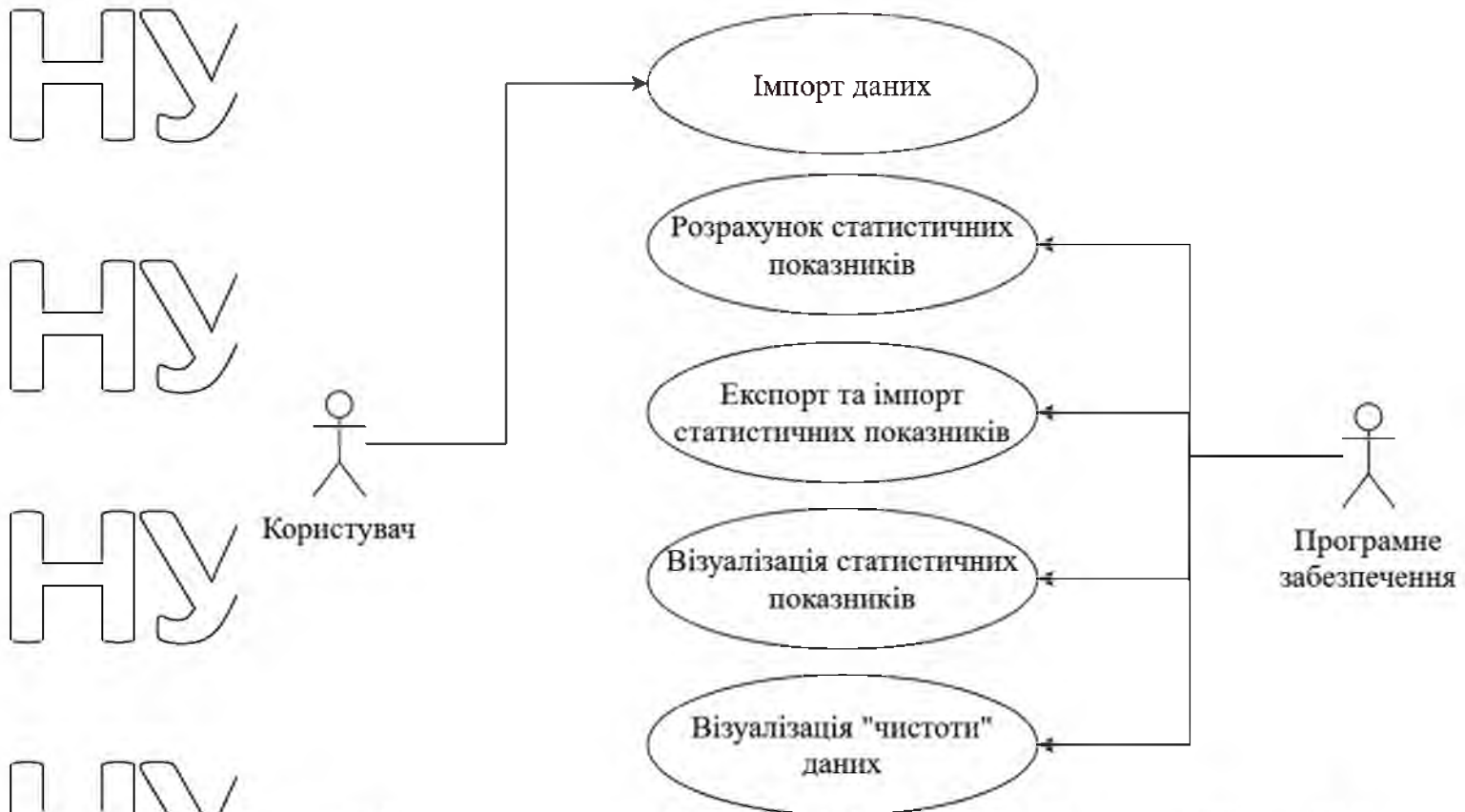


Рис. 2.5 Діаграма прецедентів

### 2.3 Діаграма послідовності

Діаграма послідовності — це тип діаграми взаємодії, оскільки вона описує, як і в якому порядку група об'єктів працює разом. Ці діаграми використовуються розробниками програмного забезпечення та бізнес-фахівцями, щоб зрозуміти вимоги до нової системи або документувати існуючий процес. Діаграми послідовності іноді називають діаграмами подій або сценаріями подій.

Діаграму послідовності можна побачити на рис. 3.5 Діаграма послідовності

## 3 РОЗРОБКА СИСТЕМИ

### 3.1 Технічне завдання

#### Загальні вимоги до ПЗ:

ПЗ повинно бути реалізовано як програмний додаток з інтерактивним меню.

В меню окремими позиціями повинні бути реалізовані всі функції програмного забезпечення, викладені в наступному пункті.

#### Функції програмного забезпечення:

- Імпорт даних з каталогу (директорії). Дані поіменовані згідно стандартного встановленого ідентифікатору імені файлів (карт стану посівів та файлів метаданих).

- Розрахунок комплексу статистичних показників (середнє зважене значення індексу, середньоквадратичне відхилення, частот входження унікальних значень у вибірку) по кожній карті стану посівів. Розрахунок комплексу статистичних показників має відбуватися з використанням багатепотокового режиму.

- Експорт комплексу статистичних показників в БД.

- Імпорт статистичних показників з БД для поля (по ID)
- Візуалізація статистичних показників для поля (по ID) за увесь вегетаційний сезон, за частину вегетаційного сезону. Під статистичними показниками мається на увазі середнє зважене та довірчі інтервали. Реалізувати можливість порівняння часових трендів по кільком (2 і більше) полям на 1-му графіку. У Візуалізацію включати виключно дані по полям, що є хоча б на 50% безхмарними.

- Візуалізація «чистоти» даних в розрізі місяців (скільки кожного місяця по конкретному полю було імпортовано та оброблено безхмарних, частково захмарених та повністю захмарених зйомок).

### 3.2 Використані технології

#### Python

Python — високорівнева мова програмування загального призначення з динамічною строгою типізацією та автоматичним управлінням пам'яттю, орієнтована підвищення продуктивності розробника, читання коду та його якості, і навіть забезпечення переносимості написаних у ньому програм. Мова є повністю об'єктно-орієнтованою - все є об'єктами. Синтаксис ядра мови мінімалістичний, за рахунок чого рідко виникає необхідність звертатися до документації. Сама ж мова відома як інтерпретована і використовується в тому числі для написання скриптів. Недоліками мови є найчастіше нижча швидкість роботи і більш високе споживання пам'яті написаних нею програм.

Основні фактори, які наводяться користувачами Python, приблизно такі:

- Якість програмного забезпечення
- Висока швидкість розробки
- Переносимість програм
- Бібліотеки підтримки
- Інтеграція компонентів

#### Багатопотоковість в python

Потоки управління (threads) утворюються та працюють у рамках одного процесу. В однопотоковій програмі є лише один потік управління. Говорячи спрощено, при запуску програми цей потік послідовно виконує оператори, прямуючи однією з альтернативних гілок оператора вибору, проходить через тіло циклу потрібну кількість разів, вибирається до місця обробки винятку. Будь-якої миті часу інтерпретатор Python знає, яку команду виконати наступною. Після виконання команди стає відомо, якій команді передати керування. Ця ниточка безперервна в ході виконання програми та обривається лише після її завершення.

Тепер можна уявити, що в деякій точці програми ниточка роздвоюється, і кожен потік іде своїм шляхом. Кожен із потоків, що утворилися може надалі ще кілька разів роздвоюватись. (При цьому один із потоків завжди залишається головним, і його завершення означає завершення всієї програми.) кожен момент

часу інтерпретатор знає, яку команду який потік повинен виконати і приділяє квант часу кожному потоку. Таке, начебто, незначне ускладнення механізму виконання програми насправді потребує якісних змін у програмі - адже діяльність потоків має бути узгоджено. Не можна допускати, щоб потоки одночасно змінювали один і той

же об'єкт, результат такої зміни, швидше за все, порушить цілісність об'єкта. Одним із класичних засобів узгодження потоків є об'єкти, звані семафорами. Семафори не допускають виконання певної ділянки коду кількома потоками одночасно.

Підтримка багатопоточності в мові Python доступна через ряд модулів. У стандартному модулі `threading` визначено необхідні розробки багатопоточної (`multithreading`) програми класи: кілька видів семафорів (класи замків `Lock`, `RLock` і клас `Semaphore`) та інші механізми взаємодії між потоками (класи `Event` і `Condition`), клас `Timer` для запуску функції через деякий час. Для створення та (низькорівневого) управління потоками у стандартному модулі `thread` визначено клас `Thread` (рис. 3.11).



### Приклад багатопотокової програми

```

# Підключення бібліотеки для роботи з потоками
import threading

# Функція з виводу тексту в консоль
def thread(n):
    print('Поток', n)

# Створюються 2 екземпляри класу Thread
# target - об'єкт який буде викликано в методи
# run() - це функція точки входу будь-якого потоку.
# args - аргументи
t1 = threading.Thread( target=thread, args=['1'])
t2 = threading.Thread( target=thread, args=['2'])

# Запускає потік під час виклику методу run.
t1.start()
t2.start()

# Дозволяє програмі очікувати завершення потоків.
t1.join()
t2.join()

```

Рис. 3.1 Приклад багатопотокової програми

Спочатку виходить два об'єкти класу `Thread`, які потім і запускаються з різними аргументами. В даному випадку в потоках працює та сама функція `thread()`, якою передається один аргумент, заданий в іменованому параметрі `args` конструктор класу `Thread`. Неважко здогадатися, що метод `start()` служить для запуску нового потоку, а метод `join()` для очікування завершення потоків. Таким чином, у наведеному прикладі працюють три потоки: основний та два додаткових.

## Функції модуля threading

У модулі threading, що тут використовується, є функції, що дозволяють отримати інформацію про потоки.

- `activeCount()` Повертає кількість активних зараз екземплярів класу

`Thread`,

- `currentThread()` Повертає поточний об'єкт-потік, тобто відповідний потоку управління, що викликав цю функцію. Якщо потік не було створено через модуль threading, буде повернуто об'єкт-потік з скороченою функціональністю (`dummy thread object`).

- `enumerate()` Повертає список активних потоків. Завершені та ще не розпочаті потоки не входять до списку.

## Клас Thread

Примірники класу `threading.Thread` представляють потоки Python-програми. Задати дії, які виконуватимуться в потоці, можна двома способами: передати конструктору класу об'єкт, що виконується, і аргументи до нього або шляхом успадкування отримати новий клас з перевизначеним методом `run()`. Конструктор

класу `threading.Thread` має такі аргументи:

`Thread(group, target, name, args, kwargs)`

Тут `group` - група потоків, `target` - об'єкт, який буде викликаний у методі `run()`, `name` - ім'я потоку, `args` та `kwargs` - послідовність та словник позиційних та іменованих параметрів (відповідно) для виклику заданого у параметрі `target` об'єкта.

У прикладі вище були використані лише позиційні параметри, але те саме можна було виконати і із застосуванням іменованих параметрів

Життям потоків можна керувати викликом методів:

- `start()` Дає життя потоку.
- `run()` Цей метод представляє дії, які мають бути виконані в потік.
- `join([timeout])` Потік, який викликає цей метод, припиняється, очікуючи

завершення потоку, метод якого викликаний. Параметр `timeout` (число з плаваючою точкою) дозволяє вказати час очікування (у секундах), після закінчення якого призупинений потік продовжує свою роботу незалежно від завершення потоку, чий метод `join` був викликаний. Викликати `join()` деякого потоку можна багато разів.

Також не можна чекати завершення ще не запущеного потоку. Слово "join" у перекладі англійської означає "приєднати", тобто, метод, що викликає `join()`, бажає, щоб потік після завершення приєднався до викликаючого методу потоку.

- `getName()` Повертає назву потоку. Для головного потоку це "MainThread".

- `setName(name)` Надає потоку ім'я `name`.
- `isAlive()` Повертає істину, якщо потік працює (метод `run()` вже викликаний, але ще не завершився).

- `isDaemon()` Повертає істину, якщо потік має ознаку демона. Програма на Python завершується після завершення всіх потоків, які не є демонами. Головний потік демоном не є.

- `setDaemon(daemonic)` Встановлює ознаку `daemonic` того, що потік є демоном. Початкове значення цієї ознаки запозичується у потоку, що запустив цей. Ознаку можна змінювати лише для потоків, які ще не запущені.

## Таймер

Клас `threading.Timer` представляє дію, яка має бути виконана через заданий час (рис. 3.2). Цей клас є підклас класу `threading.Thread`, тому запускається методом `start()`. Наступний простий приклад, що друкує на стандартному висновку `Hello, world!` пояснює сказане.

```
# Підключення бібліотеки для роботи з потоками
import threading

# Функція з виводу тексту в консоль
def hello():
    print("Hello, world!")
# Створення екземпляру класу Timer з 2 параметрами
# 1 параметр - час очікування до виконання
# 2 параметр - функція яка буде виконана
timer = threading.Timer(10.0, hello)
# Запуск потоку так як threading.Timer це підклас threading.Thread
timer.start()
#
```

Рис. 3.2 Приклад роботи `Timer`

## Замки

Найпростіший замок може бути реалізований на основі класу `Lock` модуля `threading`. Замок має два стани: він може бути або відкритий, або замкнений. У останньому випадку ним володіє певний потік. Об'єкт класу `Lock` має наступні методи:

- `acquire([blocking=True])` Робить запит на замикання замка. Якщо параметр `blocking` не вказано або є істиною, то потік буде очікувати звільнення замка. Якщо параметр не було встановлено, метод не поверне значення. Якщо `blocking` був заданий і правдивий, метод поверне `True` (після успішного оволодіння замком). Якщо блокування не потрібне (тобто заданий `blocking=False`), метод поверне `True`, якщо замок не був замкнений і ним успішно опанував цей потік. В іншому випадку буде повернуто `False`.



• `release()` Запит на відмикання замка.  
 • `locked()` Повертає поточний стан замку (`True` - замкнений, `False` - відкритий).

Слід мати на увазі, що навіть якщо стан замку щойно перевірено, це не означає, що він збереже цей стан до наступної команди.

Є ще один варіант замку – `threading.RLock`, який відрізняється від `threading.Lock` тим, що деякий потік може вимагати його замикання багато разів.

Відмикання такого замку має відбуватися стільки ж разів, скільки було замикань.

Це може бути корисним, наприклад, всередині рекурсивних функцій.

### Коли потрібні замки?

Замки дозволяють обмежувати вхід у деяку область програми одним потоком.

Замки можуть бути потрібні для забезпечення цілісності структури даних. Наприклад, якщо для коректної роботи програми потрібне додавання певного елемента відразу в кілька списків або словників, такі операції в багатопотоковому

додатку слід обставити замками. Навколо атомарних операцій над вбудованими

типами замки ставити необов'язково. Наприклад, метод `append()` (вбудованого) списку є атомарною операцією, а той самий метод, реалізований користувальницьким класом, може вимагати блокувань. У разі сумнівів, звісно,

краще перестраховатися та поставити замки, проте слід мінімізувати загальне час

дії замку, оскільки замок зупиняє інші потоки, які намагаються потрапити до тієї ж області програми. Відсутність замку в критичній частині програми, працює над спільними для двох і більше потоків ресурсами, може призвести до випадкових помилок які складно знайти.

## Тупикова ситуація (deadlock)

Замки використовуються для керування доступом до ресурсу, який не можна використовувати спільно. У програмі таких ресурсів може бути кілька. При роботі із замками важливо добре продумати, чи не зайде виконання програми в глухий кут (deadlock) через те, що двом потокам будуть потрібні одні і ті ж ресурси, але не той, жоден не зможуть їх отримати, оскільки вони вже отримали замки. Існують методики, що дозволяють уникнути подібних тупиків.

## Події

Ще одним способом комунікації між об'єктами є події. Примірники класу `threading.Event` можуть бути використані для передачі інформації про настання деякої події від одного потоку одному або кільком іншим потокам. Об'єкти-події мають внутрішній прапор, який може перебувати у встановленому чи скинутому стані. При своєму створенні прапор події перебуває у скинутому стані. Якщо прапор у встановленому стані, очікування не відбувається: потік, що викликав метод `wait()` для очікування події, просто продовжує свою роботу. Нижче наведено методи екземплярів класу `threading.Event`:

- `set()` Встановлює внутрішній прапор, що сигналізує про настання події. Всі потоки, що чекають даної події, виходять зі стану очікування.
- `clear()` Скидає прапор. Усі події, які викликають метод `wait()` цього об'єкта-події, будуть перебувати у стані очікування доти, доки прапор скинутий, або після заданого таймауту.
- `isSet()` Повертає стан прапора.
- `wait([timeout])` Переводить потік у стан очікування, якщо прапор скинуто, і відразу повертається, якщо прапор встановлений. Аргумент `timeout` задає таймаут в секундах, після якого очікування припиняється, навіть якщо подія не настала.

## УМОВИ

Більш складним механізмом комунікації між потоками є механізм умов. Умови подаються у вигляді екземплярів класу `threading.Condition` та, подібно до щойно розглянутих подій, сповіщають потоки про зміну деякого стану.

Конструктор класу `threading.Condition` приймає необов'язковий параметр, що задає замок класу `threading.Lock` або `threading.RLock`. За замовчуванням створюється новий екземпляр замку класу `threading.RLock`.

Методи об'єкта-умови описані нижче:

- `acquire(...)` Запитує замок. Фактично викликається однопоточний метод об'єкта-замка, що належить об'єкту-умові.

- `release()` Знімає замок.

- `wait([timeout])` Переводить потік у режим очікування. Цей метод може бути викликаний тільки в тому випадку, якщо потік, що викликає його, отримав замок. Метод знімає замок та блокує потік до появи оголошень, тобто викликів методів `notify()` та `notifyAll()` іншими потоками. Необов'язковий аргумент `timeout` задає тайм очікування в секундах. При виході з очікування потік знову просить замок і повертається з методу `wait()`.

- `notify()` Виводить із режиму очікування один із потоків, що очікують дані умови. Метод можна викликати, тільки оволодівши замком, асоційованим з умовою. Документація попереджає, що в майбутніх реалізаціях модуля цілей оптимізації цей метод перериватиме очікування відразу кількох потоків. Сам собою метод `notify()` не призводить до продовження виконання очікуваних умови потоків, тому що цьому перешкоджає зайнятий замок. Потоки отримують керування тільки після зняття замка потоком, що викликав метод `notify()`.

- `notifyAll()` Цей метод аналогічний методу `notify()`, але перериває очікування всіх, хто чекає виконання умови потоків.

## Модуль `thread`

Порівняно з модулем `threading`, модуль `thread` надає низькорівневий доступ до потоків. Багато функцій модуля `threading`, який розглядався до цього, реалізований на базі модуля `thread`. Тут варто зробити деякі зауваження щодо застосування потоків взагалі. Документація з Python попереджає, що використання потоків має особливості:

- Виняток `KeyboardInterrupt` (переривання від клавіатури) може бути отримано будь-яким із потоків, якщо у поставці Python немає модуля `signal` (для обробки сигналів).
- Не всі вбудовані функції, які блокуються очікуванням введення, дозволяють іншим потокам працювати. Щоправда, основні функції на кшталт `time.sleep()`, `select.select()`, метод `read()` файлових об'єктів не блокують інші потоки.

- Неможливо перервати метод `acquire()`, оскільки виняток `KeyboardInterrupt` порушується лише після повернення з цього методу.

- Небажано, щоб головний потік завершувався раніше за інші потоки, оскільки не будуть виконані необхідні деструктори та навіть частини `finally` в операторах `try-finally`. Це з тим, що майже всі операційні системи завершують додаток, у якого завершився головний потік.

**GDAL** (англ. Geospatial Data Abstraction Library — бібліотека абстракції геопросторових даних) — бібліотека для читання та запису растрових та векторних геопросторових форматів даних, що випускається під Open Source ліцензією X/MIT організацією Open Source Geospatial Foundation (OSGeo). Бібліотека надає єдиним абстрактним моделям даних для всіх підтримуваних форматів. Під час складання можна також увімкнути додаткові утиліти. За допомогою цих утиліт можна виконувати конвертацію та обробку даних, використовуючи інтерфейс командного рядка.



Супутня бібліотека OGR, що є частиною дерева вихідних кодів GDAL, надає схожі можливості векторних даних.

Пакет GDAL/OGR вважається важливим проектом у Open Source, а також у комерційних колах ГІС у зв'язку з широким поширенням та великим набором функціональності.

**PyCharm** - інтегроване середовище розробки для мови програмування Python. Надає засоби для аналізу коду, графічний оглядчик, інструмент для запуску юніт-тестів і підтримує веб-розробку на Django.

**SQLite** - компактна вбудована реляційна база даних. Слово «вбудовується» означає, що SQLite не використовує парадигму клієнт-сервер. Тобто движок SQLite не є окремо працюючим процесом, з яким взаємодіє програма, а надає бібліотеку, з якої програма компонується і движок стає складовою частиною програми. Таким

чином, в якості протоколу обміну використовуються виклики функцій (API) бібліотеки SQLite. Такий підхід зменшує накладні витрати, час відлику і спрощує програму. SQLite зберігає всю базу даних (включаючи визначення, таблиці, індекси і дані) в єдиному стандартному файлі на тому комп'ютері, на якому виконується

програма.

**Створення інформаційної бази**  
 Створюємо базу даних для зберігання результатів обробки статистичних показників. На рис. 3.3 зображено фрагмент коду для створення бази даних.

```
conn = sqlite3.connect('database.db')
cursor = conn.cursor()
cursor.execute("""CREATE TABLE IF NOT EXISTS meta_info
                (date date, id text, filename text, cloud_coverage real,
                avg_ndvi real, AvgLinearDeviation real, AvgQuadraticDeviation real,
                ConfidenceIntUpperLimit real, ConfidenceIntLowerLimit real)""")
```

Рис. 3.3 Створення бази даних

	date	id	filename	cloud_coverage	avg_ndvi	AvgLinearDeviation	AvgQuadraticDeviation	ConfidenceIntUpperLimit	ConfidenceIntLowerLimit
31	2018-01-08	22r4	08012018_22...	0.023	0.1587	0.0947774940230001	0.307859438742748	0.762104499935787	-0.4444704499935787
32	2018-01-08	29r2	08012018_29...	0.0	0.1806	0.1265658303	0.355760917330727	0.877891397968224	-0.516691397968225
33	2018-01-08	30r2	08012018_30...	0.0	0.3466	0.0617619481760001	0.248519512666511	0.833688244826362	-0.140498244826361
34	2018-01-08	31r2	08012018_31...	0.0	0.1916	0.215823814032	0.46456841695492	1.10215409723164	-0.718954097231643
35	2018-01-08	32r2	08012018_32...	0.0	0.3025	0.0708997897500001	0.266270133041616	0.824389460761568	-0.219389460761568
36	2018-01-08	33r2	08012018_33...	0.0	0.614	0.1761484748	0.419700458422432	1.43661289850797	-0.208612898507988
37	2018-01-08	34r2	08012018_34...	0.0	0.4275	0.0753554443750001	0.274509461357892	0.985538544261469	-0.110538544261469
38	2018-01-08	35r2	08012018_35...	0.0	0.1558	0.190183329776	0.43610013732628	1.01055626915951	-0.698956269159509
39	2018-01-08	36r2	08012018_36...	0.0	0.455	0.0907204000000001	0.301198273567429	1.04534861619216	-0.135348616192162
40	2018-01-08	37r2	08012018_37...	0.0	0.5313	0.14683690015	0.383193032491459	1.28235834368326	-0.219758343683259
41	2018-01-08	38r2	08012018_38...	0.0	0.3923	0.0582078690650001	0.241263070246982	0.865175617684084	-0.0805756176840843
42	2018-01-08	39r2	08012018_39...	0.0	0.2252	0.2499867908	0.499986790625513	1.20517410962601	-0.754774109626005
43	2018-01-08	40r2	08012018_40...	0.0	0.2013	0.169142879	0.41126983595785	1.00738887855174	-0.604788878551739
44	2018-01-08	41r2	08012018_41...	0.0	0.5083	0.151961091984	0.389821872121101	1.27235086935736	-0.255750869357358
45	2018-01-08	42r2	08012018_42...	0.0	0.4556	0.0639854634240001	0.252953480750908	0.95138882227178	-0.04018882227178
46	2018-01-08	66r2	08012018_66...	0.0	0.5322	0.151456809144	0.389174522732411	1.294982064555526	-0.230582064555526
47	2018-01-08	1031r1	08012018_10...	0.0	0.1569	0.211186103096	0.459549891846359	1.05761778801886	-0.743817788018864
48	2018-01-08	1032r1	08012018_10...	0.0	0.2347	0.169902826635	0.412192705703282	1.04258730317845	-0.573192703178451
49	2018-01-08	11r2	08012018_11...	0.8529	0.1345	0.048170727750001	0.219479308210629	0.564677484092834	-0.295677484092834
50	2018-01-08	12r2	08012018_12...	0.3555	0.1494	0.118830373056	0.344717816563055	0.825046920463588	-0.526246920463588
51	2018-01-08	19r2	08012018_19...	0.0	0.431	0.0830098893300002	0.288114358024726	0.995704141728464	-0.133704141728464

Рис. 3.4 Таблиця мета дані

На рис. 3.4 зображено створену таблицю з наступними полями:

- Data – дата знімку
- ID – ідентифікаційний код знімку
- Filename – назва файлу
- Cloud coverage - Хмарне покриття
- Avg NDVI - середнє зважене значення індексу
- Avg Linear Deviation – Лінійне відхилення

• Avg Quadratic Deviation - середньоквадратичне відхилення  
 • Confidence Int Upper limit - верхній довірчий інтервал  
 • Confidence Int Lower limit - нижній довірчий інтервал

## 3.3 Алгоритм роботи Графічний опис алгоритму

Для графічного зображення алгоритму обробки я обрав діаграму послідовності. Не зовсім блок-схема, звісно. Менше можливостей описувати цикли та розгалуження кожного "потoku" більше можливостей описувати взаємодію між ними. Діаграма послідовності зображено на рис 3.5 Діаграма послідовності.

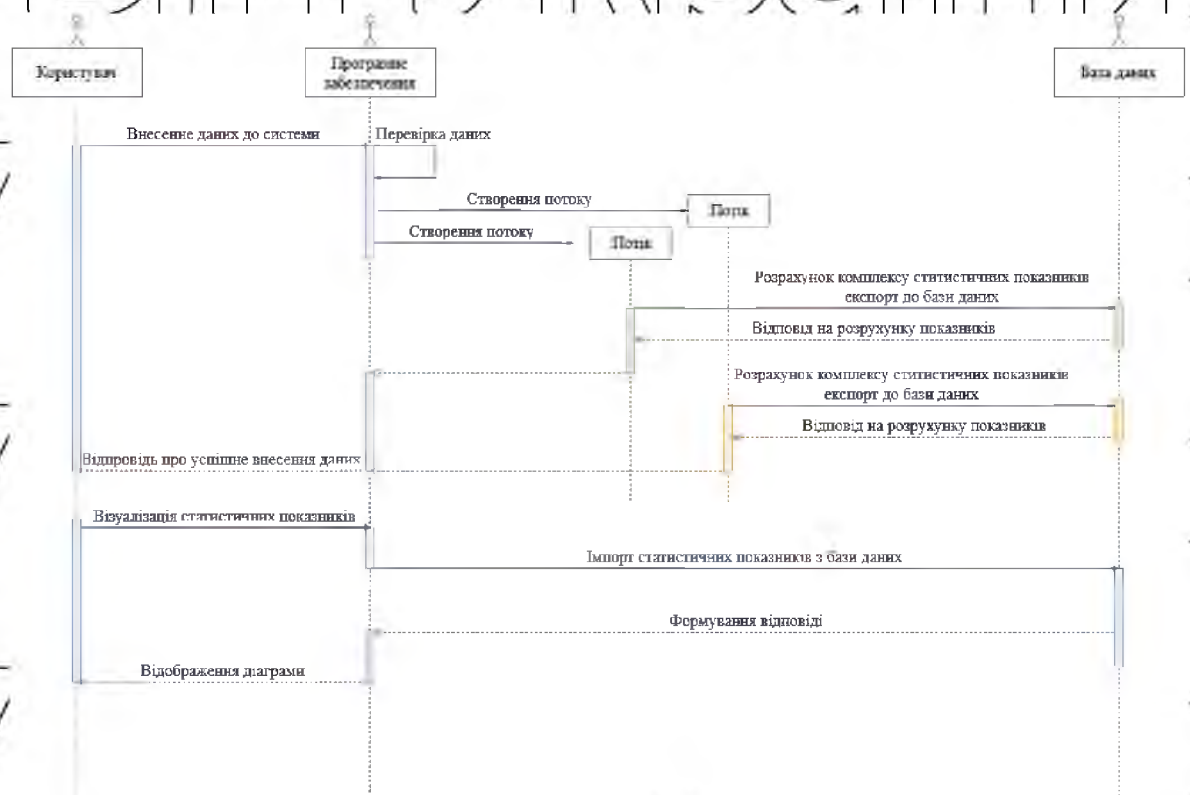


Рис. 3.5 Діаграма послідовності





НУ

НУ

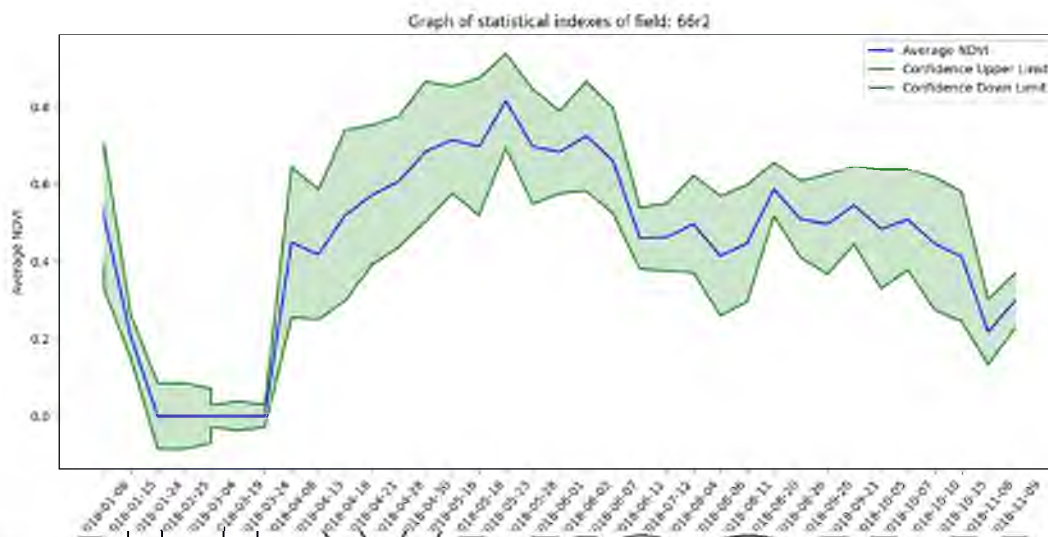


Рис. 3.8 результати роботи функції візуалізація статистичних показників

Функція візуалізація чистоти даних допомагає користувачеві зрозуміти

наскільки були захмарені знімки, частково захмарені або повністю чистими (рис. 3.9 Результати роботи функції візуалізація чистоти даних).

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України



Graph of cloudiness of field: 66r2

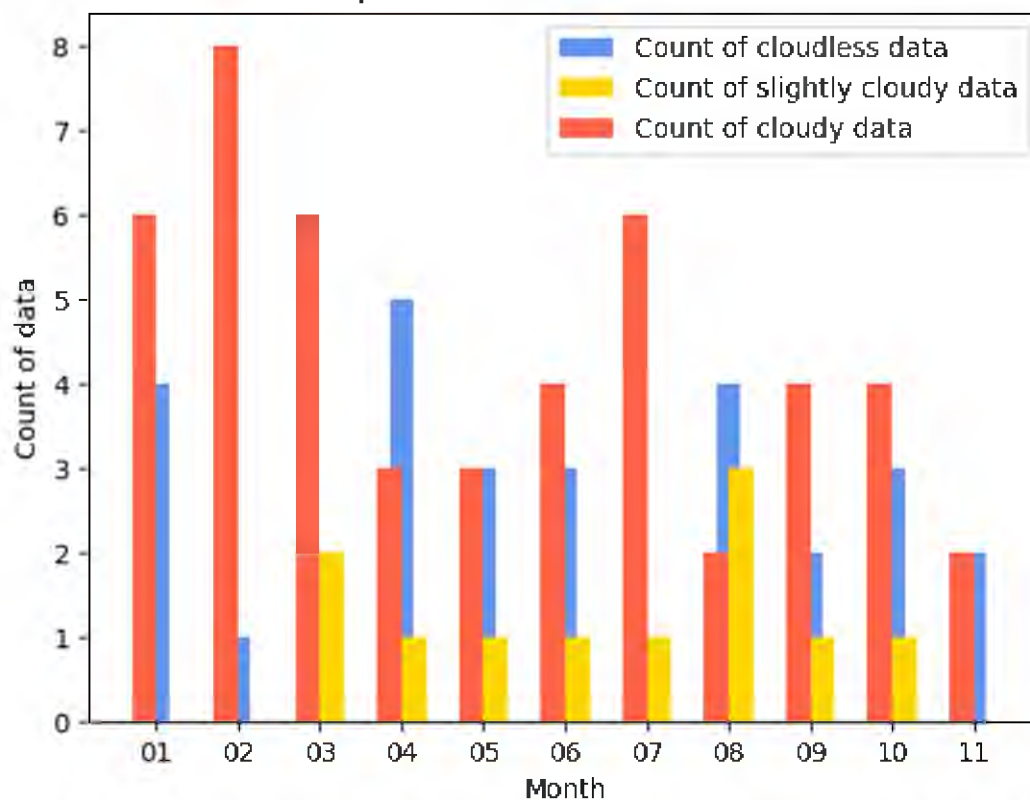


Рис. 3.9 Результати роботи функції візуалізація чистоти даних

НУБІП України

НУБІП України

НУБІП України

## 4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

### 4.1 Реалізація програмного коду

Методів, принципів та алгоритмів обробки зображень є чимала кількість. У даному випадку ми будемо оброблювати супутникові зображення для розрахунку вегетаційного індексу NDVI (рис. 4.1).

$$NDVI = \frac{NIR - RED}{NIR + RED}$$

Рис. 4.1 Формула вегетаційного індексу NDVI

NDVI (Нормалізований вегетаційний індекс, Нормалізований вегетаційний індекс) – це числовий показник якості та кількості рослинності на ділянці поля. Він розраховується за супутниковими знімками і від того, як рослини відбивають та поглинають світлові хвилі різної довжини.

Об'єктами дослідження виступають супутникові зображення наступного вигляду (рис. 4.2):



Рис. 4.2 Приклад супутникового зображення

Також деяка інформація зберігається у файлах типу JSON.

Реалізація програмного коду з обробки супутникових зображень (рис. 4.3).

НУБ

```

for root, dirs, files in os.walk(path):
    for file in files:
        if file.endswith("meta.json"):
            actual_dir = root.replace("\\", "/")
            file = '{}/{}'.format(actual_dir, file)
            with open('{}{}'.format(actual_dir, file), 'r', encoding="utf-8") as json:
                data = json.load(json)
                avg = 0
                res = 0
            file_path = os.path.basename(file)
            filename = os.path.basename(file).split('.')
            date = filename[0][4:] + '-' + filename[0][2:4] + '-' + filename[0][:2]
            try:
                avg = data['avg_ndvi']
                cloud_coverage = data['cloud_coverage']
                count = 0.01
                for i in data['by_1_100'].values():
                    if count > 1.01:
                        break
                    res = res + (((count - avg) ** 2) * i)
                    count += 0.01
                res2 = res ** (1 / 2)
                interval_d = avg - res2 * 1.96
                interval_u = avg + res2 * 1.96
                cursor.execute("Select * from meta_info where date == ? and id == ? and "
                               "filename == ? and cloud_coverage == ? and avg_ndvi == ? and "
                               "AvgLinearDeviation == ? and AvgQuadraticDeviation == ? and "
                               "ConfidenceIntUpperLimit == ? and ConfidenceIntLowerLimit == ? ",
                               (
                                   date, filename[1], file_path, cloud_coverage, avg, res, res2,
                                   interval_u, interval_d))
            except:
                conn.rollback()

```



НУБ



НУБ



НУБ

Рис. 4.3 Програмний код розрахунку статистичних показників

НУБІП України

НУБІП України

НУБІП України



Реалізація обробки використовуючи багатопотоковість (рис. 4.4, рис. 4.5):

```

import sqlite3
import json
import os
import time
import threading

part1 = r'geo1'
part2 = r'geo2'
part3 = r'geo3'
part4 = r'geo4'

def imageProcess(path, thread):...

start = time.perf_counter()

t1 = threading.Thread(target=imageProcess, args=[part1, '1'])
t2 = threading.Thread(target=imageProcess, args=[part2, '2'])
t3 = threading.Thread(target=imageProcess, args=[part3, '3'])
t4 = threading.Thread(target=imageProcess, args=[part4, '4'])

t1.start()
t2.start()
t3.start()
t4.start()

t1.join()
t2.join()
t3.join()
t4.join()

end = time.perf_counter()
print(f'Finished in {round(end - start, 2)} second(s)')

```

Рис. 4.4 Багатопотокова обробка

НУБІП Україні

```

НУБЕ import sqlite3
НУБЕ import json
НУБЕ import os
НУБЕ import time
НУБЕ import threading
НУБЕ from concurrent.futures import ThreadPoolExecutor

```

И

```

НУБЕ part1 = r'geo1'
НУБЕ part2 = r'geo2'
НУБЕ part3 = r'geo3'
НУБЕ part4 = r'geo4'

```

И

```

НУБЕ paths = [r'geo1', r'geo2", r'geo3', r'geo4']
НУБЕ def imageProcess(path):...
НУБЕ     start = time.perf_counter()

```

И

```

НУБЕ result = []
НУБЕ with ThreadPoolExecutor(max_workers=4) as exe:
НУБЕ     result = exe.map(imageProcess, paths)

```

И

```

НУБЕ end = time.perf_counter()
НУБЕ print(f'Finished in {round(end - start, 2)} second(s)')

```

И

Рис. 4.5 Багатопотокова обробка реалізація ThreadPool

НУБІП Українни

НУБІП Українни



## 4.2 Результат використання багатопотокового режиму в обробці супутникового зображення

На рис. 4.6 зображено графік результатів тестування.



Рис. 4.6 Графік результатів тестування

Як ми можемо бачити з результатів, пришвидшити обробку супутникових знімків нам не вдалося. Це залежно від кількості потоків швидкість варується від 4.5 хвилин до 6.2 хвилин. Причина по якій ми не змогли збільшити швидкість обробки полягає в GIL.

## 4.3 GIL - Global Interpreter Lock

Python Global Interpreter Lock (GIL) – це своєрідне блокування, яке дозволяє лише одному потоку керувати інтерпретатором Python. Це означає, що у будь-який час буде виконуватися лише один конкретний потік.

Робота GIL може здаватися несуттєвою для розробників, які створюють однопотоківі програми. Але в багатопотокових програмах відсутність GIL може негативно позначатися на продуктивності процесорно-залежних програм

Оскільки GIL дозволяє працювати тільки одному потоку навіть у багатопотоковому додатку, він отримав репутацію «сумно відомі» функції.

### Що за проблему в Python вирішує GIL?

Python підраховує кількість посилань для коректного керування пам'яттю. Це означає, що створені Python об'єкти мають змінну підрахунку посилань, в якій зберігається кількість всіх посилань на цей об'єкт. Як тільки ця змінна стає рівною нулю, пам'ять, виділена під цей об'єкт, звільняється.

Ось невеликий приклад коду, що демонструє роботу змінних підрахунку посилань (рис. 4.7 Приклад підрахунку посилань):

```
import sys

a = []
b = a

print(sys.getrefcount(a))
```

Рис. 4.7 Приклад підрахунку посилань

У цьому прикладі кількість посилань на порожній масив дорівнює 3. На цей масив посилаються: змінна `a`, змінна `b` і аргумент, переданий функції `sys.getrefcount()`.

Проблема, яку вирішує GIL, пов'язана з тим, що в багатопотоковому додатку відразу кілька потоків можуть збільшувати або зменшувати значення цього лічильника посилань. Це може призвести до того, що пам'ять очиститься неправильно і видалятиметься той об'єкт, на який ще існує посилання.

Лічильник посилань можна захистити, додавши блокування на всі структури даних, які поширюються кількома потоками. У такому випадку лічильник змінюватиметься виключно послідовно.

Але додавання блокування до кількох об'єктів може призвести до появи іншої проблеми - взаємоблокування (англ. *deadlocks*), яке виходить тільки якщо

блокування є більш ніж на одному об'єкті. До того ж, ця проблема теж знижувала б продуктивність через багаторазову установку блокування.

GIL - це одиночний блокувальник самого інтерпретатора Python. Він додає правило: будь-яке виконання байткоду в Python вимагає блокування інтерпретатора. У такому випадку можна виключити взаємоблокування, тому що GIL буде єдиним блокуванням у додатку. До того ж, його вплив на продуктивність процесора зовсім не критичний. Однак варто пам'ятати, що GIL впевнено робить будь-яку програму однопоточною.

Незважаючи на те, що GIL використовується і в інших інтерпретаторах, наприклад Ruby, він не є єдиним рішенням цієї проблеми. Деякі мови вирішують проблему потокобезпечного звільнення пам'яті за допомогою збирачів сміття.

З іншого боку, це означає, що такі мови часто повинні компенсувати втрату однопоточних переваг GIL додаванням якихось додаткових функцій підвищення продуктивності.

### **Чому для вирішення проблеми було обрано саме GIL?**

Отже, чому ж це не дуже гарне рішення використовується в Python? Наскільки для розробників це рішення є критичним?

Архітектурне рішення GIL це одна з тих речей, які зробили Python популярним.

Python існує з тих часів, коли в операційних системах не було поняття про потоки. Ця мова розроблялася у розрахунок на легке використання та прискорення процесу розробки. Дедалі більше розробників переходило на Python.

Багато розширень, яких потребував Python, було написано для вже існуючих бібліотек на C. Для запобігання незгодженим змінам, мова C вимагала потокобезпечного управління пам'яттю, яку змогла надати GIL.

Python. Він збільшував продуктивність однопотоків додатків, оскільки управління велось лише одним блокувальником.

Ті бібліотеки на C, які не були потокобезпечними, стало легше інтегрувати.

Ці розширення на C стали однією з причин, чому Python-спільнота стала розширюватися.

Як можна зрозуміти, GIL - фактичне вирішення проблеми, з якою зіткнулися розробники CPython на початку життя Python.

### Вплив GIL на багатопотокові програми

Якщо дивитися на типову програму (не обов'язково написану на Python) — є різниця, чи ця програма обмежена продуктивністю процесора або I/O.

Операції, обмежені продуктивністю процесора (англ. CPU-bound) — це обчислювальні операції: перемноження матриць, пошук, обробка зображень тощо.

Операції, обмежені продуктивністю I/O (англ. I/O-bound) - це операції, які часто перебувають в очікуванні чогось від джерел вводу/виводу (користувач, файл,

БД, мережа). Такі програми та операції іноді можуть чекати довгий час, доки не отримають від джерела те, що їм потрібно. Де з тим, що джерело може проводити власні (внутрішні) операції, як він буде готовий видати результат.

GIL не сильно впливає на продуктивність I/O-операцій у багатопотокових програмах, тому що в процесі очікування від I/O блокування поширюється потоками.

Однак програма, потоки якої будуть працювати виключно з процесором (наприклад обробка зображення частинами), через блокування не тільки стане однопотоковою, але і на її виконання буде витрачатися більше часу, ніж якщо вона спочатку була строго однопотоковою.

Таке збільшення часу — це результат появи та реалізації блокування.

## ВИСНОВКИ

В даній роботі було проведено системний аналіз предметної області: короткий опис поняття дистанційного зондування Землі та історія ДЗЗ, систем дистанційного зондування за допомогою яких можливо отримати зображення

Землі, ресурси, що надають дані ДЗЗ а також аналіз методів обробки даних ДЗЗ. Було описано проблему обробки великої кількості даних і запропоновано підхід до її вирішення, а саме використання багатопотокового режиму при обробці супутникових зображень.

За проведеним дослідження було виявлено, що використання багатопотокового режим не є ефективним рішенням для вирішення даної проблеми через особливості роботи з потоками в мові програмування Python.

НУБІП України

НУБІП України

НУБІП України

НУБІП України



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кохан С.С., Востоков А.Б. Дистанційне зондування Землі : теоретичні основи : підруч. — К. : Вища школа, 2009. — С. 16.

2. Схема получения и обработки данных ДЗЗ [Электронный ресурс] — Режим доступу до ресурсу: [http://files.school-collection.edu.ru/dlrstore/b82\\_2b11f-492c-4270-b01e-377de68b3f1d/files/9.htm](http://files.school-collection.edu.ru/dlrstore/b82_2b11f-492c-4270-b01e-377de68b3f1d/files/9.htm).

3. Класифікація способів отримання зображень [Электронный ресурс] — Режим доступу до ресурсу: <https://helpiks.org/6-22680.html>.

4. Дрони і супутники: моніторинг стану посівів впродовж сезону [Электронный ресурс]. — 2017. — Режим доступу до ресурсу: <https://smartfarming.ua/ua-blog/monitoring-sostoyaniya-posevov-vtechenie-sezona>.

5. Аерокосмічні методи [Электронный ресурс] — Режим доступу до ресурсу: <http://ecology.udau.edu.ua/assets/files/024.pdf>.

6. Вовчик Е.Б. Що таке ПГС/І як їх спостерігають [Электронный ресурс] / Е.Б. Вовчик. — 2015. — Режим доступу до ресурсу: <http://astro.lnu.edu.ua/wp-content/uploads/2015/04/vstup-do-praktyky.pdf>

7. Гарбук С.В., Гершензон В.Е. Космические системы дистанционного зондирования Земли. - М.: Издательство АшБ, 1997.

8. Капкин В.Б., Сухинин А.И. Дистанционное зондирование Земли из космоса. Цифровая обработка изображений: Учебное пособие. - М.: Логос, 2001.

9. Киенко Ю.П. Основы космического природоведения: Учебник для вузов. - М.: «Картоцентр» – «Геодезиздаг», 1999. - 285 с.

10. Кравцова В.И. Космические методы картографирования / Под редакцией Ю.Ф.Книжникова. - М.: Изд-во МГУ, 1995.

11. Кравцова В.И. Космические методы исследования почв / Учебное пособие для студентов вузов. - М.: Аспект Пресс, 2005. - 190 с.

12. Кронберг П. Дистанционное изучение Земли. Основы и методы дистанционных исследований в геологии. - М.: Мир, 1988.

13. Петрусевич М.Н. Аэрометоды при геологических исследованиях / Под редакцией А.А.Богданова. - М.: Госгеолтехиздат, 1961.

14. Савиных В.П., Кучко А.С., Стеценко А.Ф. Аэрокосмическая фотосъемка. Учебник. - М.: Картогеоцентр-Геодезиздат, 1997.

15. Інформаційні системи і технології на підприємствах [Електронний ресурс] // сайт “Навчальні ресурси СумДУ”. - Режим доступу:

[http://elearning.sumdu.edu.ua/free\\_content/lectured\\_de1c9452f2a161439391120eef364d88ce4f8e5e720160217112601200847/index.html](http://elearning.sumdu.edu.ua/free_content/lectured_de1c9452f2a161439391120eef364d88ce4f8e5e720160217112601200847/index.html)

16. Палмер С. Функціонально-орієнтоване відслідкування [Електронний ресурс] : навч. посіб. / С. Палмер, Р. Фелсинг, М. Джон // Практичний посібник по

функціонально-орієнтованій розробці - 2002. - С. 304. - Режим доступу до навч. посіб.

<https://books.google.com.ua/books?id=3MU9WAC9MIMC&pg=PA22&lpg=PA22&dq=fdd>

17. Функціональне моделювання [Електронний ресурс] // Сайт “Посібники статті та книги для студентів”. - Режим доступу: <http://posibniki.com.ua/posibniki/funkcionalne-modelyuvannya>

18. Діаграма прецедентів. Роль прецедентів при розробці ПС. Види прецедентів і відносини між ними. Правила опису [Електронний ресурс] // сайт

“Учебні Матеріали для студентів і школярів України”. - Режим доступу: <https://studopedia.net/5-71809-diyagrami-precedenti-v-Use-case-diagram.html>

19. Автоматизоване проектування інформаційних систем [Електронний ресурс] - Режим доступу до ресурсу: <http://posibniki.com.ua/catalog-avtomatizovane-proektuvannya-informaciynih-sistem---denisova-o-o>



20. Основні поняття ER-діаграм сайт [Електронний ресурс] // “easycode”. – Режим доступу: <http://easy-code.com.ua/2012/09/elementi-modeli-sutnist-zvyazok-integraciya-dodatkov-i-danix-bazi-danix-stati/>

21. Розробка ER-моделі предметної області сайт [Електронний ресурс] // –

Режим доступу: [https://web.pos.bivkv.vntu.edu.ua/fik/10savchuk-organizaciya-bazdanih-znan/gl\\_26.html](https://web.pos.bivkv.vntu.edu.ua/fik/10savchuk-organizaciya-bazdanih-znan/gl_26.html)

22. Уніфікована мова моделювання [Електронний ресурс] // сайт бібліотеки

МГПУ ім. Богдана Хмельницького. Режим доступу: <http://um.co.ua/8/8-13/8-132563.html>

23. Діаграма прецедентів. Роль прецедентів при розробці ПС. Види прецедентів і відносини між ними. Правила опису [Електронний ресурс] // сайт

“Учбові Матеріали для студентів і школярів України”. Режим доступу: <http://um.co.ua/8/8-2/8-213194.html>

24. ВЕРИФІКАЦІЯ І ВАЛІДАЦІЯ В ТЕСТУВАННІ [Електронний ресурс] // – Режим доступу: <https://training.qatestlab.com/blog/technical-articles/verification-validation-testing/>

25. Поняття якості та культури програмного забезпечення – StudFiles [Електронний ресурс] Режим доступу: <https://studfiles.net/preview/5851327/page:8/>

26. NDVI – теорія і практика. [Електронний ресурс]. URL: <https://gis-lab.info/qa/ndvi.html>

27. Вегетаційний індекс. [Електронний ресурс]. URL: <https://gis-lab.info/qa/vi.html>

28. Дистанционное зондирование Земли из космоса. Цифровая обработка изображений Кашкин В.Б., Сухинин А.И., Логос, Москва, 2001 г.

29. Черепанов А.С. Спектральные свойства растительности и вегетационные индексы А.С. Черепанов, Е.Г. Дружинина // Геоинформатика, 2009. – № 3 – С. 28–32.

30. Документація Python: [Електронний ресурс] – Режим доступу:

<https://www.python.org/doc/>

31. Михайлов С.И. Применение данных дистанционного зондирования Земли для решения задач в области сельскохозяйственного производства // Земля из космоса. – 2011. – Выпуск 9. – С. 17-23. 2

32. Чандра А.М., Гош С.К. Дистанционное зондирование и географические информационные системы. / Москва Техносфера, 2008. – 312 с.

33. Плуталова, Т. Г. Мониторинг сельскохозяйственных культур сухостепной зоны Кулундинской равнины на основе данных спутниковой группировки RapidEye [Текст] / Т. Г. Плуталова, Н. М. Ковалевская // Дистанционное зондирование Земли из космоса: алгоритмы, технологии, данные. Материалы молодежной школы-семинара / Под ред. А. А. Лагутина. – Барнаул: АЗБУКА, 2013. – С. 16–23. 43

34. Посевные площади, валовые сборы и урожайность сельскохозяйственных культур в 2015 году (предварительные данные) [Электронный ресурс] / Федеральная служба государственной статистики. – Режим доступа : [http://www.gks.ru/free\\_doc/doc\\_2015/bul\\_dr/sx/posev-2015.rar](http://www.gks.ru/free_doc/doc_2015/bul_dr/sx/posev-2015.rar).

35. Принципы агроландшафтного районирования пахотных земель Северного Казахстана по данным LANDSAT и MODIS [Текст] / А. Г. Терехов, И. С. Витковская, М. Ж. Батырбаева, Л. Ф. Сливак // Современные проблемы дистанционного зондирования Земли из космоса. – 2010. – Т. 7, № 3. – С. 292–304.

36. Прогнозирование урожайности зерновых и зернобобовых культур в центральных черноземных областях на основе комплексирования наземных и спутниковых данных [Текст] / А.И. Страшная, Л. Л. Тарасова, Н. А. Богомолова, Т.



А. Максимова, О. В. Береза // Труды Гидрометцентра России. 2015. – Вып. 353. – С. 128–153.

37. Лагутин, А. А. Дистанционное зондирование Земли из космоса: алгоритмы, технологии, данные [Текст] : учеб. пособие для слушателей

молодежной школы-семинара / А. А. Лагутин, Р. И. Райкин, Т. В. Чимитдоржиев. – Барнаул : Изд-во Алт. ун-та, 2013. – 151 с.

38. Леонтьев, А. А. Система спутникового мониторинга состояния полей и прогнозирования урожайности [Текст] / А. А. Леонтьев // Геоматика. – 2013. – № 2

(19). – С. 77–79.

39. Михайлов С. И. Применение данных дистанционного зондирования Земли для решения задач в области сельскохозяйственного производства [Текст] /

С. И. Михайлов // Земля из космоса. Наиболее эффективные решения. – Вып. 9. – Весна 2011. – С. 17–23.

40. Прасул Ю.І., Коланіщина К.М. Дистанційне зондування Землі. Робота з аеро- та космоснімками (матеріали до модуля 2). – Харків : Форт, 2009. – 51 с.

41. Кочуб Е.В., Топаз А.А. Анализ методов обработки материалов дистанционного зондирования земли // Вестник Полоцкого государственного университета. Серия F. – 2012. – С. 132–140.

42. Первинна обробка (нормалізація) даних ДЗЗ [Електронний ресурс]. – Режим доступу до ресурсу: <https://helpiks.org/6-84343.html>.

43. Системний аналіз та проектування ГІС. – Електронний навчальний посібник / Є.М. Крижановський, В.Б. Мокич, А.Р. Янцолт, Л.М. Скорина. –

Вінниця : ВНТУ, 2015. – 127 с.

44. Світличний О.О., Плотницький С. В. Основи геоінформатики : навч. посіб. / За заг. ред. О.О. Світличного. – Суми : ВТД «Університетська книга»,

2006. – С. 217–219.



45. GIS Implementation [Электронный ресурс] — Режим доступа до ресурсу: <http://gmtgis.com/gmtwp/services/gis-implementation>

46. Даджион Д., Мерсеро Р. Цифровая обработка многомерных сигналов. - М.: Мир, 1988.

47. Ярославский Л.П. Введение в цифровую обработку изображений. - М.: Сов. радио, 1979.

48. Прэтт У. Цифровая обработка изображений. Кн.1. - М.: Мир, 1982.

49. Методы передачи изображений. Сокращение избыточности / У.К.

Прэтт, Д.Д., Сакрисон, Х.Г., Д. Мусманн и др. Под ред. У.К.Прэтта. - М.: Радио и связь, 1983.

50. Джайн А.К. Сжатие видеоинформации // ТИИЭР. Т.69. - № 3

51. Горбунов Б.А., Дементьев В.Н., Пяткин В.П. Распознавание изображений в дистанционном зондировании // Автоматизированная обработка изображений природных комплексов Сибири. - Новосибирск: Наука, 1988.

52. Левин Б.Р. Теоретические основы статистической радиотехники. Кн.1. - М.: Сов. радио, 1966.

53. Ярославский Л.П. Введение в цифровую обработку изображений. - М.: Сов. радио, 1979.

54. Прэтт У. Цифровая обработка изображений. Кн.2. - М.: Мир, 1982.

55. Спектор А.А. Рекуррентная фильтрация гауссовских дискретных полей, наблюдаемых в гауссовских шумах // Радиотехника и электроника, 1994, №7.

56. Даджион Д., Мерсеро Р. Цифровая обработка многомерных сигналов. - М.: Мир, 1988.

57. Jones B. Vision Of The Future [Электронный ресурс] / Brad Jones // Futurism — 2018. — Режим доступа до ресурсу: <https://futurism.com/elon-musk-lidar-tech-autonomous-vehicles/>

58. Вчені розповіли, скільки тонн космічного сміття на орбіті Землі [Електронний ресурс]— 2018.— Режим доступу до ресурсу: <http://uagolos.com/vcheni-rozповіli-skilki-tonn-kosmichnogo-smittja-na-orbiti-zemli/>.

59. Новгородова Г. Про світло. Основи ДЗЗ [Електронний ресурс]/ Г. Новгородова//50north.— 2015.— Режим доступу до ресурсу: <http://www.50northspatial.org/ua/concerning-the-light/>.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України