

МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ  
ДЕРЖАВНИЙ НАУКОВО-ДОСЛІДНИЙ ІНСТИТУТ  
ЛУГАНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ВНУТРІШНІХ СПРАВ ІМЕНІ Е.О. ДІДОРЕНКА

## **Інформаційні технології в кримінальному аналізі**

**(практична частина)**



Надруковано за сприяння та міжнародної технічної допомоги  
Консультативної місії Європейського Союзу

УДК 343.1.85.(477)

Рекомендовано до друку:

Вченою радою Державного науково-дослідного інституту МВС України  
(протокол від 14 грудня 2022 року № 8)

Вченою радою Луганського державного університету  
внутрішніх справ ім. Е.О. Дідоренка  
(протокол від 28 липня 2022 року № 15)

Рецензенти:

**Микола КАРЧЕВСЬКИЙ**, доктор юридичних наук, професор, полковник поліції, перший проректор Луганського державного університету внутрішніх справ ім. Е.О. Дідоренка

**Карен ІСМАЙЛОВ**, кандидат юридичних наук, підполковник, начальник 5-го відділу (інформаційних технологій та програмування) в південному регіоні 3-го управління (інформаційних технологій та програмування) Департаменту кіберполіції Національної поліції України

**Користін О.Є., Тімошин А.С.** Інформаційні технології в кримінальному аналізі (практична частина): Навчальний посібник. 2022, 102 с.  
**ISBN 978-617-616-110-3**

У навчальному посібнику «Інформаційні технології в кримінальному аналізі (практична частина)» розглянуто практичні аспекти здійснення кримінального аналізу в межах інформаційно-аналітичного забезпечення правоохоронної діяльності, викладено ключові методологічні засади аналізу та візуалізації інформації, а також методика роботи з сучасним програмним забезпеченням.

Даний навчальний посібник призначений для практичних працівників Національної поліції України, викладачів, курсантів, та слухачів закладів вищої освіти й усіх, кого цікавить проблематика безпеки.

© Користін О. Є., Тімошин А. С.,  
ДНДІ МВС України,  
ЛДУВС ім. Е.О. Дідоренка, 2022

## ЗМІСТ

<b>ВСТУП</b> .....	4
<b>1. Структуровані дані</b> .....	6
1.1. Табличні формати .....	6
1.2. Радіотехнічна розвідка .....	7
Питання для самоперевірки .....	12
<b>2. Інструментарій для аналізу даних</b> .....	13
2.1. Імпорт даних в Excel .....	13
2.2. Імпорт даних в i2 Analyst's Notebook .....	16
2.3. Таблиця банківських транзакцій .....	30
2.3.1. Аналіз таблиці банківських транзакцій в Excel .....	30
2.3.2. Імпорт таблиці банківських транзакцій в i2 Analyst's Notebook .....	38
2.4. Таблиця трафіку з'єднань .....	43
2.4.1. Аналіз таблиці трафіку з'єднань в Excel .....	43
2.4.2. Імпорт таблиці трафіку з'єднань в i2 Analyst's Notebook .....	50
2.5. Аналіз даних за допомогою функцій Excel .....	52
Питання для самоперевірки .....	57
<b>3. Аналіз даних в Python</b> .....	58
3.1. Відомості про Python .....	58
3.2. Підготовка даних до імпорту в Python .....	74
3.3. Використання бібліотеки OPENPYEX .....	75
3.3.1. Аналіз банківських транзакцій .....	75
3.3.2. Аналіз трафіку з'єднань .....	85
3.4. Інтерактивна візуальна аналітика в Python .....	89
Питання для самоперевірки .....	93
Додаток А. Скорочений довідник функцій в Excel .....	95
Додаток Б. «Гарячі» клавіші в Excel .....	100
<b>СПИСОК ВИКОРИСТАНОЇ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ</b> .....	102

## ВСТУП

Застосування інформаційних технологій в кримінальному аналізі є невід’ємною частиною аналітичної роботи і передбачає досить широкий спектр задач і інструментів. Особливо це стосується аналітичної роботи з великими масивами даних.

Серед інструментів, які використовуються в кримінальному аналізі, найбільш розповсюдженими є додаток MS Excel та програма i2 Analyst's Notebook. Але, досить потужним доповненням до Excel може бути Python – високорівнева мова програмування загального призначення.

Більшість даних, які правоохоронці отримують для аналітичної роботи, пов’язані з структурованими табличними даними. Як правило, це файли формату txt, csv або xlsx. Прикладом табличних даних, які мають відношення до кримінального аналізу, є таблиця банківських транзакцій та таблиця трафіку з’єднань, аналіз яких ретельно розглянуто в посібнику.

Існують певні процедури при отриманні того або іншого набору даних для аналізу. Наприклад, отримання даних про трафік з’єднань на початку пов’язано з проведенням радіотехнічної розвідки. В розділі 1 надається інформація про порядок проведення радіотехнічної розвідки та прилади, за допомогою яких проводиться така розвідка. За результатами розвідки про місцезнаходження та параметри базових станцій мобільного зв’язку, оператори зв’язку надають слідчому відомості (у вигляді файлу) про всі з’єднання, які були зафіксовані на певній території за певний час. Також, в розділі 1, розкрито напрями використання отриманого аналізу трафіка. Додатково роз’яснюється, як можна визначити місцезнаходження мобільного телефону.

У розділі 2 викладено інструментарій аналізу даних, надано покрокові інструкції імпорту структурованих табличних даних в додаток Excel та i2 Analyst's Notebook. На прикладі вказаних таблиць, розглянуто аналіз даних в Excel за допомогою таблиць Excel, які передбачають фільтрацію даних, сортування, створення рядка підсумків та зведених таблиць. Треба зазначити, що в посібнику представлено чіткий блок завдань, які розкривають зміст аналізу банківських транзакцій та трафіку з’єднань. Наведено приклади виконання цих завдань, які містять необхідні пояснення та рисунки. Особлива увага приділена побудові Google карти з вказанням місцезнаходження базових станцій, і, відповідно, потенційних учасників події. Розглянуто функції Excel

для пошуку значень та підрахунку комірок за певними умовами, умовне підсумування.

З приводу застосування мови Python для аналізу структурованих табличних даних, на початку розділу 3 було розглянуто загальні відомості щодо редакторів коду на Python, а саме ILDE (Python 3.8.0 Shell), PyCharm, Spyder, Jupyter Notebook. Викладені відомості щодо синтаксису мови, функцій вводу та виводу, підключення модулів, операторів розгалуження та циклів. Суттєва інформація надана стосовно структур даних, а саме рядків, списків, кортежів, діапазонів, множин, словників. Представлена значна кількість прикладів щодо задання вказаних структур, виконання операцій з елементами структур, методів структур.

Для роботи з табличними даними в Python, а саме з файлами `xlsx` та таблицями в них, розглянута бібліотека `openpyxl`. Представлено фрагменти коду, які вказують на те, як здійснювати імпорт таблиці з Excel в Python. Показано, як за допомогою модуля `openpyxl` можна відшукувати в книзі `xlsx` листи за їх ім'ям або вмістом певних комірок, будувати нові книги, створювати листи в книгах. Детально описано створення такої структури, як словники, які дозволяють виконувати сортування та фільтрацію даних в Python. Вказані методи роботи з модулем `openpyxl` розглянуто виключно на прикладах таблиці банківських транзакцій та таблиці трафіку з'єднань. Для побудови географічних карт, в аспекті аналізу трафіка з'єднань, розглянута бібліотека `folium`. При аналізі даних засобами Python обговорюються ті елементи, які підкреслюють перевагу мови Python по відношенню к аналізу в Excel.

Матеріал посібника підготовлено у відповідності до програми навчальної дисципліни «Основи кримінального аналізу», яка викладається для здобувачів вищої освіти спеціальності 262 Правоохоронна діяльність та спеціальності 081 Право. Посібник може бути корисним також і практичним працівникам підрозділів Національної поліції, які здійснюють свою діяльність у сфері протидії організованій злочинності на підставі сучасного програмного забезпечення, засобів та методів кримінального аналізу.

# 1. Структуровані дані

## 1.1. Табличні формати

Велика кількість інформації, яка підлягає аналізу, подана текстовими файлами з структурованими даними. Це типи даних, які мають внутрішню структуру та можуть бути сконструйовані з простих типів даних.

Структурованими типами даних можуть бути одновимірні масиви (рядки, стовпці), багатовимірні масиви (матриці), множини (набори неупорядкованих унікальних значень), файли.

Прикладом структурованих даних є табличні формати. Текстовий табличний формат – це структура даних, яка представлена рядками з окремих полів даних (значень). Поля розділені спеціальними символами (delimiter). Зазвичай, роздільником може бути знак табуляції, кома, крапка з комою, пробіл. Додатково потрібно мати опис структури таблиці – назви та формат полів (текстовий, числовий, грошовий, дата і т.п.).

Табличні формати можуть зберігатися в текстових файлах формату txt, або csv. Такі файли можна відкрити і проглянути звичайним блокнотом (Notepad). Різниця між цими файлами (якщо там дані структуровані) незначна, і, частіше полягає в тому, що вони мають різні роздільники. Наприклад, якщо таблиця з Excel експортована в файл формату txt, то роздільником буде знак табуляції, а якщо експорт відбувся в файл типу csv, то роздільником буде крапка з комою (рис.1.1). Крім того, файл типу csv за замовчуванням відкривається додатком Excel.

Джерело	Призначення	Дата час	Сума коштів
01223854840	01429256054	01.01.2020 9:43	2345,00
01223854840	01364558175	01.01.2020 19:54	1365,00
01223854840	01914715106	01.01.2020 22:42	19785,00

```
Джерело;Призначення;Дата час;Сума коштів  
01223854840;01429256054;01.01.2020 9:43;2345,00  
01223854840;01364558175;01.01.2020 19:54;1365,00  
01223854840;01914715106;01.01.2020 22:42;19785,00
```

Рисунок 1.1

## 1.2. Радіотехнічна розвідка

В якості прикладу текстового файлу з структурованими даними розглянемо трафік з'єднань, який надається оператором мобільного зв'язку з приводу встановлення тих абонентів мобільного зв'язку, які знаходилися на місці вчинення злочину.

Позиціонування підозрюваних, потерпілих та свідків з використанням технічних засобів операторів мобільного зв'язку базується на періодичності подачі реєстраційних сигналів мобільним телефоном на базову станцію, фіксуючись в комп'ютерній базі операторів, що дозволяє, використовуючи технології визначати місцезнаходження абонента з точністю до 10 м в міських умовах та 800 м та більше в сільській місцевості.

Через те, що на початковому етапі досудового розслідування підозрюваний поки невідомий, запитавши елементи трафіку з'єднань, суб'єкти досудового провадження з боку обвинувачення отримують масив даних про проведені з'єднання абонентів за вказаний період часу в межах відповідної змісту вчинення кримінального правопорушення базової станції. Аналізуючи отримані дані, суб'єкти розслідування виокремлюють збіги за датою, часом та місцем, після чого відбувається відпрацювання конкретних осіб. Якість аналітичного матеріалу буде залежати від здатності слідчого виокремити номери абонентів на підставі визначених подією кримінального правопорушення критеріїв і може бути використана за такими напрямками:

- нейтралізація впливу на учасників кримінального провадження шляхом встановлення осіб, які його здійснюють, та запобігання їх діям;
- викриття дачі неправдивих показань та встановлення всіх учасників кримінального правопорушення. Наприклад, огляд протоколів деталізації з'єднань абонентських номерів дає можливість підтвердження: а) факту знайомства співучасників; б) регулярності спілкування, а саме з метою підготовки та вчинення злочинів; в) місця та часу вчинення епізодів кримінально-протиправної діяльності;
- подолання зайнятої позиції протидії під час допиту;
- виявлення та нейтралізація зв'язку затриманих підозрюваних (обвинувачених) зі співучасниками та іншими особами з метою здійснення протидії розслідуванню;
- викриття неправдивого алібі;
- встановлення всіх учасників організованої групи та здійснення протидії розслідуванню.

Для вирішення цієї задачі, працівниками поліції проводиться радіотехнічна розвідка на місці скоєння даного кримінального правопорушення, в ході якої встановлюються параметри базових станцій операторів рухомого (мобільного) зв'язку, які могли приймати сигнали телефонних викликів на момент злочину.

Радіотехнічна розвідка полягає в застосуванні технічного обладнання для локалізації місцезнаходження радіоелектронного засобу, у тому числі мобільного терміналу, системи зв'язку та інших радіо випромінювальних пристроїв, активованих у мережах операторів мобільного зв'язку (ст. 268 КПК, п. 1.14.2 Інструкція про проведення негласних слідчих (розшукових) дій та використання їх результатів у кримінальному провадженні).

Основні засади встановлення місцезнаходження радіообладнання (радіоелектронного засобу):

- може проводитися у кримінальному провадженні про злочини незалежно від їх тяжкості;

- підставою проведення є ухвала слідчого судді, в якій обов'язково зазначаються ідентифікаційні ознаки, які дозволяють унікально ідентифікувати абонента, мережу, кінцеве обладнання;

- захід можливо провести до постановлення ухвали слідчим суддею в порядку ст. 250 КПК України;

- метою є встановлення місцезнаходження радіообладнання (радіоелектронного засобу), маршруту його пересування, з'ясування технічних параметрів засобу зв'язку, розглядається як допоміжний захід під час організації та проведення інших НС(Р)Д;

- не передбачає розкриття змісту повідомлень, які передаються і, відповідно, втручання у приватне спілкування, під час проведення зазначеного заходу, не допускається.

До параметрів базової станції відносять:

MCC (Mobile Country Code) – код, що визначає країну, в якій знаходиться оператор мобільного зв'язку. Наприклад, для України він дорівнює 255, США – 310, Угорщина – 216, Китай – 460, Білорусь – 257. Росія – 250.

MNC (Mobile Network Code) – унікальний код, який присвоюється кожному оператору мобільного зв'язку в конкретній країні. Наприклад,

MCC	MNC	Бренд, оператор
255	01	Vodafone, ПрАТ «ВФ Україна»
255	02	Київстар, ПрАТ «Київстар»

LAC (Location Area Code) – код локальної зони (об'єднання деякої кількості базових станцій, які обслуговуються одним контролером базових станцій (BSC).

CID – ідентифікатор соти (сектор базової станції). Цей параметр представлений в десятковому вигляді.

Параметри MCC та MNC називають параметрами мережі оператора зв'язку. Параметри мережі оператора зв'язку як правило об'єднуються і позначаються в через PLMN. Наприклад, PLMN мережі Vodafone є 25501. Зрозуміло, що найбільш цікавими є параметри базових станцій LAC і CID.

Радіотехнічна розвідка відбувається із застосуванням нетмоніторів, які дозволяють визначити параметри базових станцій. Прикладом такого додатку є мобільний додаток G-MoN Pro. Залежно від типу мобільної мережі, нетмонітор може відображати різні параметри. Наприклад, якщо ми говоримо про LTE (Long-Term Evolution) – мобільну мережу четвертого покоління, то мають місце наступні параметри.

1. TAC (Tracking Area Code) – 16-розрядне ціле число. Це код зони відстеження, унікальний у межах мережі оператора.

2. eNBID (eNodeB Identifier) – 6-розрядне ціле число. Унікальний у межах мережі оператора, аналог базової станції в LTE.

3. LCID – 2-розрядне ціле число.

4. Ідентифікатор соти CellID (Cell Identifier) це eNBID-LCID (шаблон: xxxxxx-xx).

Як видно, LAC у мережах LTE не передбачено. Це викликано тим, що мережа передачі даних LTE спрощена і складається лише з мережі базових станцій (eNodeB) і виділеного ядра пакетної передачі даних. Проте аналог LAC у мережі LTE існує – це TAC (практично їх ідентифікують). TAC у мережах LTE служить для логічного поділу мережі на зони відстеження, на відміну LAC, який обумовлений, швидше, фізичним поділом мережі.

Сторони кримінального провадження можуть одержати від операторів мобільного зв'язку інформацію щодо:

- відомостей, які дозволяють унікально ідентифікувати абонента та обладнання, що ним використовується;
- відомостей про оплату наданих послуг;
- деталізації телефонних з'єднань.

На відміну від тимчасового доступу до електронних інформаційних систем або їх частин, мобільних терміналів систем зв'язку шляхом зняття копії інформації, які містяться в таких електронних інформаційних системах або їх частинах, мобільних терміналах систем зв'язку, без їх вилучення, що

регламентується ч. 2 ст. 159 КПК України, установлення місцезнаходження радіоелектронного радіообладнання (радіоелектронного засобу) як НС(Р)Д полягає в одержанні інформації, яка одержується (формується) оператором зв'язку після прийняття рішення про проведення цієї слідчої дії. А тимчасовий доступ до інформації стосується відомостей, які одержані та зберігаються у оператора зв'язку до надходження чи виконання ухвали суду про тимчасовий доступ.

Після визначення параметрів базових станцій мобільного зв'язку, слідчий, керуючись п. 7 ч. 1 ст. 162, ч. 5 та 6 ст. 163, ст.ст. 233, 234 КПК України звертається до слідчого судді з клопотанням про тимчасовий доступ до даних (трафіку з'єднань), які перебувають у володінні встановлених операторів телекомунікації (мобільного зв'язку). Такими даними є дата, час, тривалість з'єднання, номер абонента з'єднань «А», IMEI (міжнародний ідентифікатор мобільного приладу) їх мобільних терміналів та IMSI (міжнародний індивідуальний номер) абонента «А», номер абонента «Б», CellID (або LAC і CID), азимуті ретрансляційних антен та адреси (або координати) розташування базових станцій для зручності виявлення мобільного терміналу.

Отже, трафік з'єднань зберігає наступну інформацію:

- ідентифікатори мобільних приладів (IMEI, IMSI), якими користувалися особи;
- дзвінки (номер, на який або з якого дзвонили, час, тривалість);
- SMS (від кого або кому, розмір SMS, час відправлення/отримання);
- інтернет-сесії (обсяг сесії в КБ, час);
- параметри базових станцій (LAC, CID або Cell ID, азимут, адреса, координати), які працюють в зоні події.

Азимут базової станції – це напрямок розташування антени, а точніше, лінія центру сегмента (сектора охоплення) антени. Азимут обчислюється в градусах за годинниковою стрілкою від північного напрямку як нульової точки.

Треба зауважити, що сигнал абонента, що телефонує, приймається менш завантаженою станцією в зоні покриття, що визначається спеціальною комп'ютерною програмою. Це може бути не обов'язково найближча до абонента станція.

Відносно задачі встановлення точних координат телефону, навіть якщо абонент не телефонує, можна надати наступне.

Одним з способів визначення місцезнаходження мобільного телефону є відстеження сигналу з вишок стільникового зв'язку. У всіх сучасних мережах

мобільного зв'язку *оператор* може обчислити, де знаходиться телефон конкретного абонента, якщо телефон увімкнений і знаходиться в мережі.

Один із способів, яким оператор може скористатися, – це спостереження за рівнями сигналу від телефону конкретного абонента, що отримується різними вежами. Можливість зробити це залежить від того, як побудовано мобільну мережу, а сама технологія називається *триангуляцією*. Зазвичай, маючи мінімум 3 вежі зв'язку, оператор може визначити місцезнаходження з точністю до кілометра. Це робиться за допомогою вимірювання кута надходження (Angle of Arrival, або AoA).

Для сучасних мереж та мобільних телефонів також використовується технологія *трилатерації*. Цей спосіб використовується там, де підтримується функція «locationInfo-r10». Ця функція надає звіт, що містить точні GPS координати телефону.

Такі спеціалізовані системи, відповідно до вимог Постанови Верховної Ради України № 2471-ХІІ від 17.06.1992, Положення про порядок розроблення, виготовлення, реалізації та придбання спеціальних технічних засобів для зняття інформації з каналів зв'язку, інших засобів негласного отримання інформації, затвердженого Постановою Кабінету Міністрів України від 27.10.2001 № 1450, Постанови Кабінету Міністрів України від 22.09.2016 № 669, не можуть перебувати у власності громадян, громадських об'єднань, міжнародних організацій та юридичних осіб інших держав на території України. Їх технічні характеристики та особливості можуть становити державну таємницю.

Комплекс моніторингу мобільного зв'язку виробництва компанії «Neosoft» та аналогічні йому здатні визначати радіоелектронну обстановку мобільних терміналів із телекомунікаційної стільникової мережі, переводити зареєстровані мобільні термінали у режим радіовипромінювання та визначати місцезнаходження мобільного терміналу за його ідентифікаторами та рівнем сигналу.

Для проведення радіоелектронної розвідки використовуються малогабаритні тактичні системи радіорозвідки «Пластун РР-525», «Пластун-РР-3000» виробництва «НВЦ Інфозахист», їх аналоги, що здатні пеленгувати системи зв'язку, здійснювати автоматизований швидкісний радіомоніторинг, обробку та реєстрацію даних перехоплення, запис аналогових та цифрових каналів передачі даних.

Проведення зазначеного НС(Р)Д не передбачає одержання інформації про зміст розмов, повідомлень, які передаються і не передбачає втручання в приватне спілкування, тому може проводитися: 1) на підставі ухвали слідчого

судді; 2) за поствою слідчого та прокурора, у невідкладних випадках з одержанням у подальшому дозволу слідчого судді; 3) за заявою власника радіообладнання.

Протокол проведеного зазначеного НС(Р)Д має містити відомості про підставу його проведення (ухвалу чи постанову), її реквізити, які мають містити, як і протокол, відомості про ідентифікаційні ознаки, які дозволяють унікально ідентифікувати абонента спостереження, електронну комунікаційну мережу, кінцеве обладнання, застосоване технічне обладнання та МНІ, на які зафіксовано одержану інформацію.

Якщо зазначене НС(Р)Д проводиться на підставі постанови, вона також має містити відомості про підстави, які вказують на невідкладність випадку.

Інший спосіб визначення місцезнаходження мобільного телефону пов'язано з витоком інформації про місцезнаходження з додатків на телефоні, заснованих на локації, та веб браузерів. Сучасні смартфони застосовують кілька способів для визначення свого власного розташування, часто використовуючи GPS, а іноді й інші сервіси визначення локації, які називають «Службами позиціонування». Як правило, «модель дозволів» вимагає від додатків запитувати право на використання служб позиціонування, однак деякі додатки можуть вимагати дозвіл на доступ до інформації про місцезнаходження більш агресивно, ніж інші. Деякі з цих програм потім надсилають інформацію про місцезнаходження телефону постачальнику послуг, що, у свою чергу, надає можливість третій стороні відстежувати об'єкт. Відстеження розташування, а значить і хронологія переміщень, може дати значну інформацію про діяльність об'єкта та визначити тих осіб, які контактували з ним.

Одна з можливостей встановити місцезнаходження телефону, це використати включений на смартфоні Wi-Fi. Смартфон періодично передає «пробний запит», що включає MAC-адресу. Тим самим він повідомляє про свою присутність усім пристроям поблизу. Bluetooth пристрої роблять щось подібне. Ці ідентифікатори традиційно використовуються як інструмент для збору інформації про переміщення пристроїв в зоні дії Wi-Fi (розважальні заклади, кафе, магазини).

### **Питання для самоперевірки:**

1. Приклади структурованих типів даних.
2. Властивості табличного формату.
3. Найбільш розповсюджений тип файлів, які зберігають табличні дані.
4. Різниця між форматами txt та csv, які містять табличні дані.

5. Як може бути використано аналітичний матеріал трафіку з'єднань?
6. В чому полягає радіотехнічна розвідка?
7. Параметри базових станцій мобільного зв'язку.
8. Застосування нетмоніторів.
9. Які параметри мобільної мережі четвертого покоління?
10. Інформація, яка надається оператором зв'язку в трафіку з'єднань.
11. Що таке азимут базової станції?
12. Технологія триангуляції для визначення місцезнаходження телефону.
13. Відстеження руху мобільних пристроїв в зоні дії Wi-Fi.

## **2. Інструментарій аналізу даних**

### **2.1. Імпорт даних в Excel**

Аналіз текстових структурованих даних, як правило, проводиться в Excel. Коли дані імпортують в Excel, з ними можуть відбутися певні зміни, навіть помилки. Excel, імпортуючи csv-файл, може некоректно «зрозуміти» зміст окремих даних. Більшість казусів пов'язано з тим, що Excel за замовчуванням перетворює рядки з набором цифр в числа.

Ось деякі приклади:

- якщо у вихідній клітинці два номери телефону записані через кому без пробілів – 9188535,9233866, то Excel перетворить це в число з двома цифрами після коми – 9188535,92;

- довгий набір цифр «413771268482» (наприклад, номер рахунку) перетворюється до експоненційної форми «4,1E+11»;

- лідируючі плюси видаляються, тому що Excel сприймає знак плюс на початку рядка з цифрами (наприклад, в номері телефону +380...) як ознаку додатного числа і відкидає його;

- видаляє лідируючі нулі. Наприклад, 0971332679 Excel перетворить в 971332679;

- такі дані, як 1/3, Excel виправить на дату.

Щоб уникнути схожих моментів некоректного імпорту, потрібно виконувати імпорт даних за допомогою Майстра імпорту. Для цього знаходимо вкладку «Дані», далі, «Отримання зовнішніх даних», вибираємо - «З тексту» (рис. 2.1.1). З'явиться діалогове вікно для вибору та завантаження

файлу. Завантаження почнеться з відкриття Майстра імпорту текстів (рис. 2.1.2).

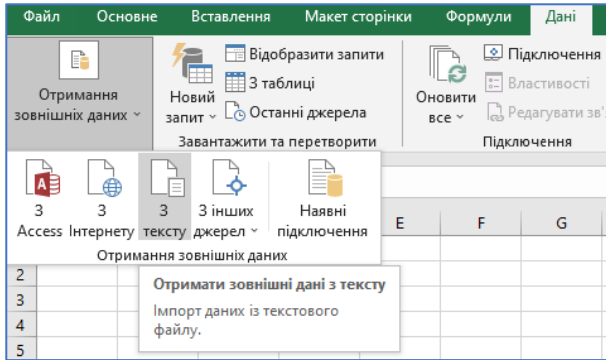


Рисунок 2.1.1 – Імпорт текстового файлу в Excel

Будемо завантажувати структурований текстовий файл банківських транзакцій формату txt. На рис.1.1 можна побачити, що файл містить структуровані дані з стовпців Джерело, Призначення, Дата Час та Сума коштів. Стовпці Джерело і Призначення містять номери рахунків.

На першому кроці імпорту Майстер запитує щодо формату даних в структурованому текстовому файлі. На рис. 2.1.3 можна бачити, що мають місце два варіанта формату – з роздільниками або фіксованої ширини. За замовчанням, формат даних структурованого текстового файлу – з роздільниками (очевидно, це найбільш розповсюджений випадок). Крім того, можна вказати, що дані містять заголовки. В нашому випадку формат даних з роздільниками та міститься заголовок.

У наступному, 2-му кроці, Майстер імпорту пропонує вказати тип символу-роздільнику (знак табуляції, крапка з комою, кома, пробіл, інше). В нашому випадку знак табуляції.

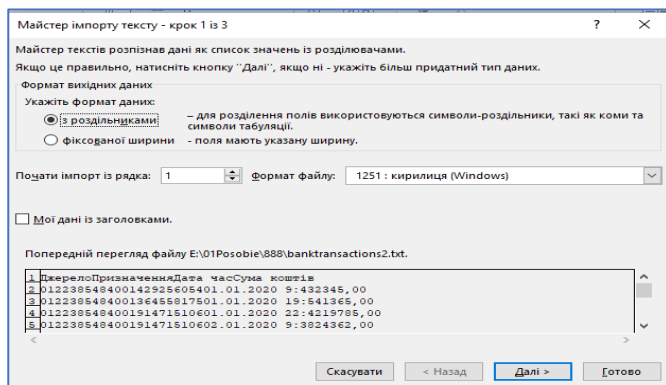


Рисунок 2.1.2 – Майстер імпорту (крок 1)

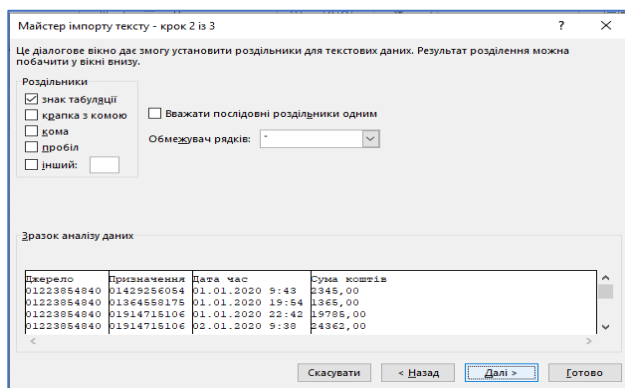


Рисунок 2.1.3 – Майстер імпорту (крок 2)

Далі (крок 3), щоб уникнути втрати інформації, для кожного поля потрібно правильно вказати формат або тип даних (рис. 2.1.4). Майстер підказує, що загальний формат є найбільш універсальним, тобто числові значення перетворюються в числа, дати – в дати, все інше в текст. У нашому прикладі 1-й та 2-й стовпці містять числові значення. Але, якщо їх формат залишити за замовчанням як загальний, то в результаті імпорту Excel відкине нуль, який стоїть першим в номері рахунку. Тим самим, ми втрачаємо частину даних. Отже, для цих стовпців потрібно вибрати формат текстовий (!). Ще раз нагадаємо, що це стосується також номерів телефонів, номерів банківських карток, номерів будинків, паролів (коли вони складаються тільки з цифр) і т.п.

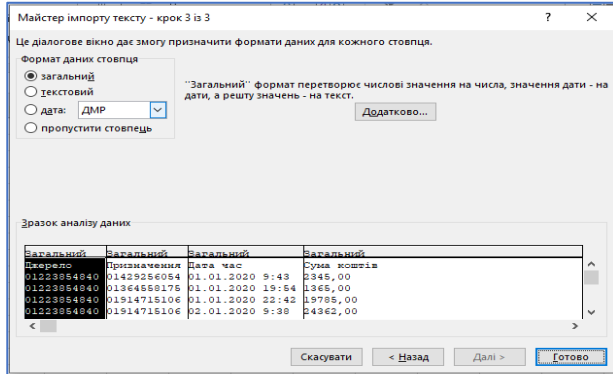


Рисунок 2.1.4 – Майстер імпорту (крок 3)

## 2.2. Імпорт даних в i2 Analyst's Notebook

Програмне забезпечення i2 Analyst's Notebook пропонує багато можливостей для створення різних схем і має в своєму розпорядженні додаткові функції для роботи з великим обсягом інформації [11].

Наступні відомості щодо роботи в i2 Analyst's Notebook відповідають версії 9.0.4 цього додатку (рис. 2.2.1).

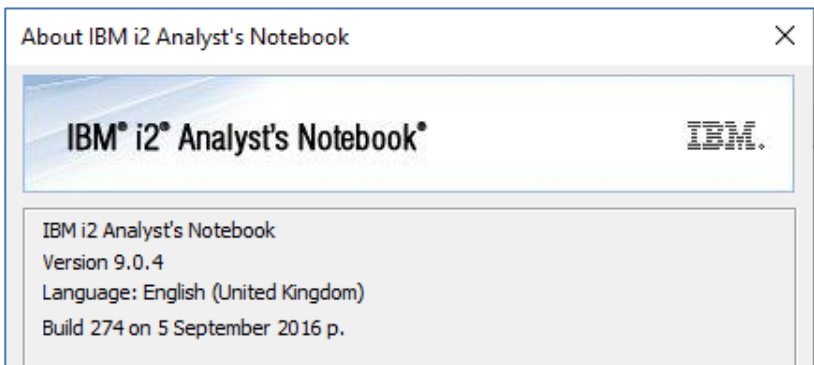


Рисунок 2.2.1.

Після завантаження додатку i2 Analyst's Notebook переходимо на закладку Home і розглянемо імпорт даних (рис. 2.2.2). Метод імпорту даних залежить від їх формату. Якщо дані в одному з форматів XML, який відомий i2 Analyst's Notebook, програма майже всю роботу виконає самостійно. Якщо інакше, потрібно створити специфікацію імпорту та визначити в ній, яким чином інтерпретувати такі дані. Специфікації імпорту можна зберігати для повторного використання з такими ж або схожими за структурою даними.

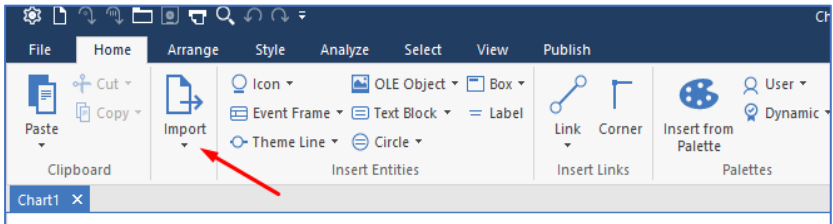


Рисунок 2.2.2.

Вибір джерела даних для імпорту виконується за допомогою команди Import from file... (рис. 2.2.3).

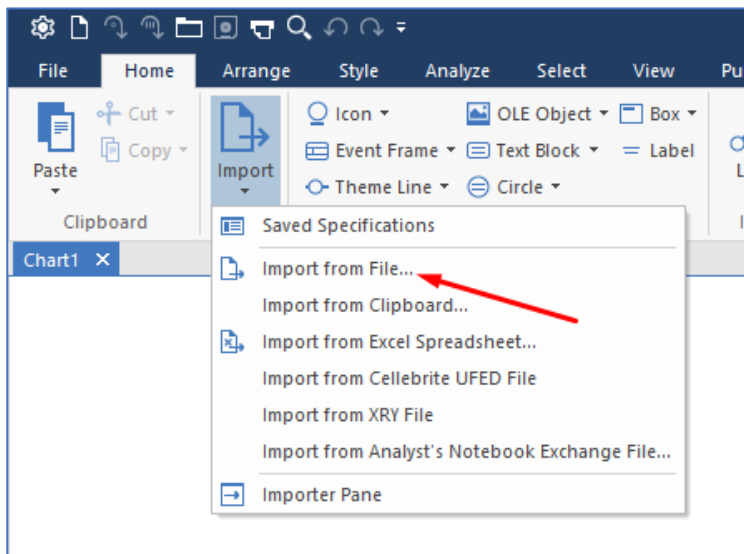


Рисунок 2.2.3.

Якщо є дані у вигляді текстового файлу або електронної таблиці Excel, можливо імпортувати ці дані безпосередньо в i2 Analyst's Notebook. Для того, щоб зручно та раціонально виконати імпорт файлу у форматі Excel в середовищі i2 Analyst's Notebook, його можна заздалегідь підготувати в Excel. Це не є обов'язковим, тому що i2 Analyst's Notebook дозволяє, використовуючи власні інструменти, виконати в певній мірі всі необхідні дії. Тому користувач вправі обрати той спосіб, який зручніший саме йому.

Можна вказати джерело даних для імпорту, а потім визначити параметри специфікації імпорту, які підходять до даних. Після завдання джерела даних i2 Analyst's Notebook виконає пошук існуючих специфікацій імпорту, які є можливим використати або налаштувати.

Можна вибрати джерело даних і створити нову специфікацію імпорту, використати існуючу специфікацію або створити нову специфікацію, використовуючи існуючу специфікацію як шаблон.

Будемо вважати, що ми не маємо відповідної специфікації для імпорту даних нашої таблиці, тобто таблиця такої структури імпортується в додаток вперше. Створення специфікації імпорту починається з вікна на рис. 2.2.4.

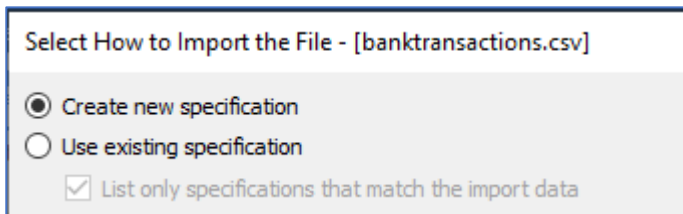


Рисунок 2.2.4

Специфікація імпорту визначає спосіб перетворення текстового файлу або файлу електронної таблиці в елементи схеми, які можна аналізувати. Коли імпортуються дані в одному з цих форматів, є можливим або створити специфікацію імпорту, або змінити існуючу специфікацію відповідно до потреб.

Після завдання джерела даних до складу створення або редагування специфікації імпорту потрібно визначити:

- структуру імпортованих даних;
- типи створюваних об'єктів і зв'язків;
- відображення імпортованих даних на властивості об'єктів і зв'язків;
- параметри, такі як формати дати та часу.

При створенні або редагуванні специфікації імпорту в i2 Analyst's Notebook можна подивитися як визначення в специфікації імпорту можуть вплинути на спосіб імпорту ваших даних.

Вся процедура імпорту розподілена на 6 закладок. На кожній з них необхідно обрати запропоновані програмою i2 Analyst's Notebook налаштування або на свій розсуд задати параметри чи виконати певні дії.

#### Крок 1. Закладка **Define Columns**.

Потрібно вказати, роздільник між полями (рис. 2.2.5). У більшості випадків потрібно вказати роздільник – semicolon (крапка з комою).

#### Крок 2. Закладка **Select Rows**.

Виконується вибір і виключення рядків для імпорту (рис. 2.2.6.). Якщо рядок даних не призначений для імпорту, можна конфігурувати засіб імпорту, щоб він виключав такі рядки з імпорту. Наприклад, рядок в даних імпорту може містити заголовки колонок, що описують дані колонок. Слід зазначити, що не завжди у файлі, що експортується заголовки можуть починатися з першого рядка. Буває, що файл, який імпортується, не містить рядків із заголовками, у такому разі галочка напроти «Extract column headers from row» не ставиться (рис. 2.2.7).

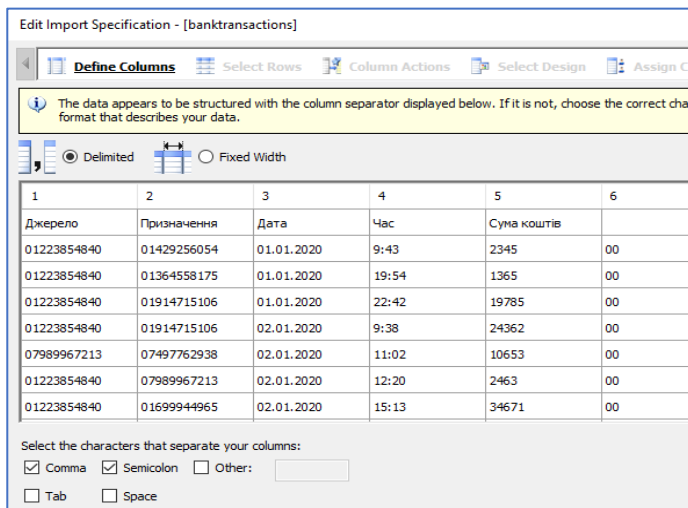


Рисунок 2.2.5

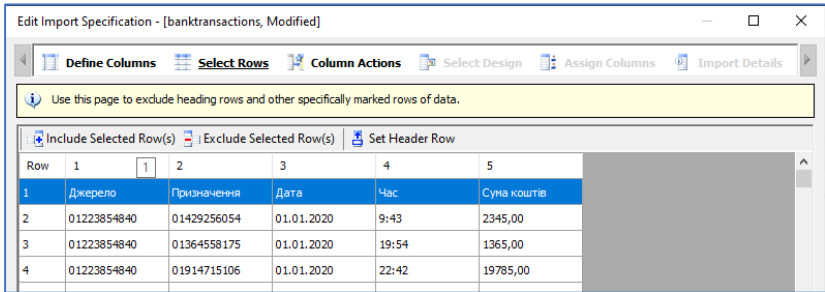


Рисунок 2.2.6

Рядки можуть містити коментарі, призначені для читача, а не фактичні дані. Щоб виключити рядки, у яких в якості префікса використовується спеціальний символ, виберіть опцію «Ignore rows starting with» і введіть символ в сусідньому полі (рис. 2.2.7).

Щоб вилучити конкретні рядки з імпорту, виберіть рядки в таблиці і натисніть праву клавішу, потім «Exclude Selected Row(s)» (рис. 2.2.8). Рядки, які виключені з імпорту, закреслюються в таблиці. Щоб включити їх знову, натисніть кнопку «Include Selected Row(s)».

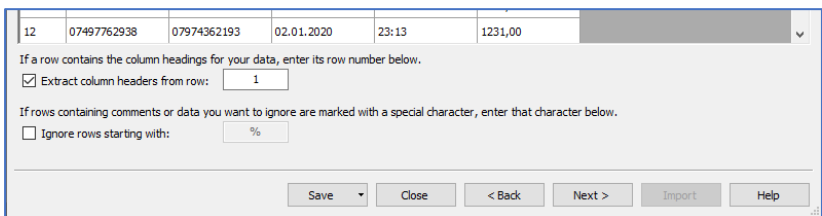


Рисунок 2.2.7

2	01223854840	01429256054	01.01.2020	9:43	2345,00
3	01223854840	01364558175	01.01.2020	19:54	1365,00
4	01223854840	01914715106	01.01.2020	22:42	19785,00
5	01223854840	01914715106	02.01.2020	9:38	24362,00
6	07989967213	07497762938	02.01.2020	11:02	10653,00
7	012238548		02.01.2020	12:20	2463,00
8	012238548		02.01.2020	15:13	34671,00
9	01223854840	07497762938	02.01.2020	16:38	24531,00
10	01223854840	01347578654	02.01.2020	19:58	1362,00
11	01223854840	01660686894	02.01.2020	20:32	2463,00

Рисунок 2.2.8

### Крок 3. Закладка **Column Action**.

Виконуємо форматування даних імпорту (рис. 2.2.9). Можна об'єднати дві колонки, додати префікс до даних у колонці або перевести дані у верхній регістр. Можна також експортувати перетворені дані як запис застосованих дій.

В залежності від того, який результат необхідно отримати, можна перейменувати заголовки стовпчиків, відокремити інформацію про дату та час у різні стовпчики, якщо вони надавалися в одному.

Текст у колонці даних можна замінити на значення, що витягнуто з файлу підстановок. Наприклад, текст Female можна замінити на F. Можна також видалити порядкові суфікси, такі як st, nd, rd і th з дат англійською мовою. Такі перетворення називаються «діями»; будь-яке число дій можна застосувати до окремих колонок даних. Всі перетворення ідентифікуються в таблиці попереднього перегляду даних графічним позначенням.

Щоб створити дію в колонці і застосувати її до колонки:

1. Виберіть колонку, яку ви хочете змінити.
2. Зі списку «Available Actions» виберіть дію, а потім за допомогою кнопки зі стрілками перемістіть її до списку «Action Applied». У заголовку вибраної колонки виводиться значок, який показує, що задано певну дію в колонці. Наприклад, якщо ви виберете дію «Add Prefix», треба ввести префікс, який ви бажаєте додати.

3. Конфігурація для кожної колонки буде своя. Ця можливість програми i2 Analyst's Notebook дуже корисна та дієва.

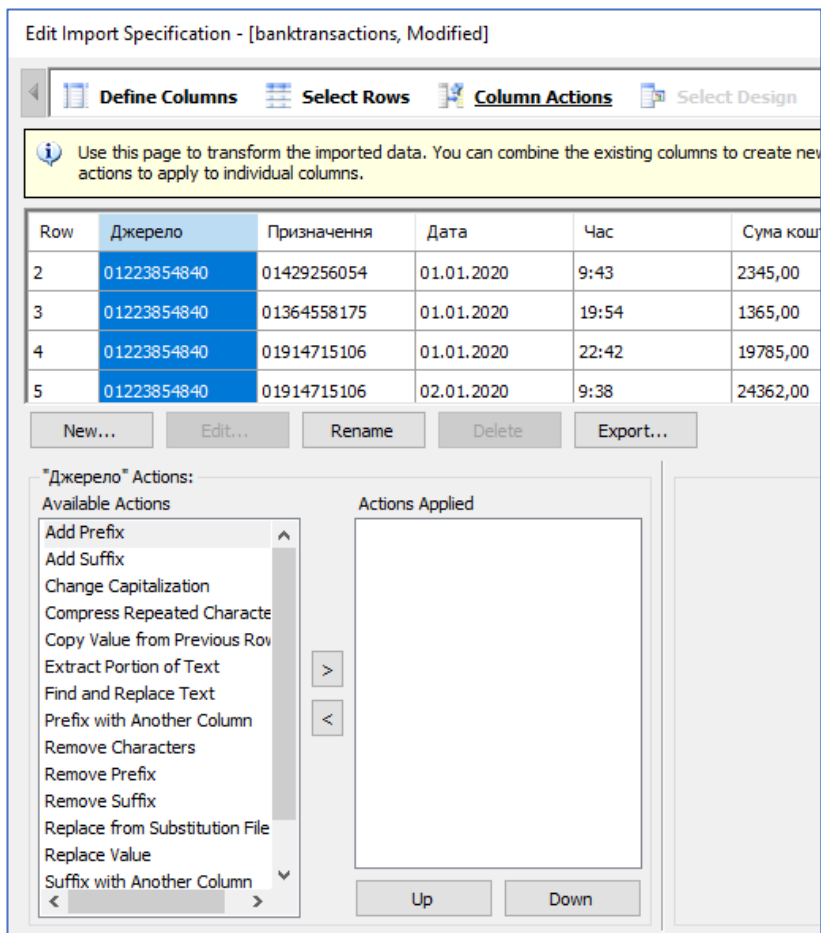


Рисунок 2.2.9

Розглянемо приклад імпорту файлу в форматі Excel з трафіком телефонних з'єднань. У стовпчику з номерами абонентів можуть міститися одні й ті ж самі номери, проте вони можуть бути надані в різних форматах (+380950000001, 380950000001, 80950000001, 0950000001 та 950000001) і програма вважає ці номери різними. Щоб привести їх до одного формату, виконаємо наступні дії:

1. «Remove prefix», з префіксом «+».
2. «Remove prefix», з префіксом «0».
3. «Remove prefix», з префіксом «80».

4. «Remove prefix», з префіксом «380».

Після цього у всіх рядках міститимуться номери телефонів у форматі 950000001.

Дії в колонках застосовуються послідовно в тому ж порядку, в якому вони показані в списку «Available Actions».

У наведеній нижче таблиці описано доступні дії для зміни даних імпорту:

Назва дії	Пояснення
Додати префікс <a href="#">Add Prefix</a>	Додає символи безпосередньо перед даними. Наприклад, Code 01234, Code 01235 і Code 01236, де префікс - "Code".
Додати суфікс <a href="#">Add Suffix</a>	Додає символи відразу ж після даних. Наприклад, ABC Inc., BCD Inc. і CDE Inc., де суфікс - "Inc.".
Змінити регістр <a href="#">Change Capitalization</a>	Змінює регістр тексту. Можна змінити регістр тексту на нижній, на верхній, перевести початкові літери у верхній регістр або звернути поточний регістр.
Об'єднати повторювані символи <a href="#">Compress Repeated Characters</a>	Замінює повторювані символи на один екземпляр символу. Наприклад, в кінці даних, що автоматично генеруються, можуть міститися зайві символи пробілу, які за допомогою цієї дії колонки можна видалити.
Скопіювати значення з попереднього рядка <a href="#">Copy Value from Previous Row</a>	Замінює дані в «сегменті» даними в «сегменті» над нею.
Витягти частину тексту <a href="#">Extract Portion of Text</a>	Використання з кожного «сегмента» тільки конкретної частини даних. Наприклад, щоб витягти з повних телефонних номерів місцевих номерів, відкинувши міжміські телефонні коди, налаштуйте дію колонки для вилучення останніх семи символів.
Знайти і замінити текст <a href="#">Find and Replace Text</a>	Замінює послідовність символів іншою послідовністю символів. Наприклад, щоб видалити суфікси порядкових числівників дат англійською, можна замінити st, nd, rd і th на порожню послідовність. Прим.: Ця дія в колонці може призвести до непередбачуваних змін. Наприклад, при заміні

	символом нового рядка суфікса порядкових числівників дат st зміняться такі слова, як August.
Додати значення іншої колонки на початок <a href="#">Prefix with Another Column</a>	Додає символи з іншої колонки на початок даних в обраній колонці. Наприклад, щоб створити об'єднану колонку дати і часу, до даних часу можна додати в якості префікса дані дати. Потім можна задати символ-роздільник для використання між даними з кожної колонки.
Видалити символи <a href="#">Remove Characters</a>	Видаляє з даних всі входження зазначених символів. Цю дію колонки можна конфігурувати для видалення пробілів або символів табуляції, а можна також ввести символи для видалення. Наприклад, щоб видалити символи ")" і "(" з (863) 555 0140, в поле Remove Characters введіть "()".
Видалити префікс <a href="#">Remove Prefix</a>	Видаляє символи з початку даних. Наприклад, можна видалити "MR" з MR SMITH.
Видалити суфікс <a href="#">Remove Suffix</a>	Видаляє символи із закінчення даних. Наприклад, можна видалити "Esq." з Andrew SMITH Esq.
Замінити з файлу заміни <a href="#">Replace from Substitution File</a>	Замінює послідовність символів на іншу послідовність символів. У файлі підстановок можна вказати кілька заміни. Наприклад, коди цивільного стану можна замінити такими словами: X = Неодружений, Ж = Одружений, P = Розлучений. <ul style="list-style-type: none"> <li>• Файл підстановок - це файл зі значеннями, розділеними комами (.csv), файл зі значеннями, розділеними табуляцією (.tsv) або текстовий файл (.txt). У кожному рядку він містить пари послідовностей символів.</li> <li>• Кожна пара складається з послідовності символів, які слід знайти в даних, символу-роздільника і послідовності символів для заміни знайдених символів. Цей роздільник може бути символом табуляції, коми (,), знаком рівності (=) або двокрапкою (:).</li> <li>• Щоб використовувати символ-роздільник в складі послідовності символів, його треба укласти в подвійні лапки (") або забезпечити як префікс зворотною дробовою рисою. Тексту, який повинен ігноруватися</li> </ul>

	засобом імпорту (наприклад, з коментарем), повинна передувати крапка з комою (;).
Замінити значення <a href="#">Replace Value</a>	Замінює слова повністю на інше слово.
Додати значення іншої колонки в кінець <a href="#">Suffix with Another Column</a>	Додає символи з іншої колонки до закінчення даних в обраній колонці. Наприклад, щоб створити об'єднану колонку дати і часу, можна додати дані дати як суфікс до даних часу. Потім можна задати символ- роздільник для використання між даними з кожної колонки.
Видаляти символи <a href="#">Trim Characters</a>	Підрізає на початку і в кінці колонки даних непотрібні символи, такі як пробіли, символи табуляції або конкретні символи (поле «Trim other characters listed below»)

Можна конфігурувати дію в колонці, щоб змінювати дані тільки при виконанні певних умов (рис. 2.2.10). Можна створити кілька умов.

- a. У списку ... Condition (Умова) вибрати умову.
- b. Ввести потрібне значення у полі Value (Значення).

У таблиці імпорту даних виводяться змінені дані.

Якщо включений перемикач «Stop processing subsequent actions when conditions are met», дія застосовується до будь-якого «сегменту» у колонці, яка задовольняє умовам, але подальші дії до таких «сегментів» не застосовуються.

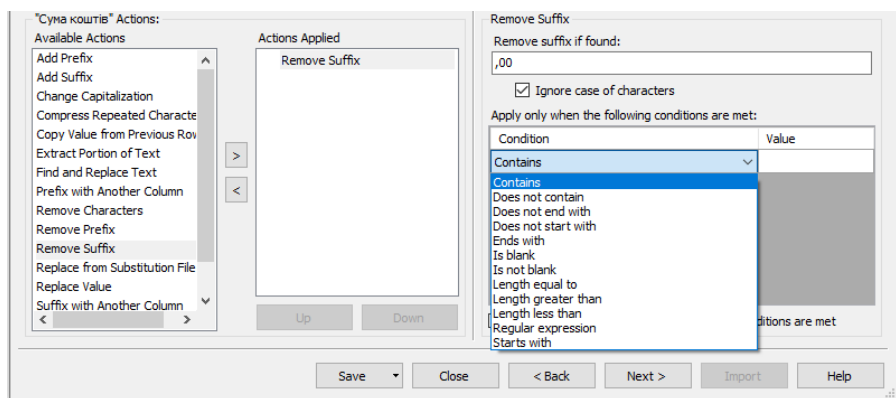


Рисунок 2.2.10

У наступній таблиці описано доступні умови дій в колонках:

Стан	Пояснення
Містить	Дані містять задану послідовність символів.
Не містить	Дані не містять задану послідовність символів.
Не закінчується на	Дані не закінчуються на задану послідовність символів.
Не починається з	Дані не починаються за заданою послідовністю символів.
Закінчується на	Дані закінчуються на задану послідовність символів.
Є порожньою	«Сегмент» порожній.
Чи не є порожньою	«Сегмент» не порожній.
Довжина дорівнює	Довжина даних (у символах) дорівнює заданому числу.
Довжина більша ніж	Довжина даних (у символах) більша заданого числа.
Довжина менша ніж	Довжина даних (у символах) менша заданого числа.
Регулярний вираз	Дані відповідають шаблону, що визначено введеним регулярним виразом.
Починається з	Дані починаються за заданою послідовністю символів.

Розглянемо об'єднання колонок при імпорті даних. Можливим є створення нового елемента даних, об'єднавши дані з інших колонок. Наприклад, можна об'єднати колонки "Код регіону" і "Місцевий номер" при імпорті даних, щоб створити колонку "Повний номер телефону".

Спочатку тиснемо на кнопку «New...» (рис. 2.2.11). Відкривається вікно «New column». У полі «Enter the name of the new column» можна вказати ім'я змішаної колонки. Далі, з вікна «Source Columns» відправляємо потрібні колонки у вікно «Joined Columns». При цьому, між даними колонок можна вставляти роздільники (space, comma, period, ...). Кнопки Up і Down дозволяють міняти місцями вибрані елементи. Потім тиснемо ОК.

Після створення нової колонки становляться активними кнопки «Edit...» та «Delete» (рис. 2.2.12), які дозволяють продовжити редагування створеної колонки (або, її видалення).

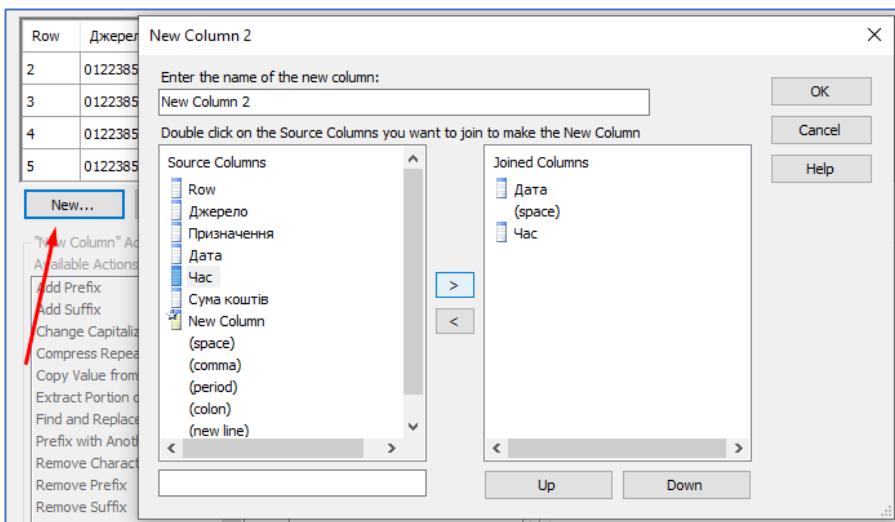


Рисунок 2.2.11

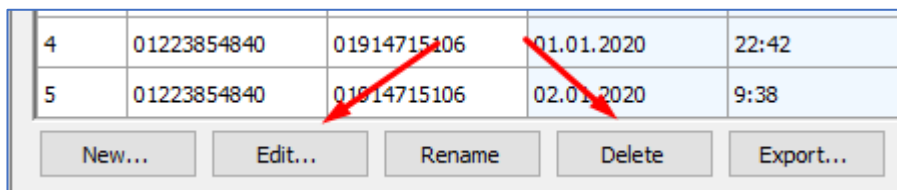


Рисунок 2.2.12

#### Крок 4. Закладка **Select Design**.

На цьому кроці визначається структура майбутньої схеми.

Залежно від структури даних і поставленої задачі може бути вибрана одна з запропонованих схем (рис. 2.2.13).

Наприклад, якщо необхідно проаналізувати трафік телефонних з'єднань, то доцільно вибрати структуру імпорту – «Network of telephone calls». У прикладі з банківськими транзакціями можна вибрати «Network of transactions» або «Sequence of transactions». Зробивши вибір можна переходити до наступної закладки.

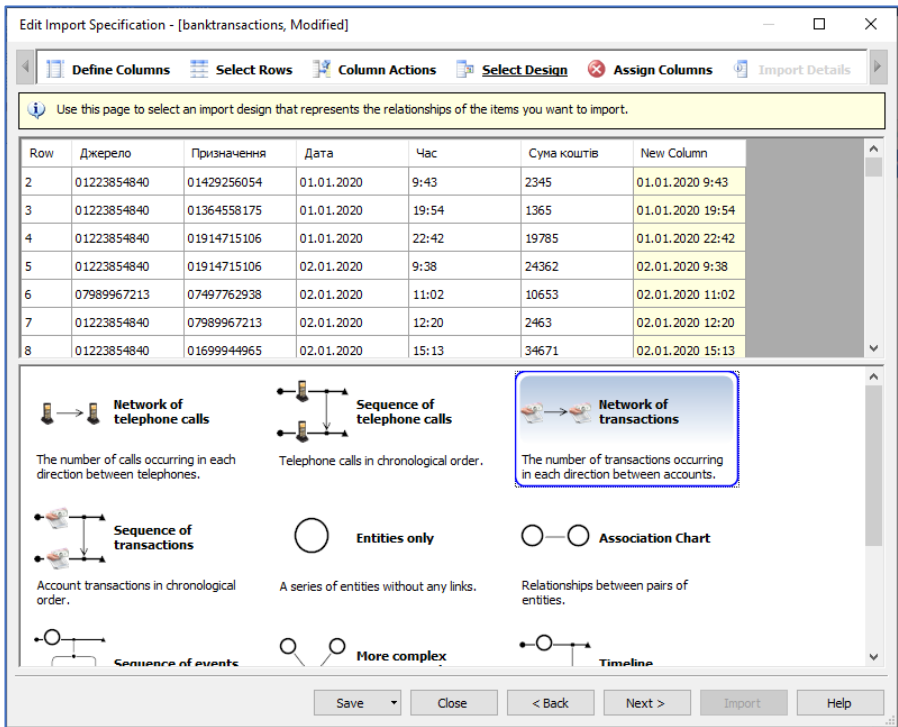


Рисунок 2.2.13

### Крок 5. Закладка **Assign Columns**.

Ця вкладка передбачає призначення (або зіставлення) колонок (стовпців таблиці) елементам схеми. На рис. 2.2.14 ми бачимо окремий фрагмент схеми, де елементам схеми ще не призначені колонки. Відповідно до конкретних даних виконуємо операцію зіставлення. У прикладі з банківськими транзакціями, очевидно, об'єкту зліва зіставляємо колонку «Джерело», об'єкту справа – колонку «Призначення». Зв'язок між цими об'єктами (спрямоване ребро) може містити таку інформацію про транзакцію, як дата або сума коштів (рис. 2.2.15). Зіставлення виконуємо перетягуванням колонки на об'єкт.

Після призначення елементам схеми певних колонок, назва закладки (Assign Columns) змінює колір на більш темний, і кнопка Next стає активною.

Наступними двома кроками завершується імпорт даних та схема додається на робоче вікно додатку i2 Analyst's Notebook.

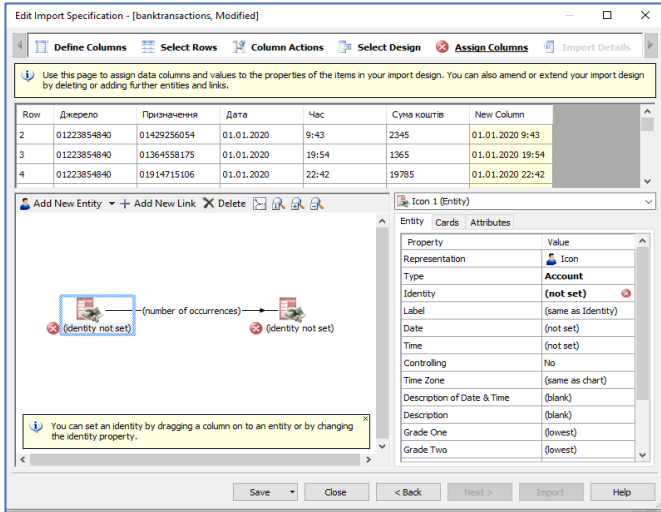


Рисунок 2.2.14

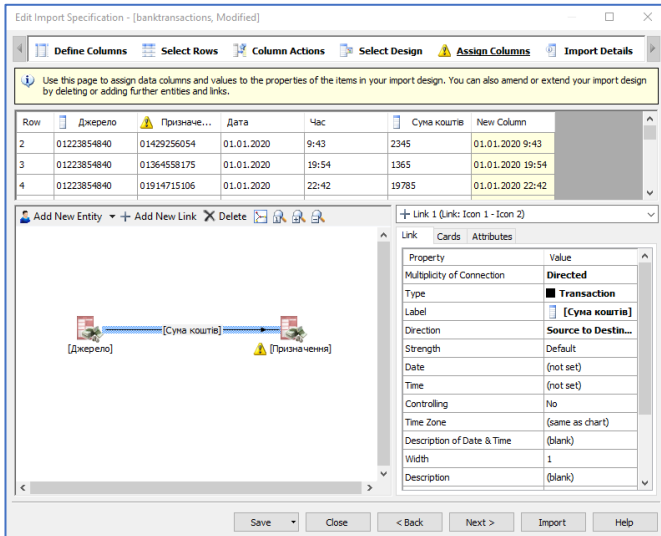


Рисунок 2.2.15

## 2.3. Таблиця банківських транзакцій

### 2.3.1. Аналіз таблиці банківських транзакцій в Excel

Після імпорту текстового структурованого файлу в Excel ми отримуємо *діапазон* комірок з даними, який краще перетворити в *таблицю*. Для цього треба помістити курсор в будь-яку комірку діапазону з даними та відкрити вкладку Вставка. Виберемо на вкладці кнопку команди Таблиця (рис. 2.3.1).

В результаті виконання цієї команди діапазон з даними прийме вид, як на рис. 2.3.2, з характерним стилем та кнопками фільтру. Крім того, з'явиться вкладка Конструктор таблиць (!) (ця вкладка зникає, коли курсор поміщається за межами таблиці).

На вкладці Конструктор таблиць міститься декілька важливих команд, а саме, кнопка фільтра, рядок підсумків, видалити дублікати, зведена таблиця (рис. 2.3.3). До речі, кнопка фільтра і рядок заголовків на вкладці Конструктор активовані відразу (поставлені прапорці).

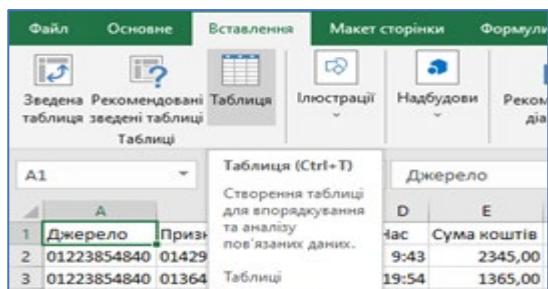


Рисунок 2.3.1 – Вставка > Таблиця

	A	B	C	D
1	Джерело	Призначення	Дата час	Сума коштів
2	01223854840	01429256054	01.01.2020 9:43	2345
3	01223854840	01364558175	01.01.2020 19:54	1365
4	01223854840	01914715106	01.01.2020 22:42	19785
5	01223854840	01914715106	02.01.2020 9:38	24362
6	07989967213	07497762938	02.01.2020 11:02	10653
7	01223854840	07989967213	02.01.2020 12:20	2463

Рисунок 2.3.2

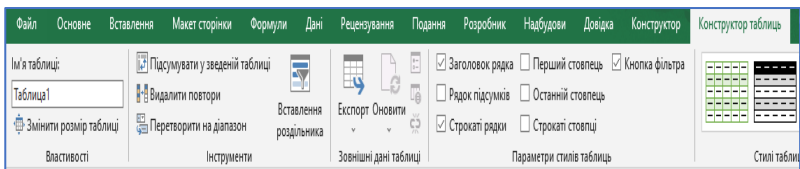


Рисунок 2.3.3 – Вкладка Конструктор

Розглянемо декілька задач щодо аналізу банківських транзакцій.

### 1) Розділення дати та часу.

Якщо ми маємо єдиний стовпець з даними дати та часу, то доволі часто виникає потреба розділити в різні стовпці дату і час. Для цього існує декілька способів.

*Спосіб 1.* Скористаємось командою «Текст по стовпцях» (вкладка Дані).

Спочатку потрібно підготувати таблицю до розділення даних. А саме, справа від стовпця з даними дати і часу, вставити порожній стовпець.

Для того щоб виконати розділення дати та часу у всьому стовпці (в нашому випадку це стовпець С, рис. 2.3.2), потрібно перед розділенням виділити весь цей стовпець. Отже, на 1-у кроці розділення тексту по стовпцях (як і в ситуації з імпортом) майстер визначається з форматом даних (в нашому випадку це роздільник, хоча можна обрати і фіксовану ширину).

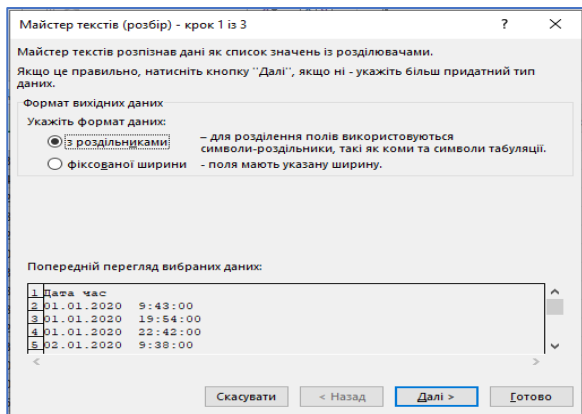


Рисунок 2.3.4 – Майстер розділення тексту по стовпцях (крок 1)

На 2-у кроці вказуємо, що символом-роздільником є пробіл, і на 3-у кроці формат даних стовпців залишаємо як загальний. В результаті отримаємо наступний результат (рис. 2.3.5).

	A	B	C	D	E	
1	Джерело	Призначення	Дата	час	Сума коштів	
2	01223854840	01429256054	01.01.2020	0:00	9:43:00	2345
3	01223854840	01364558175	01.01.2020	0:00	19:54:00	1365
4	01223854840	01914715106	01.01.2020	0:00	22:42:00	19785
5	01223854840	01914715106	02.01.2020	0:00	9:38:00	24362
6	07989967213	07497762938	02.01.2020	0:00	11:02:00	10653

Рисунок 2.3.5 – Результат розділення дати та часу

Допрацювати вид стовпців C і B (рис. 2.3.6) можна, якщо для стовпця C встановити формат «Дата», для стовпця D – формат «Час» (точніше, «усі формати» h:mm).

Для більш якісної візуалізації даних рекомендується закріпити перший рядок таблиці (рядок заголовків) (вкладка Вид, команда Закріпити області > Закріпити верхній рядок).

	A	B	C	D	E
1	Джерело	Призначення	Дата	час	Сума коштів
2	01223854840	01429256054	01.01.2020	9:43	2345
3	01223854840	01364558175	01.01.2020	19:54	1365
4	01223854840	01914715106	01.01.2020	22:42	19785
5	01223854840	01914715106	02.01.2020	9:38	24362
6	07989967213	07497762938	02.01.2020	11:02	10653

Рисунок 2.3.6

*Спосіб 2.* За допомогою формул.

Підготуємо таблицю, додавши справа від стовпця з даними дати і часу два порожні стовпця (в нашому прикладі це будуть стовпці D та E, при цьому дані «Сума коштів» опиняться в стовпці F). Встановимо для першого порожнього стовпця формат «Дата», для другого – формат «Час». В комірці D2 вводим формулу =INT(C2), а в комірці E2 вводим формулу =C2 – D2.

Шляхом протягування або одночасного занесення даних у виділений діапазон комірок, розповсюджуємо ці формули на інші комірки стовпців D та E.

## 2) Проміжні підсумки даних таблиці.

Повернемось до вкладки Конструктор і активуємо рядок підсумків (поставимо прапорець). Перед тим, як активувати рядок підсумків, ми встановили формат останнього стовпця числовим.

Серед проміжних підсумків для таблиці банківських транзакцій можна вказати наступні:

- загальна кількість транзакцій;
- загальна сума коштів для всіх транзакцій;
- кількість транзакцій з певного рахунку (або певних рахунків);
- сума коштів для транзакцій з певного рахунку (або певних рахунків);
- кількість транзакцій на певний рахунок (або певні рахунки);
- кількість транзакцій за конкретну дату (або певний термін);
- сума коштів для транзакцій за конкретну дату (або певний термін);
- максимальний (мінімальний) переказ;

Всі ці задачі вирішуються шляхом фільтрації та використанням функції SUBTOTAL, яка розташована в рядку підсумків (рис. 2.3.7). Синтаксис функції наступний: SUBTOTAL(номер\_функції; діапазон). Номер функції (обов'язковий аргумент) – число від 1 до 11 або від 101 до 111, що вказує, яку функцію використовувати для проміжних підсумків. Аргументи від 1 до 11 працюють з прихованими вручну рядками, а аргументи від 101 до 111 при підрахунку ігнорують приховані рядки. Зв'язок між номером обов'язкового аргументу функції SUBTOTAL та функцією в Excel наступний:

1 (101) – AVERAGE	7 (107) – STDEV
2 (102) – COUNT	8 (108) – STDEVP
3 (103) – COUNTA	9 (109) – SUM
4 (104) – MAX	10 (110) – VAR
5 (105) – MIN	11 (111) – VARP
6 (106) – PRODUCT	

Наприклад, функція SUBTOTAL(9;E2:E1595) підраховує суму всіх переказів в таблиці банківських транзакцій (рис. 2.3.7).

1	Джерело	Призначення	Дата	Час	Сума коштів
1587	01223854840	01309657924	30.03.2020	20:00	2463,00
1588	01223854840	01699944965	30.03.2020	21:46	34671,00
1589	01223854840	01223859234	30.03.2020	21:48	24531,00
1590	01223854840	07497762938	30.03.2020	22:20	1362,00
1591	01223854840	07989967213	31.03.2020	8:07	2463,00
1592	01638782715	01948474016	31.03.2020	9:23	1231,00
1593	01638782715	01463160895	31.03.2020	14:43	34928,00
1594	07497762938	01830520901	31.03.2020	14:44	5732,00
1595	01223854840	01885779839	31.03.2020	18:36	82413,00
1596					42578584,00

Рисунок 2.3.7

### 3) Зведені дані.

Сформулюємо наступні задачі:

- визначити рахунок, з якого надійшла найбільша (найменша) сума коштів;
- визначити рахунок, на який надійшла найбільша (найменша) сума коштів;
- визначити кількість транзакцій з кожного рахунку;
- визначити рахунок, з якого приходиться найбільша кількість транзакцій;
- визначити рахунок, на який приходиться найбільша кількість транзакцій;
- визначити дату, на яку прийшла найбільша кількість транзакцій.

Вказані задачі вирішуються з використанням зведених таблиць в Excel.

Знову повернемося до вкладки Конструктор, на якій знаходимо кнопку команди «Підсумувати у зведеної таблиці».

Створення зведеної таблиці починається з вікна майстра, в якому пропонується вибрати таблицю або діапазон даних та місце розташування зведених даних – на наявному або новому аркуші (рис. 2.3.8).

Наступним кроком з'явиться макет зведеної таблиці – це область звіту (рис. 2.3.9 зліва) (як правило, це початковий діапазон комірок з 3 стовпців і 18 рядків) та панель налаштування зведеної таблиці (рис. 2.3.9 справа).

Панель налаштування складається з області з полями таблиці та чотирма областями: Фільтри, Рядки, Стівпці, Значення.

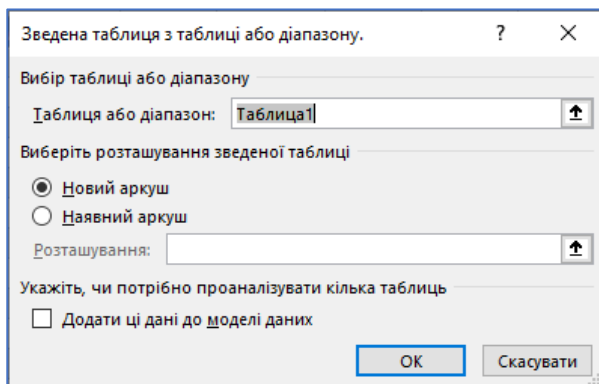


Рисунок 2.3.8

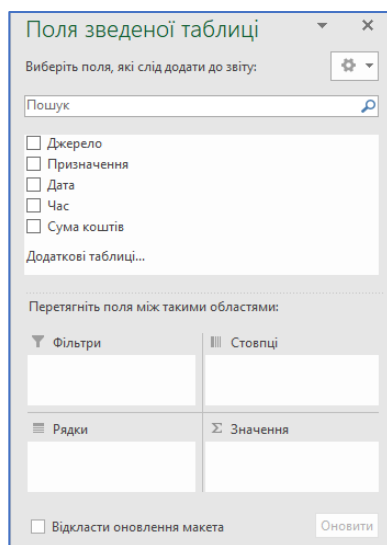
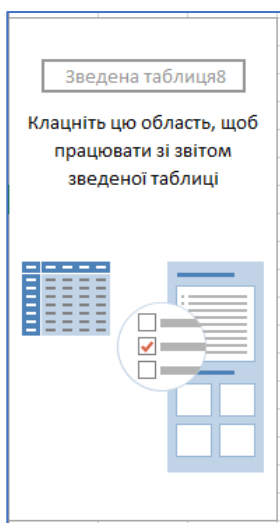


Рисунок 2.3.9

Як тільки ми поставимо прапорець на полі Джерело, отримаємо в області звіту рядки з номерами унікальних рахунків (рис. 2.3.10).

Позначки рядків
01223854840
01223859234
01638782715
07497762938
07974362193
07989967213
<b>Загальний підсумок</b>

**Поля зведеної таблиці** x

Виберіть поля, які слід додати до звіту: ⚙

Пошук

- Джерело
- Призначення
- Дата
- Час
- Сума коштів

Додаткові таблиці...

---

Перетягніть поля між такими областями:

Фільтри <div style="border: 1px solid gray; height: 20px; width: 100%;"></div>	Столпці <div style="border: 1px solid gray; height: 20px; width: 100%;"></div>
Рядки Джерело <span style="float: right;">▼</span>	Σ Значення <div style="border: 1px solid gray; height: 20px; width: 100%;"></div>

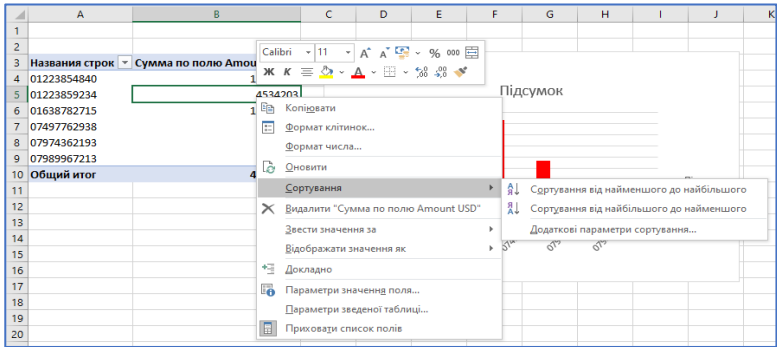
Рисунок 2.3.10

Якщо далі перетягнути в область «Значення» поле Сума коштів, то отримаємо зведену таблицю рахунків та відповідно загальну суму коштів, перерахованих з цих рахунків (рис. 2.3.11).

Позначки рядків	Сума з Сума коштів
01223854840	13664524
01223859234	4534203
01638782715	14553160
07497762938	7388109
07974362193	308873
07989967213	2129715
<b>Загальний підсумок</b>	<b>42578584</b>

Рисунок 2.3.11

При умові значної кількості рядків в зведеній таблиці, їх можна відсортувати. Для цього потрібно поставити курсор на будь-яку комірку другого стовпця, натиснути праву клавішу миші і вибрати Сортування (рис. 2.3.12).



Позначки рядків	Сума з Сума коштів
07974362193	308873
07989967213	2129715
01223859234	4534203
07497762938	7388109
01223854840	13664524
01638782715	14553160
<b>Загальний підсумок</b>	<b>42578584</b>

Рисунок 2.3.12

Тепер легко бачити, що рахунок, з якого надійшла найбільша сума коштів, це 01638782715. Тим самим, ми вирішили першу задачу.

Щоб визначити кількість транзакцій, які відбулися з кожного рахунку-відправника, на панелі налаштування виберемо два поля Джерело і Призначення. Поле Джерело перетягнемо у вікно Рядки, а поле Призначення – у вікно Значення (рис. 2.3.13). В якості параметра значення відразу з'явиться Кількість. Отримані дані в зведеній таблиці можна відсортувати. Тепер легко бачити, що найбільша кількість транзакцій відбулася з рахунку 01638782715 – це 545.

Аналогічно, можна побудувати зведені таблиці для визначення, на який (або, які) рахунки надійшла найбільша сума, або на який рахунок приходится найбільша кількість транзакцій. Для цього на панелі налаштування достатньо поміняти місцями Рядки і Значення.

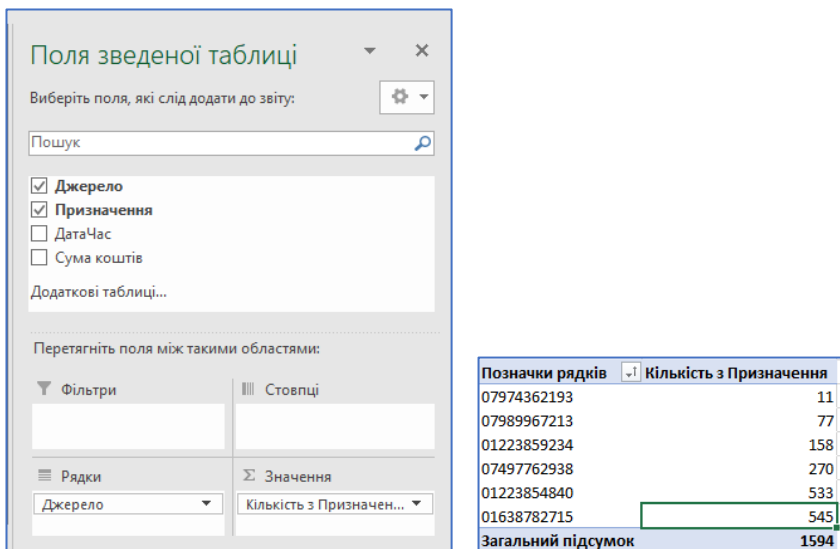


Рисунок 2.3.13

### 2.3.2. Імпорт таблиці банківських транзакцій в i2 Analyst's Notebook

Для аналізу банківських транзакцій в i2 Analyst's Notebook може бути взятий файл формату *xlsx*, або *csv*, або *txt*. Почнемо з імпорту файлу в програмі i2 Analyst's Notebook (вкладка Home) та створимо *нову специфікацію*.

На першому кроці Майстер імпорту (вікно Define Columns) запрошує вказати тип роздільнику між полями, при умові, якщо файл текстовий (формат *csv* або *txt*). Пропонується ще додатково і кома, але її потрібно «зняти», інакше останній стовпець з сумами коштів перетвориться у два стовпця.

На другому кроці (вікно Select Rows) визначається рядок заголовку, його місцезнаходження, або виключаються певні рядки з подальшого імпорту даних. Наша таблиця має заголовок, тому ми вказуємо на це, ставлячи галочку в поле «Extract column headers from row».

Третій крок (вікно Column Actions) дозволяє розділити дату та час (якщо імпортувався файл формату *xlsx*).

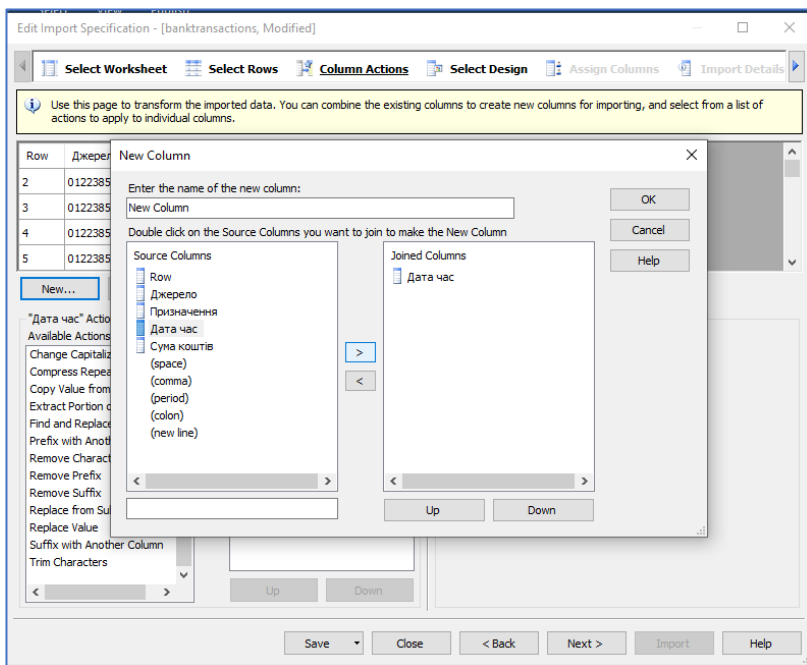


Рисунок 2.3.14

Спочатку ми створюємо два нових стовпця на підставі стовпця Дата час (рис. 2.3.14). Далі, до одного стовпця застосовуємо дію Extract Portion of Text з параметрами First Character – 1, Last Character – 6. Отримаємо дату. Назву стовпця перейменуємо в Дата. До іншого стовпця застосовуємо дію Extract Portion of Text з параметрами First Character – 11, Last Character – 1. Отримаємо час, назву стовпця перейменуємо в Час (рис. 2.3.15).

Цікаво те, що розділення дати та часу на два різні стовпця відбувається значно простіше, якщо ми імпортуємо структуровані дані в форматі csv. Вже на першому кроці імпорту, коли треба визначитися з типом роздільника, в полі Space треба поставити галочку. Програма сприймає це як додатковий роздільник і розміщує дату і час в різні стовпці (початково дата і час розділені пробілом та представлені одним полем).

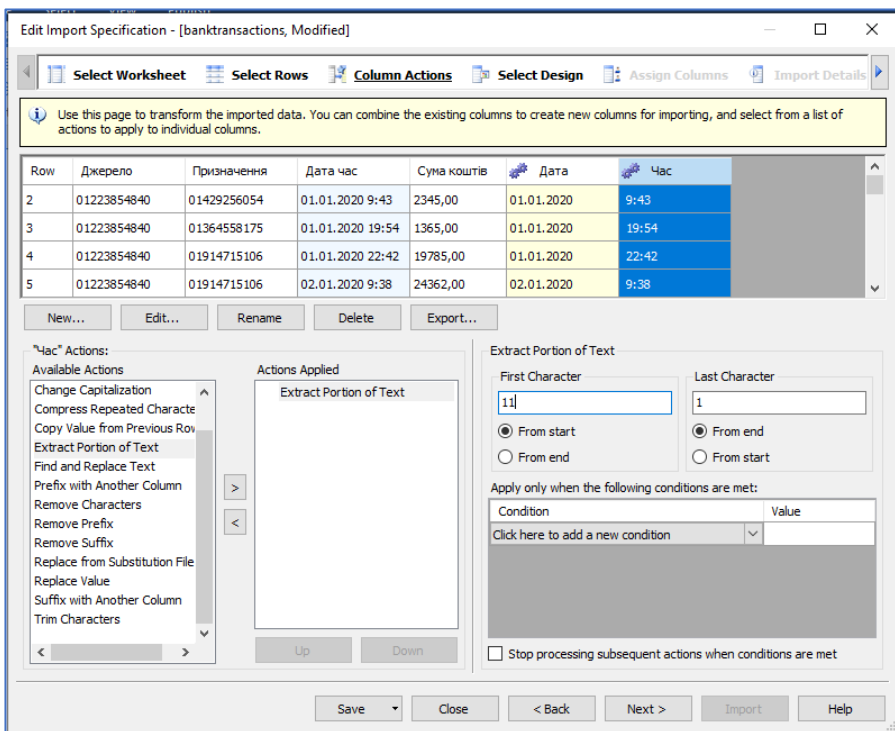


Рисунок 2.3.15

Наступний крок (вікно Select Design) дозволяє вибрати тип зв'язків між об'єктами. Для нашого прикладу візьмемо тип зв'язків Network of transactions (рис. 2.3.16).

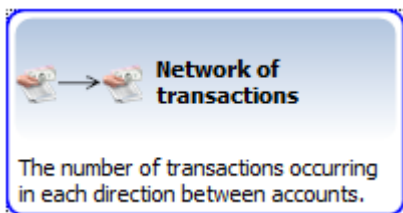


Рисунок 2.3.16

Далі, (вікно Assign Columns) потрібно зіставити дані (стовпці) нашої таблиці банківських транзакцій тим об'єктам, які складають мережу, вибрану на попередньому кроці. На рис. 2.3.17 представлено (зверху) початковий «порожній» зв'язок, та визначений (знизу), на якому лівому об'єкту зіставлено перший стовпець таблиці (Джерело), правому – другий стовпець (Призначення), і сам зв'язок – це п'ятий стовпець (Сума коштів).

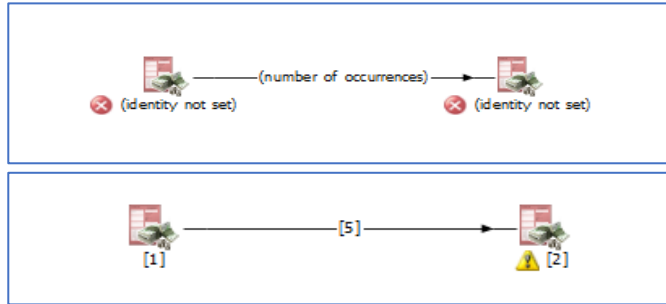


Рисунок 2.3.17

Нарешті, залишилося виконати імпорт даних та додати їх до діаграми (Add to Chart) в робочому вікні програми (рис. 2.3.18).

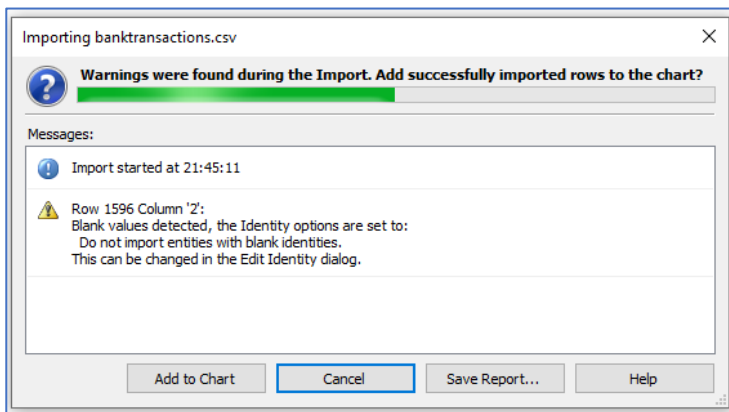


Рисунок 2.3.18

Остаточно, одержимо схему банківських транзакцій (рис. 2.3.19). Схема чітко демонструє найбільш активні рахунки (бачимо три таких рахунки). Більш того, простим наведенням курсора на рахунок або зв'язок та

натисканням на ліву кнопку миші можна проглянути будь-яку транзакцію, або цілий список транзакцій з певного рахунку (або, на певний рахунок).

На вкладці Analyze є кнопка Bar Charts and Histograms, яка дозволяє переглянути деякі кількісні показники схеми. До речі, треба сказати, що в таблиці банківських транзакцій можна бачити повтори транзакцій з одного рахунку на інший з різницею в часі та розміру перерахованих коштів. В схемі це позначається однією стрілкою, але знову ж таки, можна проглянути всі перекази в спливаючому діалоговому вікні. Отже, в вікні Bar Charts and Histograms програма показує узагальнені операції з перерахуванням коштів.

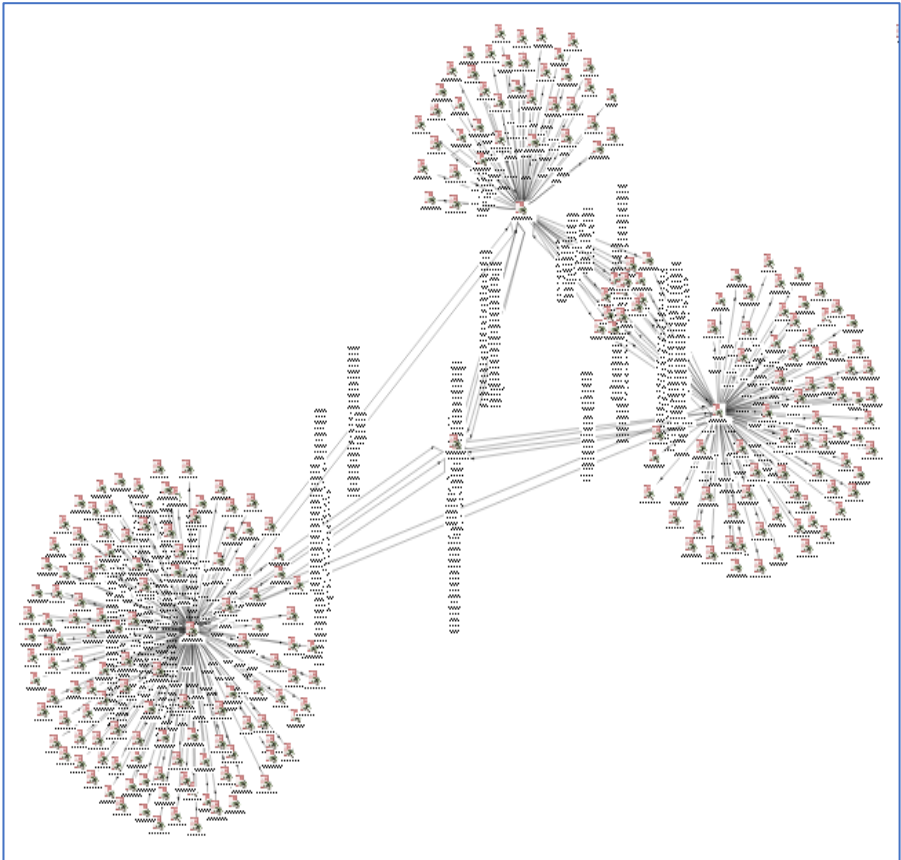


Рисунок 2.3.19

## 2.4. Таблиця трафіку з'єднань

### 2.4.1. Аналіз таблиці трафіку з'єднань в Excel

Орієнтовно, таблиця трафіку з'єднань має наступні поля: Тип, Дата, Час, Тривалість, Абонент А, ІМЕІ А, Абонент Б, ІМЕІ Б, CellID, Азимут, БС.

Розглянемо більш детально поля таблиці. Поле «Тип» містить тип з'єднань (вхідний, вихідний, вх. СМС, вих. СМС, gprs), формат поля – текстовий. Поле «Тривалість» містить тривалість розмови, як правило задається в секундах, формат поля – числовий. Поля «Абонент А» і «Абонент Б» містять абонентські номери, формат поля – текстовий (не числовий – це важливо!). Формат поля ІМЕІ (міжнародний ідентифікатор мобільного приладу, 15-16 цифр) теж текстовий. Поле «CellID» має текстовий формат (шаблон з цифр – xxxxxx-xx). Поле «Азимут» вказує азимут базової станції, формат поля – числовий. Поле БС – це адреса (або, координати) базової станції, формат поля – текстовий (або, числовий).

Але, таблиця трафіку з'єднань може мати трохи і іншу структуру. Наприклад, нехай таблицю трафіку з'єднань збережено у файлі формату xlsx (рис. 2.4.1). В даному випадку в якості абонентів А виступає лише один абонент (об'єкт). Відомості про розташування базових станцій представлено координатами.

	A	B	C	D	E	F	G	H	I
	Об'єкт	Абонент Б	Дата	Час	Тривалість	Тип	Напрямок	Широта	Довгота
2	366916502	366966012	01.01.2017	1:26:00		1 СМС	вихідний	44,92624033	26,06092339
3	366916502	366913695	01.01.2017	0:00:00	16	голосовий	вихідний	44,94545701	26,03350719
4	36691	352069133	01.01.2017	16:39:00	664	голосовий	вихідний	44,92624033	26,06092339
5	366916502	351239369	02.01.2017	11:50:00	20	голосовий	вихідний	44,92624033	26,06092339
6	366916502	366913695	02.01.2017	0:00:00	39	голосовий	вхідний	44,92624033	26,06092339
7	366916502	366913695	02.01.2017	0:00:00	18	голосовий	вихідний	44,92624033	26,06092339
8	366916502	366913695	02.01.2017	18:38:00	27	голосовий	вихідний	44,92624033	26,06092339
9	366916502	366966012	03.01.2017	10:56:00	1	СМС	вихідний	44,95802693	25,99793693
10	366916502	322005621	03.01.2017	10:59:00	1	СМС	вхідний	44,95802693	25,99793693
11	366916502	322005613	03.01.2017	11:00:00	1	СМС	вхідний	44,95802693	25,99793693
12	366916502	322005613	03.01.2017	11:00:00	1	СМС	вихідний	44,95802693	25,99793693
13	366916502	366966012	03.01.2017	0:00:00	1	СМС	вихідний	44,95802693	25,99793693
14	366916502	366966012	03.01.2017	0:00:00	1	СМС	вихідний	44,95802693	25,99793693

Рисунок 2.4.1

Сформулюємо низку задач, які пов'язані з аналізом трафіку з'єднань:

1) визначити кількість з'єднань (голосових та СМС) об'єкта з конкретним абонентом Б;

- 2) порівняти кількість з'єднань (голосових та СМС) об'єкта з абонентами Б;
- 3) порівняти кількість з'єднань (голосових та СМС) об'єкта з абонентами Б по датам;
- 4) визначити тривалість вхідних (вихідних) голосових викликів об'єкта з конкретним абонентом Б;
- 5) визначити кількість прийнятих (відправлених) СМС об'єктом стосовно конкретного абонента Б;
- 6) порівняти тривалість вхідних (вихідних) голосових викликів об'єктом стосовно всіх абонентів Б;
- 7) порівняти кількість прийнятих (відправлених) СМС об'єктом стосовно всіх абонентів Б.

Для розв'язування більшості вказаних задач достатньо скористатися фільтром та рядком підсумків. Для тих задач, які потребують порівняння даних, краще скористатися зведеними таблицями. Наприклад, задача 7) може бути вирішена за допомогою зведених таблиць (рис. 2.4.2).

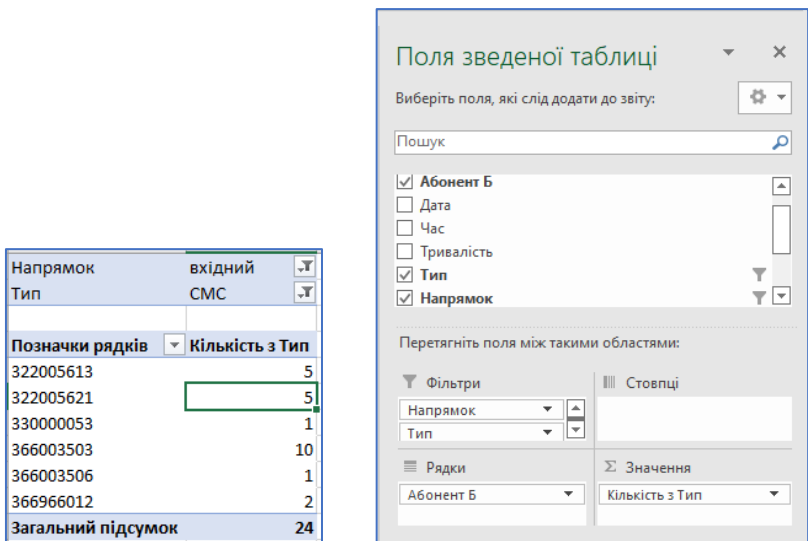


Рисунок 2.4.2

Зведена таблиця побудована з використанням трьох полів: Абонент Б, Тип і Напрямок. Поле Абонент Б утворює рядки зведеної таблиці. Поле Тип використовується для підрахунку кількості з'єднань (в даному випадку це

вхідні СМС) об'єкта з абонентами Б. Крім того, поле Тип додано як фільтр. Нарешті, поле Напрямок вибрано для фільтрації. Отже, в даному прикладі можна проглянути (навіть без сортування) кількість СМС, отриманих об'єктом від абонентів Б, та порівняти ці дані.

Іноді потрібно просто проглянути ті рядки таблиці, які містять або певний номер абонента Б, або певну дату (час) з'єднання, або, навіть, певні координати базової станції (якщо потрібно відслідкувати місцезнаходження об'єкта в певний час). В такому випадку можна скористатися *умовним форматуванням* (рис. 2.4.3). Умовне форматування дозволяє змінювати формат комірок (а саме, їх фон та колір тексту) залежно від того, що вони містять. Команда Умовне форматування знаходиться на вкладці Основне. Найчастіше використовують правило виділення клітинок – «текст, який містить...».

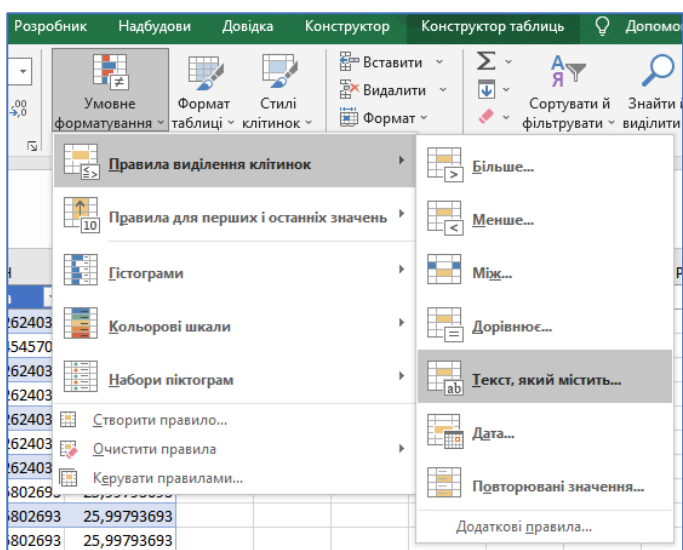


Рисунок 2.4.3

Наступна задача, яку ми розглянемо щодо аналізу трафіку з'єднань, пов'язана з побудовою карти розташування базових станцій, координати яких задано в нашій таблиці (рис. 2.4.1).

Спочатку створимо нову книгу xlsx (нехай ім'я файлу буде coordbzs.xlsx), в яку скопіюємо координати з заголовками, тобто два стовпця (широта і довгота) з таблиці трафіку з'єднань. Як правило, об'єкт

пересувається в границях певної території, що означає повторення координат базових станцій в таблиці. Отже, потрібно спростити таблицю координат в файлі coordbzs.xlsx, а саме, *видалити повтори*. Для цього, спочатку перетворимо діапазон координат в таблицю, як це ми робили в підрозділі 2.3.1. Потім відкриємо вкладку Конструктор та знайдемо на ній функцію Видалити повтори. Для нашого прикладу ми отримали наступне:

	A	B
1	Широта	Довгота
2	44,92624	26,060923
3	44,945457	26,033507
4	44,958027	25,997937
5	44,951316	26,000145
6	44,941558	26,023627
7	44,948592	26,008707
8	44,956351	26,017293
9	44,935154	26,025078
10	44,939827	26,048881
11	44,927998	26,033306
12	44,966631	25,969838
13	44,945924	26,019625
14	44,959869	25,971302

Рисунок 2.4.4

Тепер, коли файл координат вже підготовлено, ми переходимо до побудови карти.

- 1) Заходимо на Google Диск і вибираємо «Карти» (рис. 2.4.5).

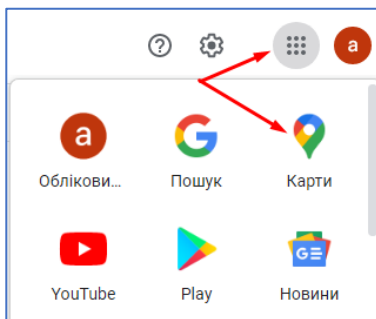


Рисунок 2.4.5

2) У вікні з картою Google відкриваємо Меню (рис. 2.4.6).

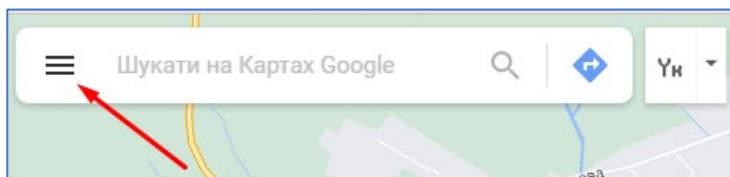


Рисунок 2.4.6

3) На вкладці Меню вибираємо «Ваші місця» (рис. 7) > «Карти» > «Створити карту». В браузері з'явиться нова вкладка з ім'ям карти (рис. 2.4.8).

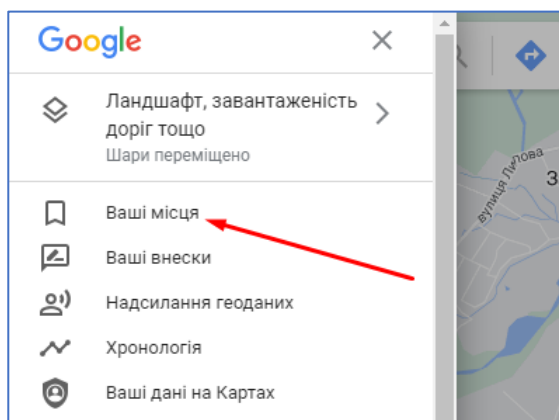


Рисунок 2.4.7

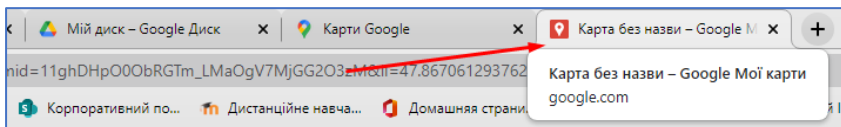


Рисунок 2.4.8

4) В новій вкладці з картою ми побачимо діалогове вікно (рис. 2.4.9). Можна редагувати назву та опис карти (при необхідності). Наступна операція це Імпорт даних з файлу *xlsx*, який містить координати.

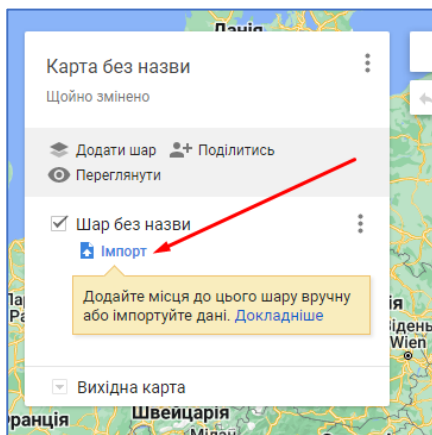


Рисунок 2.4.9

5) Після натискання на активне посилання «Імпорт» (синього коліру) відкриється вікно для вибору файлу з координатами (рис. 2.4.10). В нашому випадку ми маємо файл *coordbzs.xlsx*, який зберігається на нашому пристрої.

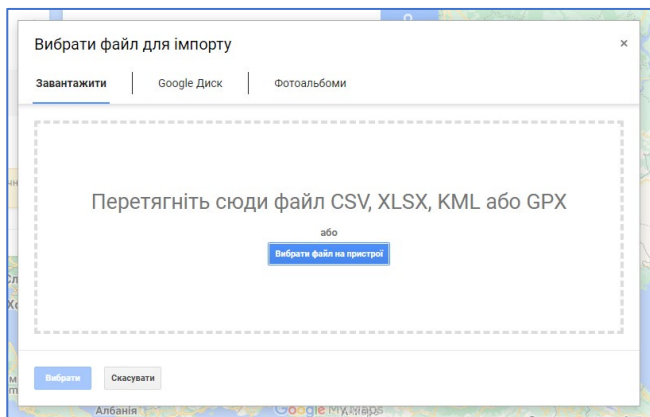


Рисунок 2.4.10

6) Подальша операція імпорту файлу супроводжується вікном (рис. 2.4.11), в якому треба зіставити широту та довготу відповідним стовпцям з таблиці у файлі coordbzs.xlsx. В нашому прикладі назви стовпців в заголовку таблиці (рис. 2.4.4) такі ж самі як у діалоговому вікні.

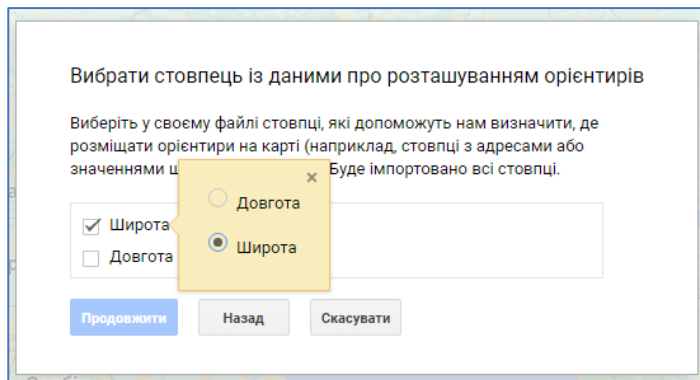


Рисунок 2.4.11

7) Після остаточного імпорту файлу з координатами з'явиться карта (рис. 2.4.12). На карті будуть позначки, які відповідають координатам в імпортованому файлі.

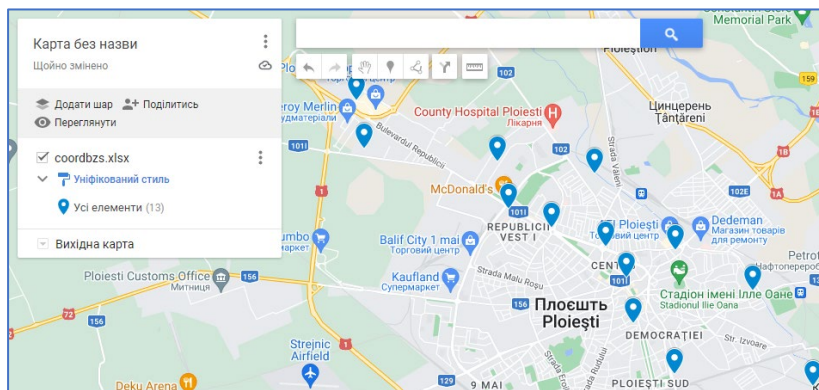


Рисунок 2.4.12

Побудову карти з позначками розташування базових станцій мобільного зв'язку можна зробити трохи простіше, використовуючи 3D-карту на вкладці Вставлення в самій книзі Excel з таблицею трафіку з'єднань.

На початку потрібно щоб курсор знаходився на якійсь комірці нашої таблиці, і далі тиснемо піктограму 3D-карта на вкладці Вставлення. Після цього з'явиться нове вікно «trafik.xlsx – 3D Maps» з картою (рис. 2.4.13).

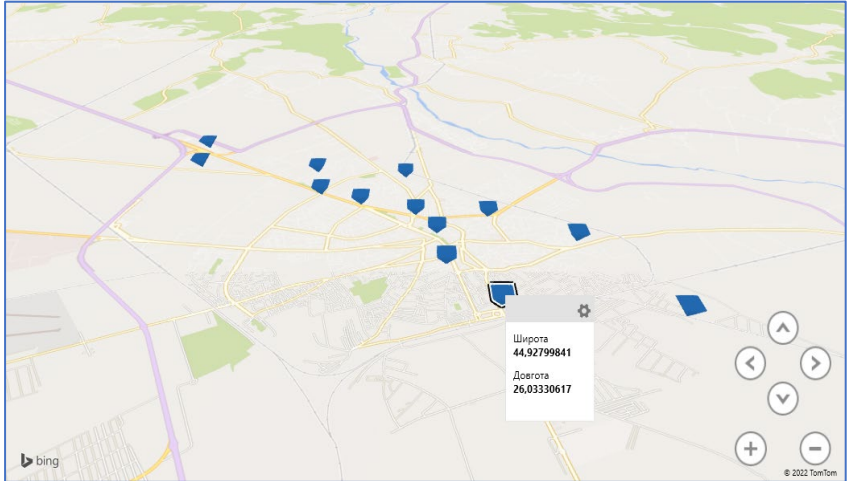


Рисунок 2.4.13

Якщо натиснути на будь-яку позначку, то можна отримати координати цього місця. Можна також додати на карту текст. Інші налаштування 3D-карти не розглядатимуться.

#### 2.4.2. Імпорт таблиці трафіку з'єднань в i2 Analyst's Notebook

Для роботи з програмою i2 Analyst's Notebook виконаємо в Excel експорт файлу xlsx з даними трафіку з'єднань в файл формату csv.

Імпорт csv-файлу в програмі i2 ANB виконаємо з *новою специфікацією*.

На вкладці Define Columns виберемо символ роздільнику тільки крапка з комою (щоб забезпечити правильний імпорт координат).

На вкладці Select Design візьмемо тип зв'язків Network of telephone calls (рис. 2.4.14).

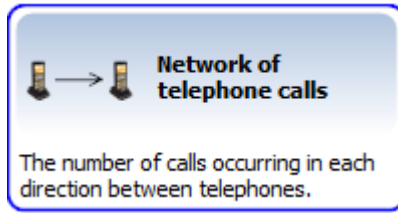


Рисунок 2.4.14

На вкладці Assign Columns зіставимо лівому об'єкту перший стовпець таблиці (Абонент А), правому – другий стовпець (Абонент Б), зв'язок – шостий стовпець (Тип) (рис. 2.4.15).

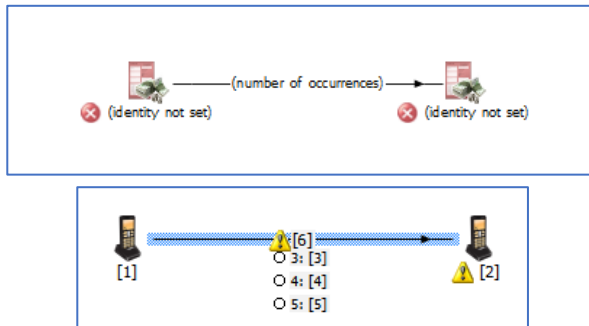


Рисунок 2.4.15

Завершуємо імпорт даних та додаємо діаграму на робоче вікно програми (рис. 2.4.16).

На етапі Assign Columns зв'язок (стовпець [6] Тип) отримав атрибути – дату, час і тривалість з'єднання (рис. 2.4.15 знизу). На діаграмі ( рис. 2.4.16) ми бачимо атрибут дати (стовпець [3]) поряд з ребром, направленим від об'єкта до абонента Б.



Рисунок 2.5.1

	A	B	C	D	E	F	G	H	I
1	Об'єкт	Абонент Б	Дата	Час	Тривалість	Тип	Напрямок	Широта	Довгота
2	366916502	366966012	01.01.2017	1:26:00	1	СМС	вхідний	44,92624033	26,06092339
3	366916502	366913695	01.01.2017	0:00:00	16	голосовий	вихідний	44,94545701	26,03350719
4	366916502	352069133	01.01.2017	16:39:00	664	голосовий	вихідний	44,92624033	26,06092339
5	366916502	351239369	02.01.2017	11:50:00	20	голосовий	вхідний	44,92624033	26,06092339
6	366916502	366913695	02.01.2017	0:00:00	39	голосовий	вхідний	44,92624033	26,06092339
7	366916502	366913695	02.01.2017	0:00:00	18	голосовий	вихідний	44,92624033	26,06092339
8	366916502	366913695	02.01.2017	18:38:00	27	голосовий	вихідний	44,92624033	26,06092339
9	366916502	366966012	03.01.2017	10:56:00	1	СМС	вихідний	44,95802693	25,99793693

Рисунок 2.5.2

1. **VLOOKUP** (ВПР) – функція пошуку та вилучення значення зі стовпця. Підтримує підстановочні знаки «\*» та «?».

Нагадаємо, що знак «\*» означає довільну кількість символів, а знак «?» - один довільний символ. Наприклад, фраза «\*хідний» може означати «вхідний», «вихідний», «західний». Фраза «?хідний» може означати «вхідний», «східний».

Синтаксис функції:

=VLOOKUP(потрібне значення; діапазон; номер стовпця в діапазоні зі значенням, що повертається; TRUE (приблизний збіг) або FALSE (точний збіг)).

Особливість функції – пошук потрібного значення завжди відбувається в першому стовпці вказаного діапазону. Відповідно, значення, які ми можемо отримати, знаходяться в стовпцях справа (!) від першого стовпця діапазону.

*Приклад 1.1.* Знайдемо напрямок виклику на номер 352069133 (рис. 2.5.2):

=VLOOKUP("352069133"; B2:G10;6; FALSE)

Замість номера телефону можна використати адрес комірки, це B4. Ще краще, використати комірку за межами таблиці (наприклад M2) для запису номера телефону, та вказати у формулі цю адресу замість номера телефону (саму формулу можна внести в комірку M3 або N2, тобто поряд з аргументами). Пошук необхідного значення відбувається до першого співпадіння потрібного значення зі значенням у стовпці B. Замість слова FALSE можна писати 0 (нуль), а замість TRUE – 1.

*Приклад 1.2.* Знайдемо тип виклику на номер, який на кінці містить цифри 33 (рис. 2.5.2):

=VLOOKUP("\*33";B2:G10;5;0)

*Приклад 1.3.* Знайдемо тип виклику на номер з комірки \$B3, якщо з'єднання відбулося за датою з комірки С6:

{=VLOOKUP(\$B\$3; IF(C2:C20=C6;B2:G20; "");5;0)}

В даному прикладі ми маємо використання функції VLOOKUP по двом умовам за допомогою формули масиву, на що вказують фігурні дужки. Коли формулу введено, то для отримання результату натискають комбінацію клавіш CTRL+Shift+Enter (фігурні дужки з'являються автоматично).

2. **MATCH** (ПОИСКПОЗ) – функція повертає номер (відносно положення) комірки у діапазоні (в рядку або в стовпці), значення якої відповідає шуканому критерію.

Синтаксис функції:

=MATCH(потрібне значення; діапазон, що переглядається; [тип зіставлення -1, 1 або 0])

*Приклад 2.1.* Знайдемо відносно положення номеру в стовпці В (рис. 2.5.2), який занесено в комірку К2:

=MATCH(К2; Таблиця2[Абонент Б];0)

Якщо комірка К2 містить номер 351239369, то результатом виконання функції буде число 4 (рис. 2.5.2).

3. **INDEX** (ИНДЕКС) – функція повертає значення комірки, використовуючи для її пошуку значення зміщення у рядку та стовпці заданого діапазону.

Синтаксис функції:

=INDEX(діапазон; номер рядка; [номер стовпця])

*Приклад 3.1.* Знайдемо вміст комірки з діапазону В2:G10 (рис. 2.5.2), яка розташована в цьому діапазоні у 2 рядку та 4 стовпці (фактично, це комірка Е3):

=INDEX(В2:G10;2;4).

Формула повертає значення 16.

*Приклад 3.2.* Розглянемо приклад спільного використання функцій INDEX та MATCH. Нехай формула повинна відшукати потрібний номер у стовпці В та повернути тип з'єднання. Щоб краще організувати пошук, у комірку К8 занесемо номер телефону та у комірку К9 – назву стовпця, тобто текст «Тип». До речі, ми можемо поміняти вміст комірки К9, якщо нам потрібно знайти дату або напрямок та т.і.:

=INDEX(В2:G20;MATCH(К8; Таблиця2[Абонент Б];0);MATCH(К9;В1:G1;0))

Якщо порівнювати комбінацію INDEX+MATCH з функцією VLOOKUP, то треба вказати на певні обмеження останньої, в той час коли вказана комбінація виконує навіть «лівий VLOOKUP».

*Приклад 3.3.* Знайдемо номер абонента Б, з яким відбувся зв'язок об'єкта на певну дату. Таких зв'язків впродовж доби може бути декілька, але ми отримаємо тільки перший номер. У комірку K12 занесемо дату:

=INDEX(Таблиця2[Абонент Б];MATCH(K12; Таблиця2[Дата];0))

Якщо занесена дата 03.01.2017, то отримаємо номер 366966012 (рис. 2.5.2).

4. **COUNTA** (СЧЁТЗ) – функція повертає число непорожніх комірок у вказаному діапазоні. Приклад: COUNTA(A3:D8). Формула підраховує комірки, що містять числові значення, текст і логічні значення. Якщо в комірці міститься формула, яка повертає порожній рядок, то вона теж рахується, незважаючи на те, що в комірці нічого не відображається.

Синтаксис функції:

=COUNTA(діапазон)

*Приклад 4.1.* Знайдемо кількість непорожніх комірок у першому стовпці таблиці (відповідно, це кількість рядків таблиці):

=COUNTA(Таблиця1[Джерело];)

5. **COUNT** (СЧЁТ) – функція повертає число комірок у вказаному діапазоні, які містять тільки числові значення. Комірки, що містять значення дат і часу, також вважаються числовими. Комірки, що містять логічні значення (FALSE, TRUE), такими не вважаються.

Синтаксис функції:

=COUNT(діапазон)

*Приклад 5.1.* Знайдемо кількість непорожніх комірок у стовпці з датами в таблиці банківських транзакцій (рис. 2.5.1):

=COUNT(Таблиця1[Дата])

*Приклад 5.2.* Знайдемо кількість комірок з числовими значеннями у рядку таблиці банківських транзакцій:

=COUNT(A2:E2)

Таким чином, можна перевірити кількість комірок таблиці числового формату.

6. **COUNTIF** (СЧЁТЕСЛИ) – функція підраховує кількість комірок в одному діапазоні, значення в якому задовольняють певному критерію.

Синтаксис функції:

=COUNTIF(діапазон; критерій)

*Приклад 6.1.* Знайдемо кількість транзакцій, які відбулися з певного рахунку, наприклад '07989967213', який збережено у комірці G5:

=COUNTIF(Таблиця1[Джерело]; G5)

*Приклад 6.2.* Знайдемо кількість транзакцій, які перевищують певну суму переказу (вказану у комірці G7):

=COUNTIF(Таблиця1[Сума коштів]; ">"&G7)

*Приклад 6.3.* Знайдемо кількість комірок у рядку таблиці трафіку з'єднань, які мають текстовий формат:

=COUNTIF(A2:I2;"\*")

Результатом виконання функції буде число 4.

*Приклад 6.4.* Знайдемо в таблиці трафіку з'єднань кількість СМС, які надійшли об'єкту, або були їм відправлені:

=COUNTIF(Таблиця2[Тип];"СМС")

Для підрахунку числа **унікальних** значень в діапазоні використовують формулу масиву:

{=SUM(1/COUNTIF (масив; масив))}.

*Приклад 6.5.* Знайдемо число унікальних рахунків, з яких перераховуються кошти:

=SUM(1/COUNTIF(Таблиця1[Джерело]; Таблиця1[Джерело]))

**7. COUNTIFS (СЧЁТЕСЛИМН)** – функція підраховує кількість комірок в декілька діапазонах на підставі критеріїв до кожного з них. Це приклад множинного критерію.

Синтаксис функції:

=COUNTIFS(діапазон1; критерій1; діапазон2; критерій2;...)

*Приклад 7.1.* Знайдемо кількість вхідних голосових з'єднань в таблиці трафіку з'єднань:

=COUNTIFS(F2:F120;"голосовий";G2:G120;"вхідний")

До речі, той же результат можна отримати, якщо використати наступну формулу:

=SUMPRODUCT((Таблиця2[Тип]="голосовий")\*(Таблиця2[Напрямок]="вхідний"))

**8. SUMIF (СУММЕСЛИ)** – функція умовного підсумовування.

Синтаксис функції:

=SUMIF(діапазон; критерій; діапазон підсумовування)

*Приклад 8.1.* Знайдемо суму переказів (рис. 2.5.1) з певного рахунку, заданого в комірці G5:

=SUMIF(Таблиця1[Джерело];G5;Таблиця1[Сума коштів])

*Приклад 8.2.* Знайдемо суму тих переказів (рис. 2.5.1), які перевищують певне значення (яке задане в комірці H2):  
=SUMIF(Таблиця1[Сума коштів];">"&H2)

### **Питання для самоперевірки:**

1. Приклади некоректного імпорту структурованих даних в Excel.
2. Опис кроків майстра імпорту текстових файлів в Excel.
3. Опис кроків імпорту табличних даних в i2 Analyst's Notebook.
4. Перетворення діапазону з даними в таблицю Excel.
5. Характерні ознаки таблиці Excel на відміну від звичайного діапазону з даними.
  6. Команди вкладки Конструктор.
  7. Способи розділення дати і часу на різні стовпці.
  8. Зміст проміжних підсумків щодо таблиці банківських транзакцій.
  9. Аргументи функції SUBTOTAL.
  10. Зміст задач при аналізі банківських транзакцій, які вирішуються з використанням зведених таблиць.
  11. Які коректні дані можна отримати в рядку підсумків для таблиці трафіку з'єднань?
  12. Які задачі при аналізі трафіку з'єднань можуть бути вирішені за допомогою зведених таблиць?
  13. Які задачі при аналізі банківських транзакцій та трафіку з'єднань можна вирішити за допомогою умовного форматування?
  14. Як можна видалити повтори даних в рядках таблиці?
  15. Основні кроки створення особистої Google карти.
  16. Побудова 3D-карти.
  17. Особливості створення нової специфікації при імпорті даних трафіку з'єднань в i2 Analyst's Notebook.
  18. В чому особливість функції VLOOKUP?
  19. Як використати функцію VLOOKUP по двом умовам?
  20. Чим відрізняється відносно положення комірки з шуканим значенням від абсолютного?
  21. Що повертає функція MATCH – положення чи значення комірки?
  22. Що повертає функція INDEX – положення чи значення комірки ?
  23. Чим схожі та чим відрізняються функції COUNTA та COUNT?
  24. Приведіть приклади критеріїв для функції COUNTIF.
  25. Чим схожі та чим відрізняються функції COUNTIF та COUNTIFS?
  26. Приведіть приклади критеріїв для функції SUMIF.

### 3. Аналіз даних в Python

#### 3.1. Відомості про Python

Для того щоб працювати з Python, потрібно встановити його інтерпретатор на своєму комп'ютері. Заходимо на офіційний сайт <https://www.python.org/downloads/windows/> (рис. 3.1.1), виконуємо загрузку файлу `python-3.10.5-amd64.exe` та встановлюємо. Як правило, разом з інтерпретатором ми побачимо Python Module Docs та IDLE.

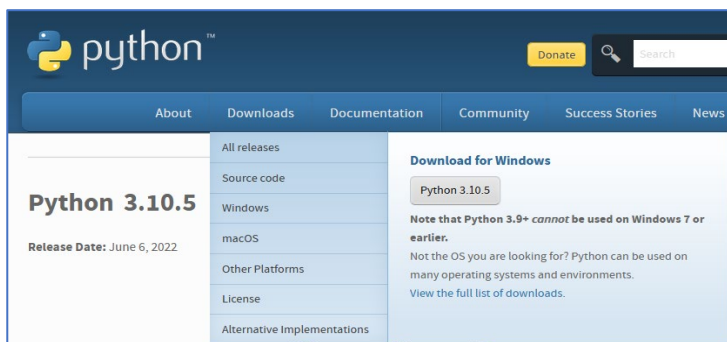


Рисунок 3.1.1

1. **IDLE** (Integrated Development and Learning Environment) – це інтегроване середовище розробки та навчання, яка є альтернативою командному рядку (рис. 3.1.2). IDLE створене за допомогою бібліотеки Tkinter, яка вбудована в Python в якості стандартного модуля. До речі, Tkinter (від англійської Tk interface) – це багатоплатформна бібліотека, яка служить для створення програм з графічним інтерфейсом.

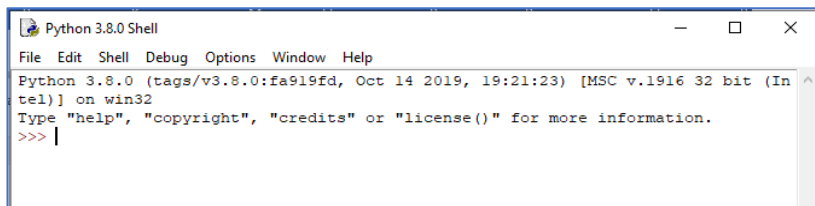


Рисунок 3.1.2

IDLE є найпростішим програмним редактором, який дозволяє писати програми, зберігати у файли та запускати їх. Редактор коду використовує підсвічування синтаксису. Однак, існують інші редактори, які дозволяють робити розробку максимально продуктивною завдяки функціям автодоповнення та аналізу коду. До таких редакторів можна віднести PyCharm, який доведеться встановлювати окремо.

**2. Anaconda.** Можна об'єднати всі зусилля по встановленню інтерпретатора, пошуку редактору коду, встановленню додаткових модулів і бібліотек, якщо встановити пакет Anaconda – дистрибутиви Python [9, 10]. Цей пакет надає все необхідне для вирішення завдань щодо аналізу та обробки даних. До пакету Anaconda входить Anaconda Navigator – графічний інтерфейс користувача, який дозволяє запускати пов'язані програми та керувати пакетами (рис. 3.1.3). У Навігаторі за замовчуванням доступні такі програми, як Jupyter Notebook та Spyder. Зокрема, Spyder і є кросплатформова інтерактивна IDE для наукових розрахунків мовою Python.

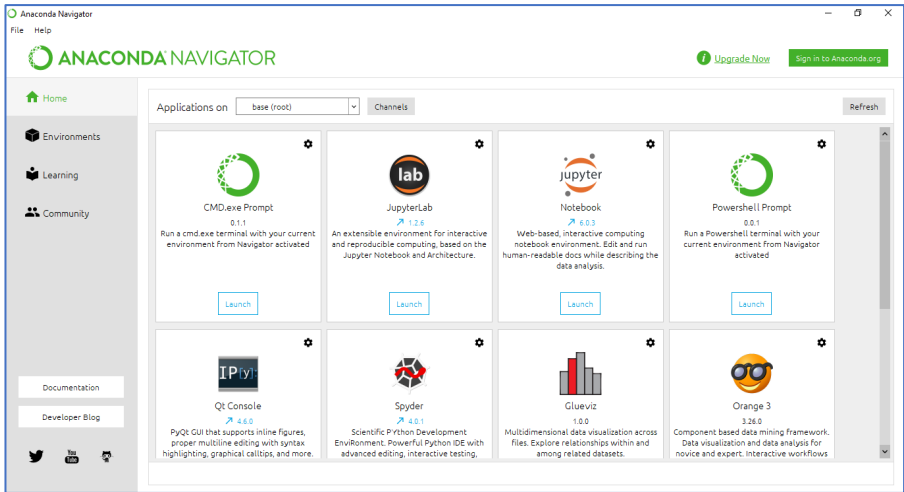


Рисунок 3.1.3

**3. Spyder.** Робоче вікно редактора Spyder розділено на дві панелі (рис. 3.1.4). В панелі зліва розміщують скрипти, а в правій панелі проглядають результати виконання коду. Права панель фактично є вбудованою консоллю (IPython console), в якій, крім виводу виконання коду, можна виконувати різні

скрипти (команди). Наприклад, отримати довідку для команди `pip`, яка призначена для встановлення та управління різними пакетами та бібліотеками Python.

Однією з цікавих особливостей Spyder є можливість перегляду значень змінних. У процесі виконання програми змінні виводяться на панелі Variable Explorer (справа, зверху) у вигляді списку з інформацією про ім'я, тип, розмір і значення змінної.

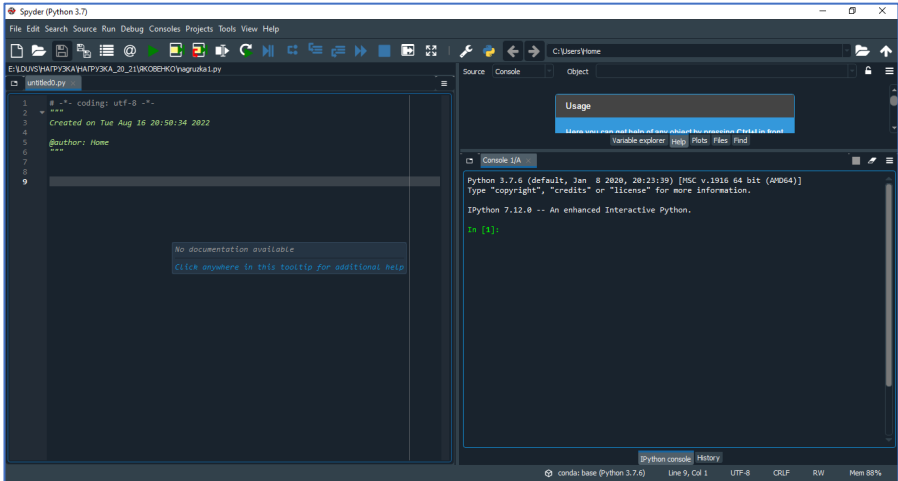


Рисунок 3.1.4

Взагалі, Spyder має унікальне поєднання можливостей – просунуте редагування, аналіз та глибока інспекція коду, гарна візуалізація.

**4. Синтаксис** мови Python має певні особливості, основні з яких ми розкриємо.

- Оператором присвоєння значень змінним є знак «`=`».
- Кінець рядка є кінцем інструкції, причому прикінцеві символи непотрібні.
- Можна записати кілька інструкцій (операторів) в одному рядку, розділяючи їх крапкою з комою.
- Можна записувати одну інструкцію в декілька рядків. Для цього необхідно розмістити її в парі круглих, квадратних або фігурних дужок.
- Якщо є основна інструкція та вкладені до неї інструкції (вкладений блок інструкцій), то основна інструкція завершується двокрапкою, наступні

рядки вкладеної інструкції розташовують з однаковим відступом (як мінімум 4 пробіли) від початку рядків по відношенню до основної інструкції. Якщо тіло вкладеної інструкції містить єдиний оператор, то його можна записати в тому ж рядку, що і основна інструкція.

- На початку нового рядка, або в рядку в кінці інструкцій можна розміщувати коментар після символу # (назва символу # – решітка, шарп, хеш). При виконанні коду, Python ігнорує текст, наступний за символом #. Так можна «коментувати» попередні варіанти коду (щоб в разі необхідності можна було до них повернутися), коли пробуються різні рішення.

Запуск програми для виконання коду (набір команд або інструкцій) на Python можна організувати в IDLE Python 3.8.0 Shell, Spyder, Jupyter Notebook, командному рядку операційної системи (shell або термінал), файлі з текстом програми (пакетний режим). У випадку командного рядка, спочатку виконують команду зі словом python, а потім з'явиться запрошення для вводу команд у вигляді >>>.

Надалі, для демонстрації прикладів, ми будемо користуватись редактором коду Spyder.

**5. Ідентифікатор** – це ім'я деякої сутності (змінної, функції, класу) в програмі для її позначення. При виборі ідентифікаторів першим символом може бути буква з алфавіту або символ підкреслення «\_». Інша частина ідентифікатора може складатися з букв, символу підкреслення або цифр. Python чутливий до регістру букв в іменах змінних, функцій, операторів, класів (великі і маленькі літери вважаються різними).

**6.** В мові Python використовують неявну строгу динамічну типізацію, тобто при оголошенні змінної її тип не вказують (при явній типізації тип змінної вказується обов'язково). Для мов з динамічною типізацією тип змінної визначається безпосередньо при виконанні програми і може довільно змінюватися по ходу виконання коду, коли змінна починає посилатися на інший об'єкт. Строга типізація не дає можливості проводити операції у виразах з даними різних несумісних типів. Отже, тип даних – це множина значень і операцій на цими значеннями.

**7. Виведення та введення даних до програми.** Для виведення результату роботи програми використовують функцію print(), яка має наступний формат:

```
print(value1,...)[,sep=' '][,end='\n'][, file=sys.stdout]
```

Функція `print()` може виводити одне або відразу декілька значень `value` різного типу, відділених один від одного комами. На друку, один від одного значення будуть відокремлені пробілом, якщо параметру `sep` не було привласнено якогось строкового значення. Після виведення всіх значень кінець рядку буде доповнений літералом (символом чи декількома символами), вказаними в параметрі `end` (за замовчуванням символ закінчення рядка та переходу на новий).

За замовчуванням, функція `print()` виводить значення на екран (в стандартній пристрій виведення `sys.stdout`). Параметр `file` дозволяє перенаправити виведення значень в файл.

Якщо на початку, або в процесі роботи програми, необхідно надати значення деякій змінній, то використовують функцію `input()`. Потрібно пам'ятати, що функція `input()` повертає значення рядкового типу. Пояснимо,

```
In [1]: a = input("введіть значення a - ");
b = input("введіть значення b - "); print("a + b = ", a + b)
введіть значення a - 7
введіть значення b - 9
a + b = 79
```

Щоб отримати суму чисел (а не з'єднання двох символів), змінні потрібно перетворити у цілі числа, наприклад,

```
a = int(input("введіть значення a - "))
```

## 8. Структури даних.

У мові Python існують вбудовані структури даних:

- Рядки (Text Sequence Type): `str`.
- Послідовності (Sequence Type): `list` – список, `tuple` – кортеж, `range` – діапазон.
- Множини (Set Types): `set`.
- Словники (Mapping Types): `dict`.

### 8.1. Рядки `str`.

Рядок – це набір символів (букв, цифр, спеціальних символів), або звичайний текст. Python сприймає набір символів як рядок, якщо його заключено в одинарні або подвійні рядки.

Рядковий тип є незмінюваним, тобто зміна значення символу рядка чи його вилучення є неможливими.

Порожній рядок задається так: `s = ''`. Числове значення можна перетворити в рядок або так: `'4534'`, або `str(4534)`.

Деякі рядки можна об'єднати в один за допомогою знака «+».

Багаторазове повторення рядка або дублювання рядка можна виконати за допомогою операції множення «\*»:

```
In [2]: s = 'POL'*3; print(s)
'POLPOLPOL'
```

Кожен символ рядка має свій індекс (порядковий номер). Отже, доступ до символів рядка відбувається за їх індексами. Нумерація символів рядка починається з нуля. Для того, щоб звернутися до певного символу рядка, необхідно вказати ім'я рядкової змінної та в квадратних дужках індекс необхідного символу. Індеси можуть бути від'ємними, тоді нумерація буде відбуватися з кінця (кількість символів рядка + від'ємний індекс). Якщо вказати індекс, який виходить за межі рядка, то генерується виняток `IndexError`.

Перевірити приналежність деякого символу або підрядка до рядка, використовують оператор `in`.

Для отримання з рядка групи символів використовують зрізи (slice). Зріз рядка також є рядком. Вкажемо можливі варіанти утворення зрізу для рядка `T`:

<code>T[start: stop: step]</code>	<code>T[start:]</code>
<code>T[start: stop]</code>	<code>T[:stop]</code>
<code>T[start:: step]</code>	<code>T[::-step]</code>
<code>T[:stop: step]</code>	

де `start` і `stop` є індекси, які вказують з якого по який символ зробити зріз рядку, `step` – крок руху по рядку. За замовчуванням, `start = 0`, `stop =` кількість символів рядку, `step = 1`. Приклади:

```
In [3]: T = 'Information'; TS1 = T[3:9:2]; TS1
Out[3]: 'omt'
```

```
In [4]: TS2 = T[2:7]; TS2
Out[4]: 'forma'
```

```
In [5]: TS3 = T[2:]; TS3
Out[5]: 'formation'
```

```
In [6]: TS4 = T[:4]; TS4
Out[6]: 'Info'
```

```
In [7]: TS5 = T[::3]; TS5
Out[7]: 'Ioao'
```

Кількість символів у рядку (довжину рядка) знаходять за допомогою функції `len`. Наприклад,

```
In [8]: len(T)
Out[8]: 11
```

Для роботи з рядками в Python передбачена велика кількість вбудованих методів, які, по-більшості, відшукують розташування певного підрядка в рядку, або визначають певні властивості рядка (непорожній, складається лише з цифр і т.п.)

## 8.2. Послідовності.

**Списки (list)** – структура даних для зберігання елементів (об'єктів) одного або різних типів. Приклади задання списку з наперед заданим набором елементів:

```
a = [0994432781, 0662431882, 0953332991, 0971533770, 0502538786]
b = [5, 'Тип', 'Координати', 'Об'єкт', 23]
c = [[44.92624033, 26.06092339], [44.94545701, 26.03350719], [44.92624033, 26.06092339], [44.92624033, 26.06092339]]
```

Порожній список можна задати так: `a = []` або `a = list()`.

Приклад створення списку з інших структур даних – рядка і діапазону:

```
In [9]: b2 = list("Traffic"); print(b2)
['T', 'r', 'a', 'f', 'f', 'i', 'c']
```

```
In [10]: c2 = list(range(7)); c2
Out[10]: [0, 1, 2, 3, 4, 5, 6]
```

Нові списки можна створити методом конкатенації (об'єднання) декількох списків в один за допомогою операції додавання («+»):

```
In [11]: d = b2 + c2; print(d)
['T', 'r', 'a', 'f', 'f', 'i', 'c', 0, 1, 2, 3, 4, 5, 6]
```

За допомогою операції множення «\*» можна створити список з повторів деякого списку:

```
In [12]: a3 = ['N', 'V']*3; print(a3)
['N', 'V', 'N', 'V', 'N', 'V']
```

Доступ до елементів списку відбувається за їх індексами. Нумерація елементів починається з нуля. Приклад:

```
In [13]: b3 = ['T', 'r', 'a', 'f', 'f', 'i', 'c']; b3[0]
Out[13]: 'T'
```

Індекси можуть бути від'ємними, в такому випадку нумерація буде відбуватися з кінця (кількість елементів списку + від'ємний індекс):

```
In [14]: b3[-1]
Out[14]: 'c'
```

Використовуючи оператор `in`, можна перевірити приналежність деякого елемента до списку:

```
In [15]: 'a' in b3
Out[15]: True
```

Для протилежної перевірки, що елемент не належить списку, може бути використаний оператор `not in`.

Виведення елементів списку окремими по рядкам реалізується за наступною конструкцією:

```
In [16]: for i in b3: print(i)
```

Виведення елементів списку через пробіл:

```
In [17]: for i in b3: print(i, end=' ')
```

За допомогою **зрізів** можна отримати список елементів, вибраних з даного списку за певним правилом [2, с. 76].

Елементи списку можна змінювати звичайним присвоєнням, використовуючи індекс елемента. Наприклад,

```
In [18]: b4 = [1, 0, 3, 5, 8]; b4[1] = 2; print(b4)
[1, 2, 3, 5, 8]
```

Команда `del` дозволяє вилучати елементи зі списку:

```
In [19]: del b4[4]; print(b4)
[1, 2, 3, 5]
```

Зміна або вилучення відразу декількох елементів виконується аналогічно за допомогою зрізів.

### Методи списків.

- `list.append(x)`. Додає елемент `x` в кінець списку `list`.

```
In [20]: A = [2,3]; A.append(5); A
Out[20]: [2, 3, 5]
```

- `list.insert(n, x)`. Вставляє в список `list` елемент `x` в позицію `n` (індекс елемента, після якого буде вставлений елемент).
- `list.remove(x)`. Вилучає перше входження елемента `x` зі списку `list`.

```
In [21]: B = ['a', 'f', 'f', 'f', 'c']; B.remove('f'); B
Out[21]: ['a', 'f', 'f', 'c']
```

- `list.pop([n])`. Вилучає з списку `list` елемент з позиції `n` та повертає його, як результат виконання функції. Якщо використовувати метод без параметру, то буде вилучений останній елемент списку.
- `list.clear()`. Очищує список `list` (вилучає всі елементи зі списку).

- `list.index(x[, start[, end]])`. Повертає індекс першого входження елемента `x` в зрізі `list[start: end]`. Значення, що повертається, є індексом списку `list`. Якщо елемента в списку не знайдено, виникає виняток `ValueError`.
- `list.count(x)`. Повертає кількість входжень елемента `x` в список.

```
In [22]: K = B.count('f'); K
```

```
Out[22]: 2
```

▪ `list.sort(key=None, reverse=False)`. Відсортовує елементи списку (аргументи методу можуть бути використані для налаштування сортування). За замовчуванням сортування відбувається за зростанням. Для сортування в зворотному порядку використовуйте параметр `reverse = True`. В результаті сортування змінюється сам список.

▪ `list.reverse()`. Змінює порядок розташування елементів у списку на зворотний. Змінюється сам список.

▪ `list.copy()`. Повертає копію списку.

**Кортеж (tuple)** – це незмінна структура даних, яка схожа на список. Кортеж може містити елементи різних типів. Кортеж записується, як перелік елементів, розділених комою та взятих в круглі дужки: ('l', 'm', 4, 'F').

Преваги кортежів:

- Змінити елементи заданого кортежу неможливо.
- Кортеж в пам'яті займає менший об'єм чим список з тими ж елементами.
- Кортежі працюють швидше, ніж списки.
- Кортежи доцільно використовувати в якості ключа у словнику.

Для задання порожнього кортежу можна скористатися однією з наступних команд: `t = ()` або `t = tuple()`.

Будь-який набір різних об'єктів, розділених комами та не виділених будь-якими дужками (тобто ні квадратними, ні фігурними, ні круглими) за замовчуванням буде вважатися кортежем.

Для створення кортежу з одного елемента треба ставити кому (,) після елемента, наприклад: `t = (5,)`

Кортеж можна отримати з елементів об'єкта, що може ітеруватися (діапазон, рядок, словник, множина, кортеж, файл), використавши функцію `tuple([iterable])`:

```
In [23]: t1 = tuple(range(1,10,2)); print(t1)
```

(1, 3, 5, 7, 9)

Нові кортежі можуть бути створені методом об'єднання декількох кортежів за допомогою знаку «+».

Можна виконати багаторазове повторення елементів за допомогою знаку «\*»:

```
In [24]: t2 = ('m',)*4; print(t2)
('m', 'm', 'm', 'm')
```

Доступ до елементів кортежу здійснюється за їхніми індексами:

```
In [25]: t3 = ('1', 'm', 4, 'F'); print(t3[2])
Out[25]: 4
```

Можна задати зріз кортежа:

```
In [26]: print(t3[1:3])
('m', 4)
```

Можна перевірити приналежність деякого елемента до кортежу, використовуючи оператор in.

Нагадаємо, що при спробі змінити чи вилучити елемент кортежу виникне виняток TypeError:

```
In [27]: t3[1] = 'n'
TypeError: 'tuple' object does not support item assignment
```

```
In [28]: del(t3[1])
TypeError: 'tuple' object doesn't support item deletion
```

З кортежу можна створити список (функція list), як і навпаки – зі списку можна створити кортеж (функція tuple).

Враховуючи незмінюваність кортежів, вони мають лише такі методи:

- tuple.index(x[, start[, end]])
- tuple.count(x)

### Діапазон (range).

Діапазон є незмінною послідовністю цілих чисел. Для задання діапазону пишуть: `range(M)` – послідовність цілих чисел від 0 до  $M - 1$  з кроком 1.

Функція `range(A, M, K)` задає послідовність цілих чисел від  $A$  до  $M - 1$  з кроком  $K$ . Приклади:

`range(5)` – це діапазон 0, 1, 2, 3, 4

`range(1, 6)` – це діапазон 1, 2, 3, 4, 5

`range(0, 10, 3)` – це діапазон 0, 3, 6, 9

### 8.3. Множини (set).

Множина - це структура даних, що містить непорядкований набір унікальних елементів. Множина є змінюваним типом даних, тобто до множин можна додавати нові та видаляти існуючі елементи.

Множини записуються, як перелік елементів, розділених комою та взятих в фігурні дужки: `{'a', 7, [0,1], 'world'}`.

Порожня множина задається за допомогою функції `set()`, або просто парою дужок: `{}`.

Множину можна отримати з елементів таких об'єктів, як діапазон, список, рядок, словник, кортеж, за допомогою функції `set`. Проте, до множини будуть включені лише унікальні елементи. Наприклад,

```
In [29]: a = set('bullet'); a
```

```
Out[29]: {'b', 'e', 'l', 't', 'u'}
```

```
In [30]: set(range(5))
```

```
Out[30]: {0, 1, 2, 3, 4}
```

Щоб перевірити приналежність деякого елемента до множини, використовують оператор `in`.

Для додавання елемента до множини викликають метод `.add(x)`. Для вилучення елемента із множини існує метод `.remove(x)`.

Функція `len` повертає довжину множини.

### 8.4. Словники (dict).

Словник (`dict`) – це структура даних, призначена для зберігання довільних об'єктів з доступом за ключем. Дані в словнику зберігаються в форматі – ключ : значення. Ключі в межах словника мають бути унікальними,

тобто в словнику двох однакових ключів бути не може. Ключ повинен мати незмінюваний тип даних: число, рядок, кортеж.

Словник є змінюваним типом даних, в нього можна додавати нові елементи і вилучати вже існуючі.

Порожній словник задають так: `ad = {}`, або, `ad = dict()`.

Словник записується як перелік пар (ключ:значення), розділених комою: `ad = {'01':250, '02':347, '03':119}`, або, `ad = dict{'01'=250, '02'=347, '03'=119}`.

Доступ до значення елемента словника за його ключем (використовують квадратні дужки):

```
In [31]: ad = {'01':250, '02':347, '03':119}; print(ad['02'])
347
```

Видалення елемента зі словника: `del ad[ключ]`.

```
In [32]: del ad['02']; print(ad)
{'01':250, '03':119}
```

Приклади методів словників.

- Метод `ad.keys()` повертає ключі словника `ad` (у вигляді списку):

```
In [33]: ad = {'01':250, '02':347, '03':119}; ad.keys()
dict_keys(['01', '02', '03'])
```

- Метод `ad.values()` повертає значення словника `ad` (у вигляді списку):

```
In [34]: ad = {'01':250, '02':347, '03':119}; ad.values()
dict_values([250, 347, 119])
```

- Сортування словника `ad` за ключами:

```
ad1 = dict(sorted(ad.items(), key=lambda x: x[0]))
```

- Сортування словника `ad` за значеннями:

```
ad2 = dict(sorted(ad.items(), key=lambda x: x[1]) )
```

## 9. Логічні вирази, логічні оператори.

Логічним (булевым) типом даних в мові Python є тип `bool`, що може набувати одного з двох значень: `True` (істина) або `False` (хиба).

Проте в мові Python істинним або хибним може бути не лише логічний вираз, але і об'єкт. Число не рівне нулю, або непорожній об'єкт інтерпретується як істина. Нуль, порожні об'єкти і спеціальний об'єкт `None` інтерпретуються як хиба.

В Python для порівняння об'єктів (змінних різних типів) є наступні операції порівняння: `>` (більше); `<` (менше); `>=` (більше або рівне (не менше)); `<=` (менше або рівне (не більше)); `==` (дорівнює (рівне)); `!=` (не дорівнює (не рівне)).

Результатом операції порівняння є змінна логічного типу `bool`.

```
In [35]: [1,2,3]<[1,3,2]
```

```
Out[35]: True
```

```
In [36]: list(range(1,4))==[1,2,3]
```

```
Out[36]: True
```

```
In [37]: list((1,2,3))!= [1,2,3]
```

```
Out[37]: False
```

Для побудови логічних виразів використовують логічні оператори: `not`, `or`, `and`.

Логічний оператор **not** (НЕ) називають запереченням. Пишуть: **not x**. Результатом застосування логічного оператора `not` є значення логічного типу. Якщо операнд істинний (`True`, будь-яке число не рівне нулю, або не порожній об'єкт), то оператор `not` поверне – `False`. І навпаки, якщо операнд хибний (`False`, нуль, порожній об'єкт або спеціальний об'єкт `None`), то буде повернуто `True`.

Логічний оператор **and** (І) називають кон'юнкцією або логічним множенням. Пишуть: **x1 and x2**. При обчисленні оператора `and` операнди обчислюються зліва направо. Як тільки знайдено перший об'єкт, що має хибне значення, він вважається результатом обчислення оператора `and`, і подальше обчислення завершується. Якщо серед операндів не знайдено об'єкта, що має хибне значення, то повертається крайній правий об'єкт.

Логічний оператор **or** (АБО) називають диз'юнкцією або логічним додаванням. Пишуть: **x1 or x2**. При обчисленні оператора `or` операнди

обчислюються зліва направо. Як тільки знайдено перший об'єкт, що має істинне значення, він вважається результатом обчислення оператора `or` і подальше обчислення завершується. Якщо серед операндів не знайдено об'єкта, що має істинне значення, то повертається крайній правий об'єкт.

Можна помітити певну особливість роботи операторів `and` і `or`. Вони «закорочують» обчислення своїх операндів: наступний правий операнд обчислюється лише в тому випадку, якщо його значення необхідне для отримання остаточного значення операцій `and` чи `or`. Оператори `and` і `or` не приводять свій результат примусово до значень `True` чи `False`, а повертають один із своїх операндів.

Оператор `not` має нижчий пріоритет, ніж оператори порівняння. Наприклад, інструкція `not a == b` інтерпретується як `not (a == b)`. Важливо, що, вираз `a == not b` є синтаксичною помилкою.

З булевих операторів, `not` має найвищий пріоритет, а `or` найнижчий, так що `A and not B or C` еквівалентно `(A and (not B)) or C`. Як завжди, дужки можуть бути використані для вираження бажаного пріоритету в операціях.

## 10. Оператори розгалуження та цикли

Умовний оператор `if-else`. Синтаксис оператора:

```
if логічний_вираз:  
    Блок_інструкцій_1  
else:  
    Блок_інструкцій_2
```

Якщо значення логічного виразу `True`, то виконується Блок\_інструкцій\_1. Якщо значення логічного виразу `False`, виконується Блок\_інструкцій\_2.

Цикл з передумовою `while`. Синтаксис оператора:

```
while логічний_вираз:  
    Блок_інструкцій
```

Логічний вираз називають умовою виконання циклу. Набір інструкцій виконується до тих пір, поки умова циклу істинна. При виконанні циклу спочатку обчислюється значення логічного виразу, якщо це значення є істинним, то виконується тіло циклу і відбувається повернення до перевірки логічного виразу. Процес продовжується доти, поки значення логічного виразу не набуде значення `False`. Після цього робота циклу завершиться і відбувається перехід до інструкції після тіла циклу.

Цикл for. Синтаксис оператора:

```
for ідексна_змінна in послідовність:  
    Блок_інструкцій
```

Індексна змінна – це набір, що підтримує ітерування (послідовності, рядки, множини, словники). Блок інструкцій виконується для кожного елемента набору послідовно. При такому опрацюванні не потрібно турбуватися про індекси елементів і їх кількість. Цикл for закінчує роботу при умові закінчення елементів набору.

### **Інструкції управління циклами.**

Оператор **continue**. Призначений для переривання поточної ітерації циклу і переходу до наступної ітерації (до наступного елемента послідовності). Тобто оператори, що будуть іти в тілі циклу після виклику continue, на даному кроці циклу виконуватися не будуть. Дуже часто оператору continue передує оператор if для перевірки певної умови.

Оператор **break**. Призначений для дострокового припинення роботи циклу (for або while). Інструкцію break варто викликати всередині інструкції if при виконанні якоїсь особливої умови.

## **11. Імпортування модулів.**

Доволі часто необхідне імпортування (підключення) модулів (бібліотек) до власної програми для отримання доступу до їх атрибутів (елементів). Прикладом таких модулів, які розглядаються в цьому посібнику, є openpyxl, os, matplotlib, folium. Для підключення модулів використовують наступні інструкції.

Інструкція **import**.

Приклади:

```
import openpyxl  
import folium
```

Назва модуля становиться змінною, через яку можна отримати доступ до атрибутів модуля, наприклад,  
wb = openpyxl.open("banktransactions.xlsx", data\_only=True)

Назву модуля можна замінити на **псевдонім**. Наприклад,  
import matplotlib as mpl  
import matplotlib.pyplot as plt

В такому випадку для звернення до атрибутів модуля використовують псевдонім:

```
fig = plt.figure(dpi = dpi, figsize = (512/dpi, 384/dpi))
```

Інструкція **from ... import**.

Цю інструкцію використовують якщо немає сенсу підключати весь модуль, а лише деякі атрибути модуля:

```
from openpyxl.styles import Alignment, PatternFill, Font
```

До речі, назву атрибуту модуля теж можна замінити на псевдонім.

Іноді, корисно підключити всі атрибути модуля, тоді пишуть (в кінці ставлять зірочку):

```
from назва_модуля import *
```

### 3.2. Підготовка даних до імпорту в Python

Перед тим, як починати аналіз даних в Python, потрібно підготувати ці дані. Підготовка даних потребує дотримання певних умов, які представляють собою так званий бест-практикс табличних даних.

Табличні дані в файлі `xlsx` повинні задовольняти таким умовам:

- зарезервувати перший рядок таблиці під заголовок, а перший стовпець – для ідентифікації кожного запису (при необхідності);
- уникати імен, значень або полів з пробілами. Замість пробілу краще використовувати знак підкреслення або «горбатий» регістр (коли ім'я поля складається з декілька слів, які записані без пробілів, при цьому перша літера кожного слова пишеться великою, наприклад, `Ім'яПоля`);
- віддавати перевагу коротким назвам;
- уникати використання символів `?, $, %, ^, &, *, -, #, (, ), <, >, /, |, \, [, ], {, }`;
- видалити будь-які коментарі;
- переконатися, що будь-які значення в наборі даних, що відсутні, відображаються як `NA` (спеціальна функція, яка вказує на відсутність даних в комірці).

### 3.3. Використання бібліотеки OPENPYEX

#### 3.3.1. Аналіз банківських транзакцій

Для інсталяції модуля openpyex можна перейти за посиланням <https://pypi.org/project/openpyxl/>. Детальний опис цього модуля знаходиться за посиланням <https://openpyxl.readthedocs.io/en/stable/>.

Підготуємо програму в Python, яка б відповідала на питання, які були поставлені напередодні для таблиці банківських транзакцій (і, які були розв'язані або за допомогою рядка підсумків, або за допомогою зведених таблиць), а саме:

- з якого рахунку надійшла найбільша (найменша) сума коштів;
- яка кількість транзакцій з кожного рахунку;
- з якого рахунку найчастіше надходили кошти.

Код в Python починається з підключення необхідних модулів:

```
import os
from openpyxl import load_workbook, Workbook
from os.path import join, abspath
class
NotAllData(Exception):
    pass
```

Далі, потрібно перейти на директорію, в якій розташовано файл xlsx з таблицею банківських транзакцій. Поєднуємо поточну директорію з даним файлом.

```
os.chdir("E:/Python/excel_python")
data_path = join('.', 'banktransactions.xlsx')
data_path = abspath(data_path)
```

Відкриємо файл xlsx в Python.

```
wb = load_workbook(filename = 'banktransact.xlsx',  
data_only = True, read_only = True)
```

Функція `load_workbook()` приймає ім'я файлу як аргумент та повертає об'єкт робочої книги, який представляє файл. Це можна перевірити за допомогою команди `type(wb)`. В даному випадку, в команді ми не вказуємо повний шлях до файлу `xlsx`, маючи на увазі, що файл розташовано в робочій директорії, тобто робочий простір було підготовлено заздалегідь. Крім того, використано атрибут функції `open` – це `read_only` зі значенням `True`, тим самим файл `xlsx` відкривається лише для читання. Якщо зчитувані дані з Excel містять формули, то інструкція `data_only = True` дозволить завантажити лише значення комірок.

Для того, щоб позиціонуватися на першому листі книги `xlsx` (яких в книзі може бути декілька) потрібно в об'єкті `wb` визвати метод `active`:

```
sheet = wb.active
```

Нумерація листів починається з 0 (і не залежить від ім'я листа). Якщо нам буде потрібно зробити активним другий лист книги, то команда наступна:

```
wb.active = 1  
sheet = wb.active
```

Для того, щоб звернутися до вмісту окремих комірок листа використовують метод `value` об'єкта `sheet`, а саме: `sheet['A1'].value`, або `sheet[1][0].value` (стиль R1C1).

Зауважимо, що для відносних посилань на комірки, нумерація стовпців починається з нуля (тобто, стовець А позначено як 0, В – як 1, С – як 2 і т.п.).

Якщо ми хочемо переглянути всі рядки (або їх частину) таблиці банківських транзакцій, то можна побудувати цикл:

```
for row in range(1, sheet.max_row+1):  
    print(sheet['A'+ str(i)].value, sheet['B'+ str(i)].value,  
          sheet['C'+ str(i)].value, sheet['D'+ str(i)].value)
```

або,

```
for row in range(1, sheet.max_row+1):  
    print(sheet[row][0].value, sheet[row][1].value,  
          sheet[row][2].value, sheet[row][3].value)
```

Нарешті, такий варіант

```
for row in range(1, sheet.max_row+1):  
    Джерело = sheet[row][0].value  
    Призначення = sheet[row][1].value  
    ДатаЧас = sheet[row][2].value  
    Сума коштів = sheet[row][3].value  
    print(Джерело, Призначення, ДатаЧас, Сума коштів)
```

Зауважимо, що параметр циклу `row` є зарезервованою змінною. Конструкція `sheet.max_row` повертає кількість заповнених рядків таблиці. Так як `range` «не бере» останнє значення циклу, то потрібно додати одиницю.

Для того, щоб вибрати та переглянути дані з певного діапазону таблиці, використовують конструкцію:

```
cells = sheet['B1':'C11']  
for Призначення, ДатаЧас in cells:  
    print(Призначення.value, ДатаЧас.value)
```

Якщо нам потрібно звернутися до певної комірки листа книги, то це можна зробити наступною конструкцією:

```
sheets = wb.worksheets  
print(sheets[2]['A5'].value)
```

Змінна `sheets` є список назв листів, як об'єктів, для можливості звертатися до копійок цих листів.

Можна проглянути список листів робочої книги:

```
wsn = list(wb.sheetnames)  
print(wsn)
```

Іноді в робочій книзі потрібно знайти саме той лист (коли їх декілька), який містить необхідну нам інформацію. Якщо ознакою цього листа є певний текст в відомій нам комірці, то це вирішується за допомогою циклу:

```
wodate = None  
for i in wsn:  
    if wb[i]['A1'].value == 'Джерело':  
        wodate = i  
print(wodate)
```

В даному випадку ми шукаємо лист, в якому у комірці A1 зберігається текст «Джерело». Команда `print` роздрукує назву листа. При умові, коли такого листа не знайдеться, краще додати наступну конструкцію:

```
if wsdate == None:  
    raise NotAllData('Нема такого листа')
```

Коли необхідний лист знайдено, треба створити об'єкт-лист для подальшої роботи з атрибутами цього об'єкту:

```
ws = wb[wsdate]
```

Подальший код створює список з назв стовпців вихідної таблиці:

```
shapka = [cell.value for cell in next(  
ws.iter_rows(min_row=1, min_col=1,  
max_row=1, max_col=ws.max_column))]
```

Далі створимо словник, який збирає для кожного рахунку-відправника (це ключ словника) всю іншу інформацію, тобто всі рахунки-отримувачі, дату, час і суму перерахованих коштів (це значення словника).

Для початку створюється порожній словник: `pryznachdata = {}`.

В наступному фрагменті коду словник `pryznachdata` заповнюється елементами. Цикл пробігає усі рядки таблиці (починаючи з другого) і для рахунку-відправника (змінна `priz`) збирає дані, формуючи список з даних рядків (змінна `prizdata`). Спочатку перевіряється, чи не є наступний рядок порожнім. Якщо ні, то формується ключ словника, як значення комірки першого стовпця та поточного рядка. Так як номери рахунків в першому стовпці таблиці можуть повторюватися, то для кожного такого рахунку, який вперше потрапляє в якості ключа словника, створюється порожній список `prizdata`. Блок-схема створення словника зображена на рис. 3.3.1.

```
for row in ws.iter_rows(min_row = 2, min_col = 1, max_row = ws.max_row,  
max_col = ws.max_column):
```

```
    if len(row) > 0:
```

```
        priz = row[0].value
```

```
        if priz is not None:
```

```
            prizdata = [cell.value for cell in row]
```

```
            if priz not in pryznachdata:
```

```
                pryznachdata[priz] = []
```

```
                pryznachdata[priz].append(prizdata)
```

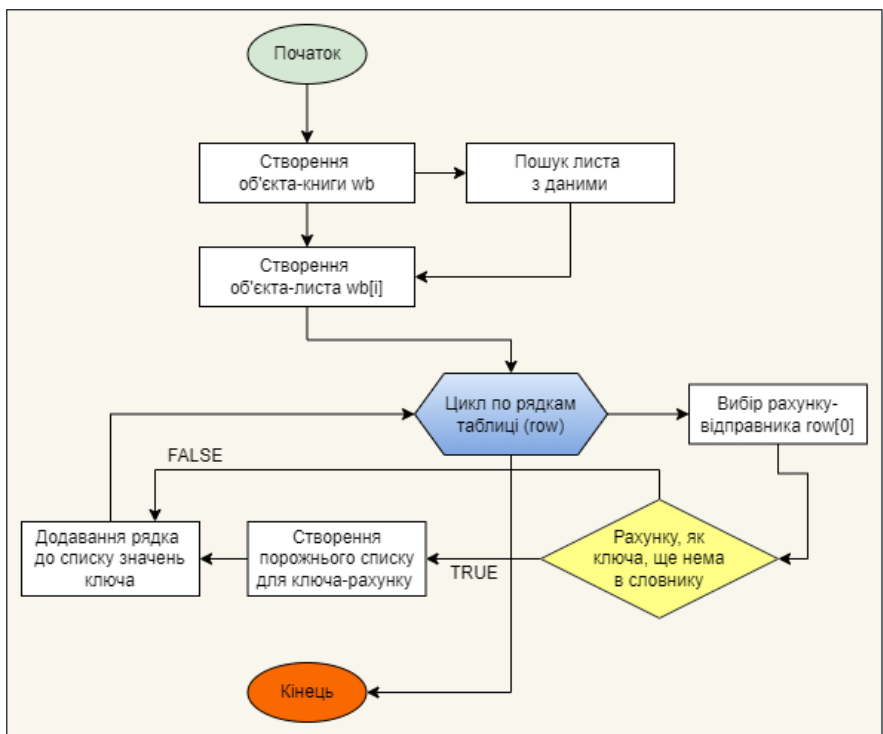


Рисунок 3.3.1

Частину словника можна бачити на наступному фрагменті:

```
('01223854840',  
[['01223854840', '01429256054', datetime.datetime(2020,1, 1, 9, 43), 2345],  
 ['01223854840', '01364558175', datetime.datetime(2020, 1, 1, 19, 54), 1365],  
 ['01223854840', '01914715106', datetime.datetime(2020, 1, 1, 22, 42), 19785],  
 ['01223854840', '01914715106', datetime.datetime(2020, 1, 2, 9, 38), 24362],  
 ['01223854840', '07989967213', datetime.datetime(2020, 1, 2, 12, 20), 2463],
```

Фрагмент словника друкувався командою `print(*pryznachdata.items(), sep='\n\n')`. Ми бачимо перший елемент (неповну першу пару) словника, в якому ключ – це рахунок-відправник ‘01223854840’, а значення ключа – список вибраних рядків таблиці, які пов’язані з цим рахунком. Можна помітити, що кожний елемент списку значення містить і рахунок-відправник (він як би дублюється з ключем). Якщо в рядку коду `prizdata = [cell.value for cell in row]` замість `row` помістити конструкцію `next(ws.iter_rows(min_row =1, min_col =2, max_row = ws.max_row, max_col = ws.max_column))`, ключ-рахунок дублюватися не буде. Суттєвим тут є те, що записи з таблиці зчитуються починаючи з другого стовпця. Дані стовпця дати та часу подано як `datetime.datetime(*,*,*,*,*)`. Але, фактично, ми маємо звичайний формат дати-часу, коли «проглядаємо» цей елемент списку за допомогою команди `print`. Надалі, можна відкоригувати формат дати на свій смак, якщо використати бібліотеку `datetime`. Взагалі, нема потреби друкувати цей словник. В даному випадку це зроблено лише для того, щоб мати уявлення про структуру словника.

Наявність побудованого словника дозволяє дати відповідь на питання з якого рахунку надійшла найбільша (найменша) сума коштів. Для цього створимо такий словник: `dicprizsum = {}`.

```
for priz in pryznachdata:  
    dicprizsum[priz] = 0  
  
    for i in range(len(pryznachdata[priz])):  
        dicprizsum[priz] = dicprizsum[priz] + pryznachdata[priz][i][3]  
        print("З рахунку", priz, "перераховано", dicprizsum[priz])
```

В кодї створення словника маємо зовнішній цикл, який «пробігає» ключі словника `pruznachdata`, та внутрішній – по кожному рядку значення ключа, звертаючись саме до перерахованих сум, і, кожного разу накопичуючи суму переказів. Елементи рядка (як списку) рахуються з 0, тому сума коштів в рядку рід номером «i» – це `pruznachdata[priz][i][3]`.

Результатом виконання цього коду буде:

```
3 рахунку 01223854840 перераховано 13664524
3 рахунку 07989967213 перераховано 2129715
3 рахунку 07497762938 перераховано 7388109
3 рахунку 01223859234 перераховано 4534203
3 рахунку 01638782715 перераховано 14553160
3 рахунку 07974362193 перераховано 308873
```

Фактично, ми маємо аналог результату в зведеній таблиці (рис. 2.3.11). Отже, проглядаючи цей фрагмент можна отримати, з якого рахунку на інші рахунки надійшла найбільша (найменша) сума коштів. Взагалі, таких рахунків може бути досить багато, і візуально знайти найбільшу суму буде складно. Але це швидко можна визначити. Використовуючи словник `dicprizsum`, створимо список зі значень сум та застосуємо до нього метод `max`.

```
valuesListSum = list(dicprizsum.values())
print('Найбільша перерахована сума складає', max(valuesListSum))
```

До того ж, використовуючи цикл по ключу в словнику, можна вказати і сам рахунок, з якого надійшла найбільша сума коштів:

```
for priz in dicprizsum:
    if max(valuesListSum)==dicprizsum[priz]:
        print("Рахунок, з якого надійшла найбільша сума - це ", priz)
```

Кількість транзакцій з кожного рахунку, визначимо за допомогою циклу по ключу в словнику `pruznachdata` та методу `len` (довжина), який застосовується до списку значень ключа `pruznachdata[priz]`.

for priz in pryznachdata:

```
print("З рахунку", priz, "відбулося транзакцій", len(pryznachdata[priz]))
```

Результат виконання цього скрипта наступний:

```
З рахунку 01223854840 відбулося транзакцій 533
З рахунку 07989967213 відбулося транзакцій 77
З рахунку 07497762938 відбулося транзакцій 270
З рахунку 01223859234 відбулося транзакцій 158
З рахунку 01638782715 відбулося транзакцій 545
З рахунку 07974362193 відбулося транзакцій 11
```

Викладені фрагменти коду для аналізу таблиці банківських транзакцій поки що «повторювали» ті елементи аналізу, які ми здійснювали в Excel. Але, потрібно зауважити, що з великими таблицями Python працює значно швидше.

Наступна задача продемонструє більш помітні можливості Python для роботи з Excel. Так як більшість питань з аналізу таблиці банківських транзакцій стосується окремих рахунків, то було би значно простіше, коли дані для певного рахунку були б розташовані на окремому листі книги Excel, або в окремій книзі. Отже, засобами Excel створимо нові книги окремо для кожного рахунка-відправника. Зрозуміло, що в кожній такій книзі будуть зібрані записи з основної таблиці тільки для певного рахунку.

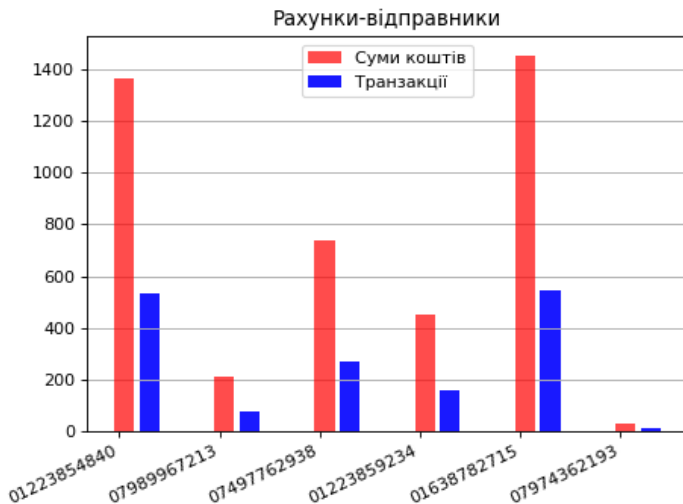
Спочатку треба закрити поточну книгу-об'єкт: `wb.close`. В наступному коді створюються книги окремо для кожного рахунка-відправника. Як ми бачимо, знову став у пригоді словник `pryznachdata`. Для кожного окремого рахунка-відправника ми створюємо нову книгу-об'єкт, вказуємо назву листа, формуємо на листі заголовки таблиці та додаємо на лист інші записи (значення ключа словника). Далі, формуємо ім'я файлу (з вказанням ім'я директорії, в яку будуть записані всі файли), зберігаємо його на диск і закриваємо об'єкт книги поточного елементу циклу.

```

for priz in pryznachdata:
    wb = Workbook()
    ws = wb.active; ws.title = "Рахунок " + priz
    ws.append(shapka)
    for row in pryznachdata[priz]:
        ws.append(row)
    exfilename = join('.', 'Pryznach', (priz + '.xlsx'))
    exfilename = abspath(exfilename)
    wb.save(exfilename)
    wb.close

```

В наступному кроці побудуємо стовпчасту діаграму, на якій можна порівняти значення сумарних перерахованих коштів з рахунків-відправників та кількість транзакцій з цих рахунків. Для цього використовуємо функцію `bar` бібліотеки `Matplotlib`. В якості даних вісі X та значень висоти стовпчиків візьмемо ключі та значення відповідних словників, перетворених у списки.



Приведений код побудови діаграм нескладний і тому ми його залишимо без пояснень.

```
data_names = list(dicprizsum.keys())
data_values_1 = list(dicprizsum.values())
data_values_2 = list(chika.values())
dpi = 80; fig = plt.figure(dpi = dpi, figsize = (512 / dpi, 384 / dpi) )
mpl.rcParams.update({'font.size': 0}); plt.title('Рахунки-відправники')
ax = plt.axes(); ax.yaxis.grid(True, zorder = 1)
xs = range(len(data_names))
plt.bar([x + 0.05 for x in xs], [d*0.0001 for d in data_values_1], width = 0.2,
color = 'red', alpha = 0.7, label = 'Суми коштів')
plt.bar([x + 0.3 for x in xs], [d*1.0000 for d in data_values_2], width
= 0.2, color = 'blue', alpha = 0.9, label = 'Транзакції',)
plt.xticks(xs, data_names); fig.autofmt_xdate(rotation = 25)
plt.legend(loc='upper center')
fig.savefig('bars.png')
```

### 3.3.2. Аналіз трафіку з'єднань

Аналізуючи трафіку з'єднань будемо розглядати певну активність кожного об'єкта (абонента А) по відношенню к одному або декілька абонентам Б. Ця активність може визначатися тривалістю вхідних (вихідних) голосових викликів або кількістю з'єднань (голосових або СМС). Саме ці задачі будемо вирішувати за допомогою Python.

Ми вже знаємо, як на Python отримати файли Excel, які містять окремо інформацію по кожному ключу словника з записами значення ключа. Але, краще, коли маємо лише одну книгу Excel (один файл), в якій створені окремі листи для кожного ключа словника з записами значення ключа. Перед тим, як реалізувати це на Python, потрібно створити відповідний словник з таблиці трафіку з'єднань.

Це реалізовано в наступному кодї:

```
slovnukA = {}  
  
for row in ws.iter_rows(min_row=2, min_col=1,  
max_row=ws.max_row, max_col=ws.max_column):  
  
    if len(row) > 0:  
  
        A = row[0].value  
  
        if A is not None:  
  
            Adata = [cell.value for cell in row]  
  
            if A not in slovnukA:  
  
                slovnukA [A] = []  
  
            slovnukA [A].append(Adata)
```

Приведений код фактично повторює те, коли створювався словник `pryznachdata` в попередньому розділі.

На підставі створеного словника `slovnukA` можна створити нову книгу Excel (файл), в якій для кожного абонента `A` відведено окремий лист.

```
i = -1  
  
wb = Workbook()  
  
for A in slovnukA:  
  
    i = i + 1  
  
    wb.create_sheet(index = i, title = A)  
  
    wb.active = i; ws = wb.active; ws.append(shapka)  
  
    for row in slovnuk[A]:  
  
        ws.append(row)  
  
    exfilename = join('.', 'Traf', ('trafikA' + '.xlsx'))  
  
    exfilename = abspath(exfilename)
```

На кожному листі маємо стовпці: Абонент Б, Дата, Час, Тривалість, Тип, Напрямок, Широта, Довгота (координати базових станцій мобільного зв'язку).

Звертаємо увагу на те, що перед тим, як створити нову книгу-об'єкт (а потім і файл Excel), необхідно закрити активну книгу-об'єкт, яка працювала з вихідною таблицею.

Змінна *i* починається зі значення -1, щоб перший лист прийшовся на перший номер абонента А. Нові листи утворюються завдяки методу `wb.create_sheet(index = i, title = A)`. Ім'я (назва) кожного наступного листа ідентифікує абонента А.

Побудована книга-об'єкт зберігається на диску – це файл з ім'ям `trafikA.xlsx`, який зберігається в директорію `Traf`.

Подальша робота з аналізу даних полягає в тому, щоб вибрати певний лист (певного абонента А). У випадку з таблицею банківських транзакцій ми відшукували той лист книги, на якому в певній комірці зберігався певний текст (до речі, це було слово Джерело). В даному випадку будемо відшукувати лист за його ім'ям. Код наступний:

```
wsn = list(wb.sheetnames)
wsdate = None
for i in wsn:
    if wb[i].title == 'A1':
        wsdate = i
print(wsdate)
if wsdate == None:
    raise NotAllData("Нема такого листа")
```

Нагадаємо знову ті задачі, які будемо вирішувати при аналізі трафіку з'єднань:

- порівняти кількість з'єднань (голосових та СМС) об'єкта з абонентами Б;
- порівняти тривалість всіх вхідних (вихідних) голосових викликів разом об'єкта з абонентами Б.

Для вирішення цих задач створимо словник:

```

slovnykB = {}

for row in ws.iter_rows(min_row=2, min_col=1,
max_row=ws.max_row, max_col=ws.max_column):
    if len(row) > 0:
        B = row[0].value
        if B is not None:
            Bdata = [cell.value for cell in row]
            if B not in slovnykB:
                slovnykB[B] = []
            slovnykB[B].append(Bdata)

```

В якості ключів цього словника маємо номера телефонів абонентів Б, а в якості значень для кожного ключа маємо список (як тип даних в Python) списків-рядків або записів таблиці на листі А1, які відносяться тільки до даного ключа (номера телефону абонента).

Кількість з'єднань (голосових та СМС) об'єкта з абонентами Б може бути знайдена на підставі такого коду:

```

for B in slovnykB:
    print("Абонент Б:", B, "має", len(slovnykB[B]), "з'єднань з абонентом А")

```

Якщо абонентів Б дуже багато, і список довгий, то порівняння даних буде скрутним, і, краще виконати якесь сортування цих значень. Для цього побудуємо такий словник:

```

medb = {}
for B in slovnykB:
    medb[B]=len(slovnykB[B])

```

Ключи словника `medb` є номери абонентів Б, а значення – кількість з'єднань з абонентом А. Виконаємо сортування словника `medb` (в зворотному порядку): `sortB = dict(sorted(medb.items(), reverse = True, key=lambda x: x[1]))`.

Тривалість всіх вхідних (вихідних) голосових викликів разом абонента А з кожним абонентом Б окремо одержимо за таким кодом:

```
goloslov = {}
for B in slovnykB:
    goloslov[B]=0
    for i in range(len(slovnykB[B])):
        if slovnykB[B][i][4]!='голосовий':
            goloslov[B] = goloslov[B] + slovnykB[B][i][3]
sortgolos = dict(sorted(goloslov.items(), reverse = True, key=lambda x: x[1]) )
for B in sortgolos:
    print("Тривалість дзвінків абонента", B, "з абонентом А", sortgolos[B], "секунд")
```

### 3.4. Інтерактивна візуальна аналітика в Python

Розглянемо побудову карти розташування базових станцій мобільного зв'язку з таблиці трафіку з'єднань засобами Python. Для цього використаємо бібліотеку (модуль) `folium`.

`Folium` – це потужна бібліотека візуалізації геопросторових даних у Python. За допомогою `folium` можна створити інтерактивну карту будь-якого розташування у світі, якщо ми знаємо його значення широти та довготи. Також можна на карту накласти маркери та кластери маркерів.

Будемо використовувати дані того ж самого файлу, з яким працювали в розділі 3.3.1. Отже, маємо файл `trafik.xlsx` з таблицею даних на першому листі (119 записів), в якій у двох останніх стовпцях (7-й та 8-й) задано широту та довготу, відповідно. Треба зауважити, що деякі координати в таблиці повторюються. Це пояснюється тим, що за певний період часу об'єкт (абонент А) пересувався в межах певної території, виходячи на зв'язок з абонентами Б, то ж неодноразово знаходився поряд з одними і тими ж базовими станціями мобільного зв'язку.

Перед тим, як безпосередньо використати модуль `folium`, потрібно підготувати список координат.

В наступному коді ми створюємо об'єкт книга, відкриваючи файл `trafik.xlsx`, робимо активним перший лист, формуємо список координат (список `coorddata`) та видаляємо дублікати координат (список `coordwd`).

```
import os

from openpyxl import load_workbook, Workbook

from os.path import join, abspath

wb = load_workbook(filename = 'trafik.xlsx', data_only = True, read_only = True)

ws = wb.active; coorddata = []

for row in
ws.iter_rows(min_row = 2, min_col = 7, ax_row = ws.max_row, max_col = ws.max_column):

    cdata = [cell.value for cell in row]

    coorddata.append(cdata)

coordwd = []

# використовуємо генератор списку

[coordwd.append(x) for x in coorddata if x not in coordwd]

print(coordwd)
```

Переходимо до роботи з модулем `folium`. На початку потрібно його додати до коду:

```
import folium

from folium.plugins import MarkerCluster

from folium.plugins import MousePosition

from folium.features import DivIcon
```

Далі необхідно визначитися з координатами того району, який охоплює базові станції, координати яких приведено у вихідній таблиці, і які будуть нанесені на карту цього району у вигляді маркерів. Для цього зробимо наступне:

```

coordwd0 = []
for x in coordwd:
    coordwd0.append(x[0])
coordwd1 = []
for x in coordwd:
    coordwd1.append(x[1])
minlat = min(coordwd0); maxlat = max(coordwd0)
latR = toFixed((minlat + maxlat)/2, 2)
minlong = min(coordwd1); maxlong = max(coordwd1)
longR = toFixed((minlong + maxlong)/2, 2)

```

В цьому кодї ми спочатку створюємо два списки. Список `coordwd0` утворено зі значень широти, які ми отримали в списку `coordwd`, а список `coordwd1` утворено зі значень довготи. В кожному з цих списків ми знаходимо мінімальне та максимальне значення, та їх середнє арифметичне (`latR` і `longR`). І ще, в значеннях `latR` та `longR` ми залишаємо тільки два десяткових знаки. Це робить функція:

```

def toFixed(numObj, digits=0):
    return f"{numObj:.{digits}f}"

```

Тепер ми маємо пару `[latR, longR]`, яку ми використаємо в головному рядку коду, який фактично створює карту:

```

map =
folium.Map(location=[latR,longR], tiles = 'openstreetmap', zoom_start = 13)

```

Параметр `tiles` використовують для зміни фону (або стилю) картки. Взагалі, існують три стиля карт. Стиль карти за замовчуванням –

OpenStreetMap (відкрита карта вулиць), яка показує вигляд та назву вулиць та межі країн світу. Стиль Stamen Toner дозволяє отримувати висококонтрастні чорно-білі карти. Цей стиль підходить для візуалізації та вивчення річкових меандрів (вигини русла річки) та прибережних зон. Стиль Stamen Terrain (тиснення місцевості) підходить для візуалізації затінення пагорбів та природних кольорів рослинності.

Параметр `zoom start` – це початковий рівень масштабування, який після відображення карти можна легко змінити шляхом збільшення або зменшення масштабу.

Перейдемо до нанесення маркерів (позначок станцій) на карту.

```
for coordinates in coordwd:  
    folium.Marker(location = coordinates, icon =  
    folium.Icon(color = 'blue')).add_to(map)
```

Як бачимо, маркери утворюються за допомогою методу `Marker` з параметрами `location` та `icon`. Ці параметри визначають координати маркерів та їх колір. Саме тут нам знадобився список `coordwd`.

За допомогою наступного коду можна бачити координати будь-якої точки на карті при наведенні на неї курсора миші. Це невеличке вікно, яке може бути розташовано, наприклад, зверху в правому куті карти - `topright`.

```
formatter = "function(num) {return L.Util.formatNum(num, 5)};"  
mouse_position = MousePosition(  
    position = 'topright', separator = ' Long: ',  
    empty_string = 'NaN', lng_first = False,  
    num_digits = 20, prefix = 'Lat:',  
    lat_formatter = formatter, lng_formatter = formatter)  
map.add_child(mouse_position)
```

Нарешті, побудовану карту треба зберегти у файлі `html` для подальшого її перегляду з браузера: `map.save(«My_map1.html»)`

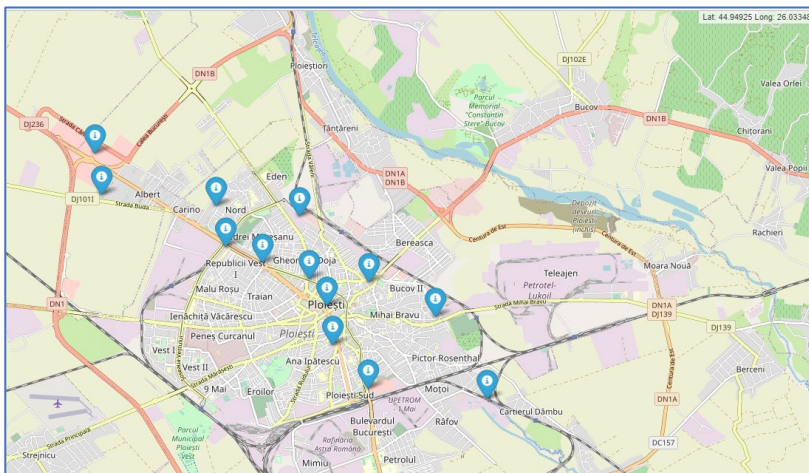


Рисунок 3.4.1

### Питання для самоперевірки:

1. Що таке IDLE і для чого служить?
2. Що містить і для чого служить пакет Anaconda?
3. Які переваги редактора Spyder?
4. Яким символом закінчується кінець рядка інструкції в Python?
5. Чи можна записати декілька команд або інструкцій в одному рядку?
6. Як записати одну інструкцію в декілька рядків?
7. Як будувється інструкція з основної та вкладеної?
8. Як розмістити коментарі в коді програми?
9. Які є можливості запуску коду на Python?
10. Що таке ідентифікатор, як його можна задати, та чи має значення регістр букв в ідентифікаторі?
11. Чи може змінюватися тип змінної по ходу виконання коду?
12. Яка особливість функції input?
13. Як розуміти те, що рядковий тип є незмінюваним?
14. Які комбінації можуть бути при заданні зрізу рядка?
15. Що можна отримати, використовуючи методи рядків?
16. Як можна створити список (list), та з яких структур?
17. З чим по-більшості пов'язані методи списків?
18. Які переваги кортежів по відношенню к спискам?
19. Які методи мають кортежі?
20. Чи може діапазон (range) складатися з букв?

21. Чи може множина (set) містити два однакових елемента?
22. Чи можна зі словника вийняти тільки ключі або тільки значення?
23. Чи можна виконувати сортування в словниках?
24. Які операції використовують для порівняння об'єктів в Python?
25. Як розуміти те, що оператори and і or «закорочують» обчислення своїх операндів?
26. В яких випадках оператор not має нищий пріоритет, а в яких найвищий?
27. Чим цикл while відрізняється від циклу for?
28. Як оператори continue та break переривають або взагалі припиняють роботу циклів?
29. Підготовка табличних даних в Excel перед зчитуванням в Python.
30. Призначення модуля openpyxl.
31. Відкриття файлу xlsx з потрібного листа.
32. Перенос даних з вихідної таблиці в окремі файли засобами Python.
33. Побудова стовпчастих діаграм в Python.
34. Перенос даних з вихідної таблиці в іншу книгу на окремі листи засобами Python.
35. Бібліотека folium, її призначення і можливості.
36. Як видалити дублікати в списках засобами Python.
37. Параметри методів Map та Marker модуля folium.

## Скорочений довідник функцій в Excel

DATE	Повертає задану дату у числовому форматі
DATEVALUE	Перетворює дату з текстового формату на числову
DAY	Перетворює дату у числовому форматі на день місяця
DAYS360	Обчислює кількість днів між двома датами на основі 360-денного року
HOUR	Перетворює дату на години
MINUTE	Перетворює дату на хвилини
MONTH	Перетворює дату на місяці
NETWORKDAYS	Знаходить кількість робочих днів між двома датами
NOW	Видає поточну дату та час
SECOND	Перетворює дату на секунди
TIME	Видає заданий час у числовому форматі
TIMEVALUE	Перетворює час із текстового формату на числовий формат
TODAY	Видає поточну дату
WEEKDAY	Перетворює дату на день тижня
WEEKNUM	Визначає номер робочого тижня року для вказаної дати
WORKDAY	Знаходить дату, віддалену від даної на задану кількість робочих днів
YEAR	Знаходить рік для заданої дати
CELL	Визначає інформацію про формат, місцезнаходження або вміст комірки
ERRORTYPE	Визначає номер, який відповідає одному з типів помилок Microsoft Excel
ISBLANK	Видає логічне значення ІСТИНА, якщо аргумент є посиланням на порожню комірку
ISERR	Видає логічне значення ІСТИНА, якщо аргумент посилається будь-яке значення помилки, крім #Н/Д

ISERROR	Видає логічне значення ІСТИНА, якщо аргумент посилається на будь-яке значення помилки
ISEVEN	Видає логічне значення ІСТИНА, якщо аргумент — парне число
ISLOGICAL	Видає логічне значення ІСТИНА, якщо аргумент посилається на логічне значення
ISNONTEXT	Видає логічне значення ІСТИНА, якщо аргумент посилається на значення, яке не є текстом
ISNUMBER	Видає логічне значення ІСТИНА, якщо аргумент посилається на число
ISODD	Видає логічне значення ІСТИНА, якщо аргумент — непарне число
ISREF	Видає логічне значення ІСТИНА, якщо аргумент посилається на посилання
ISTEXT	Видає логічне значення ІСТИНА, якщо аргумент посилається на текст
N	Перетворює задане значення на число
NA	Видає значення помилки #Н/Д
TYPE	Видає тип значення
AND	Логічна функція
IF	Виконує перевірку умови
NOT	Логічна функція
OR	Логічна функція
ADDRESS	Видає посилання на окрему комірку робочого листа у вигляді тексту
CHOOSE	Вибирає значення зі списку аргументів згідно індексу
COLUMN	Повертає порядковий номер стовпця для вказаної комірки
COLUMNS	Визначає кількість стовпців у масиві
GETPIVOTDATA	Повертає дані, що зберігаються у зведеній таблиці
HLOOKUP	Шукає значення в першому рядку масиву і видає значення з комірки в знайденому стовпці та вказаному рядку

HYPERLINK	Створює посилання, яке відкриває документ, що знаходиться на жорсткому диску, сервері мережі або в Інтернеті
INDEX	За індексом отримує значення із масиву
INDIRECT	Визначає посилання, задане текстовим значенням
LOOKUP	Шукає значення у векторі чи масиві
MATCH	Шукає значення у масиві
ROW	Визначає номер рядка, що визначається посиланням
VLOOKUP	Шукає значення в першому стовпці масиву і видає значення з комірки у знайденому рядку та вказаному стовпці
ABS	Знаходить модуль (абсолютну величину) числа
CEILING	Округлює число до найближчого цілого або до найближчого кратного вказаного значення
COMBIN	Знаходить кількість комбінацій для заданої кількості об'єктів
DEGREES	Перетворює радіани на градуси
EVEN	Округлює число до найближчого парного цілого
FACT	Обчислює факторіал числа
FLOOR	Округлює число до найближчого меншого за модулем цілого
GCD	Знаходить найбільший спільний дільник
INT	Округлює число до найближчого меншого цілого
LCM	Знаходить найменше загальне кратне
LN	Обчислює натуральний логарифм числа
LOG10	Обчислює десятковий логарифм числа
MROUND	Знаходить число, округлене з необхідною точністю
ODD	Округлює число до найближчого непарного цілого
POWER	Обчислює результат зведення числа до ступеня
PRODUCT	Обчислює добуток аргументів
RADIANS	Перетворює градуси на радіани
RAND	Видає випадкове число в інтервалі від 0 до 1
RANDBETWEEN	Видає випадкове число у заданому інтервалі

ROMAN	Перетворює число в арабському записі до римського як текст
ROUND	Округлює число до вказаної кількості десяткових розрядів
SIGN	Визначає знак числа
SQRT	Обчислює позитивне значення квадратного кореня
SUBTOTAL	Обчислює проміжні підсумки
SUM	Підсумовує аргументи
SUMIF	Підсумовує значення комірок, що задовольняють певній умові
SUMPRODUCT	Повертає суму добутків відповідних діапазонів або масивів
SUMSQ	Обчислює суму квадратів аргументів
TRUNC	Відкидає дробову частину числа
COUNTA	Повертає число непорожніх комірок у вказаному діапазоні
COUNT	Повертає число комірок у вказаному діапазоні, які містять тільки числові значення
COUNTIF	Повертає число комірок у вказаному діапазоні, які задовольняють певним критеріям в одному діапазоні
COUNTIFS	Повертає число комірок у вказаному діапазоні, які задовольняють певним критеріям в декількох діапазонах
CHAR	Визначає символ за заданим кодом
CLEAN	Видаляє всі недруковані символи з тексту
CODE	Визначає числовий код першого символу в текстовому рядку
CONCATENATE	Поєднує кілька текстових елементів в один
DOLLAR	Перетворює число на текст, використовуючи грошовий формат долара
EXACT	Перевіряє ідентичність двох текстів
FIND	Шукає входження одного тексту до іншого (з урахуванням регістру)

FIXED	Форматує число та перетворює його на текст із заданим числом десяткових знаків
LEN	Визначає кількість знаків у текстовому рядку
LOWER	Робить усі літери в тексті малими
MID	Видає певну кількість знаків з рядка тексту, починаючи із зазначеної позиції
PROPER	Робить великою перші літери в кожному слові тексту
REPLACE	Замінює символи у тексті
REPT	Повторює текст задане число разів
SEARCH	Шукає входження одного тексту до іншого (без урахування регістру)
SUBSTITUTE	Замінює в текстовому рядку старий текст на новий
TEXT	Форматує число та перетворює його на текст
TRIM	Видаляє з тексту зайві пробіли
UPPER	Робить усі літери в тексті великими
VALUE	Перетворює аргумент на числовий формат

## «Гарячі» клавіші в Excel

Ctrl + C	Копіювати
Ctrl + X	Вирізати
Ctrl + V	Вставити
Ctrl + Z	Скасувати останню дію
Ctrl + F	Знайти
Ctrl + H	Замінити
Ctrl + I	Відобразити діалогове вікно Формат комірок
Ctrl + K	Відобразити діалогове вікно для вставки гіперпосилання
Ctrl + A	Виділити всю таблицю (аркуш)
Ctrl + 9	Згорнути рядок з активною коміркою
Ctrl + 0	Згорнути стовпець з активною коміркою
Ctrl + Shift + 8	Виділити всю таблицю
Ctrl + Shift + 9	Відобразити виділені рядки
Ctrl + Shift + 0	Відобразити виділені стовпці
Ctrl + PgDn	Переміщення по робочим листам
Ctrl + PgUp	Переміщення по робочим листам
Ctrl + End	Переміщення до останньої комірки області даних
Ctrl + Home	Переміщення до комірки A1
Home	Переміщення до 1-ї комірки поточного рядка
Ctrl + пробіл	Виділити поточний стовпець
Shift + пробіл	Виділити поточний рядок
Ctrl + стрілка	Переміщення до кінця діапазону
Ctrl + Shift + стрілка	Виділення діапазону комірок
Ctrl + Enter	Заповнення однотипними даними

Ctrl + Shift + Enter	Введення формули масиву
Ctrl + /	Виділення поточного масиву
Ctrl + B	Жирний шрифт
Ctrl + i	Нахилений шрифт
Ctrl + U	Підкреслений шрифт
Ctrl + (-)	Вилучити комірки
Ctrl + 5	Перекреслити текст
Ctrl + (+)	Додати комірки
Ctrl + Shift + _	Видалити рамки
F2	Редагування комірки
Ctrl + F3	Диспетчер імен
F3	Вставити ім'я
F4	Зміна адреси в формулі з відносної на абсолютну (\$), і навпаки

## СПИСОК ВИКОРИСТАНОЇ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Джонстоун Дженніфер, Яніцкі Мірослав, Навроцкі Даріуш. Посібник з кримінального аналізу для кримінальних аналітиків Державної прикордонної служби України. Київ: ОБСЄ, 2015. 178 с
2. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Балакіна С.М., 2020. 180 с
3. Нелюбов В. О., Куруца О. С. Основи інформатики. Microsoft Excel 2016: навчальний посібник. Ужгород: ДВНЗ «УжНУ», 2018. 58 с.: іл.
4. Основи кримінального аналізу : посіб. з елементами тренінгу / Користін О. Є., С. В. Албул, А. В. Холостенко та ін. – Одеса : ОДУВС, 2016. 112 с
5. Тактичний кримінальний аналіз: теорія та практика; навчальний посібник / О.Є. Користін, Н.П. Свиридюк, О.М. Цільмак, О.М. Заєць, К.Ю. Ісмайлов, В.А. Некрасов; МВС України, ДНДІ, ОДУВС. Одеса: РВВ ОДУВС, 2019. 216 с
6. Федчак І. А. Основи кримінального аналізу : навчальний посібник. Львів : Львівський державний університет внутрішніх справ, 2021. 288 с.
7. Гвідо ван Россум, Фред Л. Дрейк. Підручник мови Python. Переклад: Сергій Кузьменко. URL: [https://uk.wikibooks.org/wiki/Підручник\\_мови\\_Python](https://uk.wikibooks.org/wiki/Підручник_мови_Python) (дата звернення: 08.07.2022)
8. Підручник з основ Microsoft Excel - Навчання використання Excel. URL: <https://websetnet.net/uk/microsoft-excel-basics-tutorial-learning-how-to-use-excel/> (дата звернення: 08.07.2022)
9. ANACONDA DISTRIBUTION. URL: <https://www.anaconda.com/products/distribution> (дата звернення: 08.07.2022)
10. Anaconda Documentation. URL: <https://docs.anaconda.com/> (дата звернення: 08.07.2022)
11. IBM i2 Analyst's Notebook. User Help. URL: [https://www.ibm.com/docs/en/SSJSV9\\_9.2.1/com.ibm.i2.anb.doc/analysts\\_notebook\\_pdf.pdf](https://www.ibm.com/docs/en/SSJSV9_9.2.1/com.ibm.i2.anb.doc/analysts_notebook_pdf.pdf) (дата звернення: 08.07.2022)