

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»



ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інформаційних технологій та комп'ютерної інженерії

ТЕХНОЛОГІЇ ХМАРНИХ ОБЧИСЛЕНЬ

Методичні рекомендації до виконання практичних робіт
для здобувачів ступеня бакалавра
спеціальностей галузі знань 12 (F) Інформаційні системи та технології

Дніпро
НТУ «ДП»
2025

Заболотний К.С.

Технології хмарних обчислень [Електронний ресурс] : методичні рекомендації до виконання практичних робіт для здобувачів ступеня бакалавра спеціальностей галузі знань 12 (F) Інформаційні системи та технології К.С. Заболотний, А. В. Кожевников, С.Л. Нікулін, Л.І. Мещеряков ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2025. – 91 с.

Автори:

К.С. Заболотний д-р техн. наук, проф.,
А.В. Кожевников, канд. техн. наук, доц.,
С.Л. Нікулін, д-р геол. наук, проф.,
Л.І. Мещеряков д-р техн. наук, проф.

Затверджено кафедрою інформаційних технологій та комп'ютерної інженерії (протокол № 14 від 23.04.2025 р.).

Уміщено теоретичні відомості за темами лекційного курсу, варіанти практичних завдань з рекомендаціями до їх виконання, контрольні питання, список використаної та рекомендованої літератури.

Зміст методичних рекомендацій відповідає програмі вибіркової навчальної дисципліни «Технології хмарних обчислень» і адресовано бакалаврам, які проходять підготовку за всіма спеціальностями факультету інформаційних технологій, а також здобувачам усіх інших спеціальностей, які можуть обрати зазначену дисципліну для вивчення в межах формування індивідуальної траєкторії навчання.

Відповідальний за випуск завідувач кафедри інформаційних технологій та комп'ютерної інженерії В. В. Гнатушенко, д-р техн. наук, проф.

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

Дисципліна “Технології хмарних обчислень” – вибіркова складова фахової підготовки бакалаврів всіх освітньо-професійних програм спеціальностей факультету інформаційних технологій. Мета дисципліни – формування у здобувачів вищої освіти необхідного обсягу теоретичних знань щодо технологій хмарних сервісів, а також умінь і практичних навичок їх реалізації у сучасних ІТ-системах.

Методичні рекомендації призначені для закріплення теоретичних знань, набутих здобувачами вищої освіти в лекційному курсі, а також формування навичок виконання практичних робіт щодо хмарних сервісів передачі, зберігання, обробки і аналізу даних.

У підсумку виконання практичних робіт здобувачі вищої освіти – майбутні фахівці повинні отримати результати навчання у відповідності з вимогами робочої програми дисципліни, які представлені в табл. 1.1.

Таблиця 1.1. Дисциплінарні результати навчання, що отримуються у підсумку вивчення дисципліни “Технології хмарних обчислень”.

Дисциплінарні результати навчання (ДРН)	
шифр ДРН	зміст
ДРН – 01	Демонструвати знання основних видів хмарних технологій та галузей їх застосування
ДРН – 02	Демонструвати знання основних видів технологій віртуалізації та практичні навички створення віртуальних середовищ
ДРН – 03	Демонструвати знання методів розробки додатків для хмарних систем
ДРН – 04	Вміти оцінювати і вибирати необхідні методи та технології для хмарних обчислень та вирішення поставленої задачі
ДРН – 05	Вміти користуватися сервісами SaaS, PaaS хмарних платформ, в тому числі з застосуванням засобів штучного інтелекту
ДРН – 06	Вміти користуватися прийомами хмарного програмування

Мета практичних робіт – поглибити і систематизувати набуті здобувачами вищої освіти на лекціях теоретичні знання з дисципліни “Технології хмарних обчислень” та сформувати у майбутніх фахівців з інформаційних технологій професійні компетентності (уміння і навички) використання методів та програмного забезпечення хмарних технологій.

Практичні роботи передбачають застосування вільно розповсюджуюваного програмного забезпечення і відкритих хмарних сервісів. Кожна практична робота має назву, ціль, постановку задачі, контрольні питання і завдання, рекомендації щодо виконання роботи та вимоги до оформлення звіту, а також варіанти індивідуальних завдань.

2. ПОСТАНОВКА ЗАДАЧ ПРАКТИЧНИХ РОБІТ, МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДЛЯ ЇХ ВИКОНАННЯ ТА ІНДИВІДУАЛЬНІ ЗАВДАННЯ

2.1. Практична робота № 1

Створення навчальних тестів в додатку Forms хмарного сервісу Microsoft 365

Об'єкт – офісні_хмарні додатки. Предмет – додаток Forms хмарного сервісу Microsoft 365. Мета – ознайомлення з засобами та методами створення навчальних тестів в додатку Forms хмарного сервісу Microsoft 365.

Стислі теоретичні відомості

Microsoft 365, колишній Office 365 – платний хмарний власницький інтернет-сервіс і програмне забезпечення компанії Microsoft, що розповсюджується за схемою SasS (Software as Services) «програмне забезпечення як послуги». Хмарний формат означає, що дані зберігаються в центрі обробки даних, а не на комп'ютері, що забезпечить користувачам доступ до документів і даних через браузер з різних пристроїв з можливістю виходу в Інтернет.

Сервіс Office 365 був анонсований в жовтні 2010 року, публічне бета-тестування розпочалося у квітні 2011. Світова прем'єра відбулася 28 червня 2011 року, коли корпорація Microsoft запустила сервіс Office 365 у 40 країнах світу. Пакет призначений для використання в компаніях від малого бізнесу до великих підприємств. Середньому та малому бізнесу він буде надаватися через партнерів корпорації, а великими клієнтами Microsoft займеться сама в рамках програми корпоративного ліцензування Enterprise Agreement, яка включає ліцензування базових продуктів Microsoft для всіх використовуваних ПК (від 250 робочих місць).

21 квітня 2020 року споживчі та малі бізнес-плани Office 365 були перейменовані в Microsoft 365, щоб підкреслити включення інших продуктів та послуг до сімейства Microsoft Office (включаючи інструменти для продуктивності на основі хмарних технологій та штучного інтелекту).

Microsoft Office постачається у кількох редакціях. Відмінності редакцій полягають у складі пакета та ціні. Найбільш повна їх містить:

- Microsoft Word – текстовий процесор. Доступний під Windows, Android та macOS. Дозволяє готувати документи різної складності. Підтримує OLE, модулі сторонніх розробників, що підключаються, шаблони та багато іншого. Основним форматом в останній версії є позиціонований як відкритий Office Open XML, який є ZIP-архівом, що містить текст у вигляді XML, а також всю необхідну графіку. Найбільш поширеним залишається двійковий формат файлів Microsoft Word 97-2003 із розширенням DOC. Продукт займає провідне становище над ринком текстових процесорів, та її формати використовуються як стандарт де-факто у документообігу більшості підприємств. Word також доступний у деяких редакціях Microsoft Works.

Програми-еквіваленти: LibreOffice Writer, OpenOffice Writer, KWord, NeoOffice Writer, Corel WordPerfect та Apple Pages (лише на платформі macOS), SoftMaker Office – TextMaker, а також, з деякими застереженнями AbiWord (в тих випадках, коли його можливостей достатньо, а малий обсяг та швидкість роботи при невисоких вимогах до ресурсів більш важливі).

- Microsoft Excel – таблицний процесор. Підтримує всі необхідні функції створення електронних таблиць будь-якої складності. Займає провідне становище на ринку. Остання версія використовує формат Office Open XML з розширенням .xlsx, раніше версії використовували двійковий формат з розширенням .xls. Доступний під Windows та Apple macOS,

Схожі за призначенням програми: LibreOffice Calc, OpenOffice Calc, KCells, Gnumeric, Corel Quattro Pro, Apple Numbers (лише на платформі macOS) та SoftMaker Office – PlanMaker.

- Microsoft Outlook (не плутати з Outlook Express) – персональний комунікатор. До складу Outlook входять календар, планувальник завдань, записки, менеджер електронної пошти, адресна книга. Підтримується спільна мережна робота.

Програми-еквіваленти: поштового клієнта – Mozilla Thunderbird/SeaMonkey, Novell Evolution, Claws Mail, Eudora Mail, The Bat!, Sylpheed; диспетчера персональних даних – Mozilla, Lotus Organizer та Novell Evolution. Доступний під Windows та під macOS.

- Microsoft PowerPoint – програма підготовки презентацій під Windows та MacOS.

Програми-еквіваленти: LibreOffice Impress, OpenOffice Impress, KPresenter, Corel WordPerfect, Apple Keynote (тільки на платформі macOS) та SoftMaker Office – Presentations.

- Microsoft Access – програма для керування базами даних (СУБД).
Програми-еквіваленти: LibreOffice Base, OpenOffice Base, Kexi.
- Microsoft InfoPath – програма для збирання даних та керування ними – спрощує процес збирання даних.
- Microsoft Communicator (Microsoft Lync) – комунікатор призначений для організації всебічного спілкування між людьми. Microsoft Office Communicator 2007 забезпечує можливість спілкування за допомогою простого обміну миттєвими повідомленнями, а також проведення голосової та відеорозмови. Ця програма є частиною програмного пакета Microsoft Office і тісно з ним інтегрована, що дозволяє йому працювати спільно з будь-якою програмою сімейства Microsoft Office.
- Microsoft Publisher – програма для підготовки публікацій.
- Microsoft Project – керування проектами.
- Microsoft Query – перегляд та вибір інформації з баз даних.
- Microsoft OneNote – програма для запису нотаток та керування ними.
- Microsoft Groove – програма підтримки спільної роботи[3].
- Microsoft SharePoint Designer – інструмент для створення програм на платформі Microsoft SharePoint та адаптації вузлів SharePoint.
- Microsoft Picture Manager – менеджер малюнків.

Програма-еквівалент: LibreOffice Draw.

- Microsoft Document Image Writer – віртуальний принтер, який друкує формат Microsoft Document Imaging Format.
- Microsoft Diagnostics – діагностика та відновлення пошкоджених програм Microsoft Office.

Раніше в Microsoft Office входила програма Microsoft FrontPage (програма для створення сайтів), проте Microsoft прийняла рішення виключити цю програму з Office і припинити її розробку. У Microsoft Office 2007 програма FrontPage була замінена на Microsoft SharePoint Designer.

Також раніше до складу Microsoft Office входив редактор формул Equation Editor 3.0 розробки компанії Design Science, який був видалений у січні 2018 через проблеми з безпекою. Користувачам Equation Editor 3.0 рекомендовано перехід на редактор

Microsoft 365, як хмарний сервіс, також включає додатки, що орієнтовані на розподілене зберігання та обробку даних. Серед них слід відзначити наступні.

OneDrive (раніше: SkyDrive, Windows Live SkyDrive і Windows Live Folders) – це файловий хостинг, що надається компанією Майкрософт як частина набору онлайн-послуг. Він дозволяє користувачам зберігати файли, а також інші особисті дані, такі як налаштування Windows або ключі відновлення BitLocker у хмарі. Файли можна синхронізувати з комп'ютером та отримувати доступ до них з Web-браузера або мобільного пристрою, а також ділитися публічно або з певними людьми. OneDrive пропонує 5 Гб вільного місця для зберігання. Додатковий обсяг пам'яті можна додати окремо або через підписку на інші служби Microsoft, включаючи Microsoft 365 та Groove Music.

Microsoft Teams – центр для командної роботи, який інтегрує користувачів, вміст і засоби, необхідні команді для ефективнішої роботи. Додаток об'єднує все в спільному робочому середовищі, що містить засіб проведення відеоконференцій, календар-щоденник з запланованими нарадами, чат, файлообмінник та корпоративні програми. Розроблений для смартфонів, що працюють на платформах Android, iOS, Windows Phone і комп'ютерів з операційною системою Windows 7 та вище або Mac OS X 10.10 та новіше. Microsoft Teams є конкурентом таких сервісів, як Slack, і є еволюційним оновленням від Microsoft Skype.

Microsoft Forms – це онлайн програма для створення форм опитувань і тестів.

Основні характеристики:

- багатий вибір типів тестів – вибір однієї чи множини відповідей, встановлення відповідності, встановлення послідовності, ессе;
- налаштування часових рамок;
- вибір користувачів;
- оцінювати отримані відповіді за допомогою вбудованих засобів аналітики;
- можливість експорту даних в таблиці Microsoft Excel;
- спільна робота над формами;

- можливість надсилання форм для копіювання.

Служба Microsoft Forms зазвичай доступна для Office 365 Education, Microsoft 365 для бізнесу та користувачів облікових записів Microsoft (Hotmail, Live або Outlook.com).

Порядок створення тесту

1. В Microsoft 365 стартується додаток Forms.
2. Обирається тип форми *Тест* (рис. 2.1.1).

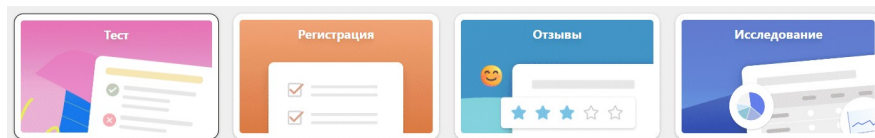


Рис. 2.1.1. Обрання типу форми *Тест*

3. Тесту надається назва (рис. 2.1.2) і натискається кнопка *Стисле керівництво з*.

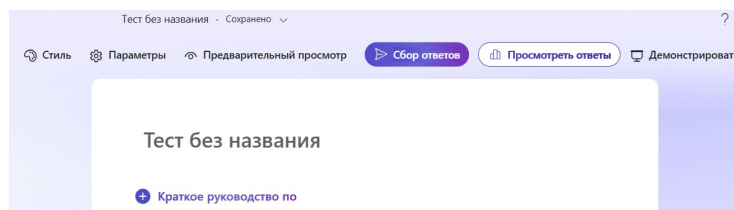


Рис. 2.1.2. Надання назви тесту

4. Обирається вид запитання і вноситься його зміст (рис. 2.1.3 – 2.1.8).

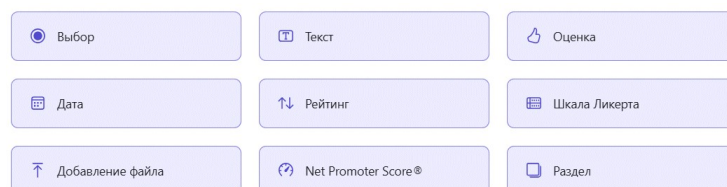


Рис. 2.1.3. Обрання виду запитання *Вибір*

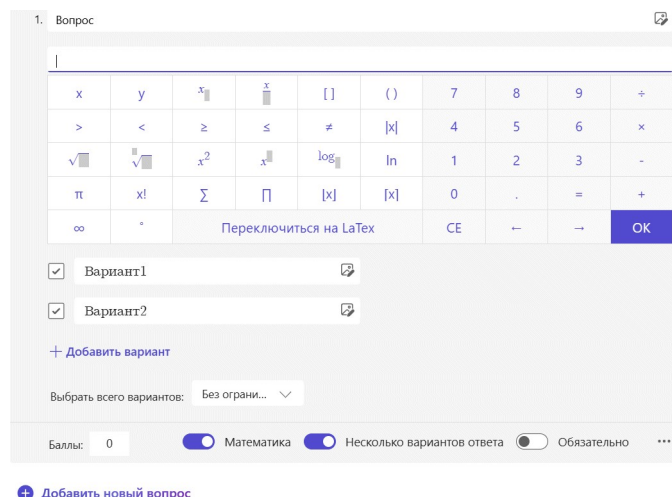


Рис. 2.1.4. Внесення змісту тесту *Вибір* (початок)

Якщо запитання містить формули, або кілька варіантів відповіді, перемикачі *Математика* та *Кілька варіантів відповіді* вмикаються.

Вводиться зміст запитання та варіанти відповідей.

Для додавання варіантів відповіді натискається кнопка *Додати варіант*.

Ліворуч від варіантів відповіді помічаються правильні відповіді.

Proba1 (Баллов: 2)

1. Оберіть корені рівняння

$x^2 - 1 = 0$

☒ 1

☒ -1

+ Додати варіант

Вибрати всего вариантов: Без ограни...

Баллы: 2

Математика

Несколько вариантов ответа

Обязательно

+ Додати новий вопрос

Активация WiP
Перейти до роз

Рис. 2.1.5. Внесення змісту тесту *Вибір* (закінчення)

Після повного введення змісту запитання вказується кількість балів за повну правильну відповідь та натискається кнопка *Ввести нове запитання*.

Рейтинг

Дії аналогічні попередньому випадку. Події про які йдеться в запитанні впорядковуються від найбільш ранньої до найбільш пізньої.

0

2. Введіть назви університету в хронологічній послідовності

Введите подзаголовок

Список параметров в правильном порядке ⓘ

Д Г І

Д Г А У

Н Г А У

Н Г У

Н Т У Д У

+ Додати варіант

Баллы: 4

Обязательно

+ Додати новий вопрос

Активация WiP
Перейти до роз

Рис. 2.1.6. Внесення змісту тесту *Рейтинг*

Шкала Ликерта

Дії аналогічні попереднім випадкам. Елементи множин, які ставляться одна одній у відповідність, вводяться у довільному порядку. Додавання та вилучення елементів множин здійснюється кнопками *Плюс* та *Корзина*.

Введите подзаголовок	Україна	Франція	Італія	Іспанія
Мадрид	☐	☐	☐	☐
Париж	☐	☐	☐	☐
Рим	☐	☐	☐	☐
Київ	☐	☐	☐	☐

Рис. 2.1.7. Внесення змісту тесту *Шкала Ликерта*

Дата

Дії аналогічні попереднім випадкам. Поле дати залишається порожнім.

Введите дату принятия Декларации про государственный суверенитет Украины

Введите подзаголовок

Введите дату (dd.MM.yyyy)

Рис. 2.1.8. Внесення змісту тесту *Дата*

Після закінчення введення змісту всіх запитань, здійснюється перехід до пункту меню *Параметри*.

Знімається помітка *Одна відповідь на людину*, встановлюються *Дати початку* (момент створення тесту) і *закінчення* (кінець літньої сесії) тестування, встановлюється помітка *Розташувати запитання у випадковому порядку*. Решту налаштувань можна залишити за замовченням (рис. 2.1.9).

Проба1 (Баллов: 14)

1. Оберіть корені рівняння (Баллов: 2)
 $x^2 - 1 = 0$

☐ 1 ✓
☐ -1 ✓
☐ 0

2. Введіть назви університету в хронологічній послідовності (Баллов: 4)

ДГІ
ДГАУ
НГАУ
НГУ
НТУ ДП

3. Поставте у відповідність країни та їх столиці (Баллов: 4)

Параметры

Автоматическая оценка
Рассчитать сумму оценок сразу же после отправки теста

Кто может заполнить эту форму
☐ Любой пользователь может ответить
☒ Оплатить могут только люди в Dnipro University of Technology
Обязательно войти для подтверждения доступа в не позднее, чем через Dnipro University of Technology

☒ Записывать имя
☐ Один ответ на человека

☐ Конкретные люди в Dnipro University of Technology могут ответить

Параметры ответов
☒ Принимать ответы
☒ Дата начала
15.01.2025 15:00
☒ Дата окончания
15.06.2025 15:00

Рис. 2.1.9. Налаштування параметрів тесту

Здійснюється перехід до пункту меню *Збір відповідей*.

Встановлюється вибір *Відповісти можуть тільки люди в Dnipro University of Technology*. Отримання посилання на тест здійснюється кнопкою *Копіювати посилання*. Посилання має вигляд

<https://forms.office.com/Pages/ResponsePage.aspx?id=p4K7bIuPIU6JlujdHwPmNgoHl2lx2u9PgXk6MziNd3ZUMFhANlIYQ0tSSDAyQkpMUIQ2UUNLQ1gyQS4u>

Для проходження тестування можна скористатися будь-яким браузером.

Перевірка статистики результатів тестування здійснюється за пунктом меню *Переглянути відповіді* (рис. 2.1.10).

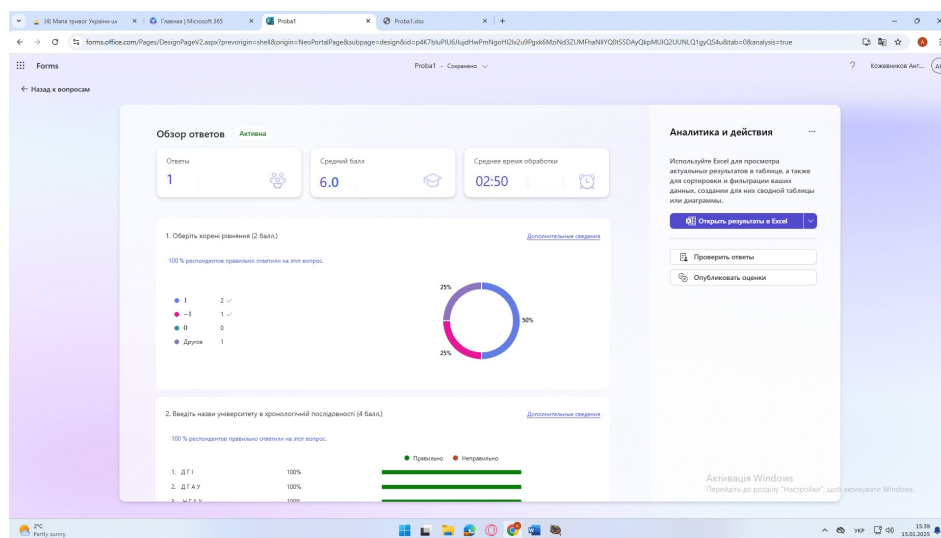


Рис. 2.1.10. Перегляд статистики результатів тестування

Експорт результатів в Excel здійснюється кнопкою *Відкрити результати в Excel* (рис. 2.1.11).

The screenshot shows an Excel spreadsheet with the following data:

Відповісти можуть тільки люди в Dnipro University of Technology	Оберіть корені рівняня (2 балл)	Введіть назви університету в хронологічній послідовності (4 балл)
1	1, -1, 0, Друге	ДГУ, ДГАУ, ДСАУ, ДСАУ

Рис. 2.1.11. Результати тестування, які експортовані в Excel

Відкриття відповідей кожного учасника тестування для перевірки здійснюється за пунктом меню *Перевірити відповіді* (рис. 2.1.12).

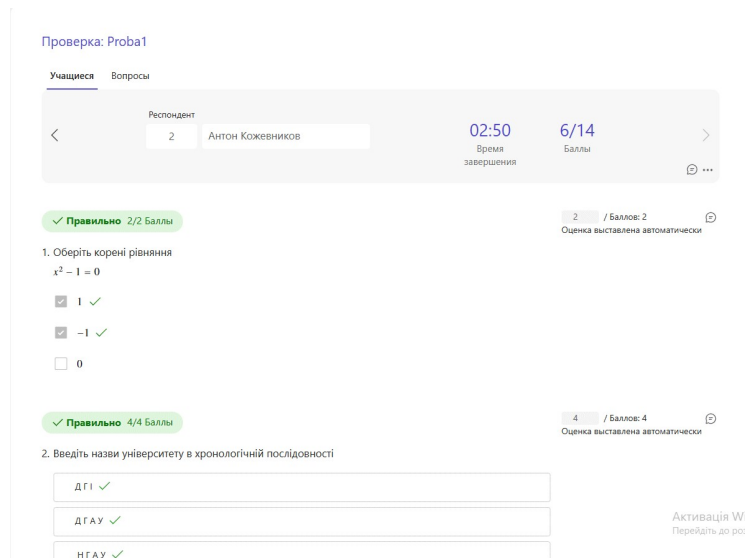


Рис. 2.1.12. Детальна перевірка відповідей

Тест повинен задовольняти наступним вимогам:

- зміст запитань повинен суворо відповідати запропонованим у таблиці темам з історії України або фізики згідно варіанту завдання, **тести, які не відповідатимуть цій умові не перевірятимуться;**
- тест повинен складатися з п'яти запитань, принаймні одне з яких повинно бути встановленням порядку (рейтингуванням подій), або встановленням відповідності між елементами двох множин (шкала Ликерта), решта запитань повинні бути у вигляді вибору з одним, або кількома правильними відповідями;
- принаймні одне запитання повинно містити одну або кілька формул, які можуть знаходитися або власно у запитанні, або у варіантах відповідей;
- якщо тема з історії України згідно варіанту завдання містити конкретну дату, одне з запитань припускається створювати у вигляді дати;
- припускається зміст запитання з встановленням порядку (рейтингуванням подій) використовувати для запитання з встановленням відповідності між елементами двох множин (шкала Ликерта), при цьому однією множиною буде порядковий номер події з другої множини.

Постановка задачі

Дано

Теми з історії України та фізики позичені з підручників, які наведені в списку використаної та рекомендованої літератури.

Потрібно

За допомогою додатку Forms хмарного сервісу Microsoft 365 створити online-тест за темами згідно варіанту завдання, який задовольняє зазначеним вище вимогам.

Рекомендації щодо виконання роботи

1. Сформулювати запитання тесту і відповіді на них згідно варіанту завдання і вимог, які зазначені вище.
2. Створити форму тесту та отримати посилання на неї.
3. Самостійно пройти тестування.
4. Надіслати посилання на корпоративну пошту викладача.
5. Після погодження викладачем, сформулювати звіт і надіслати його на корпоративну пошту викладача.

Вимоги до звіту

Звіт по роботі повинен містити:

1. Назву дисципліни та лабораторної роботи.
2. Прізвище, ім'я та по батькові здобувача вищої освіти, код групи.
3. Номер варіанту завдання і теми, які йому відповідають.
4. Знімки зображень екрану, які виникають при створенні тесту і його перевірці, з підписами (Рис. 5 – 12), при цьому кожне запитання тесту після його створення повинне бути представлено на окремому рисунку.

Контрольні питання і завдання

1. Чим відрізняються хмарні версії офісних програмних продуктів від локальних?
2. Назвіть основні складові Microsoft 365.
3. Які типи шаблонів підтримує Microsoft Forms?
4. Які типи тестів підтримує Microsoft Forms?
5. Які статистичні дані за результатами тестування надає Microsoft Forms?

Варіанти завдань

Таблиця 2.2.1. Темі тестів

№	Історія України	Фізика
1	Стародавня доба	Кінематика матеріальної точки
2	Бронзова доба. Скіфо-сарматський час	Кінематика твердого тіла
3	Античні міста-держави Північного Причорномор'я	Динаміка матеріальної точки
4	Походження та розселення слов'ян	Динаміка обертального руху
5	Заняття, побут та звичаї слов'ян. Племінні союзи	Робота і енергія
6	Народження та розбудова Київської Русі	Гармонічні коливання. Звукові хвилі
7	Економічне життя та побут Київської Русі	Механіка рідин
8	Християнізація Русі	Макроскопічні стани та параметри

9	Культура Київської Русі	Молекулярно-кінетична теорія ідеального газу
10	Причини розпаду Київської Русі. Київське, Чернігово-Сіверське, Переяславське князівства	Статистичні розподіли
11	Боротьба з монголо-татарською навалою та її світове явище	Перше начало термодинаміки
12	Походження козацтва	Теплоємність
13	Виникнення та устрій Запорозької Січі. Реєстрове козацтво	Адіабатний процес. Рівняння Пуассона
14	Морські походи (Запорізького козацтва)	Теплові машини
15	Українська Козацька держава в період національно-визвольної війни під проводом Б. Хмельницького	Цикл Карно. Теорема Карно
16	Двозначність Переяславської угоди	Рівність Клаузіуса. Ентропія
17	Етапи руйнування козацької держави	Тверді тіла і рідини
18	Приєднання українських земель до складу Російської та Австрійської імперій	Постійне електричне поле у вакуумі
19	Реформи в Російській імперії у другій половині XIX ст. та соціально-економічний розвиток України в пореформений період	Діелектрики і провідники в електростатичному полі
20	Український національно-культурний та суспільно-політичний рух. Антиукраїнська політика царизму. Виникнення українських політичних товариств і партій	Постійний електричний струм
21	Початок української революції. Створення Української Центральної Ради. I Універсал	Статичне магнітне поле у вакуумі
22	II Універсал та визнання української політичної автономії	Магнітне поле у речовині
23	III і IV Універсали. Створення УНР та проголошення незалежності України	Електромагнітна індукція
24	УНР часів Директорії	Рівняння Максвелла
25	Фастівський договір 1918 р. - крок до соборності України	Електричні коливання. Змінний струм
26	Політично-правові основи становлення Української держави та її зовнішньополітичні пріоритети	Електромагнітні хвилі. Оптичний діапазон
27	На перехресті політичних перспектив. Поширення ідей українського	Дисперсія і поглинання світла

	консерватизму	
28	Політика воєнного комунізму та її наслідки	Фотометричні поняття і величини
29	Неп у “командних висотах” народного господарства.	Геометрична оптика
30	Голод 1921–1923 рр. в Україні	Інтерференція світла
31	Перехід до планової економіки. Голодомор в Україні 1932–1933 рр.	Дифракція світла
32	Соціально-економічне й політичне становище Західної України в 20–30-х роках	Поляризація світла
33	Національно-визвольний рух. Політичні партії, їх програми і діяльність	Квантові властивості світла
34	Мюнхенська угода та її наслідки	Основні положення квантової механіки
35	Возз’єднання західноукраїнських земель у складі УРСР	Атом водню та воднеподібні атоми
36	Рух Опору в Україні (1941–1944 рр.)	Основні властивості атомних ядер
37	Окупаційний режим та визволення України	Фундаментальні взаємодії та ядерні сили
38	Суспільно-політичне життя УРСР після Другої світової війни. Голод 1946–1947 рр. в Україні	Ядерні реакції. Радіоактивність
39	Політичні зміни в УРСР у період “хрущовської відлиги”	Кінематика матеріальної точки
40	Економічний розвиток УРСР у другій половині 50-х - на початку 60-х років	Кінематика твердого тіла
41	Особливості національно-демократичного руху в Україні у 60–80-х роках	Динаміка матеріальної точки
42	Загострення кризи тоталітарної системи в Україні у другій половині 80-х - на початку 90-х років	Динаміка обертального руху
43	Відновлення незалежності України	Робота і енергія
44	Україна за часів незалежності: етапи державотворення	Гармонічні коливання. Звукові хвилі
45	Українська Помаранчева революція	Механіка рідин

2.2. Практична робота № 2

Інсталяція гостьової операційної системи у вигляді віртуальної машини Oracle VM Virtual Box

Об'єкт – засоби віртуалізації. Предмет – засіб програмної віртуалізації Oracle VM Virtual Box. Мета – проведення та виявлення її особливостей інсталяції гостьової операційної системи у вигляді віртуальної машини Oracle VM Virtual Box.

Стислі теоретичні відомості

Технологія віртуалізації призначена для реалізації можливості одночасного запуску на одному комп'ютері декількох (в тому числі, різних) операційних систем (ОС).

В ОС, реалізованій на віртуальній машині (ВМ), є можливість виконувати різні маловивчені або потенційно небезпечні для неї операції, не турбуючись про наслідки – оскільки система є віртуальною, її крах або часткове пошкодження не позначиться на роботі реальної ОС.

Найбільш розповсюджені віртуальні машини:

- VirtualBox (Oracle) (<https://www.virtualbox.org/>);
- VMware Workstation (<http://www.vmware.com/>);
- Virtual PC (Microsoft) (http://ru.wikipedia.org/wiki/Virtual_PC).

VirtualBox – це програма віртуалізації для операційних систем, розроблена німецькою фірмою Innotek, зараз вона належить Oracle Corporation. Програма встановлюється на існуючу ОС, яка називається хостовою, усередину цієї програми встановлюється друга ОС, яку називають гостьовою, що функціонує на ВМ.

В якості хостових підтримується основними ОС Linux, FreeBSD, Mac OS X, OS/2 Warp, Microsoft Windows, які працюють з гостьовими ОС FreeBSD, Linux, OpenBSD, OS/2 Warp, Windows і Solaris.

Характеристики VirtualBox:

- є крос-платформним додатком віртуалізації і працює на комп'ютерах з процесорами Intel або AMD під управлінням операційних систем Windows, Mac, Linux або Solaris;
- розширює можливості комп'ютера тим, що дозволяє працювати всередині віртуальних машин множині операційних систем одночасно;
- дозволяє встановити і запустити певну кількість віртуальних машин (ця кількість обмежується тільки розміром вільного дискового простору і пам'яті);
- надає можливість установки гостьових доповнень Guest Additions для підвищення ступеня інтеграції з хостовою ОС;
- підтримує множину знімків поточного стану гостьової системи (snapshots);
- підтримує диски динамічно розширюваного та фіксованого розміру.

Основна термінологія

Віртуальна машина (ВМ, VM) – це спеціальне середовище, яке VirtualBox створює для роботи окремої гостьової ОС.

Операційна система хоста (основна ОС, хостова ОС, host OS) – це операційна система фізичного комп'ютера, в якій встановлено програмне забезпечення для віртуалізації.

Гостьова операційна система (гостьова ОС, guest OS) – це операційна система, яка працює всередині віртуальної машини.

Гостьові доповнення (Guest Additions) – це пакет програмних продуктів, що постачаються з VirtualBox і розроблені для установки всередині ВМ, що сприяють поліпшенню продуктивності гостьової ОС та наданню додаткової функціональності.

Файл конфігурації ВМ – основа віртуальної машини, зберігає всі налаштування, що стосуються ВМ. Являє собою XML-файл, який має розширення vbox.

Розмір пам'яті ВМ – об'єм пам'яті, що виділений для ВМ. Буде не доступний для хоста.

Файл віртуального жорсткого диска (VDI, VMDK, VHD) – файл, який існує у хостовій ОС та описує віртуальний жорсткий диск, створюваний для гостьової ОС.

Знімок віртуальної машини (snapshots) – знімок стану ВМ. Він зберігає зміни, які відбулися у ВМ з моменту останнього зняття знімку.

Диференціальні диски – основа технології снапшотів. При створенні снапшоту запис у файл VHD (VDI) припиняється, і всі наступні зміни записуються в окремий файл, який має розширення VHD (VDI).

Опис ВМ

VirtualBox сприймає ВМ як множину параметрів, які визначають її поведінку і включають:

- апаратні налаштування (розмір пам'яті, виділеної для ВМ, типи жорстких дисків (ЖД), підключені CD тощо);
- поточний стан (ВМ працює, збережена, вимкнена).

Налаштування представлені у вікні менеджера VirtualBox, а також ними можна управляти через програму командного рядка VBoxManager.

Формати та типи віртуальних жорстких дисків

VirtualBox підтримує наступні види образів віртуальних жорстких дисків (ВЖД):

- VDI (Virtual Disk Image) – власний формат віртуальних дисків. Зокрема, цей формат використовується, коли ви створюєте майстром нову віртуальну машину.
- VMDK (Virtual Machine Disk) – популярний відкритий формат, який використовується у багатьох інших програмах віртуалізації, зокрема VMware.
- VHD (Virtual Hard Drive) – формат, який використовується Microsoft. З VHD-дисками можна працювати на рівні ОС, прямо до завантаження з них самої ОС.

Існує два типи віртуальних дисків:

- динамічний ВЖД;
- фіксований ВЖД.

Динамічний віртуальний жорсткий диск – динамічно розширюваний дисковий образ. Він забезпечує обсяг диска, необхідний для зберігання поточних даних. Розмір VHD-файлу в момент створення такого диска невеликий, але збільшується в міру додавання даних на диск. При видаленні даних із ВЖД розмір VHD-файлу автоматично не зменшується. Наприклад, якщо був створений динамічно розширюваний жорсткий диск, об'ємом 5 Гб, то спочатку цей файл на хостовій системі буде займати близько двох мегабайт. У міру накопичення інформації в цьому образі, його розмір буде збільшуватися, поки не досягне п'яти гігабайт. Недоліком такого типу ВЖД є зниження швидкодії гостьової ОС, оскільки з часом ВЖД фрагментується і звернення до нього займає багато часу.

Фіксований віртуальний жорсткий диск – дисковий образ фіксованого розміру. Він забезпечує необхідну ємність, використовуючи VHD-файл, розмір якого задається при створенні віртуального жорсткого диска. Розмір VHD-файлу залишається постійним незалежно від обсягу збережених даних. Наприклад, якщо був створений диск фіксованого розміру, об'ємом 5 Гб, то система створить файл, який займатиме орієнтовно 5 Гб. На відміну від динамічного, фіксований віртуальний жорсткий диск не фрагментується.

Якщо місця на жорсткому диску, який використовується для створення віртуального жорсткого диска, багато, Oracle рекомендує використовувати фіксований формат. Якщо ж місця мало – доцільно обрати динамічний формат.

Проведення інсталяції програми Virtual Box і його розширень.

Запускається інсталятор. Отримуються привітання вікна (рис. 2.2.1).



Рис. 2.2.1. Інсталяція програми Virtual Box (початок)

Здійснюється перехід на наступний крок.

На другому кроці інсталяції можливо переглянути всі компоненти програми, які будуть встановлені і каталог (директорію, папку, теку), в яку буде встановлено програму. Каталог для встановлення можна змінити, а компоненти варто встановити всі (рис. 2.2.2).

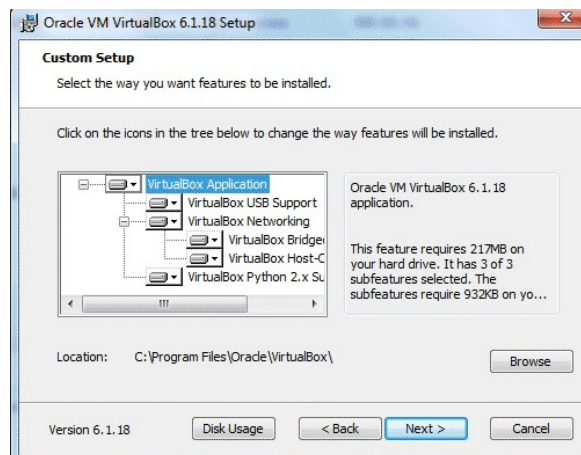


Рис. 2.2.2. Вибір компонентів, що встановлюються

Далі обирається місце розташування посилання (ярлика) програми.

На наступному кроці з'явиться попередження, що в часі встановлення можливо виникнути тимчасове відключення персонального комп'ютера від мережі (рис. 2.2.3).



Рис. 2.2.3. Інсталяція програми Virtual Vох (продовження)

Далі для встановлення програми натискається кнопка *Install* (рис. 2.2.4).

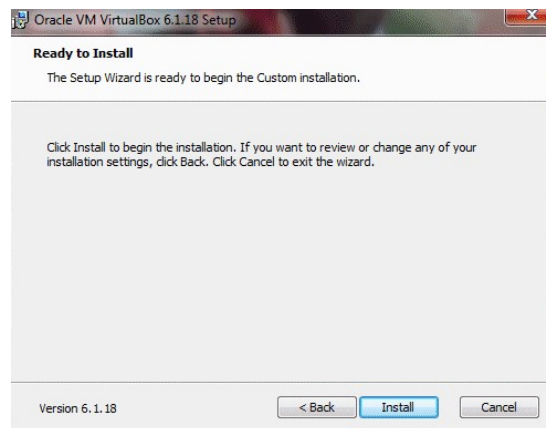


Рис. 2.2.4. Інсталяція програми Virtual Vох (продовження)

Треба зачекати, поки виконається встановлення програми (рис. 2.2.5).

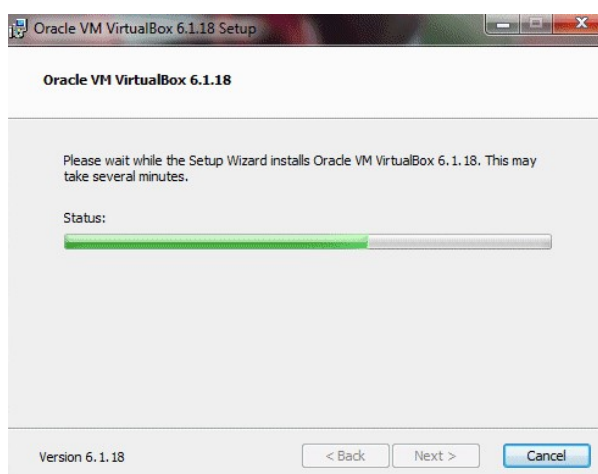


Рис. 2.2.5. Інсталяція програми Virtual Box (продовження)

При виникненні запитів на встановлення контролерів USB, мережевих адаптерів, мережевої служби на них слід відповісти позитивно (рис. 2.2.6 – 2.2.8).

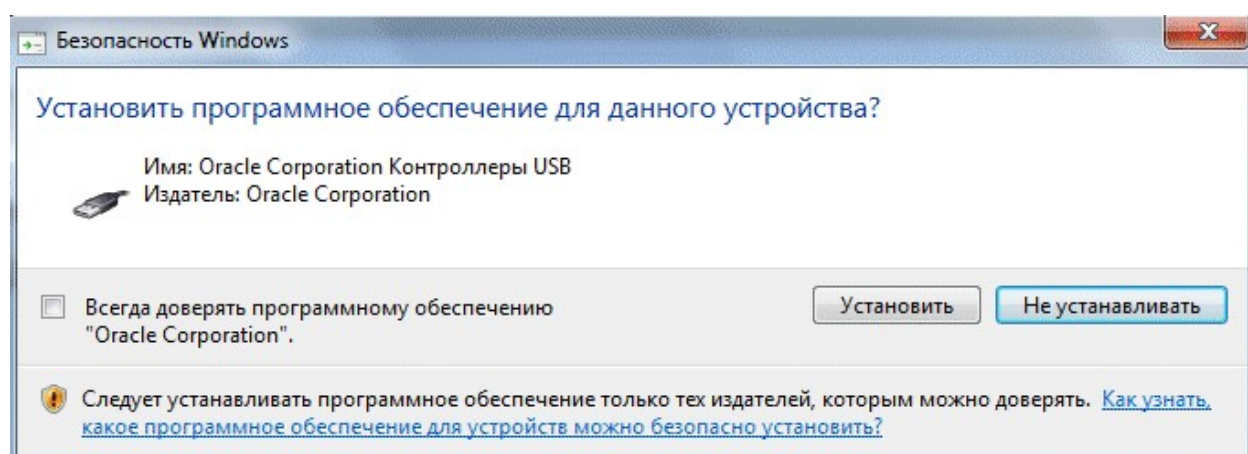


Рис. 2.2.6. Запит інсталятора на встановлення контролерів USB

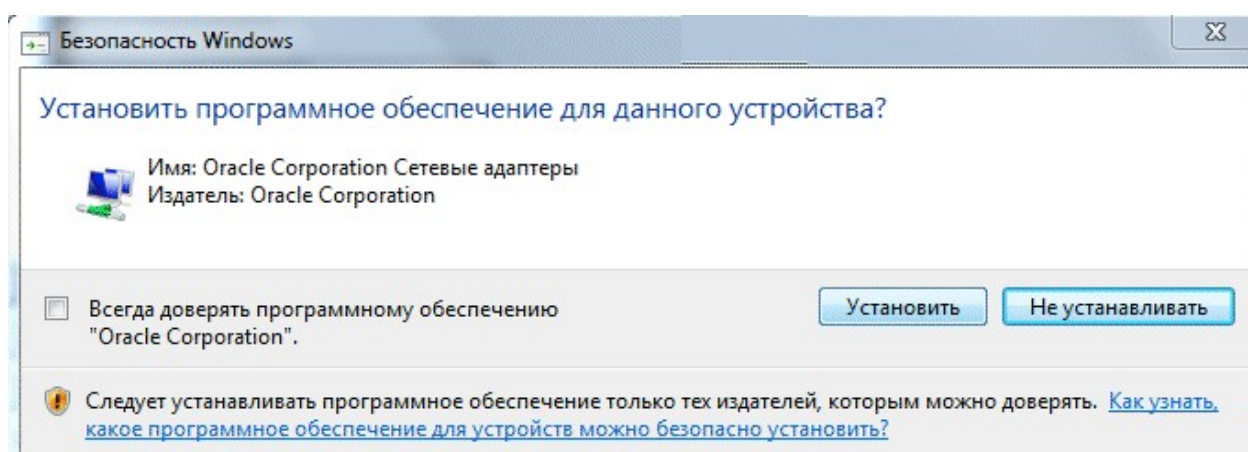


Рис. 2.2.7. Запит інсталятора на встановлення мережевих адаптерів

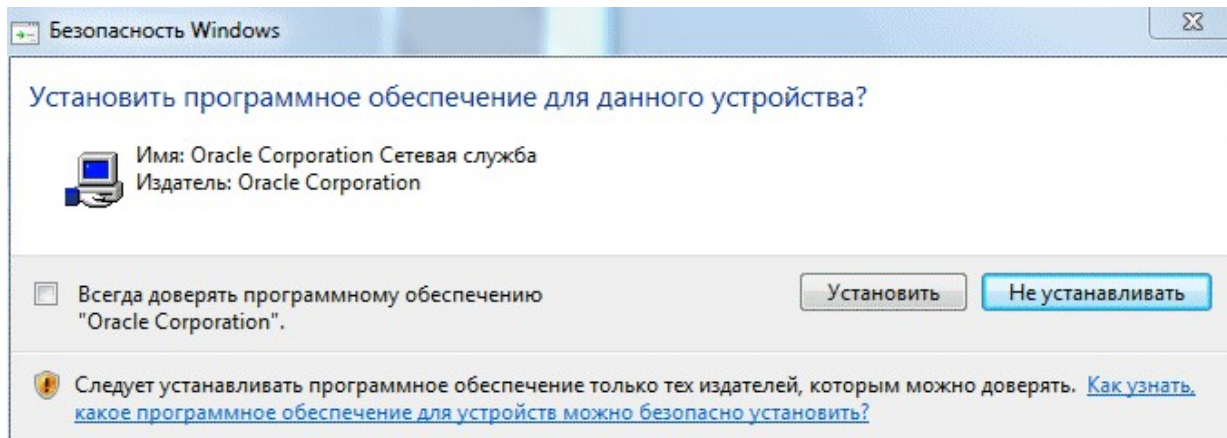


Рис. 2.2.8. Запит інсталятора на встановлення мережевої служби

Встановлення програми завершено. Треба натиснути кнопку *Finish* (рис. 2.2.9).



Рис. 2.2.9. Інсталяція програми Virtual Box (закінчення)

Віртуальна машина автоматично запуститься після закінчення встановлення. Її інтерфейс користувача зі встановленими віртуальними машинами представлений на рис. 2.2.10.

Для проведення наступних лабораторних робіт треба також встановити гостьові доповнення Virtual Box Extension, котрі обов'язково повинні бути тієї ж самої версії, що і власно Virtual Box.

Гостьові доповнення буде необхідне для створення інтерфейсів обміну даними між хостовою ОС і віртуальною машиною гостьової ОС, таких як: двосторонній буфер обміну, функція drag-n-drop, організація спільних каталогів між, організація USB інтерфейса в середовищі гостьової ОС.

Для встановлення гостьових доповнень слід скористатися пунктами меню *Файл – Налаштування – Плагіни* і для пошуку натиснути зелену кнопку на правому боці діалогового вікна (рис. 2.2.11).

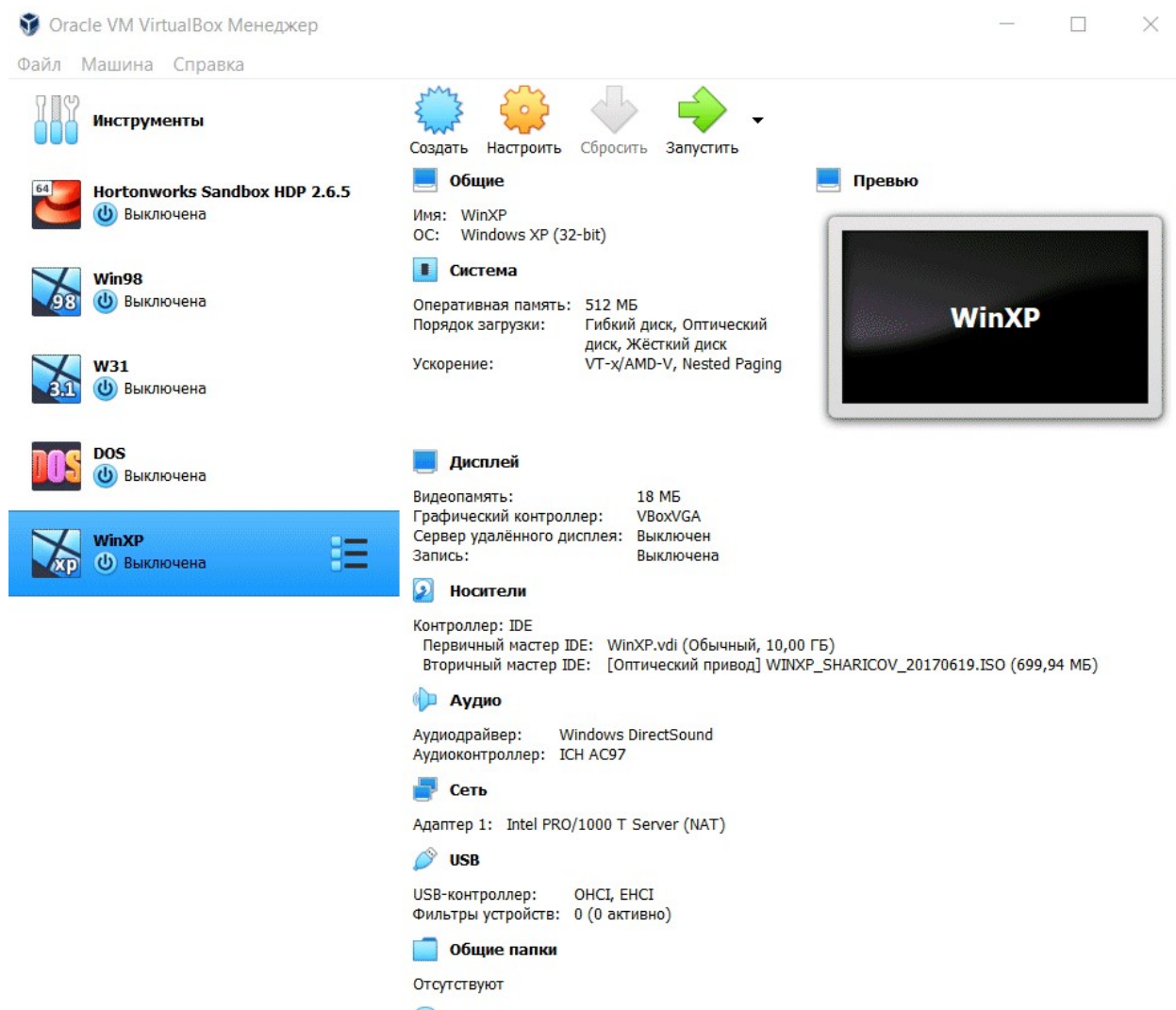


Рис. 2.2.10. Інтерфейс користувача Virtual Box зі встановленими віртуальними машинами

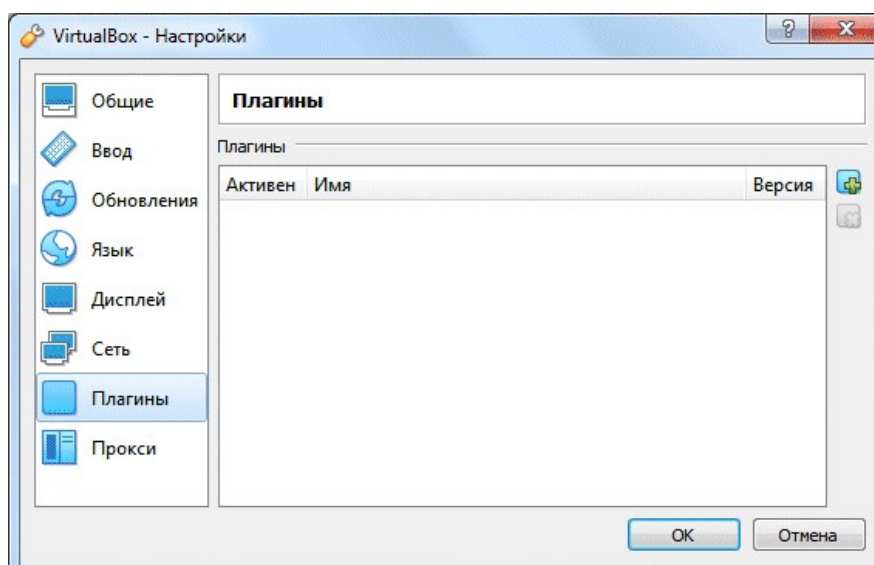


Рис. 2.2.11. Інсталяція гостьових доповнень Virtual Box (початок)

Після вказання розширення файлу доповнення, починається процес його інсталяції (рис. 2.2.12), при цьому треба погодитися з ліцензійною угодою.

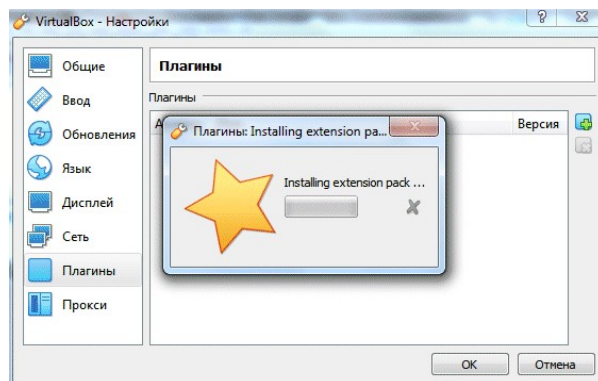


Рис. 2.2.12. Інсталяція гостьових доповнень Virtual Box (продовження)

По закінченню інсталяції доповнень, вікно встановлених плагінів буде виглядати так, як показано на рис. 2.2.13.

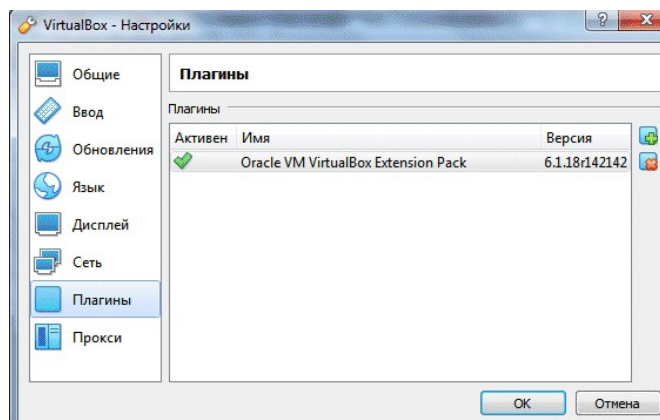


Рис. 2.2.13. Інсталяція гостьових доповнень Virtual Box (закінчення)

Проведення інсталяції гостьової ОС родини Windows (на прикладі Windows XP) у вигляді віртуальної машини Virtual Box.

Створюється віртуальна машина для Windows XP – пункти меню *Машина* – *Створити*. Назва може бути будь-якою. а от два інших пункти) – обов'язково “Microsoft Windows” і “Windows XP” (рис. 2.2.14).

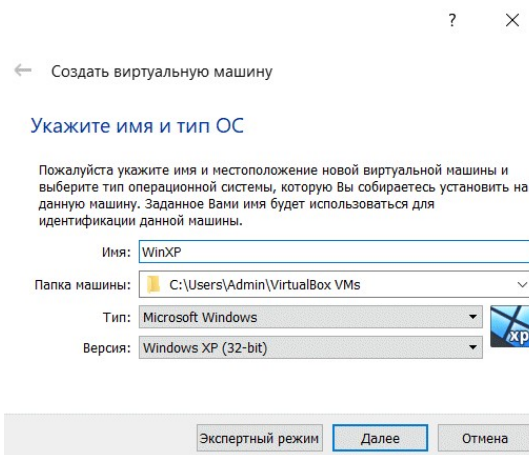


Рис. 2.2.14. Створення віртуальної машини в Virtual Box (початок)

Задається об'єм оперативної пам'яті віртуальної машини (рис. 2.2.15).

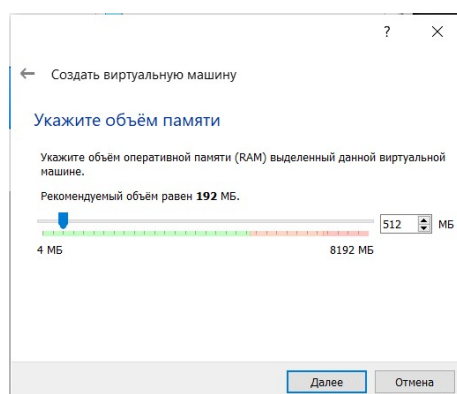


Рис. 2.2.15. Задання об'єму оперативної пам'яті віртуальної машини

Обирається чи використовується існуючий віртуальний жорсткий диск (при використанні завантажувальних жорстких дисків форматів VDI, VMDK, VHD), чи створюється новий (у разі завантажувального оптичного диску формату ISO), як показано на рис. 2.2.16.

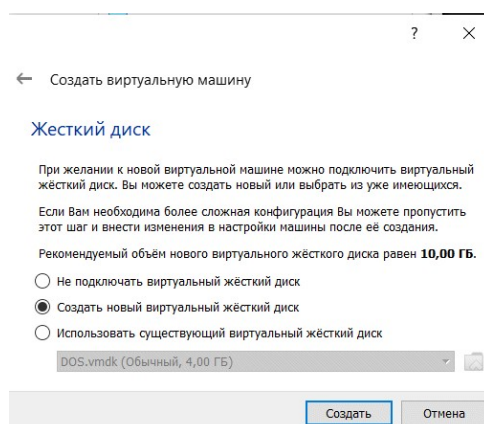


Рис. 2.2.16. Задання параметрів жорсткого диску віртуальної машини

При створенні віртуального жорсткого диска, вказується його формат і можливість зміни його об'єму (рис. 2.2.17 – 2.2.18).

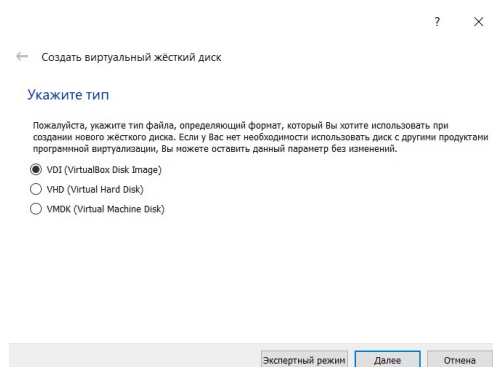


Рис. 2.2.17. Задання параметрів жорсткого диску віртуальної машини

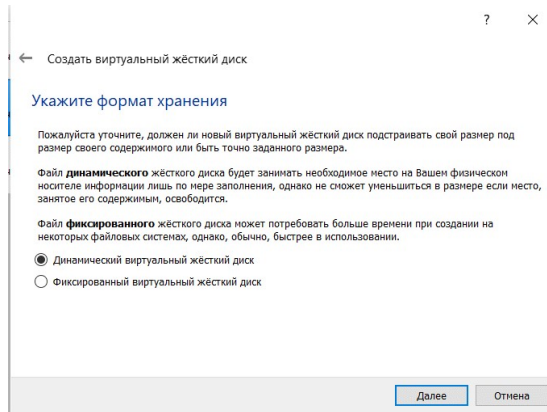


Рис. 2.2.18. Задання параметрів жорсткого диску віртуальної машини

Вказується ім'я віртуального жорсткого диска і його розмір.

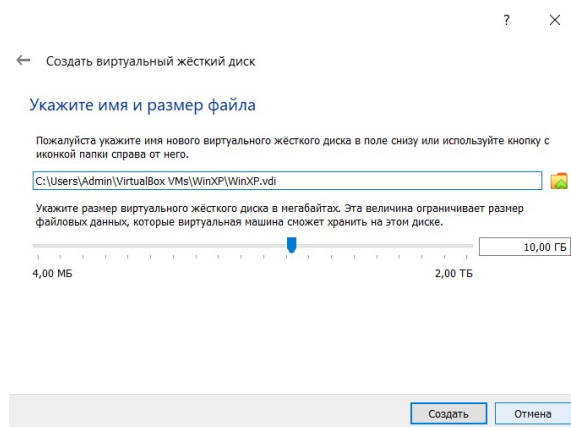


Рис. 2.2.19. Задання імені і розміру жорсткого диску віртуальної машини

Натискується кнопка *Створити*.

Далі до віртуальної машини підключається ISO-образ Windows XP – пункти меню *Налаштувати – Носії – Пусто – Оптичний привід – Вторинний майстер* (Рис. 2.2.20). Далі натискується кнопка поряд з рядком введення *Оптичний привід*, обирається пункт випадаючого меню *Оберіть файл диска*, вказується місце розташування образу віртуального диску і натискується кнопка *OK*.

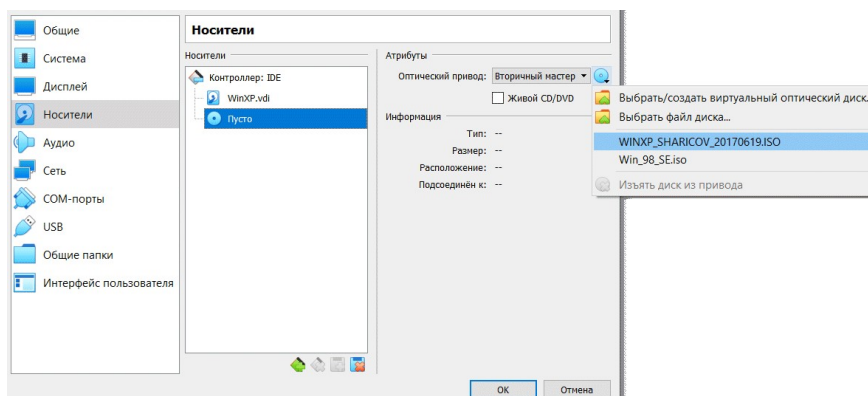


Рис. 2.2.20. Підключення до віртуальної машини образу диску гостьової ОС

Після цього в головному меню Virtual Box натискається кнопка *Запустити*. Починається інсталяція ОС. Підтверджується обраний завантажувальний диск натисненням кнопки *Продовжити* (Рис. 2.2.21).

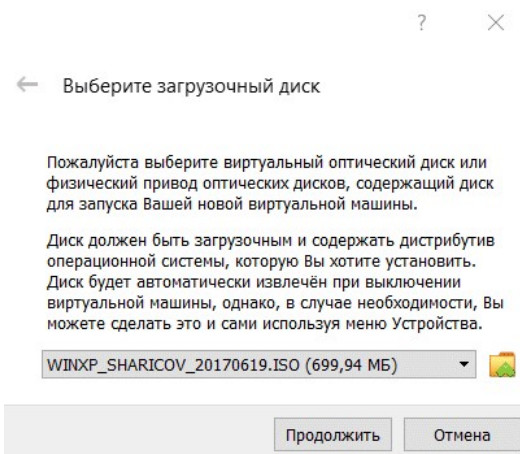


Рис. 2.2.21. Обрання завантажувального диску гостьової ОС

Далі починається власно інсталяція гостьової ОС (рис. 2.2.22 – 2.2.27), при цьому в перших двох діалогових вікнах слід указати >>> *Windows XP SP 3 Russian Установка* та *C Internet Explorer 8* (базова) відповідно.

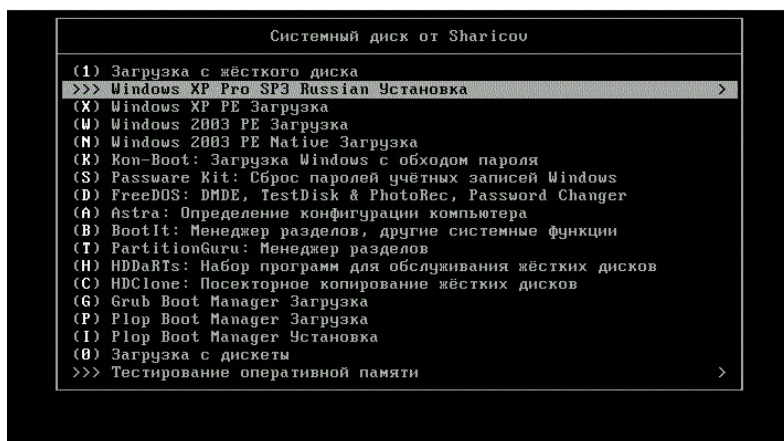


Рис. 2.2.22. Інсталяція гостьової ОС (початок)

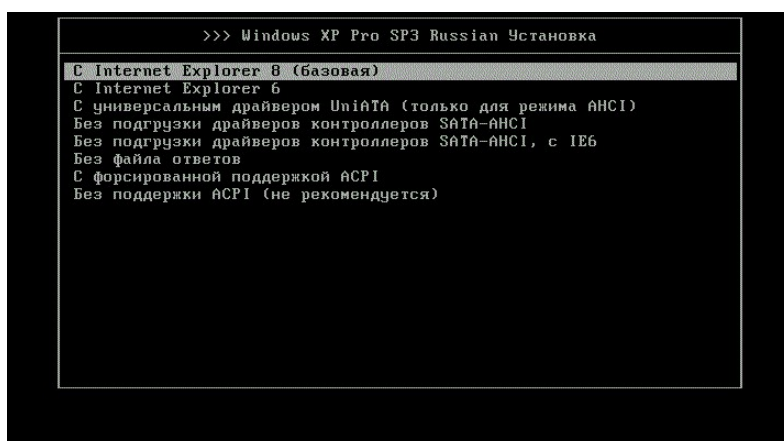


Рис. 2.2.23. Інсталяція гостьової ОС (продовження)

Отримується привітання програми установки, натискається клавіша *Введення*.

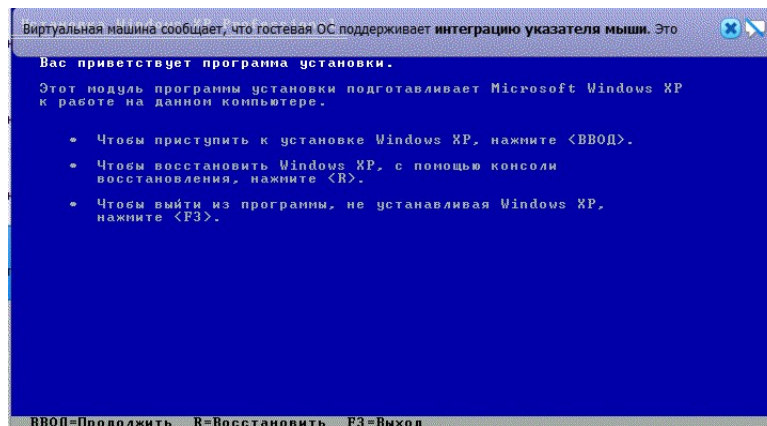


Рис. 2.2.24. Інсталяція гостьової ОС (продовження)

Визначаються розділи віртуального жорсткого диску для розміщення гостьової ОС та їх об'єми, при цьому залишаються значення за замовченням і натискається клавіша *Введення*.

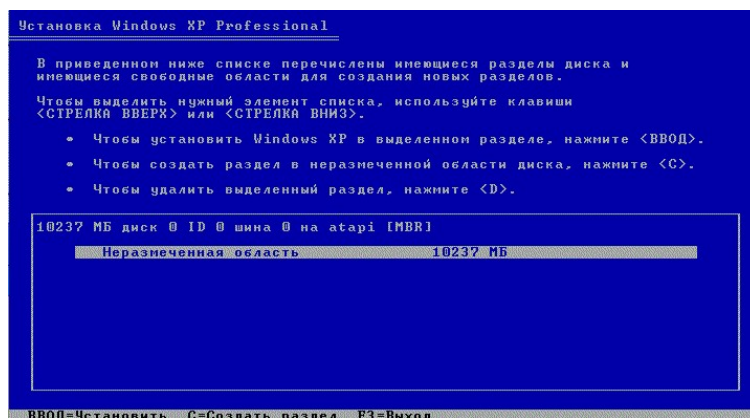


Рис. 2.2.25. Інсталяція гостьової ОС (продовження)

Оббирається повне форматування файлової системи NTFS, після чого слід дочекатися повного форматування диску і копіювання на нього інсталяційних файлів.

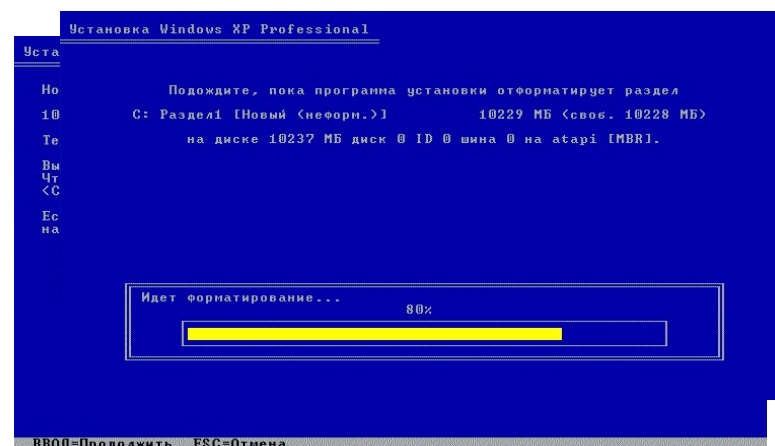


Рис. 2.2.26. Інсталяція гостьової ОС (продовження)

Після цього гостьова ОС перезавантажиться і подальший процес інсталяції триватиме в автоматичному режимі.

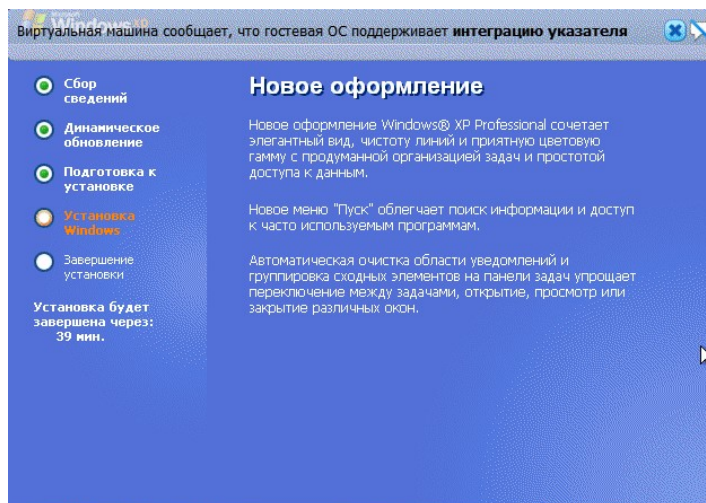


Рис. 2.2.27. Інсталяція гостьової ОС (закінчення)

Після остаточного завантаження гостьової ОС, у вікні віртуальної машини з'явиться її робочий стіл (рис. 2.2.28).

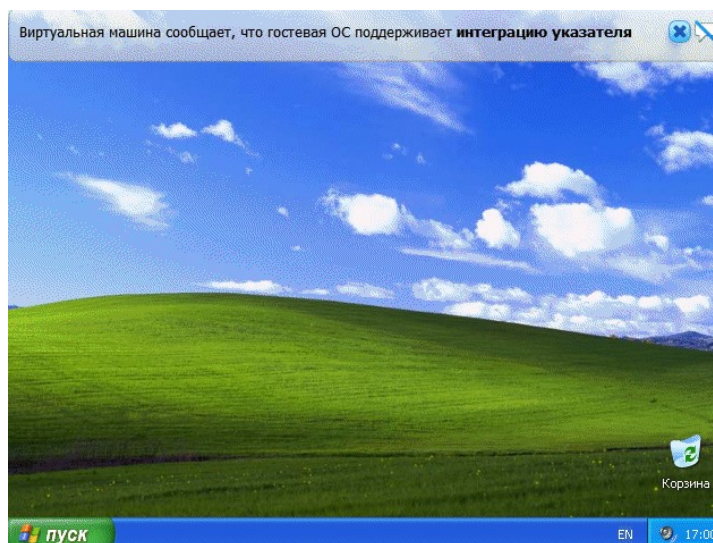


Рис. 2.2.28. Робочий стіл гостьової ОС

Далі представлено вікно гостьової ОС з працюючим додатком Блокнот в дрібному (рис. 2.2.29) та великому (рис. 2.2.30) масштабах.

Щоб включити створену віртуальну машину потрібно натиснути на кнопку Запустити, щоб припинити її роботу, закрити вікно гостьової ОС, обравши при цьому варіант завершення без збереження стану віртуальної машини чи зі збереженням.

Налаштування віртуальної машини здійснюється за допомогою пункту меню *Налаштувати VirtualBox*.

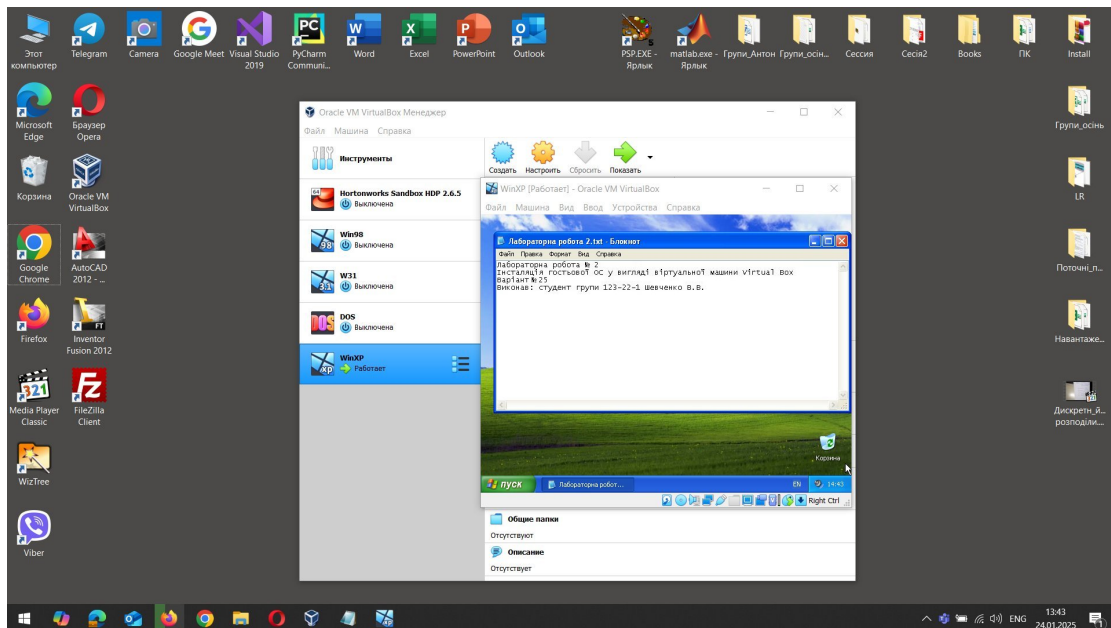


Рис. 2.2.29. Вікно гостьової ОС з працюючим додатком Блокнот в дрібному масштабі на фоні вікна додатка Virtual Box та робочого стола хостової ОС

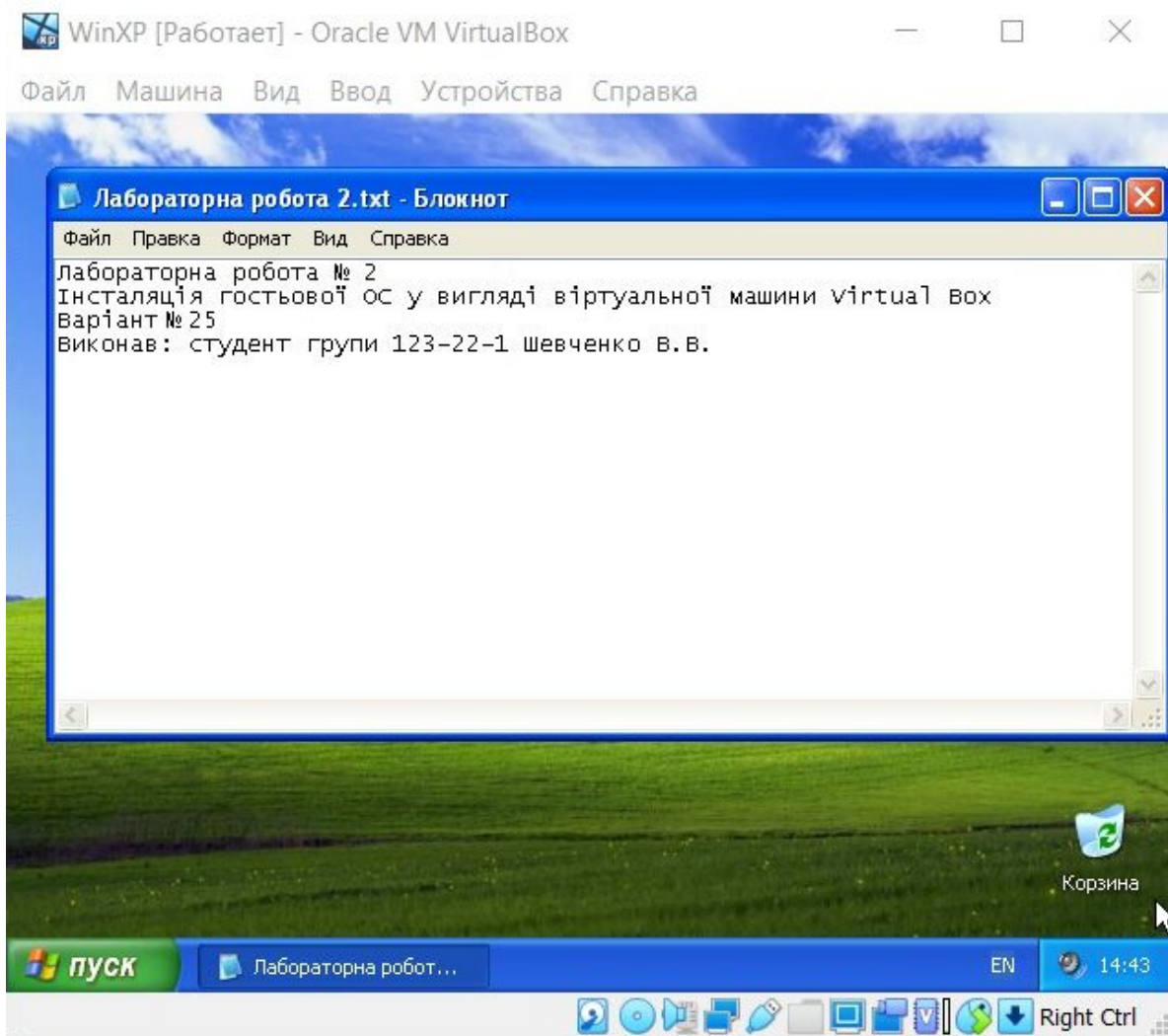


Рис. 2.2.30 Вікно гостьової ОС з працюючим додатком Блокнот в дрібному масштабі на фоні додатка Virtual Box та робочого стола хостової ОС

Основні налаштування віртуальної машини по розділах:

Загальна – ім'я машини. тип ОС. буфер обміну. змінні носії;

Система – основні налаштування апаратного забезпечення віртуальної машини: оперативна пам'ять, чіпсет. процесор, прискорення. порядок завантаження пристроїв;

Дисплей – налаштування відеопам'яті, кількості моніторів, 2D і 3D прискорення. підключення до віддаленого дисплею. захоплення відео;

Носії – контролери жорстких дисків і їх налаштування;

Аудіо – аудіодрайвер і контролер;

Мережа – налаштування мережевих адаптерів;

COM-порти – включення та налаштування COM-портів;

USB – включення контролера USB та налаштування USB-фільтрів;

Загальні папки – створення і управління загальними папками.

У разі завдання некоректних параметрів. програма проінформує про це. Нижче вікна налаштувань з'явиться напис "Виявлено неправильні налаштування".

Дано

Інсталятори Oracle VM Virtual Box, його гостьового доповнення та гостьової ОС.

Потрібно

Провести інсталяцію Oracle VM Virtual Box, його гостьових доповнень та гостьової ОС в середовищі хостової ОС.

Рекомендації щодо виконання роботи

1. Провести інсталяцію програми Virtual Box і його доповнення.
2. Провести інсталяцію гостьової ОС родини Windows згідно варіанту завдання.

3. Оформити копію звітного екрану. При цьому відкривається будь-який текстовий редактор, наприклад Блокнот і в ньому набирається наступна інформація: назва дисципліни та лабораторної роботи, прізвище та ініціали здобувача вищої освіти, код групи, номер варіанту завдання. На копії звітного екрану повинні бути видимі: робочий стіл комп'ютера здобувача вищої освіти, вікно VirtualBox, вікно гостьової ОС та вікно текстового редактору, з зазначеною вище інформацією, як наведено у зразку.

Вимоги до звіту

Звіт по роботі повинен містити:

1. Назву дисципліни та лабораторної роботи.
2. Прізвище, ім'я та по батькові здобувача вищої освіти, код групи.
3. Номер варіанту завдання.
4. Знімки зображень екрану, які виникають інсталяції і роботі використовуюваного ПЗ, з підписами (аналогічні приведені у прикладі Рис. 2.2.1 – 2.2.30).

Контрольні питання

1. Які основні функції ОС в комп'ютерній системі?

2. За якими показниками здійснюється класифікація ОС?
3. Які основні сімейства ОС персональних комп'ютерів та інших мобільних пристроїв можна виділити? Назвіть їх представників.
4. Що являє собою і для чого призначене ПЗ “віртуальна машина”?
5. Що є хостовою та гостьовою ОС?
6. Назвіть основні види віртуалізації.
7. Що таке гіпервізор? Які види віртуалізації його використовують?

Варіанти завдання

Поділяються за встановлюваною ОС.

Для здобувачів вищої освіти інституту природокористування з непарними номерами варіантів Windows 3.1, з парними – Windows 95, для здобувачів вищої освіти факультету інформаційних технологій в залежності від значення № mod 10 (залишку від цілочисельного ділення номера варіанту № на 10):

- 0 – Windows 98 (латинична);
- 1 – Windows 98 (кирилична);
- 2 – Windows NT 4 (латинична);
- 3 – Windows NT 4 (кирилична);
- 4 – Windows ME (латинична);
- 5 – Windows ME (кирилична);
- 6 – Windows 2000 (латинична);
- 7 – Windows 2000 (кирилична);
- 8 – Windows XP (латинична);
- 9 – Windows XP (кирилична).

2.3. Практична робота № 3

Організація взаємодії хостової та гостьової операційних систем, налаштування мережевих адаптерів у Oracle VM Virtual Box

Об'єкт – засоби віртуалізації. Предмет – засіб програмної віртуалізації Oracle VM Virtual Box. Мета – організація взаємодії хостової та гостьової операційних систем, налаштування мережевих адаптерів у Oracle VM Virtual Box.

Стислі теоретичні відомості

Організувати взаємодію хостової та гостьової ОС шляхом налаштування інтернету на віртуальній машині (ВМ) або створення локальної мережі (ЛМ) між кількома ВМ можна різними способами. Наприклад, у версії VirtualBox 6.1 для цього доступні такі інструменти:

- NAT (Network Address Translation) – цей спосіб використовується за замовчуванням. Для кожної машини створюється окрема внутрішня ЛМ, де машина отримує ір 10.10.0.1. Машина може зв'язатися з інтернетом, використовуючи технологію NAT, і можна звернутися до машини, використовуючи прокидання портів VirtualBox, але якщо є кілька ВМ, то між ними так взаємодіяти неможна. І якщо з хостової системи до гостьової можна звернутися, то до хостової з гостьової ні;

- Віртуальний адаптер хоста – при цьому створюється віртуальний мережний адаптер, до якого можна підключити кілька ВМ, об'єднавши їх у ЛМ. Доступ до інтернету не підтримується, але машини знаходяться в одній мережі і кожна має свою ір адресу, вони можуть взаємодіяти між собою. Хостова система теж доступна за IP 192.168.56.1. Машини доступні не лише між собою, а й з основної системи;

- Мережевий міст – при такому підключенні ВМ стає повноцінним членом ЛМ, до якої підключено хостову систему. Машина використовує мережний інтерфейс щоб отримати адресу у роутера і стає доступною для інших пристроїв, як і основний комп'ютер за своєю ір адресою.

- Внутрішня мережа – майже те саме, що й віртуальний адаптер хоста, тільки без можливості доступу до ЛМ з хостової системи, доступ до інтернету не підтримується.

- Універсальний драйвер – дозволяє використовувати драйвер із розширень VirtualBox для зв'язку між машинами, розташованими на різних фізичних хостах.

Гостьові доповнення Oracle VM VirtualBox

Для поліпшення роботи з віртуальною машиною та взаємодії з нею у гостьовій ОС потрібно встановити спеціальний набір інструментів, який зветься "Гостьові доповнення" ("Guest Additions"). Ці інструменти дозволяють підтримувати двосторонній буфер обміну, функцію drag-n-drop, організовувати спільні каталоги між хостовою та гостьовою ОС, здійснювати автоматичний вибір роздільної здатності екрану на основі розміру вікна VirtualBox тощо.

Підтримка цього набору інструментів ОС лінійки Windows приведена в табл. 2.3.1.

Таблиця 2.3.1. Підтримка гостьових доповнень Oracle VM VirtualBox операційними системами лінійки Windows

Операційна система	Статус	Примітка
Windows 11		Потрібна підтримка AMD-V або Intel VT-x.
Windows Server 2019		Потрібна підтримка AMD-V або Intel VT-x.
Windows Server 2016		Потрібна підтримка AMD-V або Intel VT-x.
Windows 10		Потрібна підтримка AMD-V або Intel VT-x.
Windows Server 2012 R2		Потрібна підтримка AMD-V або Intel VT-x.
Windows 8.1		Потрібна підтримка AMD-V або Intel VT-x.
Windows Server 2012		Потрібна підтримка AMD-V або Intel VT-x.
Windows 8		Потрібна підтримка AMD-V або Intel VT-x.
Windows Server 2008 R2		Підтримується
Windows 7		Підтримується
Windows Server 2008		Підтримується
Windows Vista		Підтримується

Операційна система	Статус	Примітка
Windows Server 2003 R2		Підтримується
Windows Server 2003		Підтримується
Windows XP		Підтримується
Windows 2000		Підтримується
Windows ME		Не підтримується
Windows 98		Не підтримується
Windows NT 4.0		Потрібен установлений Service Pack 6a
Windows 95		Не підтримується

Налаштування гостьових доповнень для віртуальної машини

В головному меню працюючої ВМ обираємо пункт *Пристрої* – *Підключити образ диска Доповнень гостьової ОС...*

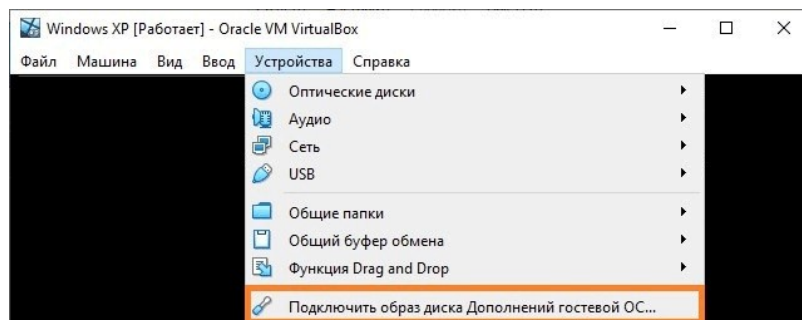


Рис. 2.3.1. Налаштування гостьових доповнень (початок)

Якщо в системі не вимкнено автозапуск, то має запуснитись установник доповнень гостьової ОС. Якщо ж у системі вимкнено автозапуск, потрібно запуснути установник вручну, знайшовши його на диску, що підключився у віртуальній машині, під ім'ям VBoxWindowsAdditions.exe.

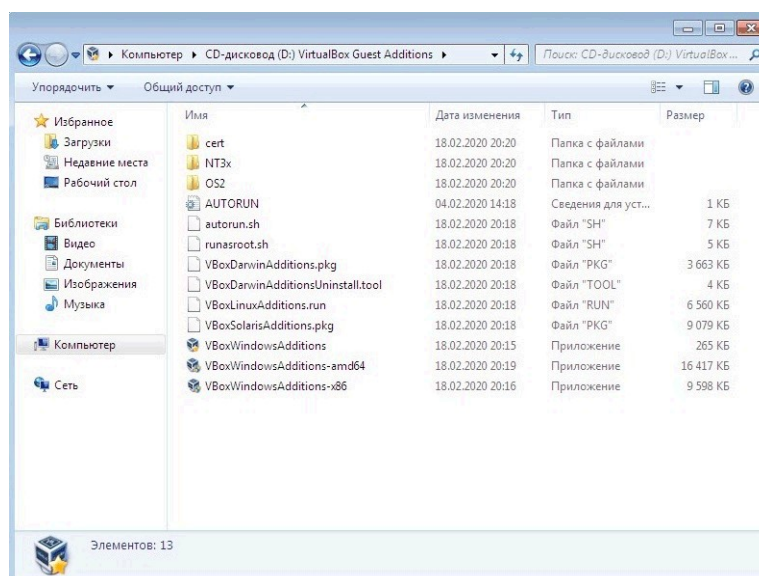


Рис. 2.3.2. Налаштування гостьових доповнень (продовження)

Налаштування гостьових доповнень максимально спрощено і по суті є натисканням кнопок *Next*. З першим вікном саме так і слід вчинити.



Рис. 2.3.3. Налаштування гостьових доповнень (продовження)

На наступному кроці обирається місце для встановлення гостьових додатків. За замовчуванням використовується каталог Program Files.

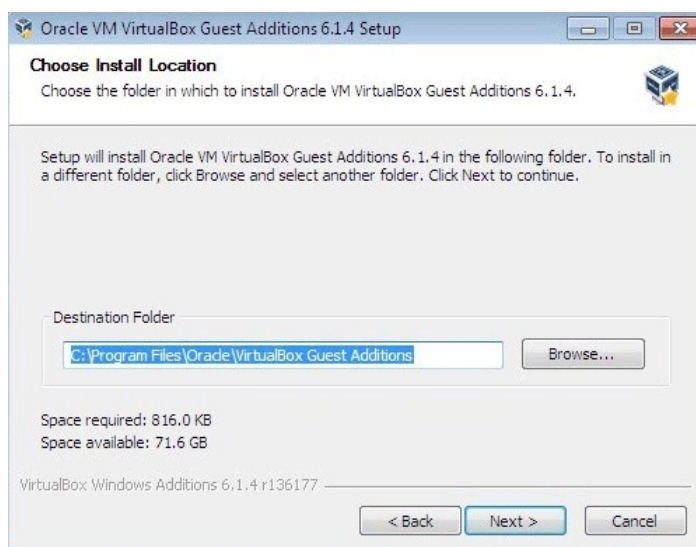


Рис. 2.3.4. Налаштування гостьових доповнень (продовження)

Останнім кроком, перед власною установкою, буде вибір встановлюваних компонентів:

- VirtualBox Guest Additions – власно гостьові доповнення VirtualBox;
- Start menu entries – ярлики в меню *Пуск*.
- Direct3D Support – третій компонент, який не вибрано за замовчуванням, є експериментальною функцією включення повноцінного 3D прискорення у віртуальній машині.

Після натискання на кнопку *Next* почнеться встановлення гостьових доповнень, під час якої може з'явитися вікно із запитом на дозволи інсталяції програмного забезпечення для пристрою (драйвер), яку необхідно схвалити.

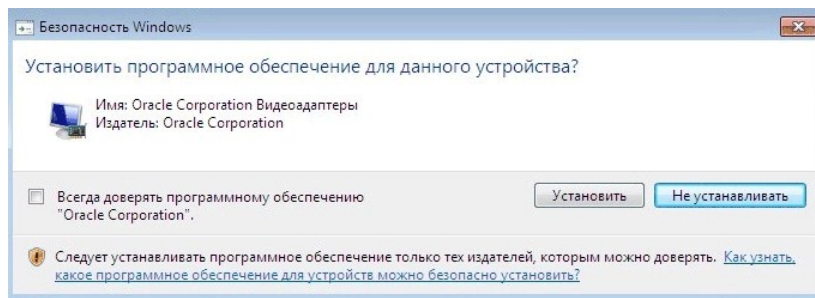


Рис. 2.3.5. Налаштування гостьових доповнень (продовження)

Після встановлення доповнень буде потрібно перезавантаження комп'ютера, яке можна виконати відразу (залишивши зазначеним пункт *Reboot now*), або виконавши пізніше самостійно (для цього потрібно відзначити пункт *I want to manually reboot later*).



Рис. 2.3.6. Налаштування гостьових доповнень (закінчення)

Буфер обміну і Drag and Drop

Буфер обміну призначений для обміну даними між основною та гостьовою ОС через спільну пам'ять, функція Drag and Drop для переміщення файлів між вікнами основної та гостьової ОС за допомогою миші. Режим роботи загального буферу обміну і Drag and Drop можуть приймати чотири значення: «вимкнено», «тільки з гостьової ОС в основну», «тільки з основної ОС в гостьову», «двонаправлений», які визначають, як буде працювати буфер обміну та Drag and Drop між хостовою та гостьовою ОС.

Буфер обміну та функція Drag and Drop можуть працювати по-різному на різних ОС (залежить від хостової та гостьової ОС). Одні ОС здатні обмінюватися тільки текстовими чи графічними даними (копіювання і вставка із текстового чи графічного редактора, знімки екранів), інші можуть копіювати і передавати об'єкти файлової системи. Для пари хостова Windows 10 та гостьова Windows 2008 буфер обміну працює в обох напрямках, функція Drag and Drop працює тільки в напрямку з хостової до гостьової ОС, через буфер обміну можна обмінюватися тільки текстовими або графічними даними, але не об'єктами файлової системи.

Робота з буфером обміну (clipboard) буде доступна тільки в тому випадку, якщо встановлено гостьові доповнення VirtualBox у гостьовій ОС.

Доступ до буферу обміну та функції Drag and Drop здійснюється за допомогою пунктів меню *ВМ Пристрої – Загальний буфер обміну – Режим роботи* та *Пристрої – Функція Drag and Drop – Режим роботи*.

На рис. 2.3.6, 2.3.7 показані хід та результат виконання функції Drag and Drop між вікнами хостової та гостьової ОС.

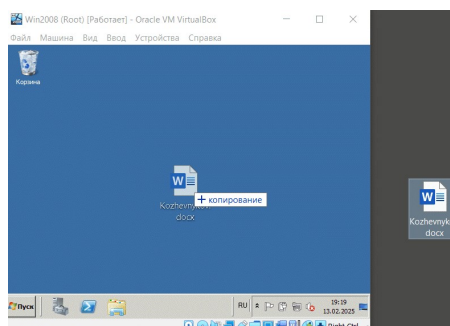


Рис. 2.3.7. Хід виконання функції Drag and Drop

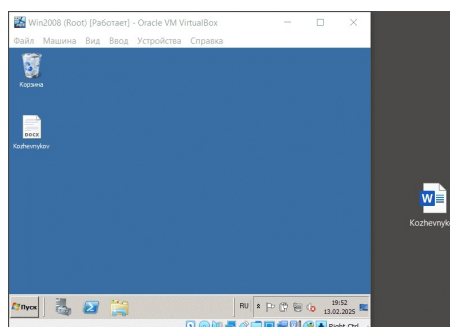


Рис. 2.3.8. Результат виконання функції Drag and Drop

Спільні каталоги

Цей механізм дозволяє створювати каталоги, які доступні як з хостової, так і гостьової ОС.

Створення спільного каталогу здійснюється за допомогою пунктів меню *ВМ Пристрої – Налаштувати загальні папки*. У відкритому вікні натиснути кнопку *Додати нову загальну папку*, яка розташована з правого боку, вказати папку у файловій системі хостової ОС, обрати автоматизування, перезавантажити ВМ. Після цього загальний каталог з'являється у вікні *Комп'ютер* як іконка мережевого диску (рис. 2.3.9).

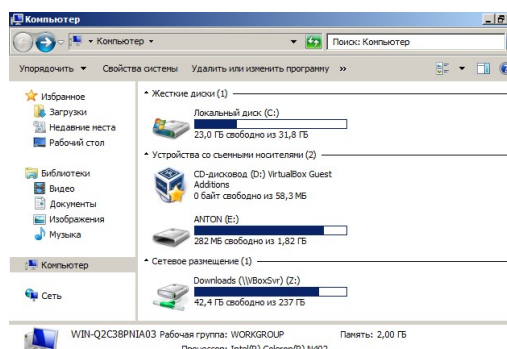


Рис. 2.3.9. Загальний каталог хостової та гостьової ОС.

Монтування USB накопичувача в гостьовій ОС

Здійснюється за допомогою пунктів меню ВМ Пристрої – USB. У списку, що відкривається, потрібно обрати ім'я USB накопичувача. Після цього його іконка з'являється у вікні *Комп'ютер* гостьовій ОС (рис. 9) і одночасно він стає недоступним для хостової ОС. Для коректного розмотування накопичувача слід скористатися пунктом *Вилучити* його контекстного меню.

Обслуговування ВМ

Експорт та імпорт. Якщо віртуальну машину VirtualBox потрібно перенести на інший хост, необхідно провести операцію експорту, а потім імпорту. Ці операції здійснюються із використанням стандарту Open Virtualization Format (OVF). При експорті конфігураційний XML-файл перетворюється у файл із розширенням OVF(OVA).

Збереження стану (Save State) – одна з корисних функцій системи віртуалізації. При збереженні стану весь вміст пам'яті віртуальної машини, регістрів процесора і т.д. зберігається у спеціальні файли, і ВМ переходить у стан "Вимкнено". Після цього можна робити все, що завгодно, аж до перезавантаження хостової машини, а потім знову запустити ВМ – і вона буде працювати в тому ж стані, у якому вона була до збереження.

Знімки ВМ. Містять інформацію про стан власної гостьової ОС і встановлене в ній користувацьке програмне забезпечення і дозволяють швидко перемикаєти ВМ в стани, які описуються цими знімками (снэпшотами). Розрізняють два типи снэпшотів: онлайновий і офлайновий. Онлайновим називають снэпшот, зроблений на віртуальній машині із запущеною гостьовою ОС. Відповідно, якщо віртуальна машина була в стані «вимкнено» – то снэпшот називатиметься офлайновим. Розрізняються вони за складом файлів, тому що при створенні снэпшотів на запущеній ВМ відбувається операція Save State, і дані Save State включаються до складу снэпшотів.

Доступ до інструментів управління знімками здійснюється за допомогою пунктів головного меню VirtualBox *Машина – Інструменти – Знімки* після виконання яких до заголовку вікна VirtualBox додаються кнопки *Зробити*, *Вилучити*, *Відновити*, *Властивості*.

Розглянемо приклад створення дерева снэпшотів в якому листи – стани ВМ будуть розрізнятися встановленими у гостьовій ОС браузерами. ЛМ ВМ функціонує у режимі NAT. Доступ до інсталяторів браузерів здійснюється через спільний каталог гостьової та хостової ОС. Початковий стан ВМ до встановлення браузерів фіксується офлайновим знімком, якому привласнюється ім'я Root. Знімок робиться при вимкнутій ВМ шляхом натиснення кнопки *Зробити*. Далі вмикається ВМ. Для входу в систему замість натиснення комбінації кнопок *Ctrl-Alt-Del* використовується пункт меню ВМ *Введення – Клавіатура – Послати Ctrl-Alt-Del*. В гостьовій ОС встановлюється браузер Mozilla і робиться знімок з відповідним іменем. ВМ вимикається, стан Root встановлюється поточним і натискається кнопка *Відновити*. При цьому відкривається вікно з попередженням про можливі втрати стану. Знімається помітка в чекбоксі «Створити знімок поточного стану машини», оскільки він вже зроблений, і натискається кнопка *Відновити*. При цьому ВМ повертається в

початковий стан. Вмикається ВМ, встановлюється браузер Opera і робиться знімок з відповідним іменем, встановлюється браузер Chrome і робиться знімок з іменем Opera+Chrome. При цьому дерево знімків набуває вигляд представлений на рис. 10 і містить 4 стани ВМ між якими можна перемикатися шляхом відновлення потрібного стану (рис. 2.3.10).

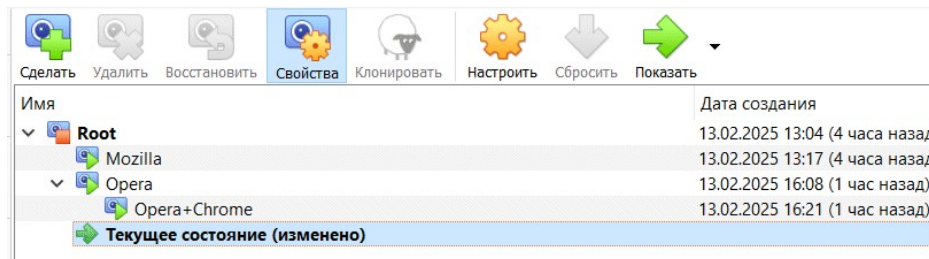


Рис. 2.3.10. Перемикач станів віртуальної машини

Дано

Засіб віртуалізації Oracle VM Virtual Box зі встановленими гостьовими доповненнями та віртуальною машиною гостьової ОС.

Потрібно

Налаштувати гостьові доповнення та віртуальної машини гостьової ОС, організувати засоби взаємодії хостової та гостьової ОС: спільний каталог, USB-інтерфейс гостьової ОС, перевірити виконання функції Drag and Drop. Ознайомитися з інструментом управління станами віртуальної машини.

Рекомендації щодо виконання роботи

1. Провести інсталяцію гостьової ОС згідно варіанту завдання.
2. Провести інсталяцію гостьових доповнень.
3. Створити на робочому столі хостової ОС текстовий файл з ім'ям у вигляді власного прізвища. Внести в нього назву лабораторної роботи. Переконавшись, що буфер обміну та функція Drag and Drop працюють у всіх можливих режимах.
4. Створити спільну папку та змонтувати USB накопичувач. Останній повинен мати ім'я, яке збігається з прізвищем здобувача вищої освіти.
5. Перевірити, що ЛМ ВМ функціонує у режимі NAT, а Internet Explorer гостьової ОС дозволяє переглядати зміст сайтів. Провести інсталяцію браузерів в гостьовій ОС, створюючи при цьому дерево снапшотів ВМ з топологією згідно варіанту завдання.

Вимоги до звіту

Звіт по роботі повинен містити:

1. Назву дисципліни та лабораторної роботи.
2. Прізвище, ім'я та по батькові здобувача вищої освіти, код групи.
3. Номер варіанту завдання.
4. Знімки зображень екрану, які виникають інсталяції і роботі використовуваного ПЗ, з підписами (аналогічні приведеним у прикладі Рис. 1 – 10).

Контрольні питання

1. Які додаткові можливості надають гостьові доповнення?
2. Які режими роботи ЛМ ВМ доступні VirtualBox?

3. Для чого потрібні знімки стану ВМ?

Варіанти завдання

Поділяються за встановлюваною ОС. Непарні номери Windows 2008 (латинична), парні Windows 2008 (кирилична).

Поділяються за встановлюваними браузером та топологією дерева снапшотів ВМ. в залежності від значення № mod 9 (залишка від цілочисельного ділення номера варіанту № на 9):

0 – root – Google Chrome, root – Mozilla Firefox;

1 – root – Google Chrome, root – Opera;

2 – root – Opera, root – Mozilla Firefox;

3 – root – Google Chrome – Mozilla Firefox;

4 – root – Google Chrome – Opera;

5 – root – Mozilla Firefox – Google Chrome;

6 – root – Mozilla Firefox – Opera;

7 – root – Opera – Google Chrome;

8 – root – Opera – Mozilla Firefox.

2.4. Практична робота № 4

Ознайомлення з методами класифікації даних за допомогою нейронних мереж

Об'єкт – нейронні мережі. Предмет – нейронні мережі перцептронів, які призначені для класифікації даних. Мета – отримання навичок програмування найпростіших нейронних мереж для класифікації даних.

Стислі теоретичні відомості

Принцип роботи штучної нейронної мережі відповідає алгоритму роботи біологічних нейронів. Нейрон – обчислювальна одиниця, яка отримує інформацію і виробляє над цією інформацією будь-які обчислення. Нейрони є найпростішою структурною одиницею будь-якої нейронної мережі. Їх множини, упорядковані певним складають в шари, які, в свою чергу поєднуються в мережу в цілому. Всі нейрони функціонують приблизно однаково, однак бувають деякі окремі випадки нейронів, що виконують специфічні функції. На рис. 2.4.1 представлений приклад топології нейронної мережі.

Розрізняють три основних типи нейронів: вхідні – нейрони шару, який одержує інформацію з зовні (синій колір), приховані – нейрони кількох шарів, що обробляють інформацію (червоний колір), вихідні – нейрони шару, що представляє результати обчислень (зелений колір на рис. 1).

Крім самих нейронів складовою частиною мережі є синапси. Синапс – зв'язок, що з'єднує вихід одного нейрона з входом іншого. Під час проходження сигналу через синапс сигнал може посилюватися або слабшати. Параметром синапсу є вага (деякий коефіцієнт, може бути будь-яким дійсним числом), через який інформація, що передається від одного нейрона до іншого, може змінюватися. За допомогою зважування вхідна інформація проходить обробку, після чого отримується результат.

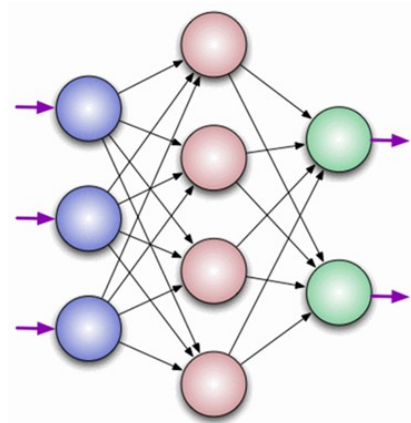


Рис. 2.4.1. Приклад топології нейронної мережі

На рис. 2.4.2 зображена графічна схема моделі штучного нейрона. На вхід подаються числа (сигнали) x_i , після вони множаться на ваги w_i (кожен сигнал на свою вагу) і підсумовуються. Функція активації вираховує остаточний сигнал y і подає його на вихід. Функція активації дозволяє проходити або не проходити сигналам від одного нейрону до іншого, або на вихід мережі в залежності від інформації, яку вони передають. Тобто якщо інформація є важливою, то функція пропускає її, а якщо інформації мало або вона недостовірна, то функція активації не дозволяє їй пройти далі.

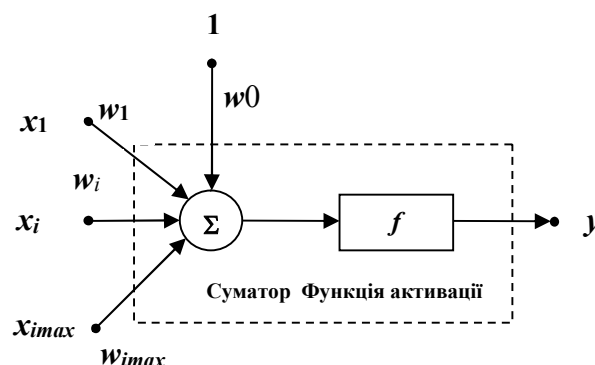


Рис. 2.4.2. Графічна схема моделі штучного нейрона.

У табл. 2.4.1 представлені деякі види активаційних функцій нейрона. Функція активації повинна бути однаковою для всіх нейронів всередині одного шару, однак для різних шарів можуть вибиратися різні функції активації. У штучних нейронних мережах вибір функції обумовлюється областю використання і є важливим завданням.

Таблиця 2.4.1. Найбільш поширені функції активації штучного нейрона

Назва	Математичне подання
Лінійна	ax
Порогова (одинична, функція Хевісайда)	$\begin{cases} 0 & \text{при } x < x_0 \\ 1 & \text{при } x \geq x_0 \end{cases}$

Функція з лінійним порогом	$\begin{cases} 0 & \text{при } x < 0 \\ \alpha x & \text{при } 0 \leq x \leq \frac{1}{\alpha} \\ 1 & \text{при } x > \frac{1}{\alpha} \end{cases}$
ReLU	$\begin{cases} 0 & \text{при } x < 0 \\ x & \text{при } x \geq 0 \end{cases}$
Leaky ReLU	$\begin{cases} \alpha x & \text{при } x < 0, \quad 0 < \alpha < 1 \\ x & \text{при } x \geq 0 \end{cases}$
Сигмоїдна	$\frac{1}{1 + \exp(-x)}$
Гіперболічний тангенс	$\frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$

Застосування нейронної мережі передбачає її попереднє навчання за даними навчальної вибірки. Схематично процес навчання представлений на рис.2.4.3.

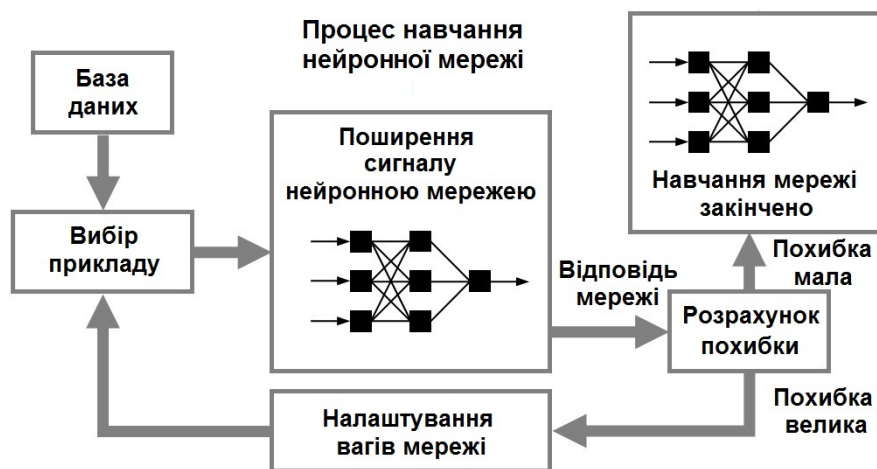


Рис. 2.4.3. Схема процесу навчання нейронної мережі

Нейронні мережі набувають все більшої популярності і область їх використання також розширюється. Список деяких областей, де застосовуються штучні нейронні мережі:

Введення і обробка інформації. Розпізнавання текстів, що подані у вигляді растрових графічних даних, розпізнавання голосових команд, голосове введення тексту.

Безпека. Розпізнавання облич і різних біометричних даних, аналіз трафіку в мережі, виявлення підробок.

Інтернет. Таргетингова реклама, капча, блокування спаму, асоціативний пошук інформації.

Зв'язок. Маршрутизація пакетів, стиснення відеоінформації, збільшення швидкості кодування і декодування інформації.

Персептрон є найпростішою моделлю нейромережі, що може складатися з одного нейрона (рис.2.4.2). Нейрон може мати довільну кількість входів, а вхідний сигнал одного з них зазвичай тотожно дорівнює 1. Цей одиничний вхід називають зсувом. Така проста мережа вирішувати прості завдання: виконувати класифікацію лінійно роздільних даних, інтерполювати часові ряди, що мають лінійний функціональний тренд. Для лінійно роздільних даних завжди можна провести пряму (або гіперплощину в багатовимірному просторі), що відокремлюють дані одного класу від даних іншого класу. Це і звуть лінійною роздільності об'єктів. Тоді модель мережі, що розглядається, зможе вирішити поставлене завдання класифікації. Якщо ж об'єкти лінійно нероздільні, то мережа з одного нейрона з ними не впорається. Для цього існують більш складні архітектури нейронних мереж що складаються з множини персептронів, об'єднаних в шари.

Розглянемо приклад застосування персептрону для класифікації лінійно роздільних даних. Дані представляють собою крапки числової площини з координатами (x_1, x_2) , які рівномірно розподілені в квадраті $x_1, x_2 \in (0,1)$. Навчальна вибірка окрім матриці вказаних вхідних даних сигналів містить вектор вихідних сигналів, які формуються за наступним правилом

$$Y_k = \begin{cases} 0 & \text{при } X_{k2} < aX_{k1} + b \\ 1 & \text{при } X_{k2} \geq aX_{k1} + b \end{cases}, \quad a, b \in (0,1), \quad (2.4.1)$$

тобто, якщо крапка знаходиться нижче прямої $x_2 = ax_1 + b$, вона відноситься до класу 0, інакше – до класу 1.

Далі будуть прийняті такі позначення $\{w_0^n\}$, $\{w_i^n\}_{i=1, \dots, imax}$ – ваги синапсів на n -му кроці навчання, $imax=2$ – кількість входів персептрону без входу зсуву, $\{X_{ki}\}_{k=1, \dots, kmax, i=1, \dots, imax}$ – навчальна вибірка вхідних сигналів, $kmax$ – об'єм вибірки, який може бути довільним, $\{Y_k\}_{k=1, \dots, kmax}$ – навчальна вибірка вихідних сигналів. При навчанні мережі здійснюється її тренування на кожній варіанті навчальної вибірки впродовж $nmax$ циклів. Фактичний вихідний сигнал персептрону на n -му циклі навчання при подачі на вхід вхідних сигналів k -ї варіанти навчальної вибірки при цьому складає

$$y_k^n = f\left(w_0^n + \sum_{i=1}^{imax} w_i^n X_{ki}\right). \quad (2.4.2)$$

Функція активації f при цьому є пороговою, а вихідний сигнал – бінарним.

Похибка мережі на n -му циклі навчання визначається як сума за всіма варіантами навчальної вибірки абсолютних значень різниць між очікуваним та фактичним значенням вихідного сигналу мережі

$$\varepsilon^n = \sum_{k=1}^{kmax} |Y_k - y_k^n|. \quad (2.4.3)$$

При цьому умовою припинення навчання може бути як досягнення необхідного значення похибки, так і просте обмеження кількості ітерацій певною величиною $nmax$. Правила налаштування ваг синапсів залежить від архітектури мережі. Для персептрона може бути використане співвідношення

$$\begin{cases} w_0^{n+1} = w_0^n + \sum_{k=1}^{kmax} \eta (Y_k - y_k^n) \\ w_i^{n+1} = w_i^n + \sum_{k=1}^{kmax} \eta (Y_k - y_k^n) X_{ki} \text{ при } 1 \leq i \leq imax \end{cases}, \quad (2.4.4)$$

де $\eta \in (0,1)$ – швидкість навчання. У якості початкових значень ваг синапсів w_i^0 беруться випадкові величини, які рівномірно розподілені на сегменті $(-0.1,0.1)$. Наведене вище правило налаштування ваг синапсів носить назву дельта-правила і є узагальненням правил Хебба. Останні були сформульовані для персептрона, який працює з бінарними вхідними і вихідними сигналами і мають вигляд:

- якщо сигнал персептрона невірний, і дорівнює нулю, то необхідно збільшити ваги тих входів, на які була подана одиниця;
- якщо сигнал персептрона невірний і дорівнює одиниці, то необхідно зменшити вагу тих входів, на які була подана одиниця.

Дійсно, якщо вихідний сигнал бінарного класифікатора невірний, і дорівнює нулю $y_k^n = 0$, $Y_k - y_k^n = 1$ і, з урахуванням невід’ємності величини X_{ki} , застосування дельта-правила веде до збільшення ваг всіх синапсів, що викликає виправлення значення вихідного сигналу на наступному циклі навчання, $y_k^{n+1} = 1$, або принаймні збільшення сигналу на виході суматора, який є аргументом функції активації. Навпаки, якщо вихідний сигнал бінарного класифікатора невірний, і дорівнює одиниці $y_k^n = 1$, $Y_k - y_k^n = -1$, застосування дельта-правила веде до зменшення ваг всіх синапсів, що викликає виправлення значення вихідного сигналу на наступному циклі навчання, $y_k^{n+1} = 0$, або принаймні збільшення сигналу на виході суматора.

Вихідний сигнал навченої мережі, який виникає при поданні на її входи сигналу $\{x_i\}_{i=1,...,imax}$ становить

$$y^{nmax} = f\left(w_0 + \sum_{i=1}^{imax} w_i^{nmax} x_i\right). \quad (2.4.5)$$

Зауважимо, що, оскільки умовою віднесення крапки, яка класифікується,

до класу 1 є $x_2 \geq ax_1 + b$, а умовою того, що вихідний сигнал мережі дорівнюватиме 1 – $w_0^{nmax} + w_1^{nmax}x_1 + w_2^{nmax}x_2 \geq 0$, для синапсів навченої мережі повинні виконуватися умови

$$a \approx -\frac{w_1^{nmax}}{w_2^{nmax}}, \quad b \approx -\frac{w_0^{nmax}}{w_2^{nmax}}. \quad (2.4.6)$$

Програмна реалізація нейронної мережі бінарного класифікатора засобами системи MathCAD включає підготовчий фрагмент у якому задаються параметри мережі, формуються навчальна та тестова вибірки, визначаються початкові ваги синапсів, а також програмні блоки навчання та тестування мережі. Вхідними даними підготовчого блока відповідно виступають $a, b, \eta, kmax, mmax, nmax$. На його виході формуються значення $\{X_{ki}\}_{k=1,...,kmax, i=1,...,imax}$,

$\{Y_k\}_{k=1,...,kmax}, \{X_{tmi}\}_{m=1,...,mmax, i=1,...,imax}, \{Y_{tm}\}_{m=1,...,mmax}, \{w_0^0\}, \{w_i^0\}_{i=1,...,imax}$. На виході

програмного блока навчання отримуються значення $\{w_0^{nmax}\}, \{w_i^{nmax}\}_{i=1,...,imax}$ та ε^{nmax} , а на виході програмного блока тестування – значення $\{y_m^{nmax}\}_{m=1,...,mmax}$, які відповідають елементам тестової вибірки.

Мережа швидко навчається, для отримання нульової похибки потрібно близько десяти циклів навчання, при цьому ваги синапсів збігаються до значень, які задовольняють співвідношенням (2.4.6), що свідчить про правильність функціонування мережі (рис. 2.4.4).

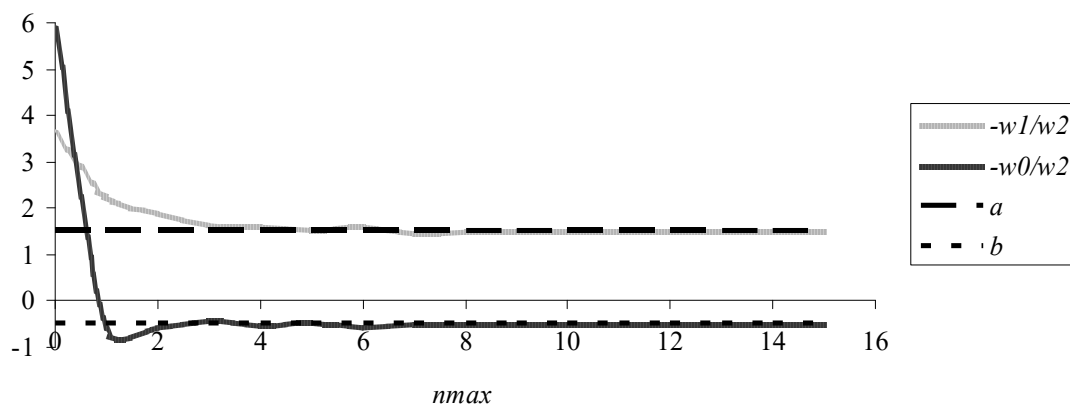


Рис. 2.4.4. Залежності ваг синапсів нейронної мережі від кількості циклів навчання

Багатошаровий персептрон є узагальненням одношарового персептрона і складається з вхідного шару, одного чи декількох прихованих шарів обчислювальних нейронів і вихідного шару. Топологія персептрона з одним прихованим шаром представлена на рис. 2.4.1.

Багатошаровий персептрон також зветься нейронною мережею прямого поширення, для нього використовується аббревіатура MLP (Multiple Layer Perceptron).

Якщо записати вихідний сигнал $\vec{y}$ навченої мережі, яка складається з одного шару, що містить довільну кількість нейронів у векторному вигляді

$$\vec{y} = f(\mathbf{w}\vec{x} + \vec{w}0), \quad (2.4.7)$$

для вихідного сигналу $\vec{y}_n$ мережі, яка складається з довільної кількості n таких шарів, можна отримати рекурентне співвідношення

$$\vec{y}_n = f_n(\mathbf{w}_n\vec{y}_{n-1} + \vec{w}0_n), \quad (2.4.8)$$

де f_n – функція активації n -го (вихідного) шару, $\vec{y}_{n-1}$ – вихідний сигнал останнього прихованого шару (вхідний сигнал вихідного шару), $\mathbf{w}_n$, $\vec{w}0_n$ – матриця ваг синапсів інформаційних входів і вектор ваг синапсів входів зсувів n -го (вихідного) шару.

У додатку наведений програмний код мовою Python, який:

- здійснює генерацію набору даних (крапок на площині), що характеризується двома ознаками та може бути розбитий на два класи, при цьому ознаками є координати крапок, класи представляють собою множини крапок у вигляді двох напівмісяців з розмитими межами;
- будує і навчає нейронну мережу у вигляді багатошарового персептрона з одним прихованим шаром;
- візуалізує межі розділу класів, що побудовані на основі навчальної вибірки і крапок даних навчальної (або контрольної) вибірки.

На рис. 2.4.5 а) представлений результат роботи програми з візуалізацією межі розділу класів, що побудовані на основі навчальної вибірки і крапок даних навчальної вибірки, а на рис. 3.4.5 б) – крапок даних контрольної вибірки.

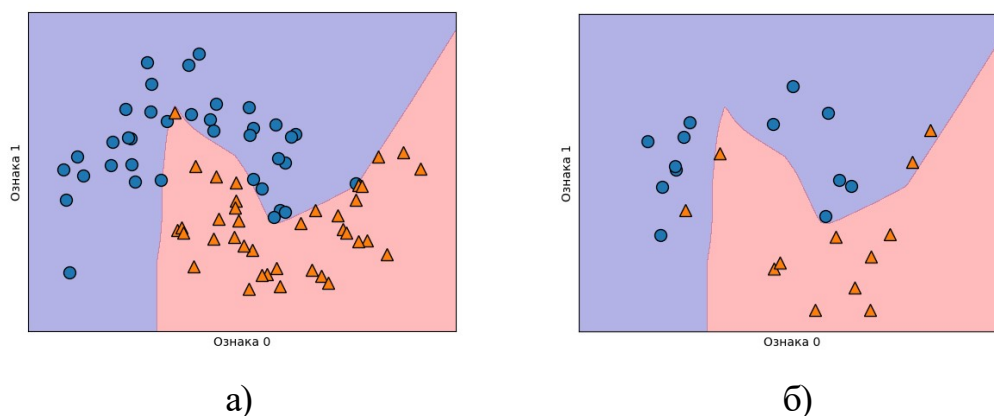


Рис. 2.4.5. Крапки вхідних даних, що класифікуються, межа розділу класів, яку будує нейромережа з кількістю нейронів прихованого шару 100 і функцією активації ReLU

З них можна бачити, що по-перше межа розділу класів досить якісно дискримінує дані різних класів для обох вибірок, а по-друге, що нейронна мережа будує нелінійну і досить гладку межу розділу. Насправді межа є кусково лінійною і гладкість пояснюється значною кількістю прихованих вузлів мережі (нейронів), оскільки за замовчуванням MLP використовує 100 прихованих вузлів. При зменшенні їх числа їх числа до 10-ти якість дискримінації суттєво не погіршується, але проявляється кускова лінійність межі, оскільки тепер вона складається з 10-ти прямолінійних відрізків (рис. 2.4.6 а, б).

Зміни в програмному коді:

```
mlp = MLPClassifier(solver='lbfgs', max_iter=100000, random_state=0,
hidden_layer_sizes=[10])
```

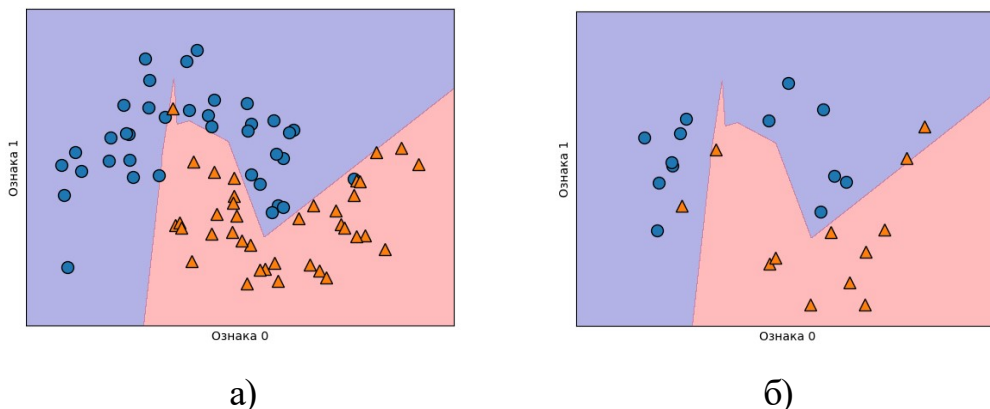


Рис. 2.4.6. Крпки вхідних даних, що класифікуються, межа розділу класів, яку будує нейромережа з кількістю нейронів прихованого шару 10 і функцією активації ReLU

Якщо необхідно отримати більш гладку межу розділу класів, замість збільшення кількості прихованих вузлів, можна додати другий прихований шар (рис. 2.4.7 а, б).

Зміни в програмному коді:

```
mlp = MLPClassifier(solver='lbfgs', max_iter=100000, random_state=0,
hidden_layer_sizes=[10, 10])
```

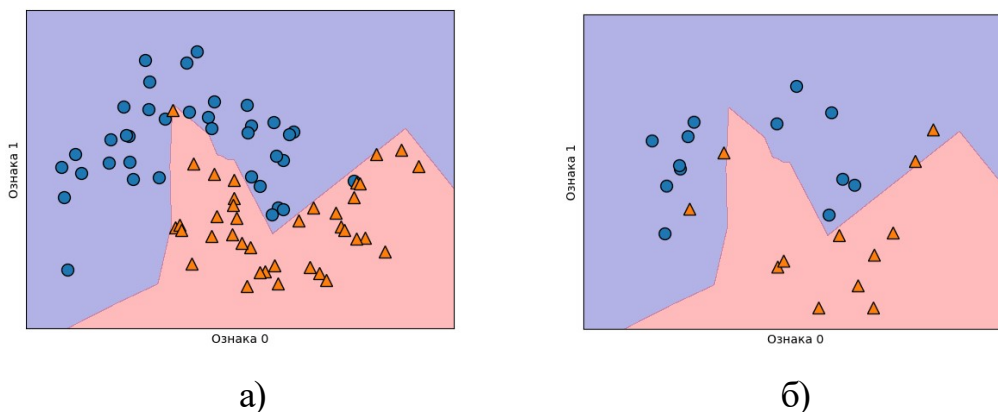


Рис. 2.4.6. Крпки вхідних даних, що класифікуються, межа розділу класів, яку будує нейромережа з двома прихованими шарами з кількістю нейронів 10,10 і функцією активації ReLU

У якості альтернативного варіанту можна використовувати більш гладку функцію активації, наприклад гіперболічний тангенс (рис. 2.4.8 а, б).

Зміни в програмному коді:

```
mlp = MLPClassifier(solver='lbfgs',max_iter=100000, activation='tanh',
random_state=0,hidden_layer_sizes=[10])
mlp.fit(X_train, y_train)
```

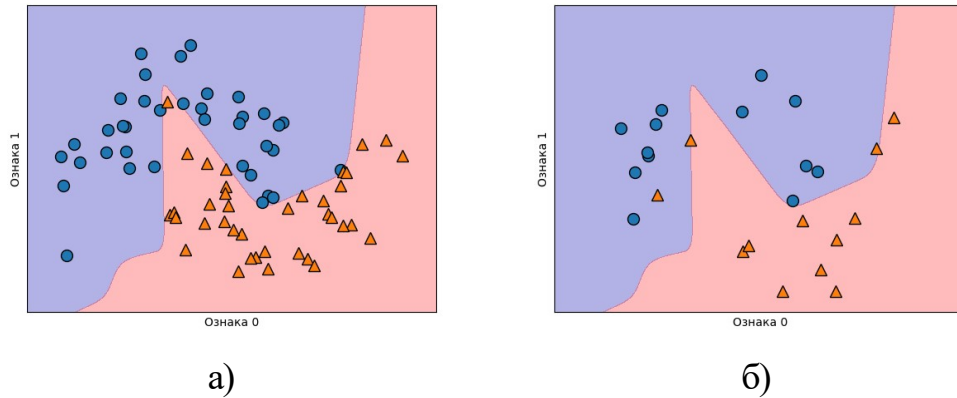


Рис. 2.4.8. Крпки вхідних даних, що класифікуються, межа розділу класів, яку будує неймережа з кількістю нейронів прихованого шару 10 і функцією активації гіперболічний тангенс

Одним із завдань практичної роботи є створення нейронної мережі багат шарового персептрону, яка реалізує бінарну класифікацію випадкових даних. При цьому імовірнісний розподіл даних дещо відрізняється від розглянутого вище. Дані представляють собою крпки площини, які рівномірно розподілені у прямокутній області $[x_{\min}; x_{\max}] \cdot [y_{\min}; y_{\max}]$, в яку вписаний графік функції $y=f(x)$. Межі незалежної змінної $[x_{\min}; x_{\max}]$ та вид функції залежать від варіанту завдання. Належність крпки до певного класу (0 або 1) визначається її розташуванням нижче, або вище графіку функції.

Для розіграшу координат точок, які рівномірно розподілені у прямокутній області, може бути використана відома формула

$$x = x_{\min} + (x_{\max} - x_{\min}) \cdot \xi_1; \quad y = y_{\min} + (y_{\max} - y_{\min}) \cdot \xi_2, \quad (2.4.9)$$

де ξ_1, ξ_2 – незалежні псевдовипадкові числа, які рівномірно розташовані на сегменті $[0,1]$.

Для розіграшу координат точок, які рівномірно розподілені у круговій області радіусом R з центром в точці (x_0, y_0) використовується співвідношення

$$x = x_0 + \sqrt{\xi_1} \cdot R \cdot \cos(2 \cdot \pi \cdot \xi_2); \quad y = y_0 + \sqrt{\xi_2} \cdot R \cdot \sin(2 \cdot \pi \cdot \xi_2). \quad (2.4.10)$$

Рис. 2.4.9 ілюструє помилкову класифікацію об'єктів за синусоїдною функцією, що визначає істинні межі класів. Невірно класифіковані об'єкти показані зеленим кольором.

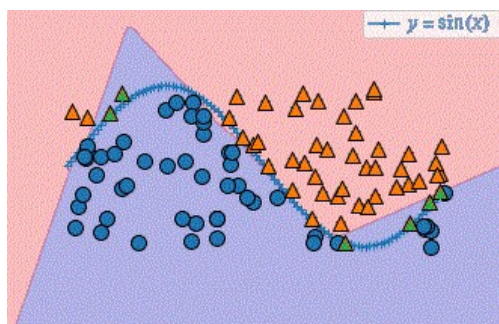


Рис. 2.4.9. Крпки вхідних даних, що класифікуються, межа розділу класів, яку будує неймережа з кількістю нейронів прихованого шару 2 і функцією активації ReLU та графік дискримінуючої функції $y=\sin(x)$ з виділенням помилкових класифікацій, які показані зеленим кольором.

Дано

Середовище програмування мовою Python, дискримінуюча функція, яка визначає приналежність до певного класу даних, які утворюються та межі варіації її незалежної змінної.

Потрібно

Створити Python-програму, що реалізує нейронну мережу у вигляді трьохшарового перцептрону, який призначений для бінарної класифікації даних згідно варіанту завдання.

Рекомендації щодо виконання роботи

1. Передбачається, що здобувач вищої освіти має засоби компіляції мови Python, які при відсутності треба інсталиювати.

2. Згідно варіанту завдання запрограмувати нейронну мережу багатошарового перцептрону, яка реалізує бінарну класифікацію даних. Дані представляють собою крпки площини, які рівномірно розподілені у прямокутній області $[x_{\min}; x_{\max}] \cdot [y_{\min}; y_{\max}]$, де $[y_{\min}; y_{\max}]$ – межі зміни дискримінуючої функції $y=f(x)$ на сегменті $[x_{\min}; x_{\max}]$ згідно варіанту завдання. Належність крпки до певного класу (0 або 1) визначається її розташуванням нижче, або вище графіку дискримінуючої функції. Сумарний об'єм навчальної і контрольної вибірок для всіх іспитів прийняти таким, що дорівнює 100. Використовувати один прихований шар, кількість його нейронів прийняти: 2, 5, 10. Скористуватися наступними функціями активації: 'identity', 'logistic', 'tanh', 'relu' (остання за замовченням).

3. Дослідити якість роботи нейронної мережі. Скласти таблицю з заголовком згідно табл. 2.4.1.

Таблиця 2.4.1. Заголовок таблиці для внесення результатів досліджень

Кількість нейронів прихованого шару	Функція активації	Кількість помилкових класифікацій
-------------------------------------	-------------------	-----------------------------------

4. Побудувати в єдиному полі сімейство графіків залежностей кількості помилкових класифікацій від кількості нейронів прихованого шару для різних функцій активації.

Вимоги до звіту

Звіт по роботі повинен містити:

1. Назву дисципліни та лабораторної роботи.
2. Прізвище, ім'я та по батькові здобувача вищої освіти, код групи.
3. Номер варіанту завдання.
4. Код програми, що реалізує поставлені завдання.
5. Знімки зображень екрану, які виникають в процесі чисельного експерименту.

6. Таблицю і сімейство графіків, які ілюструють якість роботи нейронної мережі.

Контрольні питання

1. Назвіть приклади галузей застосування нейронних мереж.
2. Назвіть типові функції активації нейронних мереж.
3. Вербально опишіть основні кроки навчання нейронної мережі.
4. Сформулюйте правила Хебба. Для якого виду персептрону вони використовуються?
5. Запишіть співвідношення, яке може бути використане для обчислення похибки мережі на певному циклі навчання.
6. Поясніть сенс параметру `random_state` функції `make_moons`. В яких випадках його доцільно використовувати?
7. Запишіть рекурентне співвідношення для вихідного сигналу багат шарового персептрону.

Варіанти завдань

Таблиця 2.4.2. Дискримінуюча функція та межі варіації її незалежної змінної

№	Функція $y=f(x)$	Межі незалежної змінної $[x_{\min}; x_{\max}]$
1	$\operatorname{atan}(x)$	$[-2; 2]$
2	$\operatorname{sh}(x)$	$[-2; 2]$
3	$\operatorname{ch}(x)$	$[-2; 2]$
4	$\sin(x)$	$[-\pi; \pi]$
5	$ x $	$[-1; 1]$
6	$\ln(x)$	$[1; 10]$
7	$1 - x^2$	$[-1; 1]$
8	$\cos(x^2)$	$[-1; 1]$
9	$\sqrt{x}$	$[0; 1]$
10	$x \cdot \sin(x)$	$[0; 2\pi]$
11	$\cos(x)^2$	$[0; \pi]$
12	$\sqrt[3]{x}$	$[0; 8]$
13	$\exp(-x) \cdot \sin(x)$	$[0; \pi]$

14	$\cot(x)$	$\left[\frac{\pi}{2}; 3\right]$
15	$\frac{1}{\sin(x)}$	$\left[\frac{\pi}{2}; 3\right]$
16	$\frac{1}{1-x}$	$[-1; 0,5]$
17	$\frac{1}{x}$	$[0,5; 2]$
18	$\frac{x-1}{x+1}$	$[0; 1,5]$
19	$\cos(x)$	$[0; 2\pi]$
20	$\frac{\sin(x)}{x^2}$	$[0,5; 2\pi]$
21	$\sin\left(\frac{1}{x}\right)$	$\left[0,3; \frac{\pi}{2}\right]$
22	$ \sin(x) $	$[-\pi; \pi]$
23	x^2	$[0; 2]$
24	$\cos(x) + \sin(x)$	$\left[0; \frac{\pi}{2}\right]$
25	$\sqrt{\sin(x)}$	$\left[0; \frac{\pi}{2}\right]$
26	$\exp\left(\frac{1}{x}\right)$	$[0,5; 2]$
27	$\exp(-x^2)$	$[-1; 1]$
28	x^3	$[-1; 1]$
29	$x \cdot \exp(-x)$	$[0,5; 2]$
30	$ x \cdot \sin(x)$	$[-\pi; \pi]$
31	$\tan(x)$	$[-1; 1]$
32	$\sin(x)$	$[-\pi; \pi]$
33	$\ln\left(\frac{1}{x}\right)$	$[0,1; 5]$
34	$\frac{\exp(-x)}{x}$	$[0,2; 2]$
35	$\exp(x)$	$[0; 1,5]$
36	$\sqrt{1-x^2}$	$[-1; 1]$
37	x^x	$[0; 1,5]$

38	$\sqrt{x} \cdot \exp(-x)$	[0;1]
39	$\exp\left(-\frac{1}{x}\right)$	[0,1;0,3]
40	$\frac{x+1}{x-1}$	[0,1;0,6]

2.5. Практична робота № 5

Інтеграція інформаційних сервісів за допомогою хмарної платформи Make

Об'єкт – хмарні платформи інтеграції інформаційних сервісів. Предмет – хмарна платформи інтеграції інформаційних сервісів Make. Мета – отримання навичок інтеграції інформаційних сервісів за допомогою платформи Make.

Стислі теоретичні відомості

Make – це платформа для автоматизації та оптимізації багатьох бізнес процесів за допомогою блочного конструктора. Вона дозволяє створювати інтеграції для багатоетапних робочих процесів та зв'язків, без використання коду. За допомогою Make можна пов'язати більш ніж 1500 додатків, сервісів та смарт-пристроїв, включаючи різні сервіси розсилки та створення чат-ботів, eCommerce платформи та інструменти CRM (Customer Relationship Management, управління відносинами з клієнтами) систем. Прикладами можуть бути: e-mail, Google Sheets, Google Calendar, Google Drive, Google Forms, One Drive, Office 365 Excel, ChatGPT, Telegram Bot, Facebook.

Основні поняття

Сценарій: автоматизований робочий процес або процес, який створюється за допомогою Make. Він визначає, як сервіси інтегруються. Як правило, сценарій опрацьовує дані – він запускається автоматично, коли отримує дані з однієї програми (наприклад, при надходженні нового повідомлення у месенджер), перетворює дані їх, скажімо фільтрує, а потім зберігає дані в іншій програмі (наприклад, Google Sheets).

Модуль: представляє окремий крок у автоматизованому процесі. Модулі визначають конкретні функції інструменту або програми/послуги. Немає обмежень щодо кількості модулів, які ви можете додати до свого сценарію.

Операція: кожен раз, коли модуль у сценарії виконує дію на даними, це рахується як операція. Наприклад, якщо сценарій перевіряє вхідні електронні листи, отримує кожен вкладений файл і завантажує їх у папку Google Disk, кожна з них вважається однією операцією. Якщо є 4 файли, їхнє завантаження займе 4 операції.

Пакет: фрагмент даних, використовуються в модулях, і їх опис. Вхідні групи пакетів показують дані, що надходять до певного модуля, а вихідні групи показують дані, що надходять із певного модуля.

Тригер: процедура, яка автоматично виконується при виникненні події, що фіксується модулем, наприклад передача наступному модулю змісту нового повідомлення при надходженні його у месенджер.

Реєстрація на платформі Make

Доступною є вільна реєстрація на сайті make.com, при цьому потрібна поштова скринька Gmail. Слід мати на увазі, що безкоштовний профіль має обмеження в 2 поточних активних сценарії, 1000 операцій на місяць і 512 Мегабайт переданих даних на місяць. Поточний стан витрачених ресурсів можна контролювати пунктом меню *Organization*, яке розташоване на лівому боці вікна платформи.

Фрагмент вікна, яке виникає при вході на платформу Make має вигляд, представлений на рис. 2.5.1.

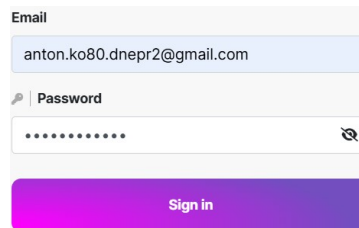


Рис. 2.5.1. Фрагмент вікна, яке виникає при вході на платформу Make

Приклад створення сценарію

Сценарії, які розміщені на профілі можуть бути переглянуті за пунктом меню *Scenarios* (рис. 2.5.2).

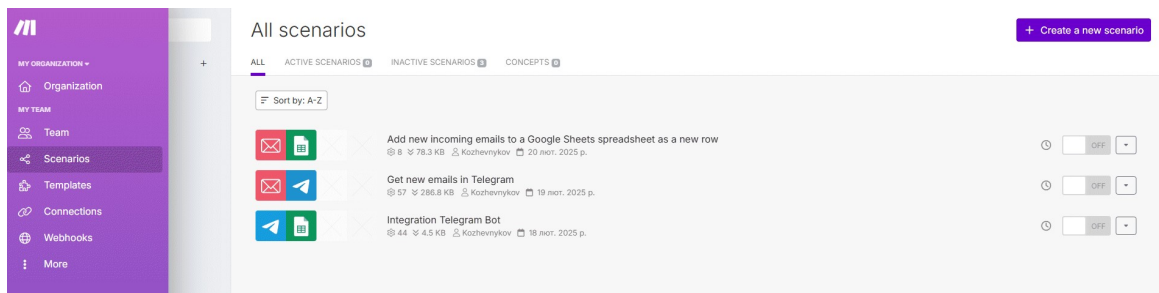


Рис. 2.5.2. Вікно перегляду сценаріїв, які розміщені на профілі

Праворуч розташовано кнопка створення сценарію “з чистого аркушу” *Create a new scenario* та перемикачі активності сценаріїв OFF/ON. Якщо сценарій активний, то активовані і тригери модулів і через кожний певний проміжок часу (для безкоштовного профілю мінімально 15 хв.) автоматично проводиться перевірка наявності подій на які реагують тригери (наприклад надходження нового листа в скриньку чи повідомлення в чат бот). Кожна така перевірка є операцією і тому для запобігання їх зайвої витрати сценарії, що в певний час не використовуються варто деактивувати.

Створення сценарію можна починати з натискування зазначеної вище кнопки, або за допомогою шаблону – пункт меню *Templates*.

Розглянемо приклад створення за допомогою шаблону сценарію, який контролює надходження нових листів в поштову скриньку та розміщення їх атрибутів та змісту у Google Sheets. Після вибору відповідного шаблону у вікні платформи виникає послідовність модулів, що зображена на рис. 2.5.3.

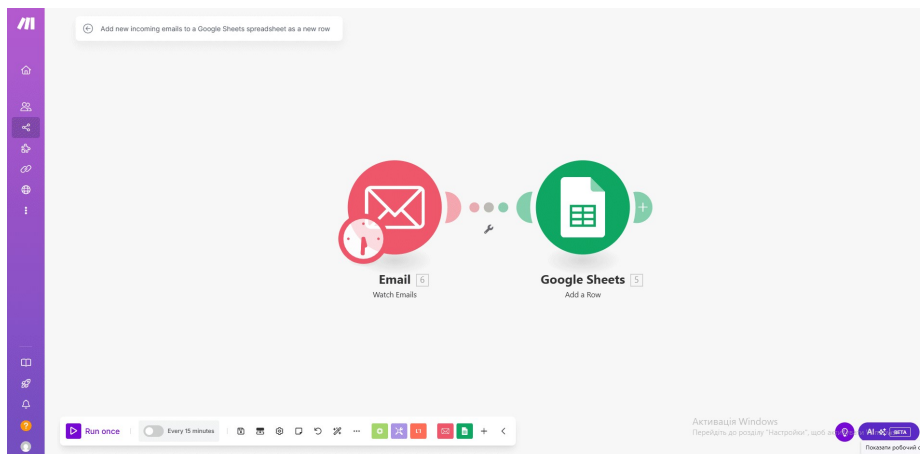


Рис. 2.5.3. Приклад послідовності модулів сценарію

Далі, як і в інших середовищах візуального програмування, слід задати властивості візуальних компонентів – модулів. Вікно властивостей модуля стає доступним, якщо зробити клік на відповідному модулі. Для модуля Email обов’язковими властивостями є (рис. 2.5.4):

Connection – вид з’єднання. Тут треба обрати Microsoft SMTP/IMAP OAuth connection – перевірку справжності OAuth для підключення за протоколами SMTP/IMAP та доступу до даних електронної пошти для користувачів Office 365. При цьому слід мати на увазі, що платформа Make запросить доступ до даних профілю користувача Office 365;

Folder – папка скриньки, що буде контролюватися;

Criteria – критерій відбору листів, що будуть контролюватися.

Також варто вказати Recipient email address – адресу поштової скриньки, яка контролюється. Якщо це поле залишити порожнім, то судячи зі всього буде контролюватися Gmail скринька, що вказана при реєстрації.

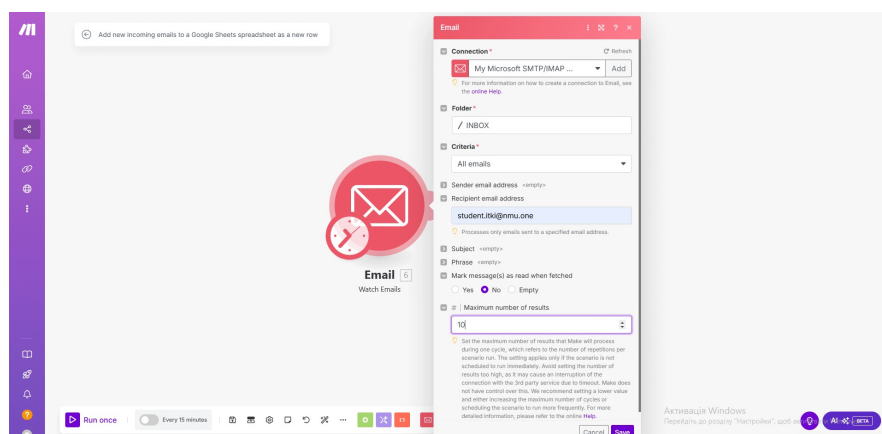


Рис. 2.5.4. Властивості модуля Email

Після встановлення значень властивостей треба натиснути кнопку *Save*. Якщо все встановлено вірно, в верхній правій частині модуля з’явиться чорний кружечок з зірочкою, якщо ні, то червоний з цифрою, яка позначає кількість невірно або неповно встановлених властивостей.

Перед встановленням властивостей модуля Google Sheets на ресурсі <https://docs.google.com/spreadsheets/create> слід створити електронну таблицю з іменем e-mail, іменем аркуша Mail, заголовками стовбців: Дата, Відправник, Тема, Адреса, Зміст.

Для модуля Google Sheets обов'язковими властивостями є (рис. 2.5.5):

Spreadsheet ID – ім'я електронної таблиці;

Sheet Name – ім'я аркуша;

Table contains headers – ознака наявності заголовків електронної таблиці.

Слід заповнити поля властивості Values – величини, які будуть розміщуватися у визначених стовбцях електронної таблиці. Значення цих величин можуть бути обрані зі значень, які передаються з модуля Email і стають доступними для вибору після кліку у відповідному полі властивості Values (рис. 2.5.6).

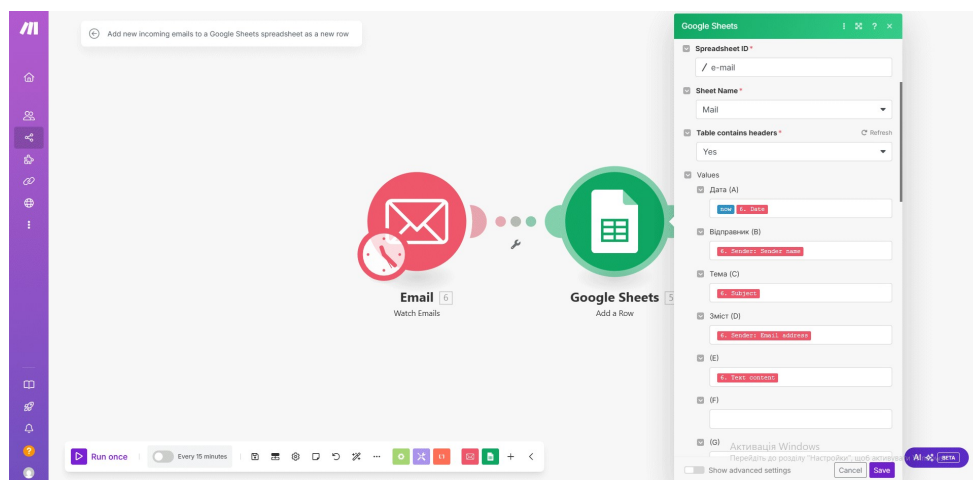


Рис. 2.5.5. Властивості модуля Google Sheets

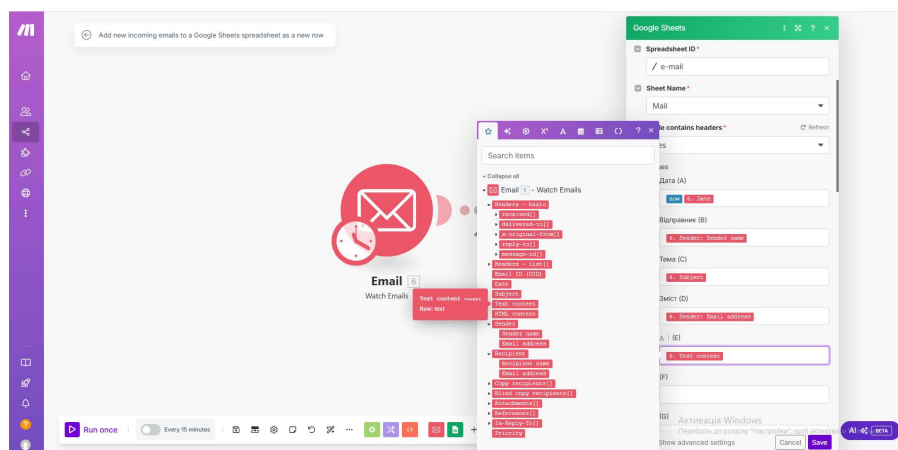


Рис. 2.5.6. Поля властивості Values модуля Google Sheets

Для створеного сценарію функціонування кожного модуля можна перевірити окремо за допомогою пункту *Run this module only* контекстного меню модуля.

Перед активуванням сценарію треба зберегти внесені в нього зміни натисканням кнопки *Save changes* у вікні (рис. 2.5.7), яке виникає при переході до вікна, що зображене на рис. 2.5.2.

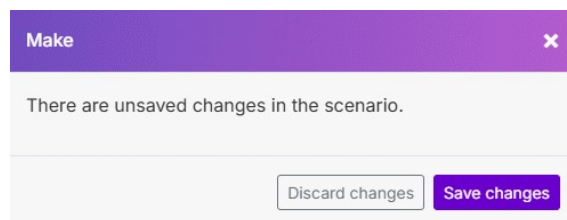


Рис. 2.5.7. Вікно збереження змін у сценарії

Сценарій активується за допомогою відповідного перемикача у стан ON. При цьому розклад перевірки виникнення подій спрацювання тригерів за замовченням встановлюється як регулярне опитування через кожні 15 хвилин. Цей розклад можна змінювати (рис. 2.5.8).

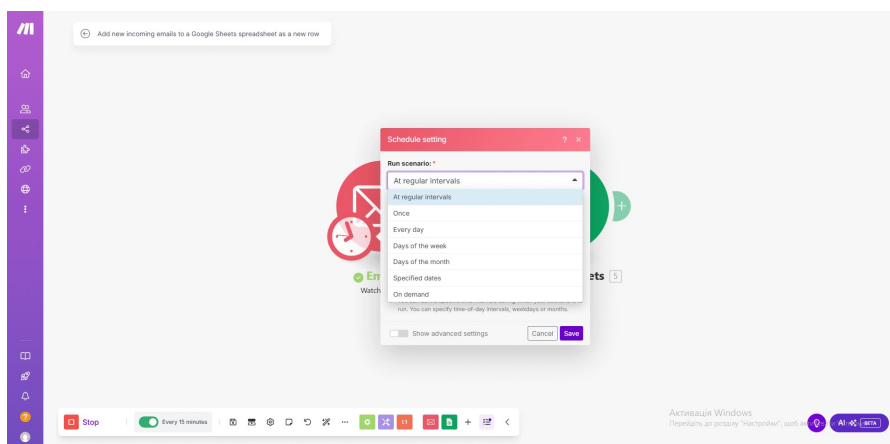


Рис. 2.5.8. Вікно розкладу перевірки виникнення подій спрацювання тригерів

Переведення перемикача розкладу (на рис. 8 він має надпис Every 15 minutes) в положення OFF деактивує сценарій. Натиснення кнопки *Run once* у неактивному сценарії (рис. 2.5.2) викликає однократну перевірку виникнення подій спрацювання тригерів, при цьому активації сценарію не відбувається.

При спрацюванні тригерів сценарію можна передивитися пакети оброблених даних (рис. 2.5.9).

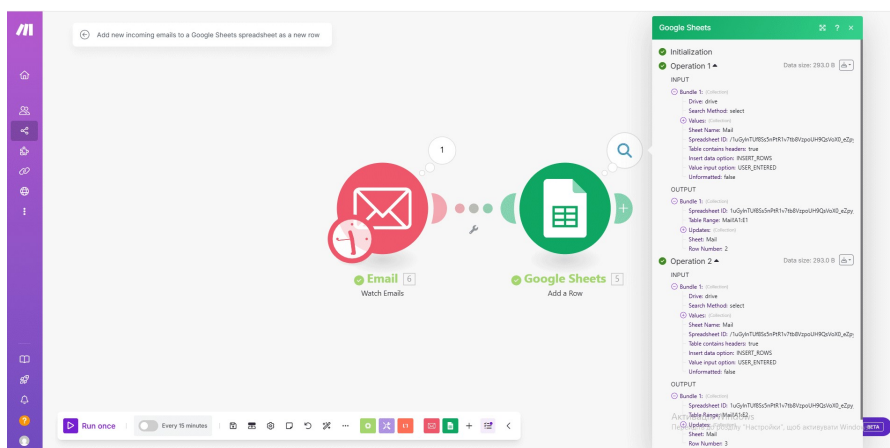
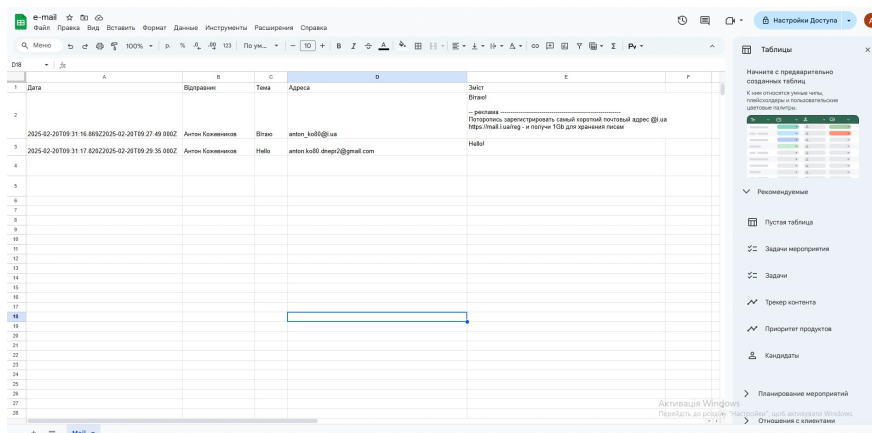


Рис. 2.5.9. Вікно перегляду пакетів даних модуля Google Sheets

Кінцевий результат обробки двох листів представлений у вигляді електронної таблиці на рис. 2.5.10.



The screenshot shows a Google Sheets spreadsheet with the following data:

Дата	Відправник	Тема	Адреса	Зміст
2025-02-20T09:31:16.66822025-02-20T09:27:49.0802	Антон Кошовець	Відео	anton_koshov@ua	... реклама - Потрібно зареєструвати свій короткий поштовий адрес @ на https://mail.i.ua/ua/ - і отримувати ТОВ для країнних послуг
2025-02-20T09:31:17.62822025-02-20T09:29:35.0802	Антон Кошовець	Ніло	anton_koshov@gmail.com	Ніло!

Рис. 2.5.10. Результат обробки двох листів електронної пошти представлений у вигляді електронної таблиці Google Sheets

Задання властивостей інших модулів

Модуль Telegram Bot

Чат-бот в Telegram – це автоматизований аккаунт в месенджері, який запрограмований на виконання певних дій: продаж товарів, залучення користувачів, пошук корисної інформації тощо. Працює за принципом «запит – відповідь»: користувач вводить текстову команду, бот обробляє запит, знаходить відповідний сценарій і відповідає за ним користувачеві.

В найпростішому варіанті в контексті роботи з платформою Make чат-бот або отримує текстове повідомлення зі зовнішнього середовища і передає його іншим модулям платформи, або навпаки отримує від них повідомлення і оприлюднює їх в месенджері.

Якщо модуль працює як джерело повідомлень, які передає іншим модулям, обов'язковими властивостями є (рис. 2.5.11):

Webhook, Webhook name – спосіб отримання повідомлень чат-ботом та його ім'я – встановлюються як My Telegram Bot Updates webhook;

Connection та Connection name – з'єднання (внутрішнє ім'я чат-боту, яке визначається при його створенні) та назва з'єднання – встановлюється як My Telegram Bot connection;

Token – токен чат-боту.

Токен – це унікальний цифро-буквений ідентифікатор, який необхідний для авторизації і взаємодії з Telegram bot API. При створенні чат-бота йому автоматично створюється і присвоюється певний токен.

Процедура створення чат-бота і отримання токена полягає у наступному:

- в пошуковому рядку Telegram знаходиться сервіс BotFather;
- вводиться команда /newbot, внутрішнє name і унікальне зовнішнє ім'я (яке будуть бачити інші користувачі) username, наприклад Finance і KozhevnykovBot;
- якщо все пройшло успішно, BotFather виводить повідомлення про успішне

створення бота, а також його токен, який виглядає на кшталт 8653301867:AAETdoy89PaiRiH7SmCSvib7kMI21jF7_Qw

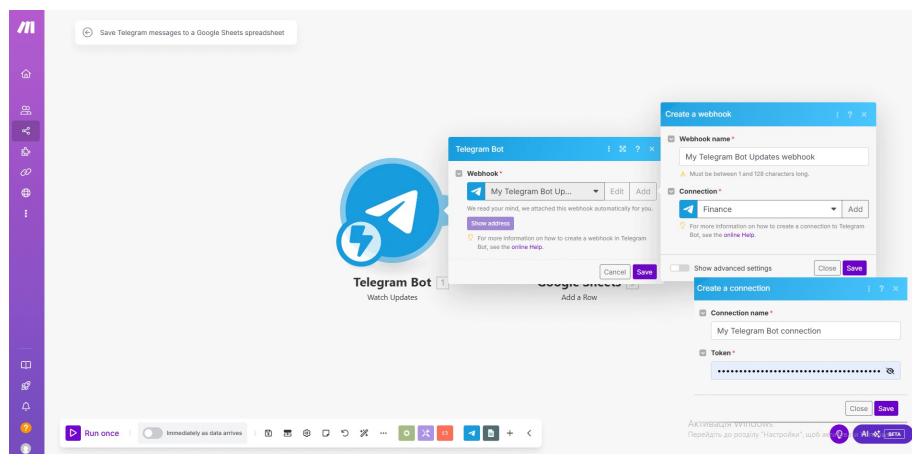


Рис. 2.5.11. Властивості модуля Telegram Bot в якості джерела повідомлень

Приклад результатів роботи сценарію, який передбачає отримання повідомлення, яке введене в чат-боті, модулем Telegram Bot, його передачу та відображення модулем Google Sheets представлено на рис. 2.5.12 (пакет Telegram Bot), рис. 2.5.13 (пакет Google Sheets), рис. 2.5.14 – фрагмент вихідної електронної таблиці, що відповідає повідомленню.

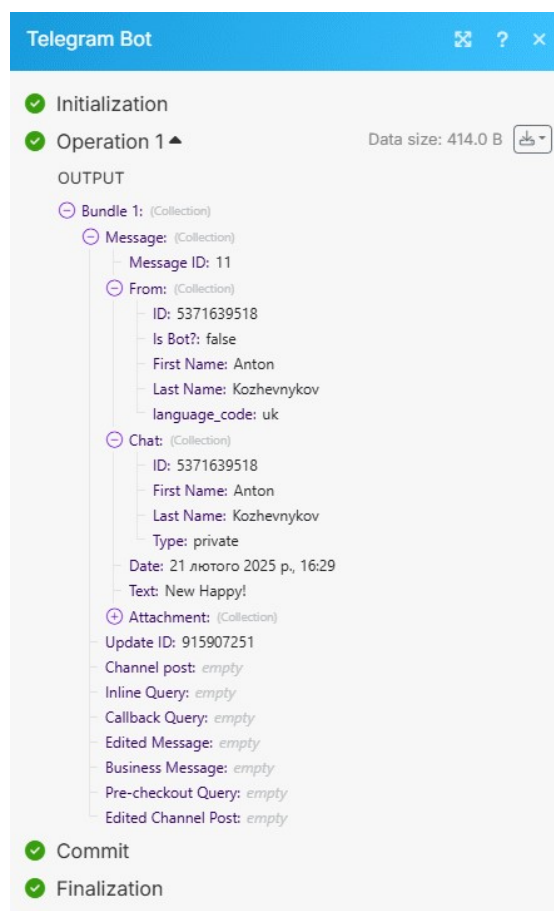


Рис. 2.5.12. Приклад пакету даних модуля Telegram Bot

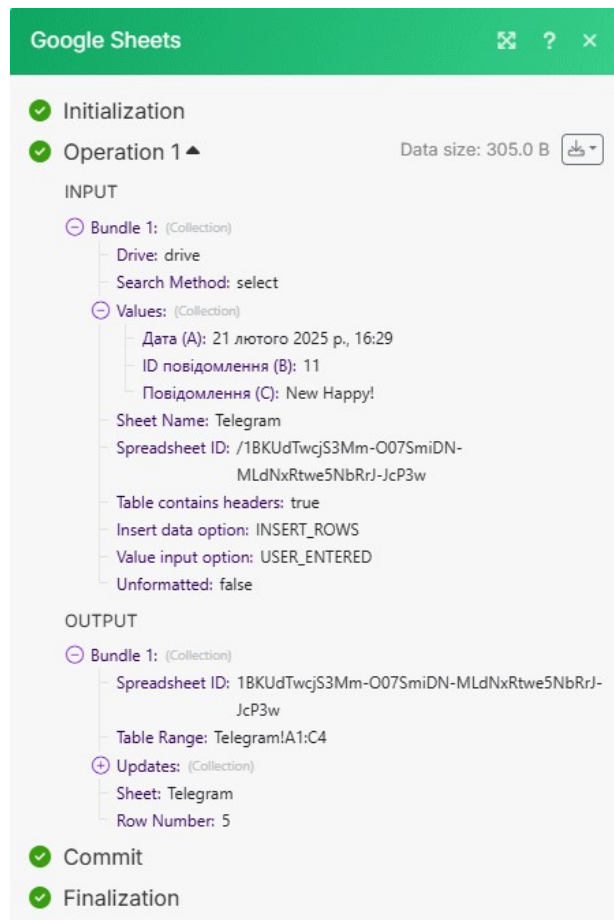


Рис. 2.5.13. Приклад пакету даних модуля Google Sheets

	A	B	C
1	Дата	ID повідомлення	Повідомлення
2	2025-02-21T14:29:55.000Z	11	New Happy!

Рис. 2.5.14 Приклад фрагменту вихідної електронної таблиці Google Sheets, що відповідає повідомленню, яке отримано з Telegram Bot

Якщо модуль працює як отримувач повідомлень, обов'язковими властивостями є (рис. 2.5.15):

Connection – з'єднання (внутрішнє ім'я чат-боту, яке визначається при його створенні), якщо користувач має кілька чат-ботів, повідомлення будуть надсилатися до чату, який визначений цією властивістю;

Chat_ID – ID чата користувача;

Text – текст повідомлення, яке буде надіслано від чат-бота.

Процедура визначення ID чата користувача полягає у наступному:

- в пошуковому рядку Telegram вводиться ім'я @getmyid_bot;
- вводиться команда /start;
- у відповіді відображається ID користувача та ID чата користувача:

Your user ID: 5371639518

Current chat ID: 5371639518

На наступних рисунках представлений приклад результатів роботи сценарію, який передбачає отримання повідомлення, що надійшло в поштову скриньку (рис. 2.5.16), його передачу модулем Email (рис. 2.5.17 пакет Email) та відображення модулем Telegram Bot (рис. 2.5.18 пакет Telegram Bot). Вихідним повідомленням при цьому буде:

2025-02-21T12:05:31.000ZВітаюВітаю!

Рис. 2.5.15 Властивості модуля Telegram Bot в якості отримувача повідомлень

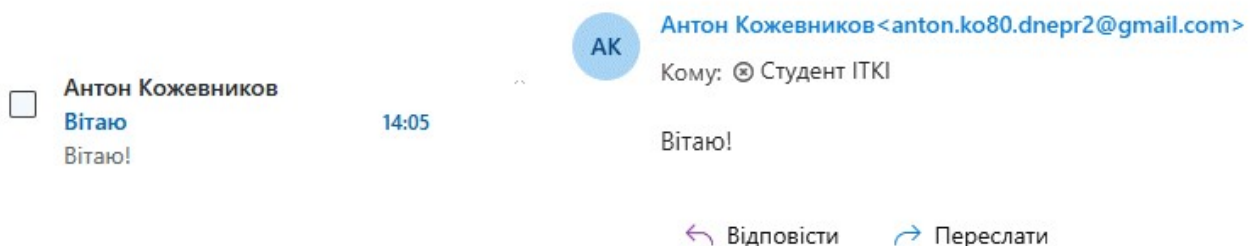


Рис. 2.5.16. Приклад вхідного повідомлення в електронній пошті

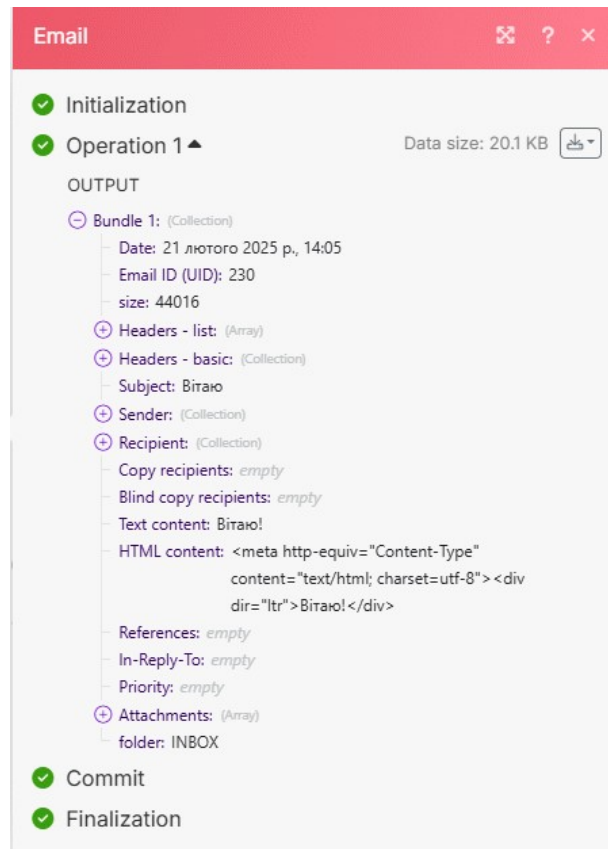


Рис. 2.5.17. Приклад пакету даних модуля Email

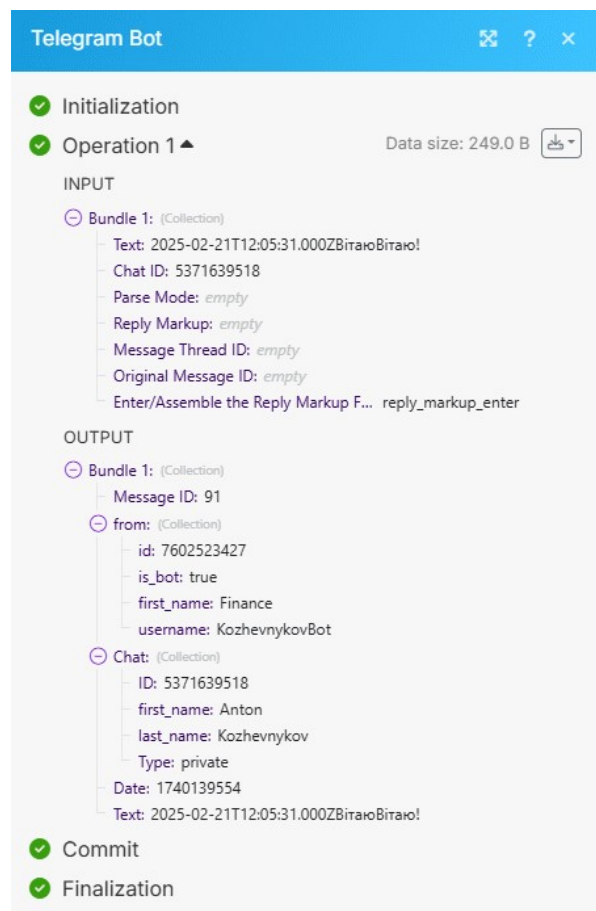


Рис. 2.5.18. Приклад пакету даних модуля Telegram Bot

Модуль Google Calendar

Доступ до Google Calendar здійснюється на ресурсі <https://workspace.google.com/intl/uk/products/calendar/>

Для модуля Google Calendar обов'язковими властивостями є (рис. 2.5.19):

Connection – з'єднання, яке встановлюється в значення My Google Connection і прив'язується до профілю електронної пошти Gmail;

Calendar – вид календаря, яке встановлюється в значення Primary Calendar;

Watch Events – події, що викликають спрацювання тригерів сценарію, яке встановлюється в значення By Created Data.

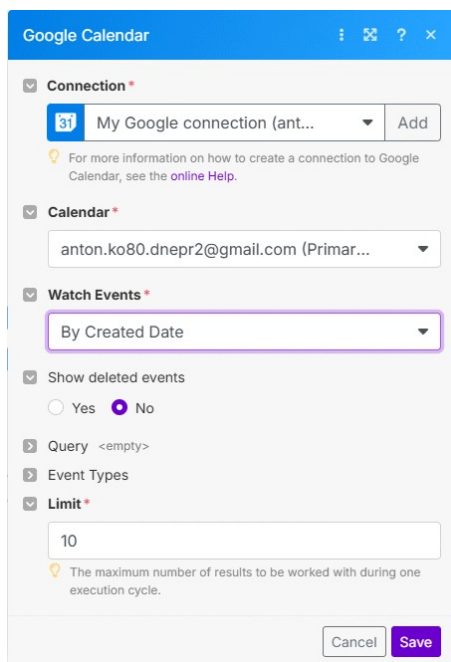


Рис. 2.5.19. Властивості модуля Google Calendar

На наступних рисунках представлений приклад результатів роботи сценарію, який передбачає для подій, що введені у календар (рис. 2.5.20 вікно події, рис. 2.5.21 пакет Google Calendar), їх передачу і відображення у Google Sheets (рис. 2.5.22 пакет Google Sheets).

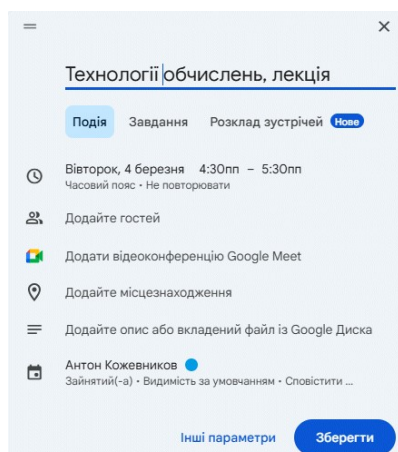


Рис. 2.5.20. Приклад вікна події, що введена у календар

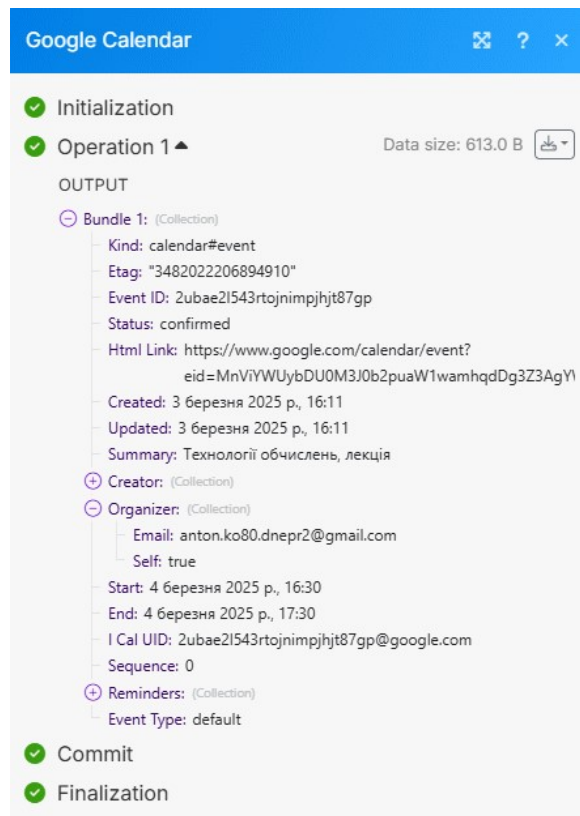


Рис. 2.5.21. Приклад пакету даних модуля Google Calendar

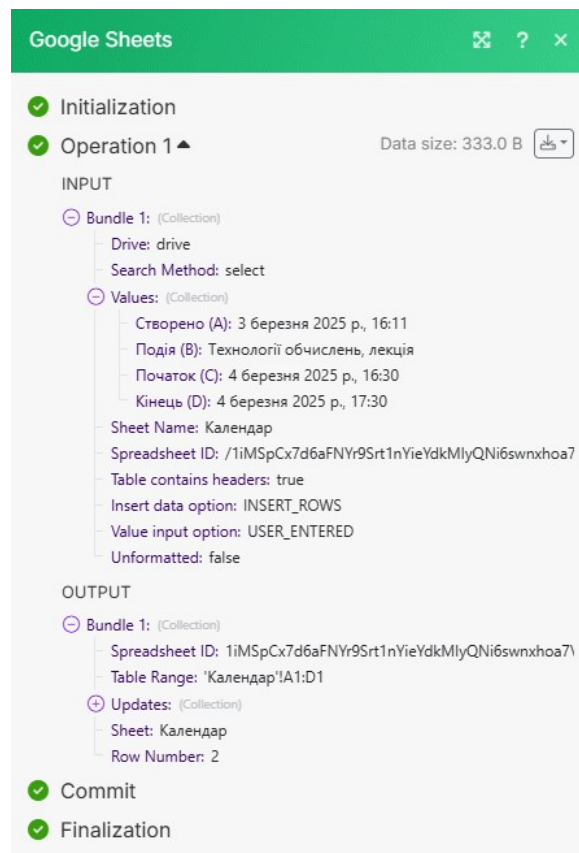


Рис. 2.5.22. Приклад пакету даних модуля Google Sheets

Відображення введеної у календар події у вигляді фрагменту електронної таблиці на представлено рис. 2.5.23.

A	B	C	D
Створено	Подія	Початок	Кінець
2025-03-03T14:11:43.000Z	Технології обчислень, лекція	2025-03-04T14:30:00.000Z	2025-03-04T15:30:00.000Z

Рис. 2.5.23. Фрагмент повідомлення, яке представлено в електронній таблиці Google Sheets

Дано

Доступ до платформи інтеграції інформаційних сервісів Make.

Потрібно

Створити сценарій збирання, передачі, обробки і зберігання даних, який інтегрує різноманітні інформаційні сервіси.

Рекомендації щодо виконання роботи

1. Завдання на лабораторну роботу полягає у створенні сценарію, який передбачає для подій, що введені у календар передачу і відображення у Google Sheets і наступну передачу і відображення у Telegram Bot. Для цього потрібно зробити наступне.

2. Створити безкоштовний акаунт на платформі Make.

3. Створити порожній Google Calendar.

4. Створити електронну таблицю Google Sheets. Ім'я книги і аркуша таблиці повинно збігатися з прізвищем здобувача вищої освіти.

5. Створити чат-бот Telegram.

6. Об'єднати в сценарії Make модулі Google Calendar, Google Sheets, Telegram Bot і налаштувати їх властивості.

7. Ввести в Google Calendar розклад занять на тиждень поточної чверті. Розклад занять повинен відповідати офіційному, який оприлюднено на ресурсі <https://old.nmu.org.ua/ua/content/students/schedule> без фактичних відхилень. здобувачі вищої освіти з непарними номерами вводять розклад числівника, а парних – знаменника.

8. Простежити відображення розкладу в Google Sheets та Telegram Bot.

Вимоги до звіту

1. Звіт по роботі повинен містити:

2. Назву дисципліни та лабораторної роботи.

3. Прізвище, ім'я та по батькові здобувача вищої освіти, код групи.

4. Номер варіанту завдання.

5. Знімок зображення фрагменту вікна, яке виникає при вході на платформу Make (Рис. 1).

6. Знімок зображення вікна Google Calendar з введенням розкладом занять.

7. Знімки зображень фрагментів вікон, які виникають при налаштуванні властивостей і роботі модулів сценарію, з підписами (аналогічні приведені у прикладі Рис. 2.5.19, 2.5.21 (Google Calendar), 2.5.5, 2.5.22 (Google Sheets), 2.5.15, 2.5.18 (Telegram Bot).

8. Знімки зображень вікон Google Sheets та Telegram Bot з відображеним розкладом занять.

Контрольні питання

1. Яке основне призначення платформи Make?

2. Які обмеження має безкоштовний акаунт платформи Make?
3. Назвіть найбільш уживані модулі платформи Make.
4. Назвіть обов'язкові властивості певного модулю платформи Make.
5. Дайте визначення тригеру сценарію.
6. Для чого використовуються чат-боти?
7. Дайте визначення токена чат-боту.
8. Опишіть процедуру створення чат-боту Telegram і отримання його токена.
9. Опишіть процедуру отримання в Telegram ID користувача та ID чата користувача.

2.6. Практична робота № 6

Розробка сценарію хмарної платформи Make з використанням засобів штучного інтелекту Eden AI

Об'єкт – хмарні платформи інтеграції інформаційних сервісів з використанням штучного інтелекту. Предмет – хмарна платформа інтеграції інформаційних сервісів Make і хмарна платформа забезпечення штучного інтелекту Eden AI. Мета – отримання навичок розробки сценаріїв платформи Make з використанням засобів штучного інтелекту.

Стислі теоретичні відомості

Eden AI (Artificial Intelligence) – це платформа, призначена для спрощення інтеграції можливостей штучного інтелекту (ШІ) в різні інформаційні системи, програми та робочі процеси. Він служить агрегатором API та API AI, дозволяючи користувачам у єдиний доступ до технологій та постачальників ШІ.

- генерація контенту: використання API-інтерфейсів генеративного штучного інтелекту для створення різних типів контенту, наприклад, зображень та тексту;
- розбір документа: автоматизація вилучення, класифікації і аналізу даних із документів за допомогою API-інтерфейсів аналізу документів;
- аналіз тексту: доступ до широкого спектру механізмів аналізу тексту та NLP, NER, визначення мови, вилучення та класифікацію ключових слів, тощо.

NLP (Natural Language Processing), або обробка природної мови, – це галузь штучного інтелекту, мета якої в полягає в наданні інформаційним системам можливість розуміти та обробляти природну мову.

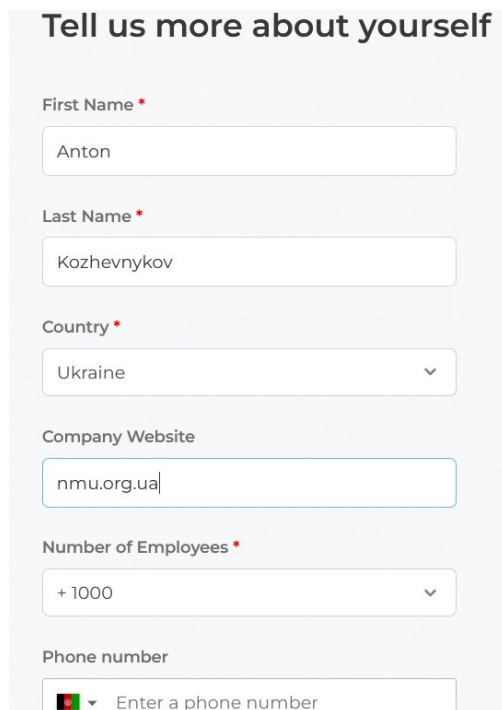
NER (Named Entity Recognition) – це завдання в галузі NLP, спрямоване на виділення та класифікацію іменованих сутностей у тексті, таких як імена людей, назви організацій, дати, місця розташування, суми грошей та інші типи специфічних об'єктів.

Реєстрація на платформі Eden AI

Доступною є вільна реєстрація на сайті app.edenai.run, при цьому потрібна поштова скринька Gmail. Безкоштовний профіль забезпечує до 60 викликів API

за хвилину і пропонує вільне використання плагінів для підтримки Zapier, Make, Bubble, Microsoft та Discord.

При реєстрації потрібно окрім даних акаунту Gmail вказати певні особисті дані (рис. 2.6.1, 2.6.2).



The image shows a registration form titled "Tell us more about yourself". It contains the following fields:

- First Name ***: A text input field containing "Anton".
- Last Name ***: A text input field containing "Kozhevnykov".
- Country ***: A dropdown menu showing "Ukraine".
- Company Website**: A text input field containing "nmu.org.ua".
- Number of Employees ***: A dropdown menu showing "+ 1000".
- Phone number**: A field with a country code dropdown (showing a flag) and a text input area with the placeholder "Enter a phone number".

Рис. 2.6.1. Вікно реєстрації на хмарній платформі забезпечення штучного інтелекту Eden AI (початок)

Модуль Eden AI платформи Make

На платформі Make доступною є низка модулів з функціями ШІ, які працюють з відповідними даними і їх структурами: Open AI (текст, зображення, аудіо), Dumpling AI (текст, зображення, аудіо, Web, бази знань), Brain Pod AI (текст), TextCortex AI (текст), Azure Open AI (текст, зображення, аудіо), Schogini AI Wizard (текст), Google Vertex AI (аудіо, відео) тощо.

Використовуваний в лабораторній роботі модуль Eden AI платформи Make має величезну кількість методів роботи, які дозволяють здійснювати інтелектуальну обробку тексту (як то виділення ключових слів із тексту, перевірка граматики, анонімізація тексту, виділення загальні теми в тексті, виявлення текстів, що створені ШІ), зображень (виявлення об'єкта на зображенні, виявлення на зображенні популярних природніх та штучних структур, анонімізація зображення), аудіо (перетворення аудіо на текст і навпаки), відео (визначення людських обличчя на відео, розпізнавання обличчя в існуючій базі даних), оптичне розпізнавання тексту, переклад тексту з однієї мови іншою.

Далі використовується метод Summarize a Text, який резюмує текст, тобто виявляє найважливіші фрагменти тексту з метою створення його стислої версії.

Customize your experience

What is your team? *

Other

What's your technical background? *

☐ I'm a tech expert
I have the skills to fully implement an external API in an existing project.

☒ I'm a hands-on tech-savvy
I can test things out, use APIs but I might not be ready to integrate it in production.

☐ I'm non-technical
I don't have the technical skills to test or implement an API on my own.

What do you need Eden AI for *

☐ Automate process in my company

☒ Integrate AI in a software I am building

Рис. 2.6.2. Вікно реєстрації на хмарній платформі забезпечення штучного інтелекту Eden AI (закінчення)

Перед встановленням властивостей модуля Eden AI на одноіменному ресурсі слід отримати токен API за пунктом меню *API keys* (рис. 2.6.3).

API Settings

Features settings | Bring your own account

Settings

Region: World

Save my requests: ☐ Enable History

Cache responses: ☐ Enable Cache

API Keys

API token: [Redacted]

Seperate monitoring

With a single API you can enable monitoring for each of your customers and control costs and usage separately. [Check documentation.](#)

Additional API Keys: [Create a new API Key]

Рис. 3 Вікно отримання токену API

Для модуля Eden AI обов'язковою властивістю є Connection – з'єднання, яке встановлюється в значення Eden AI Connection, при цьому вводиться отримане значення токену API (рис. 2.6.4);

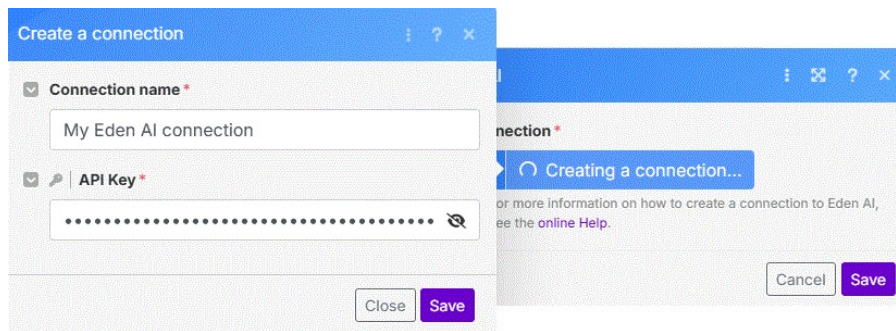


Рис. 2.6.4. Налаштування властивості Connection модуля Eden AI

Інші обов'язкові властивості представлені на рис. 2.6.5

Рис. 2.6.5. Властивості модуля Eden AI

Text – правила за якими робиться резюме тексту, що аналізується;

Output sentences – кількість речень вихідного резюме;

Providers – ім'я провайдеру ШІ;

Language – мова тексту, що аналізується.

Наприклад правила формування резюме мають вигляд:

Проаналізуйте ці витрати "*Вхідне повідомлення*" в форматі "сума, категорія витрат". Сума без валюти – це лише число. Як категорію витрат, візьміть категорії "продукти, напої, кафе, транспорт, одяг, взуття, спорт, подорожі, здоров'я, краса, комунальні платежі, інші". Повернути лише суму та категорію, без пояснень.

Або англійською:

Analyze these expenses "*Input message*" in the format "amount, expense category". The amount without currency is just a number. As an expense category, take categories "groceries, drinks, cafes, transport, clothes, shoes, sports, travel, health, beauty, municipal payments, others". Return only the amount and category, without explanations.

Приклад створення сценарію, який включає модуль Eden AI

Розглянемо створення сценарію, який передбачає для повідомлень, що надійшли у Telegram Bot і містять інформацію про оплачені користувачем товари і послуги та розміри витрат, їх поділ за певними споживчими категоріям та резюмування за допомогою Eden AI і подальше відображення у Google Sheets.

Типові пакети оброблених даних для сценарію, що розглядається представлені на рис. 2.6.6 – 2.6.8.

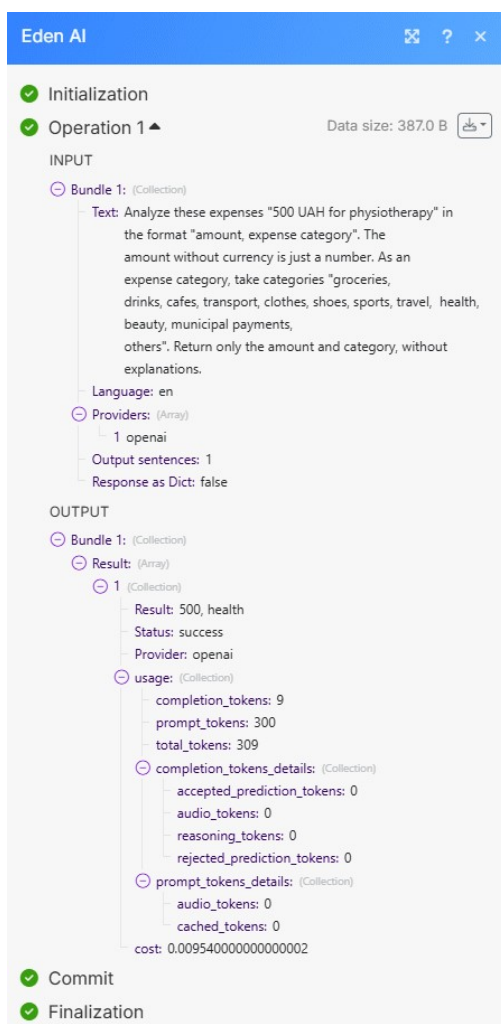


Рис. 2.6.6. Приклад пакету даних модуля Eden AI

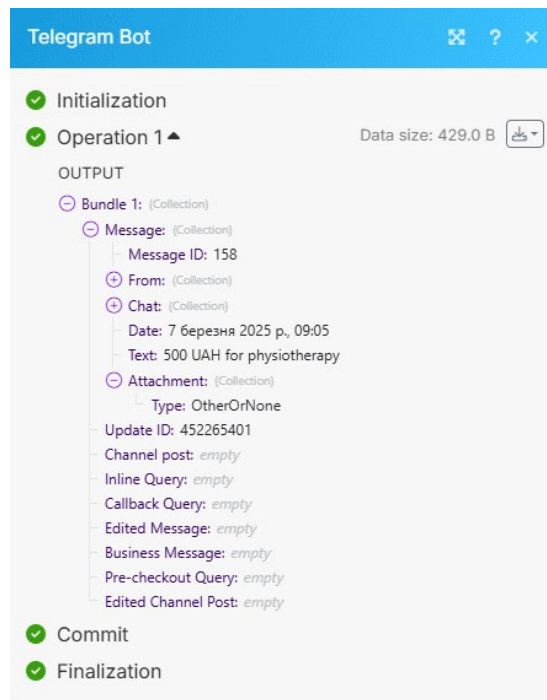


Рис. 2.6.7. Приклад пакету даних модуля Telegram Bot

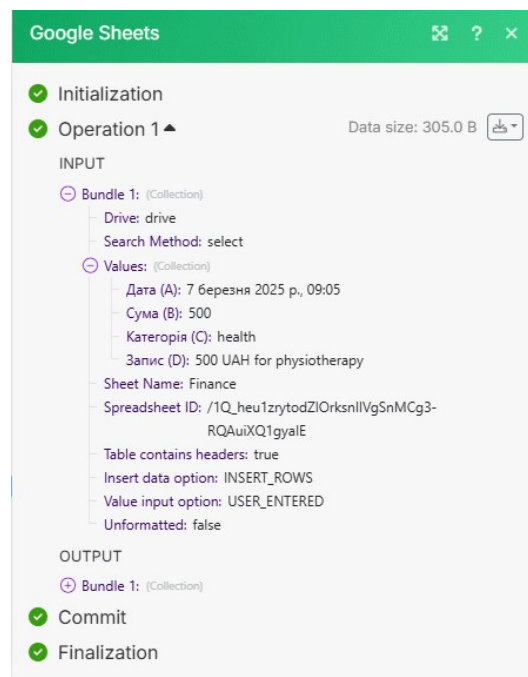


Рис. 2.6.8. Приклад пакету даних модуля Google Sheets

Налаштування властивостей модулів Telegram Bot і Google Sheets було розглянуто у лабораторній роботі № 5. Але з рис. 6 випливає, що властивість результатів аналізу повідомлення за допомогою III *Result* модулю Eden AI, яка повинна відображатися в електронній таблиці у вигляді обсягу та категорії витрати, представляє собою складну вкладену структуру. Тому для вилучення зазначених величин обсягу та категорії з властивості *Resul* в стовбцях *Сума* та *Категорія* електронної таблиці використовується низка вкладених функцій (рис. 9).

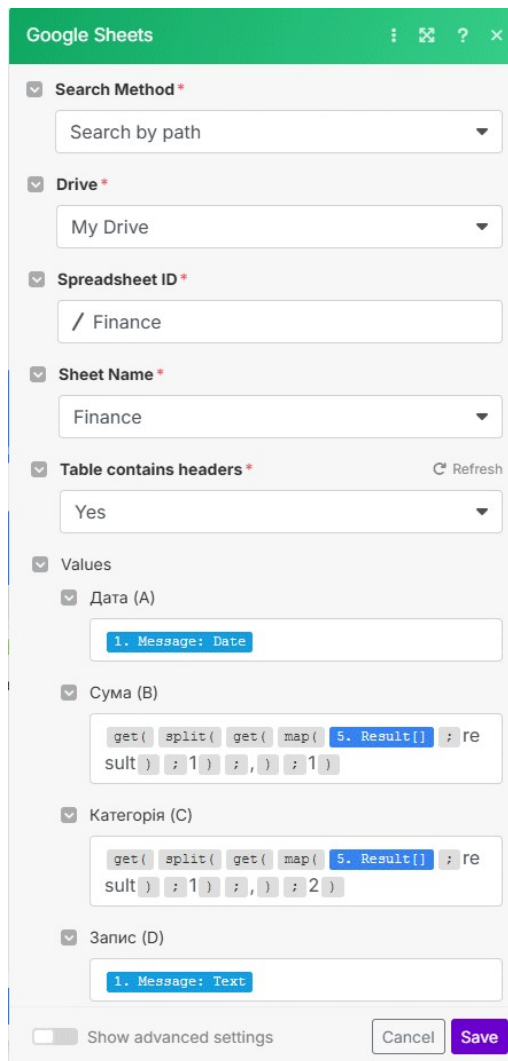


Рис. 2.6.9. Налаштування властивостей модуля Google Sheets

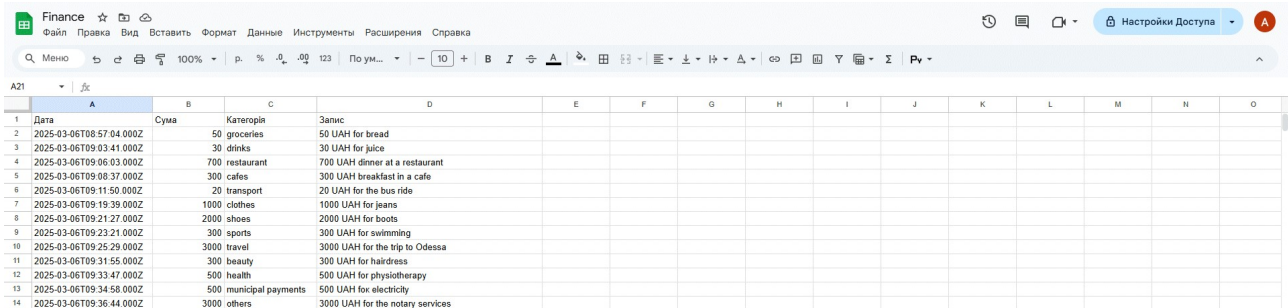
При цьому використовуються функції `get`, `map` та `split`. Фрагменти, які в рядках Сума (B) та Категорія (C) не виділені, або виділені сірим вводяться вручну, а ті, які виділені синім – вставляються з пакету Eden AI. Для розуміння роботи цих функцій слід мати на увазі, що в сценаріях платформи Make підтримуються зокрема такі складні типи даних як колекції, прості та складні масиви. Колекції представляють собою списки властивостей об'єктів. В деяких мовах програмування вони звуться записами. Складові записів – поля можуть містити дані різних типів. Простий масив містить одне або декілька текстових значень. Складний масив містить одну або кілька колекцій одного типу. Прикладом складного масиву є вкладення електронної пошти. Кожному електронному листу відповідає масив вкладень. Кожне вкладення представляє колекцію, яка може містити назву, вміст, розмір тощо. Синтаксис функцій приведений нижче.

`get(array; position)` – повертає елемент простого масиву, який знаходиться у позиції `position`;

`map(complex array; key)` – повертає простий масив, що містить значення складного масиву у полі `key`;

split(text; separator) – перетворює рядок, який розділений символами separator, на простий масив підрядків.

Кінцевий результат обробки низки вхідних повідомлень, що належать різним споживчим категоріям представлений у вигляді електронної таблиці на рис. 2.6.10.



Дата	Сума	Категорія	Запис
2025-03-06T08:57:04.000Z	50	groceries	50 UAH for bread
2025-03-06T09:03:41.000Z	30	drinks	30 UAH for juice
2025-03-06T09:06:03.000Z	700	restaurant	700 UAH dinner at a restaurant
2025-03-06T09:08:37.000Z	300	cafes	300 UAH breakfast in a cafe
2025-03-06T09:11:50.000Z	20	transport	20 UAH for the bus ride
2025-03-06T09:19:39.000Z	1000	clothes	1000 UAH for jeans
2025-03-06T09:21:27.000Z	2000	shoes	2000 UAH for boots
2025-03-06T09:23:21.000Z	300	sports	300 UAH for swimming
2025-03-06T09:25:29.000Z	3000	travel	3000 UAH for the trip to Odessa
2025-03-06T09:31:55.000Z	300	beauty	300 UAH for hairdress
2025-03-06T09:33:47.000Z	500	health	500 UAH for physiotherapy
2025-03-06T09:34:58.000Z	500	municipal payments	500 UAH for electricity
2025-03-06T09:36:44.000Z	3000	others	3000 UAH for the notary services

Рис. 2.6.10. Кінцевий результат обробки низки вхідних повідомлень

Рекомендації щодо виконання роботи

1. Завдання на лабораторну роботу полягає у створенні розглянутого вище сценарію у відповідності до варіанту завдання. Для цього потрібно зробити наступне.

2. Створити безкоштовний акаунт на платформі Eden AI.

3. Створити електронну таблицю Google Sheets. Ім'я книги повинно бути Finance, а аркуша таблиці повинно збігатися з прізвищем здобувача вищої освіти.

4. Створити чат-бот Telegram, якщо він не був створений раніше.

5. Об'єднати в сценарії Make модулі Telegram Bot, Eden AI, Google Sheets і налаштувати їх властивості.

6. Здобувачі вищої освіти з непарними номерами створюють україномовний варіант сценарію, а парних – англomовний.

7. Ввести в чат-бот Telegram повідомлення, що містять інформацію про оплачені користувачем товари і послуги та розміри витрат по категоріях згідно варіанту завдання (табл. 2.6.1).

8. Простежити коректність відображення повідомлень в Google Sheetst.

Вимоги до звіту

1. Звіт по роботі повинен містити:

2. Назву дисципліни та лабораторної роботи.

3. Прізвище, ім'я та по батькові здобувача вищої освіти, код групи.

4. Номер варіанту завдання.

5. Знімки зображень фрагментів вікон, які виникають при реєстрації на платформі Eden AI (Рис. 2.6.1, 2.6.2).

6. Знімки зображень фрагментів вікон, які виникають при налаштуванні властивостей і роботі модулів сценарію, з підписами (аналогічні приведеним у прикладі Рис. 2.6.7 (Telegram Bot), Рис. 2.6.4 – 2.6.6 (Eden AI), Рис. 2.6.8, 2.6.9 (Google Sheets).

7. Знімок зображення вікна Google Sheets з відображеними повідомленнями (Рис. 2.6.10).

Контрольні питання

1. Назвіть основні хмарні платформи, що підтримують штучний інтелект.
2. Наведіть приклади застосування штучного інтелекту.
3. Назвіть основні модулі платформи Make з функціями ІІІ і види даних з якими вони працюють.
4. Наведіть приклади використання модулю Eden AI платформи Make для обробки різних видів даних.
5. Опишіть структуру даних, яка є результатом роботи модулю Eden AI.
6. Наведіть обов'язкові властивості модулю Eden AI.
7. Поясніть призначення та наведіть синтаксис функцій get, map та split.
8. Дайте визначення колекції, простого та складного масиву.

Варіанти завдань

Таблиця 2.6.1. Класифікація витрат за товарами та послугами

№	продукти	напої	кафе	транспорт	одяг	взуття	спорт	подорожі	здоров'я	краса	комунальні платежі
1	+		+		+		+		+		
2		+		+		+		+		+	
3	+	+	+	+	+						
4							+	+	+	+	+
5	+	+		+	+		+				
6		+	+		+	+		+			
7			+	+		+	+		+		
8				+	+		+	+		+	
9					+	+		+	+		+
10					+		+	+		+	+
11				+		+	+		+	+	
12			+		+	+		+	+		
13		+		+	+		+	+			
14	+		+	+		+	+				
15					+		+		+	+	+
16				+		+		+	+	+	
17			+		+		+	+	+		
18		+		+		+	+	+			
19	+		+		+	+	+				

20	+	+	+		+		+				
21		+	+	+		+		+			
22				+	+	+		+		+	
23					+	+	+		+		+
24	+	+							+	+	+
25		+	+					+	+	+	
26			+	+			+	+	+		
27				+	+	+	+	+			
28			+	+	+		+	+			
29		+	+	+				+	+		
30	+	+	+						+	+	
31	+							+	+	+	+
32		+					+	+	+	+	
33			+			+	+	+	+		
34					+	+		+		+	+
35				+	+		+		+	+	
36			+	+		+		+	+		
37		+	+		+		+	+			
38	+	+		+		+	+				
39	+		+	+	+		+				
40		+	+	+	+	+					
41			+	+	+	+	+				
42				+	+	+	+	+			
43					+	+	+	+	+		
44		+		+		+		+		+	
45			+	+	+	+	+				
46				+	+	+	+	+			
47					+	+	+	+	+		
48						+	+	+	+	+	
49					+		+	+	+		+
50	+	+				+				+	+
51			+		+		+		+		+
52						+		+	+	+	+

2.7. Практична робота № 7

Пісочниця Apache Hadoop і команди файлової системи HDFS

Об'єкт – хмарні платформи обробки великих даних. Предмет – пісочниця Apache Hadoop і файлова система HDFS. Мета – ознайомлення з порядком інсталяції пісочниці Apache Hadoop і вивчення за її допомогою команд файлової системи HDFS.

Стислі теоретичні відомості

Загальні відомості про проект Apache Hadoop. Проект Hadoop заснований у 2005 році, у 2008-му набув статус проекту верхнього рівня фонду Apache Software Foundation є вільно розповсюджуваним набором утиліт, бібліотек і фреймворків для розробки і експлуатації розподілених програм, що працюють на кластерах з сотень і тисяч вузлів. Використовується для реалізації пошукових і тематичних механізмів багатьох високонавантажених веб-сайтів, в тому числі, для Yahoo! і Facebook. Розроблений на Java в рамках обчислювальної парадигми MapReduce, згідно з якою додаток поділяється на велику кількість однакових елементарних завдань, здійснених на вузлах кластера та приведених до кінцевого результату. Вважається однією з основоположних технологій Big Data. Протягом 2010 року кілька підпроектів Hadoop: Avro, HBase, Hive, Pig, Zookeeper послідовно стали проектами верхнього рівня фонду Apache, що послужило початком формування екосистеми навколо Hadoop, структура якої наведена на рис. 2.7.1.

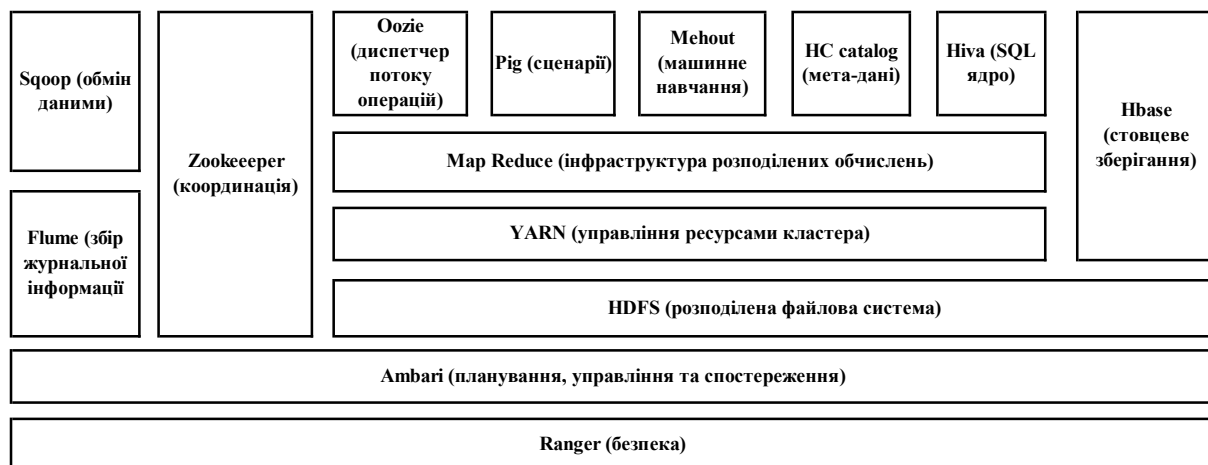


Рис. 2.7.1. Структура екосистеми Hadoop

Станом на 2014 рік, проект складався з чотирьох складових – Hadoop Common (сполучаюче програмне забезпечення – набір інфраструктурних програмних бібліотек і утиліт, використовуваних для інших модулів і споріднених проектів), HDFS (розподілена файлова система), YARN (система для планування завдань і управління кластером) і Hadoop MapReduce (платформа програмування і виконання розподілених MapReduce-обчислень), раніше в Hadoop входив цілий ряд інших проектів, які стали самостійними в

рамках системи проектів Apache Software Foundation.

В аналітичному звіті Big Data Analytics Market Study 2017-го року приводиться діаграма (рис. 2.7.2) кількості інфраструктур Big Data, впроваджених на підприємствах, що представлена в розрізі розмірів підприємств.

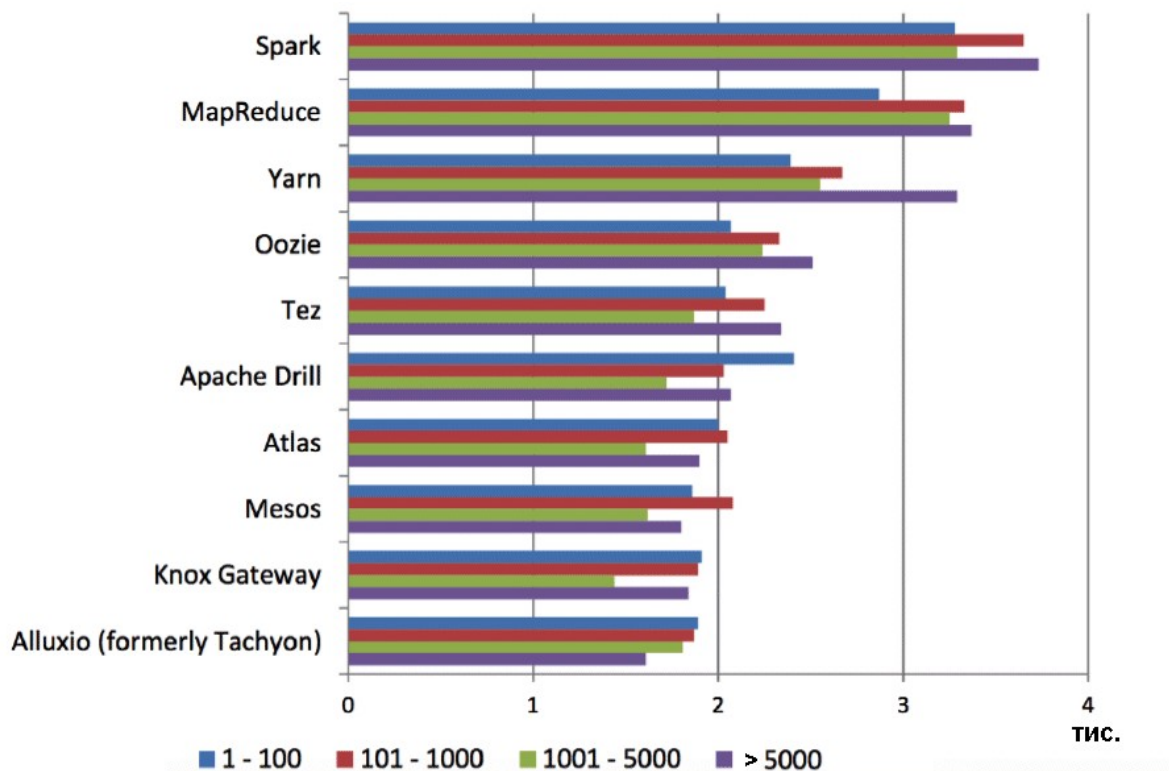


Рис. 2.7.2. Діаграма кількості інфраструктур Big Data, впроваджених на підприємствах станом на 2017 рік

Проект Hadoop має наступні цілі:

- портабельність (кросплатформеність) – можливість інтеграції в проект апаратно-програмного забезпечення різних комп'ютерних платформ.
- надійність, яка досягається резервним копіюванням даних;
- відмовостійкість – виявлення збоїв та автоматичне відновлення нормального функціонування систем;
- горизонтальна масштабованість – можливість кластера за допомогою додавання обладнання масового класу, Commodity hardware;

HDFS (Hadoop Distributed File System) – розподілена файлова система Hadoop для зберігання файлів великих розмірів з можливістю потокового доступу до інформації, поблоково розподіленої за вузлами обчислювального кластера, який може складатися з довільного апаратного забезпечення. HDFS як і будь-яка файлова система є ієрархією каталогів з вкладеними в них підкаталогами і файлами.

HDFS є невід'ємною частиною Hadoop і основою інфраструктури Big Data. Однак, Hadoop підтримує роботу і з іншими розподіленими файловими

системами, зокрема, Amazon S3 і CloudStore. Також деякі дистрибутиви Hadoop, наприклад, MapR, реалізують свою аналогічну розподілену файлову систему MapR File System.

HDFS може використовуватися не тільки для запуску MapReduce-завдань, а й як розподілена файлова система загального призначення, забезпечуючи роботу розподілених СУБД (HBase) і масштабованих систем машинного навчання (Apache Mahout).

Архітектура HDFS представлена на рис. 2.7.3.

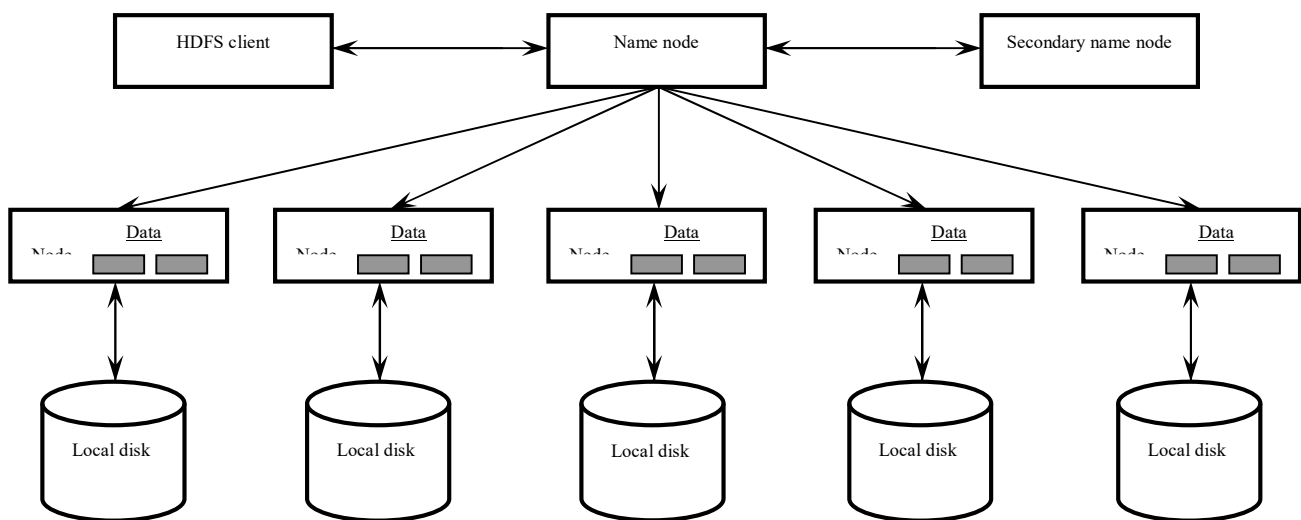


Рис. 2.7.3. Архітектура HDFS

Кластер HDFS включає наступні компоненти.

Керуючий вузол, вузол імен або сервер імен (NameNode) – окремий, єдиний в кластері, сервер з програмним кодом для управління простором імен файлової системи, який зберігає дерево файлів, а також метадані файлів і каталогів. NameNode – обов'язковий компонент кластера HDFS, який відповідає за відкриття і закриття файлів, створення і видалення каталогів, управління доступом з боку зовнішніх клієнтів і відповідність між файлами і блоками, дубльованими (репліційованими) на вузлах даних. Сервер імен розкриває для всіх бажаючих розташування блоків даних на машинах кластера.

Secondary NameNode – вторинний вузол імен, окремий сервер, єдиний в кластері, який копіює образ HDFS і лог транзакцій операцій з файловими блоками в тимчасову папку, застосовує зміни, накопичені в лозі транзакцій до образу HDFS, а також записує його на вузол NameNode і очищає лог транзакцій. Secondary NameNode необхідний для швидкого ручного відновлення NameNode в разі його виходу з ладу.

Вузол або сервер даних (DataNode, Node) – один із множини серверів кластера з програмним кодом, що відповідає за файлові операції і роботу з блоками даних. DataNode є обов'язковим компонентом кластеру HDFS, який відповідає за запис і читання даних, виконання команд від вузла NameNode по створенню, видалення і реплікації блоків, а також періодичну відправку повідомлення про стан (heartbeats) і обробку запитів на читання і запис, що

надходять від клієнтів файлової системи HDFS. З інших вузлів кластера дані проходять до клієнта повз вузла NameNode.

Клієнт (client) – користувач або додаток, який взаємодіє через спеціальний інтерфейс (API – Application Programming Interface) з розподіленою файловою системою. При наявності достатніх прав, клієнту дозволені наступні операції з файлами і каталогами: створення, видалення, читання, запис, перейменування і переміщення. Створюючи файл, клієнт може явно вказати розмір блоку файлу (за замовчуванням 64 Мб) і кількість створюваних реплік (за замовчуванням значення дорівнює 3-ом).

Завдяки реплікації блоків по вузлах даних, розподілена файлова система Hadoop забезпечує високу надійність зберігання даних і швидкість обчислень. Крім того, HDFS властива наступна низка відмінностей:

- великий розмір блоку в порівнянні з іншими файловими системами (> 64MB), оскільки HDFS призначена для зберігання великої кількості величезних (> 10GB) файлів;
- орієнтація на недорогі і, тому не найнадійніші сервера – відмовостійкість всього кластера забезпечується за рахунок реплікації даних;
- віддзеркалення і реплікація здійснюються на рівні кластера, а не на рівні вузлів даних;
- реплікація відбувається в асинхронному режимі – інформація розподіляється по декількох серверах прямо під час завантаження, тому вихід з ладу окремих вузлів даних не спричинить за собою повну втрату даних;
- HDFS оптимізована для поточкових зчитувань файлів, тому застосовувати її для нерегулярних і довільних зчитувань недоцільно;
- клієнти можуть зчитувати і записувати файли HDFS безпосередньо через програмний інтерфейс Java;
- файли пишуться одноразово, що виключає внесення в них будь-яких довільних змін;
- принцип WORM (Write-once and read-many, один раз записати – багато разів прочитати) повністю звільняє систему від блокувань типу «запис-читання». Запис в файл в один час доступна тільки одному процесу, що виключає конфлікти множинної записи.
- HDFS оптимізована під потокову передачу даних;
- стиснення даних і раціональне використання дискового простору дозволило знизити навантаження на канали передачі даних, які найчастіше є вузьким місцем в розподілених середовищах;
- самодіагностика – кожен вузол даних через певні інтервали часу відправляє діагностичні повідомлення вузлу імен, який записує логи операцій над файлами в спеціальний журнал;
- всі метадані сервера імен зберігаються в оперативній пам'яті.

У зв'язку з особливостями архітектури та принципом дії, для HDFS характерні наступні недоліки:

- сервер імен є центральною точкою всього кластера і його відмова спричинить збій системи цілком;
- відсутність повноцінної реплікації Secondary NameNode;
- відсутність можливості дописувати або залишити відкритим для запису файли в HDFS, за рахунок чого в класичному дистрибутиві Apache Hadoop неможливо оновлювати блоки вже записаних даних;
- відсутність підтримки реляційних моделей даних;
- відсутність інструментів для підтримки посилювальної цілісності даних, що не гарантує ідентичність реплік. HDFS перекладає перевірку цілісності даних на клієнтів. При створенні файлу клієнт розраховує контрольні суми кожні 512 байт, які в подальшому зберігаються на сервері імен. При зчитуванні файлу клієнт звертається до даних і контрольних сум. У разі їх невідповідності відбувається звернення до іншої репліки.

Пісочниця Apache Hadoop від Hortonworks Пісочниця Apache Hadoop від Hortonworks є локальним середовищем розробки для ознайомлення з Hadoop, розподіленою файловою системою Hadoop (HDFS) і відправкою завдань, яка дозволяє ознайомитися з екосистемою Hadoop. Розглянемо налаштування на віртуальній машині Apache Hadoop від Hortonworks. Додаткові відомості з цього питання можуть бути отримані за посиланням <https://docs.microsoft.com/ru-ru/azure/hdinsight/hadoop/apache-hadoop-emulator-get-started>.

Для доступу до пісочниці слід встановити оболонку віртуальних машин Oracle VM VirtualBox, яка дозволяє у середовищі встановленої в комп'ютері операційної системи емулювати інші операційні системи та віртуальну машину HDP Sandbox, що містить віртуальний образ операційної системи CentOS зі встановленим інтерфейсом до Hadoop. Інсталятор першого програмного продукту можна отримати за посиланням <https://www.virtualbox.org/wiki/Downloads>. Після встановлення VirtualBox platform packages потрібно також встановити пакет розширення VirtualBox Extension Pack тієї самої ж версії, що і основний. Другий програмний продукт можна отримати за адресою <https://www.cloudera.com/downloads/hortonworks-sandbox/hdp.html>. При цьому слід мати на увазі, що доступні наразі версії HDP Sandbox містяться у файлах великих розмірів: версія 2.5 – 11,5 Гб, 2.6.5 – 15,7 Гб, 3.0.1 – 21,6 Гб. В розгорнутому вигляді вони відповідно потребують: версія 2.5 – 21,8 Гб, 2.6.5 – 62,5 Гб дискового простору.

Образ віртуальної машини HDP Sandbox слід імпортувати у середовище VirtualBox. Для цього слід скористатися пунктом меню *Файл – Імпорт конфігурацій* (Ctrl+I) (рис. 2.7.4) і в відкритійся вікні діалогу вказати файл HDP Sandbox (цей файл має відкритий формат віртуалізації ova).

Далі потрібно налаштувати віртуальну машину, натиснувши на кнопку *Налаштувати* (Ctrl+S). Слід зазначити, що віртуальна машина HDP Sandbox висуває значні вимоги до апаратної частини комп'ютера, на який вона встановлюється. Приведені нижче на рис. 2.7.5, 2.7.6 параметри комп'ютера

слід вважати близькими до мінімальної необхідної конфігурації для встановлення віртуальної машини HDP Sandbox.

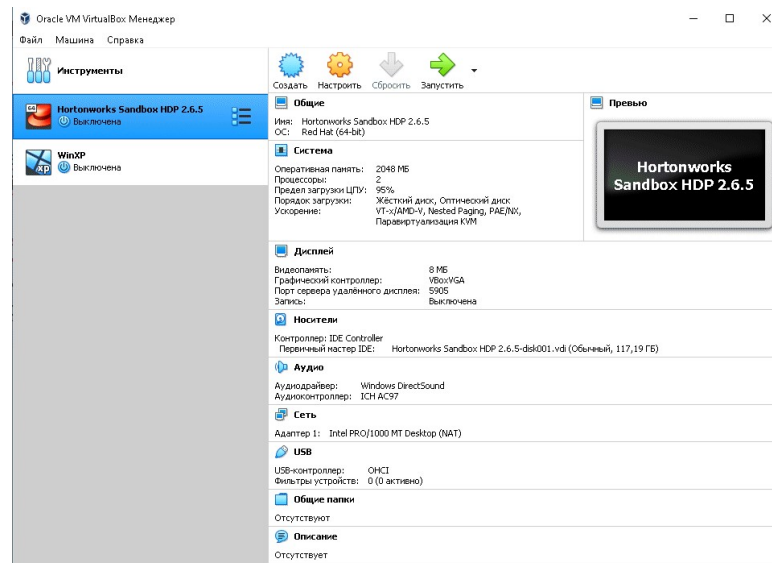


Рис. 2.7.4. Діалогове вікно середовища Oracle VM VirtualBox

Характеристики устройства

Имя устройства	CMZ2804
Процессор	Intel(R) Pentium(R) CPU 4405U @ 2.10GHz 2.11 GHz
Оперативная память	4,00 Гб
Код устройства	CBD4A885-D43B-4CED-8BD7-9FFF407A37A5
Код продукта	00331-20000-00000-AA877
Тип системы	64-разрядная операционная система, процессор x64
Перо и сенсорный ввод	Для этого монитора недоступен ввод с помощью пера и сенсорный ввод

Переименовать этот ПК

Характеристики Windows

Выпуск	Windows 10 Pro
Версия	1909
Дата установки	01.06.2020
Сборка ОС	18363.1379

Рис. 2.7.5. Параметры комп'ютера, на який встановлюється віртуальна машина HDP Sandbox (початок)

Дополнительные параметры дисплея

Выберите дисплей

Выберите дисплей, чтобы просмотреть или изменить его параметры.

Дисплей 1: AIO LCD

Сведения о дисплее

AIO LCD	Дисплей 1: подключен к Intel(R) HD Graphics 510
Разрешение рабочего стола	1920 × 1080
Активное разрешение сигнала	1920 × 1080
Частота обновления (Гц)	60 Гц
Разрядность	8-бит
Цветовой формат	RGB
Цветовое пространство	Стандартный динамический диапазон (SDR)

Свойства видеоадаптера для дисплея 1

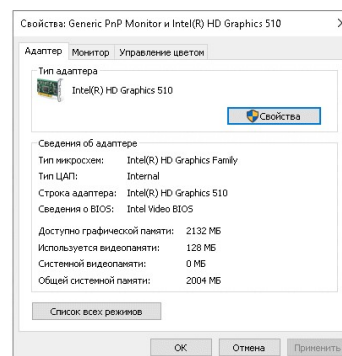


Рис. 2.7.6. Параметры комп'ютера, на який встановлюється віртуальна машина HDP Sandbox (продовження)

Найбільш критичними при налаштуванні є розділи *Система – Материнська плата* и *Система – Процесор*. Відомості про невірні налаштування містяться у спливаючих вікнах, які розташовуються знизу вікна діалогу налаштування та мають вигляд, представлений на рис. 2.7.7, 2.7.8.

Після налаштування віртуальну машину можна запустити кнопкою **Запустити**. Коли віртуальна машина стартує на екрані з'явиться вікно, представлене на рис. 2.7.9.

Для доступу до пісочниці використовується протокол SSH, широко використовуваний в операційних системах сімейства Linux для віддаленого доступу між машинами. При доступі до віддаленого комп'ютера Linux з Windows в останній повинен бути встановлений клієнт SSH. Але для версій 2.5 та 2.6.5 можна використовувати протокол SSH, наданий віртуальною машиною за адресою <http://localhost:4200/>.

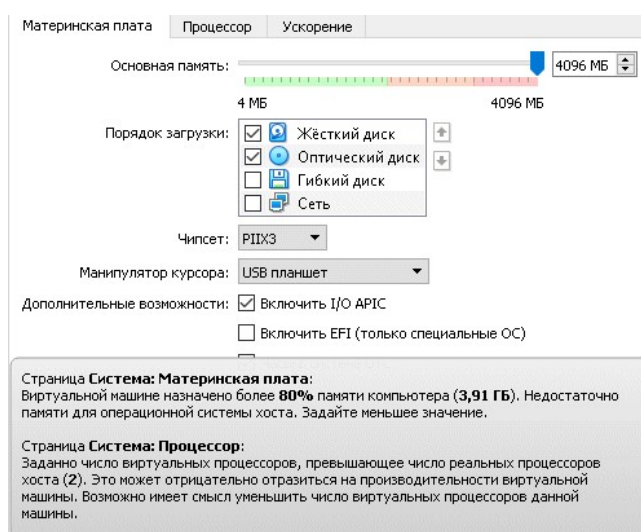


Рис. 2.7.7. Відомості про невірні налаштування комп'ютера, на який встановлюється віртуальна машина HDP Sandbox (початок)

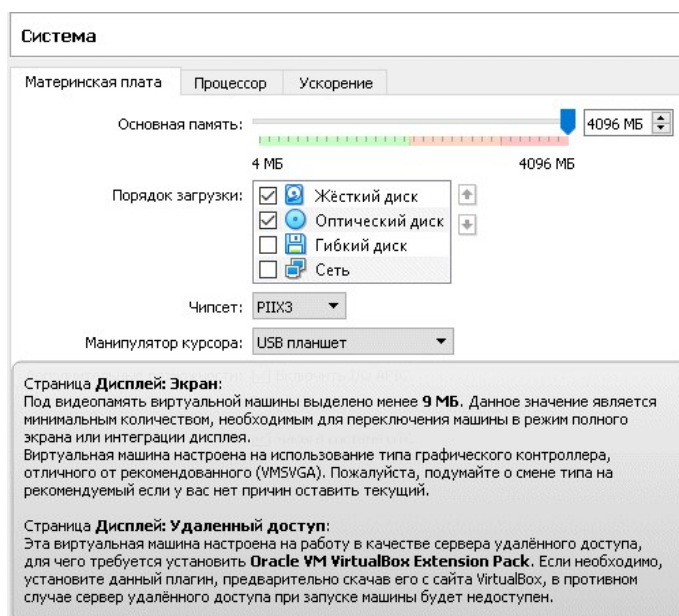


Рис. 2.7.8. Відомості про невірні налаштування комп'ютера, на який встановлюється віртуальна машина HDP Sandbox (продовження)

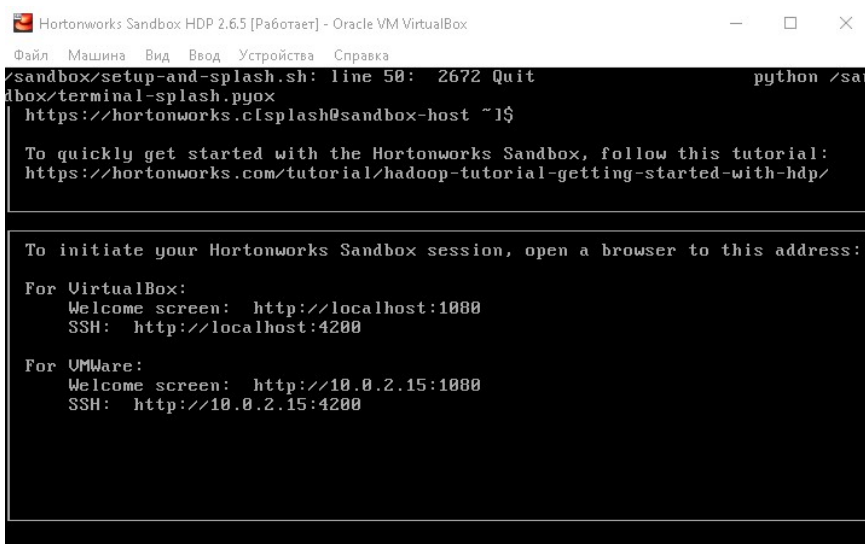


Рис. 2.7.9. Ініціалізація роботи з пісочницею Apache Hadoop

При першому підключенні до пісочниці за допомогою SSH треба ввести ім'я користувача `root` та пароль за замовченням `hadoop`, після чого буде запропоновано змінити пароль для облікового запису `root`, у відповідь на що треба ввести новий пароль, який подалі буде використовуватися при вході. Після цього відкривається доступ до командного рядка файлової системи HDFS.

Формат команд HDFS (shell команд) має наступний вигляд:

Версія 1 `hadoop fs -command -<options> <URI>`

Версія 2 `hdfs dfs -command -<options> <URI>`

Далі використовується друга версія формату.

Тут:

- `hdfs` – утиліта роботи з HDFS;
- `dfs` – спеціальний параметр, який позначає, що ми працюватимемо безпосередньо з розподіленою файловою системою. Можуть бути вказані інші параметри;
- `<command>` – команда, яку ми хочемо застосувати до файлової системи;
- `<options>` – опції команди (можуть відсутні);
- `<URI>` – шлях у вигляді URI-схеми. URI (Uniform Resource Identifier) – уніфікований ідентифікатор ресурсу – символічний рядок, що дозволяє ідентифікувати будь-який ресурс: документ, зображення, файл, службу, скриньку електронної пошти тощо, насамперед, йдеться про ресурси мережі Інтернет.

Деякі команди файлової системи HDFS в форматі версії 2 наведено в табл. 1.

Таблиця 2.7.1. Команди файлової системи HDFS

Команда	Дія, яка виконується командою
Команди виведення списків	
<code>hdfs dfs</code>	Виведення повного списку команд HDFS
<code>hdfs dfs -ls /</code>	Виведення списку всіх підкаталогів/файлів в кореневому каталозі HDFS

<code>hdfs dfs -ls /DirectoryName</code>	Виведення списку всіх підкаталогів/файлів в локальному каталозі або каталозі HDFS
<code>hdfs dfs -ls -R /DirectoryName</code>	Рекурсивне виведення списку всіх підкаталогів/файлів локальному каталозі або каталозі HDFS (виведення змісту всього дерева підкаталогів для якого зазначений каталог є вершиною)
Команди управління каталогами і файлами	
<code>hdfs dfs -rmdir /DirectoryName</code>	Вилучення локального каталогу або каталогу HDFS
<code>hdfs dfs -mkdir /DirectoryName</code>	Створення локального каталогу або каталогу HDFS
<code>hdfs dfs -chmod code /DirectoryName[/FileName]</code>	Зміна прав доступу до каталогу/файлу. При значенні <code>code=777</code> надається повний доступ (читання <code>r</code> , запис <code>w</code> , запуск <code>x</code>)
<code>hdfs dfs -cp [-f] [-p] /DirectoryName1/FileName1 /DirectoryName2/ FileName2</code>	Копіювання файлу з одного локального каталогу або каталогу HDFS в інший каталог. При копіюванні ім'я файлу може бути змінено. Необов'язковий параметр <code>f</code> дозволяє перезаписувати файл з тим же ім'ям у каталогу призначення, якщо він вже існує, параметр <code>p</code> дозволяє зберігати режим доступу до файлу
<code>hdfs dfs -mv /DirectoryName1/FileName /DirectoryName2</code>	Переміщення файлу з одного локального каталогу або каталогу HDFS в інший каталог.
<code>hdfs dfs -touchz /DirectoryName/FileName</code>	Створення порожнього файлу в заданому локальному каталозі або каталозі HDFS
<code>hdfs dfs -put /localDirectoryName/localFileName /hdfsDirectoryName/hdfsFileName</code>	Копіювання файлу з каталогу локальної файлової системи в каталог HDFS. При копіюванні ім'я файлу може бути змінено.
<code>hdfs dfs -get /hdfsDirectoryName/hdfsFileName /localDirectoryName/localFileName</code>	Копіювання файлу з каталогу HDFS в каталог локальної файлової системи. При копіюванні ім'я файлу може бути змінено.
<code>hdfs dfs -cat /DirectoryName/FileName</code>	Виведення змісту локального файлу або файлу HDFS
<code>hdfs dfs -text /DirectoryName/FileName [head -n кількість рядків]</code>	Виведення змісту локального файлу або файлу HDFS. Якщо файл є архівом, то відбувається його розархівування. Необов'язкова опція, яка записана у дужках, може бути використана для виведення заданої кількості перших рядків файлу
Адміністративні команди	
<code>hadoop version</code>	Виведення поточної версії Hadoop
<code>hdfs fsck /</code>	Перевірка стану HDFS
<code>hdfs dfs -df /</code>	Виведення обсягу загального, вільного і використовуваного простору файлової системи

Кириличні імена каталогів та файлів підтримуються.

HDP Sandbox оперує двома файловими системами: локальною і розподіленою, при заданні шляху до каталогу або файлу в першій з них власно шляху повинна передувати так звана схема `file://`. Так, наприклад команди перегляду змісту каталогів верхнього рівня в локальній і розподіленій файловій системах будуть відповідно виглядати як

```
hdfs dfs -ls file://
hdfs dfs -ls /
```

Обидві системи доступні із середовища гостьової операційної системи, якою є Linux-подібна система Cent OS. Файловий обмін між зазначеними системами можливий за допомогою команд HDFS get та put. Для завантаження файлів з зовнішнього ресурсу в локальну файлову структуру може бути використана команда

Linux wget [--no-check-certificate] URL ресурса

при цьому завантаження відбувається в каталог file:///root. Необов'язкова опція, яка записана у дужках, може бути використана якщо при завантаженні виникає помилка, яка обумовлена неможливістю перевірки сертифікату, з якого відбувається завантаження.

В приведеному на рис. 2.7.10 прикладі послідовно здійснюється:

- перегляд змісту кореневого каталогу;
- створення каталогу student;
- надання дозволу на повний доступ до каталогу student;
- створення підкаталогу documents каталогу student;
- повторний перегляд змісту кореневого каталогу;
- перегляд змісту каталогу student.

```
Last login: Fri Jan 10 17:00:43 2025 from 172.18.0.2
[root@sandbox-hdp ~]# hdfs dfs -ls /
Found 12 items
drwxr-xr-x - root hdfs 0 2025-01-10 16:56 /anton
drwxrwxrwx - yarn hadoop 0 2018-06-18 15:18 /app-logs
drwxr-xr-x - hdfs hdfs 0 2018-06-18 16:13 /apps
drwxr-xr-x - yarn hadoop 0 2018-06-18 14:52 /ats
drwxr-xr-x - hdfs hdfs 0 2018-06-18 14:52 /hdp
drwx----- - livy hdfs 0 2018-06-18 15:11 /livy2-recover
y
drwxr-xr-x - mapred hdfs 0 2018-06-18 14:52 /mapred
drwxrwxrwx - mapred hadoop 0 2018-06-18 14:52 /mr-history
drwxr-xr-x - hdfs hdfs 0 2018-06-18 15:59 /ranger
drwxrwxrwx - spark hadoop 0 2025-01-10 17:07 /spark2-histor
y
drwxrwxrwx - hdfs hdfs 0 2018-06-18 16:06 /tmp
drwxr-xr-x - hdfs hdfs 0 2018-06-18 16:08 /user
[root@sandbox-hdp ~]# hdfs dfs -mkdir /student
[root@sandbox-hdp ~]# hdfs dfs -chmod 777 /student
[root@sandbox-hdp ~]# hdfs dfs -mkdir /student/documents
[root@sandbox-hdp ~]# hdfs dfs -ls /
Found 13 items
drwxr-xr-x - root hdfs 0 2025-01-10 16:56 /anton
drwxrwxrwx - yarn hadoop 0 2018-06-18 15:18 /app-logs
drwxr-xr-x - hdfs hdfs 0 2018-06-18 16:13 /apps
drwxr-xr-x - yarn hadoop 0 2018-06-18 14:52 /ats
drwxr-xr-x - hdfs hdfs 0 2018-06-18 14:52 /hdp
drwx----- - livy hdfs 0 2018-06-18 15:11 /livy2-recover
y
drwxr-xr-x - mapred hdfs 0 2018-06-18 14:52 /mapred
drwxrwxrwx - mapred hadoop 0 2018-06-18 14:52 /mr-history
drwxr-xr-x - hdfs hdfs 0 2018-06-18 15:59 /ranger
drwxrwxrwx - spark hadoop 0 2025-01-16 11:10 /spark2-histor
y
drwxrwxrwx - root hdfs 0 2025-01-16 11:09 /student
drwxrwxrwx - hdfs hdfs 0 2018-06-18 16:06 /tmp
drwxr-xr-x - hdfs hdfs 0 2018-06-18 16:08 /user
[root@sandbox-hdp ~]# hdfs dfs -ls /student
Found 1 items
drwxr-xr-x - root hdfs 0 2025-01-16 11:09 /student/documents
[root@sandbox-hdp ~]# █
```

Рис. 10 Приклад роботи з командами файлової системи HDFS

Постановка задачі

Дано

Інсталятор віртуальної машини HDP Sandbox.

Індивідуальна адреса архіву даних у репозиторії машинного навчання.

Потрібно

Інсталювати віртуальну машину HDP Sandbox на персональному комп'ютері.

Провести маніпуляції з каталогами та файлів в локальній і розподіленій файлових системах HDP Sandbox.

Рекомендації щодо виконання роботи

1. Встановити на персональному комп'ютері віртуальну машину HDP Sandbox.

2. Вивести обсяги загального, вільного і використовуваного простору файлової системи.

3. Отримати зміст каталогів верхнього рівня в розподіленій і локальній файлової структурах, порівняти їх.

4. Створити в розподіленій і локальній файлової структурах вкладені каталоги, верхній з яких має назву у вигляді прізвища здобувача вищої освіти латиницею, а нижній – кирилицею.

5. Надати права повного доступу до створених каталогів.

6. Завантажити до локальної файлової структури архів з репозиторію машинного навчання згідно варіанту завдання (табл. 2). Переконавшись, що каталог file:///root містить завантажений архів і уточнити його ім'я. Перемістити його до створеного каталогу нижнього рівня локальної файлової структури.

7. Вивести перші 5 рядків архіву за допомогою команди text.

8. В створеному каталогі розподіленої файлової структури створити порожній файл який має назву у вигляді імені здобувача вищої освіти кирилицею.

9. Скопіювати завантажений архів в створений каталог розподіленої файлової структури, а створений файл – у відповідний каталог локальної файлової структури.

10. Рекурсивно вивести зміст створених гілок розподіленої і локальної файлової структур.

Вимоги до звіту

Звіт по роботі повинен містити:

1. Назву дисципліни та лабораторної роботи.

2. Прізвище, ім'я та по батькові здобувача вищої освіти, код групи.

3. Номер варіанту завдання і адреса архіву даних у репозиторії машинного навчання яка йому відповідає.

4. Знімки зображень екрану, які виникають при встановленні і ініціалізації на персональному комп'ютері віртуальної машини HDP Sandbox з підписами (Рис. 4 – 9) та знімки зображень екрану, які виникають при виконанні п. 2, 3, 6, 7, 10 рекомендацій щодо виконання роботи з підписами.

5. Висновки.

Контрольні питання і завдання

1. Назвіть основні складові екосистеми Hadoop.
2. Назвіть і охарактеризуйте компоненти кластеру HDFS.
3. Які основні переваги і недоліки притаманні файловій системі HDFS?
4. Наведіть загальні формати двох версій для команд HDFS.
5. Наведіть приклад будь-якої команди HDFS для каталогу або файлу в локальній і її аналог в розподіленій файловій системах.
6. Назвіть основні групи команд файлової системи HDFS.

Таблиця 2.7.2. Індивідуальні адреси архівів даних у репозиторії машинного навчання

Спільний початок всіх адрес <https://archive.ics.uci.edu/static/public/>

№	Продовження адреси
1	484/travel+reviews.zip
2	697/predict+students+dropout+and+academic+success.zip
3	397/las+vegas+strip.zip
4	468/online+shoppers+purchasing+intention+dataset.zip
5	519/heart+failure+clinical+records.zip
6	292/wholesale+customers.zip
7	560/seoul+bike+sharing+demand.zip
8	601/ai4i+2020+predictive+maintenance+dataset.zip
9	236/seeds.zip
10	267/banknote+authentication.zip
11	856/higher+education+students+performance+evaluation.zip
12	374/appliances+energy+prediction.zip
13	597/productivity+prediction+of+garment+employees.zip
14	529/early+stage+diabetes+risk+prediction+dataset.zip
15	936/national+poll+on+healthy+aging+(npha).zip
16	863/maternal+health+risk.zip
17	967/phiusiil+phishing+url+dataset.zip
18	1025/turkish+crowdfunding+startups.zip
19	563/iranian+churn+dataset.zip
20	878/cirrhosis+patient+survival+prediction+dataset-1.zip
21	225/ilpd+indian+liver+patient+dataset.zip
22	383/cervical+cancer+risk+factors.zip
23	542/internet+firewall+data.zip
24	857/risk+factor+prediction+of+chronic+kidney+disease.zip
25	571/hcv+data.zip
26	572/taiwanese+bankruptcy+prediction.zip
27	864/room+occupancy+estimation.zip
28	915/differentiated+thyroid+cancer+recurrence.zip
29	244/fertility.zip
30	837/product+classification+and+clustering.zip
31	547/algerian+forest+fires+dataset.zip

32	603/in+vehicle+coupon+recommendation.zip
33	396/sales+transactions+dataset+weekly.zip
34	851/steel+industry+energy+consumption.zip
35	409/daily+demand+forecasting+orders.zip
36	760/multivariate+gait+data.zip
37	849/power+consumption+of+tetouan+city.zip
38	488/facebook+live+sellers+in+thailand.zip
39	591/gender+by+name.zip
40	485/tarvel+review+ratings.zip
41	911/recipe+reviews+and+user+feedback+dataset.zip
42	537/cervical+cancer+behavior+risk.zip
43	211/communities+and+crime+unnormalized.zip
44	386/fma+a+dataset+for+music+analysis.zip
45	498/incident+management+process+enriched+event+log.zip
46	229/skin+segmentation.zip
47	722/naticusdroid+android+permissions+dataset.zip
48	567/covid+19+surveillance.zip
49	713/auction+verification.zip
50	482/parking+birmingham.zip

3. КРИТЕРІЇ ОЦІНЮВАННЯ

Навчальні досягнення здобувачів вищої освіти за результатами вивчення курсу оцінюватимуться за шкалою табл. 3.1.

Таблиця 3.1. Шкала оцінювання навчальних досягнень здобувачів вищої освіти

Рейтингова шкала	Інституційна шкала
90–100	відмінно
74–89	добре
60–73	задовільно
0–59	незадовільно

Здобувач вищої освіти може отримати підсумкову оцінку з навчальної дисципліни на підставі оцінювання знань та практичних навичок за умови, якщо набрана сумарна кількість балів зі складових оцінювання становитиме не менше 60 балів.

Успішність складається з оцінок за теоретичну частину курсу, практичні роботи та самостійну роботу здобувача вищої освіти з підготовки до практичних робіт. Отримані бали додаються один до одного і є підсумковою оцінкою за вивчення навчальної дисципліни. Максимально здобувач може набрати 100 балів.

Максимальне оцінювання навчальних досягнень здобувачів вищої освіти за видами занять наведено в табл. 3.2.

Таблиця 3.2. Максимальне оцінювання навчальних досягнень здобувачів вищої освіти за видами занять

Теоретична частина	Практичні роботи	Самостійна робота з підготовки до практичних робіт	Разом
37	56	7	100

Теоретична частина оцінюється за результатами online тесту, який містить 37 запитань.

Практичні роботи приймаються за звітами і контрольними запитаннями до кожної з робіт.

Критерії оцінювання теоретичної частини курсу

Для кожного запитання тесту потрібно вибрати єдину правильну відповідь. Кількість варіантів відповіді в залежності від характеру запитання може складати від 3-х до 9-ти. За правильну відповідь нараховується 1 бал. У разі неправильної відповіді на запитання, здобувач отримує за нього 0 балів.

Критерії оцінювання практичних робіт

За кожну практичну роботу здобувач може отримати до 8 балів (усього 56

балів), а саме:

8 балів: програма чисельних розрахунків правильно функціонує, супроводжується достатньою кількістю коментарів, звіт з роботи оформлений згідно з методичними рекомендаціями і містить повні, інформативні та обґрунтовані висновки.

6 балів: програма чисельних розрахунків правильно функціонує, але містить незначні помилки, які суттєво не впливають на отримувані результати, або кількість коментарів недостатня, або звіт з роботи оформлений із незначними відхиленнями від методичних рекомендацій або висновки не повністю задовольняють вимогам повноти, інформативності та обґрунтованості.

4 бали: програма чисельних розрахунків правильно функціонує, але містить незначні помилки, які суттєво не впливають на отримувані результати та має місце одна з наступних вад: або кількість коментарів недостатня, або звіт з роботи оформлений із незначними відхиленнями від методичних рекомендацій або висновки не повністю задовольняють вимогам повноти, інформативності та обґрунтованості.

2 бали: програма чисельних розрахунків функціонує, але містить помилки, які суттєво впливають на отримувані результати, або коментарі відсутні, або звіт з роботи оформлений із суттєвими відхиленнями від методичних рекомендацій, або висновки не задовольняють вимогам повноти, інформативності та обґрунтованості.

0 балів: програма чисельних розрахунків не функціонує, або звіт з роботи відсутній, або не містить висновків, або висновки протирічають фактично отриманим результатам.

У разі отримання позитивної оцінки здобувач вищої освіти може підвищити її шляхом очної співбесіди з викладачем. При цьому з кожної практичної роботи здобувач вищої освіти отримує 3 запитання з переліку контрольних запитань. За кожну правильну відповідь на запропоновані запитання нараховується 1 бал.

Критерії оцінювання самостійної роботи

За підготовку вхідних даних згідно з варіантом завдання до кожної практичної роботи здобувач отримує:

1 бал (усього до 7 балів).

0 балів: підготовку вхідних даних не виконано, дані розраховані невірно або не відповідають варіанту завдання.

СПИСОК ВИКОРИСТАНОЇ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Юрчишин В.Я. Хмарні та Грід-технології: конспект лекцій [Електронний ресурс] : навч. посіб. для студ. спец. 121 «Інженерія програмного забезпечення» / В. Я. Юрчишин – Київ: КПІ ім. Ігоря Сікорського, 2019, 264 с.
2. Zgurovsky M.Z. Big Data: Conceptual Analysis and Applications. [Текст] / M.Z. Zgurovsky, Y.P. Zaychenko // Springer, 2021, 298 p.
3. Akerkar R. Models of Computation for Big Data [Текст] / R. Akerkar // Springer, 2018, 110 p.
4. Ghavami P. Big Data Governance: Modern Data Management Principles for Hadoop, NoSQL & Big Data Analytics [Текст] / P. Ghavami // CreateSpace Independent Publishing Platform, 2016, 204 p.
5. Кожевников А.В. Створення відкритої нейронної мережі бінарного класифікатора засобами системи Mathcad/ А.В. Кожевников, І.М. Удовик // Матеріали XVI міжнародної конференції "Проблеми використання інформаційних технологій у сфері освіти, науки та промисловості" – Дніпро: НТУ «Дніпровська політехніка», 2021. – С. 27–32.
6. Кожевников А.В. Створення відкритої нейронної мережі предиктора лінійного часового ряду засобами системи Mathcad/ А.В. Кожевников, І.М. Удовик // Матеріали XVI міжнародної конференції "Проблеми використання інформаційних технологій у сфері освіти, науки та промисловості" – Дніпро: НТУ «Дніпровська політехніка», 2021. – С. 45–49.
7. Мартинов Є. Український національний ґрід – учасник міжнародного об'єднання NORDUGRID / Є. Мартинов, О. Смирнова // Вісник НАН України. – Київ, 2011. № 12. – С. 30 – 35.
8. Згуровський М.З. Створення національної grid-інфраструктури для забезпечення наукових досліджень. [Електрон. ресурс]. Режим доступу: <https://www.ite.kspu.edu/index.php/ite/article/download/594/604/> (дата звернення: 20.08.2024).
9. UCI Machine Learning Repository. [Електрон. ресурс]. Режим доступу: <http://archive.ics.uci.edu/ml/index.php/> (дата звернення: 20.08.2024).
10. Make – Automation Software [Електрон. ресурс]. Режим доступу: <http://make.com> (дата звернення: 20.08.2024).
11. Eden AI [Електрон. ресурс]. Режим доступу: <http://app.edenai.run> (дата звернення: 20.08.2024).
12. Маркова С. В. Історія України: навч.-метод. посіб. / С. В. Маркова, В. О. Парандій. – Київ : МАУП, 2007. – 266 с.
13. Дякон В.М. Основи фізики: навч. посіб. / В.М. Дякон, Л.Є. Ковальов, М.Н. Миколайчук . – Бровари: «АНФ ГРУП», 2015. – 192 с.

Програма реалізації класифікатора даних на основі багат шарового перцептрону мовою Python

```
# Імпорт бібліотеки mglearn надає можливість додаткової графічної інтерпретації
# результатів аналізу даних. Бібліотека mglearn не входить до
# середовища Anaconda за замовченням, тому її заздалегідь слід встановити командою
# з командного рядка pip install mglearn, або додати в програму рядок
#!pip install mglearn
# Імпорт бібліотеки mglearn та модуля pyplot бібліотеки matplotlib
import mglearn
import matplotlib.pyplot as plt
# Імпорт класів і функцій з модулів бібліотеки Scikit-Learn
from sklearn.model_selection import train_test_split
from sklearn.datasets import make_blobs
from sklearn.neural_network import MLPClassifier
from sklearn.datasets import make_moons
# Функція make_moons генерує набори точок такі, що в проекції на площину Оху
# виходять два півмісяці, вкладені один в одний. Параметр noise є
# середньоквадратичним відхиленням щільності ймовірності гаусового шуму. Чим він
# менше, тим менше точки відхиляються від значень, які визначаються функціями
# напівкіл. Параметр random_state повинен бути числом – початковим значенням
# генератора псевдовипадкових чисел. При використанні функції без вказівки
# параметру random_state кожного разу при запуску програми будується новий
# випадковий розподіл вхідного набору даних. Якщо ж параметр встановлено, при
# чому не важливо яке конкретно значення, розподіл буде збігатися з попереднім. Це
# корисно, якщо потрібні відтворювані результати, наприклад, при тестуванні, або
# документуванні. Вихідними параметрами є X – матриця ознак варіант – точок (їх
# горизонтальні та вертикальні координати) та y – вектор належності варіант до
# певного класу. Функція make_moons викликає функцію make_blobs, яка генерує
# набір даних. BLOB (Binary Large Object). Її парметрами є кількість ознак, якими
# характеризуються дані та кількість кластерів, до яких вони належать
X, y = make_moons(n_samples=100, noise=0.25, random_state=3)
# Функція train_test_split поділяє вхідний набір даних на навчальну і контрольну
# вибірки. Для якісного навчання імовірнісні розподіли навчальної і контрольної
# вибірок повинні бути близькі. Функція за замовчуванням автоматично перемішує
# дані, що забезпечує близькість вказаних розподілів. Параметр stratify встановлює
# ознаку даних за якою здійснюється контроль відповідності розподілів навчальної і
# контрольної вибірки за класами. У приведеному нижче випадку здійснюється
# стратифікація відповідно розподілу за ознакою y. Можна визначити або train_size
# або test_size – відсотки поділу об'ємів навчальної та контрольної (тестової) вибірок,
# але обидва вони не потрібні. Якщо явно встановлюються обидва параметри, вони
# повинні складати в сумі 1.
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, random_state=42)
```

```

#Імпорт бібліотеки numpy
import numpy as np
#Виведення розподілів навчальної і контрольної вибірки за класами.
print(f"Кількість рядків в y_train за класами: {np.bincount(y_train)}")
print(f"Кількість рядків в y_test за класами: {np.bincount(y_test)}")
#Клас MLPClassifier з використанням методу fit дозволяє здійснювати побудову і
#навчання нейронної мережі. За замовченням використовується функція активації
#ReLU. Створення змінної mlp як екземпляра класа MLPClassifier.
mlp = MLPClassifier(solver='lbfgs', max_iter=100000, random_state=0).fit(X_train, y_train)
#Алгоритм, який використовується для обчислення ваг синапсів задається за
#допомогою параметра solver. Алгоритм 'adam', використовуваний за замовчанням,
#вельми чутливий до масштабування даних (тому важливо перед класифікацією
#стандартизувати дані так, щоб кожна ознака мала середнє 0 і дисперсію 1). Інший
#алгоритм 'lbfgs' позбавлений цього недоліку, але є більш повільним і може зайняти
#багато часу в разі складних моделей або великих масивів даних. Параметр max_iter
#встановлює кількість циклів навчання.
#Візуалізація межі розділу класів, що побудована нейронною мережею на основі
#навчальної вибірки.
mglearn.plots.plot_2d_separator(mlp, X_train, fill=True, alpha=.3)
#Побудування крапок даних навчальної (або контрольної) вибірок
mglearn.discrete_scatter(X_train[:, 0], X_train[:, 1], y_train)
#mglearn.discrete_scatter(X_test[:, 0], X_test[:, 1], y_test)
#Виведення легенд осей
plt.xlabel("Ознака 0")
plt.ylabel("Ознака 1")

```

ЗМІСТ

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ	3
2. ПОСТАНОВКА ЗАДАЧ ПРАКТИЧНИХ РОБІТ, МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДЛЯ ЇХ ВИКОНАННЯ ТА ІНДИВІДУАЛЬНІ ЗАВДАННЯ	4
2.1. Практична робота № 1 Створення навчальних тестів в додатку Forms хмарного сервісу Microsoft 365	4
2.2. Практична робота № 2 Інсталяція гостьової операційної системи у вигляді віртуальної машини Oracle VM Virtual Box	15
2.3. Практична робота № 3 Організація взаємодії хостової та гостьової операційних систем, налаштування мережевих адаптерів у Oracle VM Virtual Box	30
2.4. Практична робота № 4 Ознайомлення з методами класифікації даних за допомогою нейронних мереж	38
2.5. Практична робота № 5 Інтеграція інформаційних сервісів за допомогою хмарної платформи Make	50
2.6. Практична робота № 6 Розробка сценарію хмарної платформи Make з використанням засобів штучного інтелекту Eden AI	63
2.7. Практична робота № 7 Пісочниця Apache Hadoop і команди файлової системи HDFS	73 86
3. КРИТЕРІЇ ОЦІНЮВАННЯ	88
СПИСОК ВИКОРИСТАНОЇ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	
Додаток. Програма реалізації класифікатора даних на основі багатозарового перцептронну мовою Python	89

Навчальне видання

Заболотний Костянтин Сергійович

Кожевников Антон Вячеславович

Нікулін Сергій Леонідович

Мещеряков Леонід Іванович

ТЕХНОЛОГІЇ ХМАРНИХ ОБЧИСЛЕНЬ

Методичні рекомендації до виконання практичних робіт
для здобувачів ступеня бакалавра
спеціальностей галузі знань 12 (F) Інформаційні системи і технології

Видано в авторській редакції

Електронний ресурс.

Підписано до видання 28.04.2025. Авт. арк. 6,8.

Національний технічний університет «Дніпровська політехніка».
49005, м. Дніпро, просп. Дмитра Яворницького, 19.