ЛАБОРАТОРНА РОБОТА 4

Тема: Робота з файловою системою.

Мета: навчитися використовувати можливості Java по роботі з файловою системою для зберігання даних програми на диску, а також для маніпулювання файлами і директоріями. Ознайомитися з технологією серіалізації об’єктів.

Теоретичні питання

1. Потоки введення / виведення. Класи потоків InputStream, OutputStream, Reader і Writer.
2. Буферизація введення / виведення.
3. Робота з бінарними файлами через DataInputStream і DataOutputStream.
4. Робота з текстовими файлами за допомогою PrintStream і Scanner.
5. Серіалізація об’єктів. Інтерфейс Serializable. Потоки об’єктів.
6. Робота з файлами і директоріями. Класи File і Files.

Контрольні питання

1. Коротко опишіть призначення класів-нащадків InputStream і OutputStream.
2. У чому відмінність між класами InputStream, OutputStream і класами Reader, Writer?
3. Як відкрити текстовий файл для додавання записів у кінець файла?
4. Навіщо застосовується буферизація введення / виведення? Які класи її забезпечують? Наведіть приклад.
5. Навіщо застосовується метод flush()?
6. Для чого застосовується серіалізація? Які методи включає інтерфейс Serializable і як зробити клас здатним до серіалізації?
7. У чому відмінність класів DataInputStream і DataOutputStream від класів ObjectInputStream і ObjectOutputStream?
8. Яку інформацію про фото можна отримати з об’єкта File?
9. Які операції над файлами і директоріями реалізує клас File?
10. Якими способами можна скопіювати файл у Java?

Індивідуальні завдання

Загальне завдання

1. Для класів із завдання 3 лабораторної роботи 3 (для базового і для похідного) реалізуйте механізм серіалізації.

2. Напишіть програму, яка:
   - читає набір об’єктів (базового і похідного класу) з CSV файла і поміщає їх до масиву або списку;
   - серіалізує цей масив (або список) у бінарний файл;
   - десеріалізує масив (або список) із бінарного файла.

   Імена файлів задаються у вигляді параметрів при запуску програми.

   Програма має коректно обробляти помилкові дані у вхідному файлі!

Варіант 1

3. Дано файл, що містить список директорій (кожен рядок файла містить повний шлях до певної директорії). Напишіть програму, яка знаходить у цьому списку найбільшу за об’ємом директорію та виводить її вміст на екран.

4*. (*Для програмістів*) Напишіть програму, яка вводить назву директорії і маску імені файла (наприклад, масці «*a.?xt» відповідають імена файлів a.txt і bba.xxt), а потім перейменовує всі файли директорії, імена яких відповідають масці. При перейменуванні розширення файла зберігається, а ім’я змінюється на номер файла в алфавітному порядку.

Варіант 2

3. Напишіть програму, яка вводить ім’я директорії та підраховує в цій директорії кількість файлів кожного типу. Тип файла визначається за його розширенням (наприклад, .java, .class, .txt і т. д.).

4*. (*Для програмістів*) Напишіть програму, яка вводить імена двох директорій і регулярний вираз, а потім копіює з першої директорії до другої всі файли, імена яких відповідають регулярному виразу. Програма повинна обробляти всі вкладені директорії. При копіюванні структура вкладених директорій має зберігатися.

Варіант 3

3. Напишіть програму, яка перейменовує всі файли з розширенням .txt із заданої директорії, даючи їм імена 1.txt, 2.txt, 3.txt і так далі по порядку. Виведіть старі імена файлів на екран і в текстовий файл.

4*. (*Для програмістів*) Напишіть програму, яка порівнює вміст двох заданих директорій і виводить список файлів, які відрізняються. Для порівняння файлів використовуйте ім’я, розмір і дату останньої зміни файла. Програма має обробляти всі вкладені директорії.

Варіант 4

3. Напишіть програму, яка копіює з першої заданої директорії до другої всі файли, змінені за останні 3 дні. Виведіть імена файлів, що копіюються, на екран і в текстовий файл.

4*. (*Для програмістів*) Напишіть програму, яка шукає задану фразу в усіх файлах директорії. Для кожної знайденої фрази виведіть назву файла, номер рядка та позицію в рядку. Програма має обробляти всі вкладені директорії. Виведені дані мають бути відсортовані за повним ім’ям файла.

Варіант 5

3. Напишіть програму, яка для заданої директорії знаходить максимальну глибину вкладеності піддиректорій. Виведіть імена всіх піддиректорій на екран і в текстовий файл.

4*. (*Для програмістів*) Напишіть програму, яка для заданої директорії визначає, яку частину загального обсягу (у відсотках) займають файли

кожного типу. Тип файла визначається його розширенням. Для зберігання даних використовуйте колекцію HashMap.

Варіант 6

3. Напишіть програму, яка для заданої директорії визначає кількість файлів з урахуванням вкладених директорій. Виведіть імена всіх цих файлів на екран і в текстовий файл.
4*. (*Для програмістів*) Напишіть програму, яка шукає адреси електронної пошти в усіх текстових файлах заданої директорії і виводить усі знайдені адреси на екран. При виведенні адреси відсортуйте за ім'ям домену.

   Вказівка. Визначити, чи є файл текстовим, можна за його розширенням; список допустимих розширень текстових файлів (наприклад, txt, htm, html, csv, java, ...) програма повинна читати з окремого файла.

Варіант 7

3. Напишіть програму, яка знаходить кількість рядків у кожному .txt-файлі заданої директорії. Ім'я директорії вводити з клавіатури. Результати виведіть на екран і в текстовий файл.
4*. (*Для програмістів*) Напишіть програму, яка для заданої директорії видаляє директорії найнижчого рівня вкладеності, а файли з них переміщує на один рівень вище.

Варіант 8

3. Задано ім'я файла *name* і ціле число *N*. Напишіть програму, яка розбиває заданий файл на фрагменти довжиною не більше *N* байт. Кожен фрагмент потрібно зберегти в окремому файлі з ім'ям *name-xx*, де *xx* – порядковий номер фрагмента. Виведіть імена створюваних файлів та їх розміри на екран, а також у текстовий файл.
4*. (*Для програмістів*) Напишіть програму, яка за заданим ім'ям файла-фрагмента *name-xx* шукає інші фрагменти і об'єднує їх в один файл з ім'ям

name. Якщо якийсь із фрагментів відсутній, видайте повідомлення про помилку.

Варіант 9

3. Дано файл, що містить список директорій (кожен рядок файла містить повний шлях до певної директорії). Вивести назви всіх файлів із цих директорій одним списком *в алфавітному порядку*.
4*. (*Для програмістів*) Напишіть програму, яка зменшує розмір папки проекту Visual C++ (або Visual C#), видаляючи з неї всі файли, без яких проект може бути успішно відкритий і заново компільований. Програма повинна також зберігати всі файли готових програм із підпапок Release.

Варіант 10

3. Напишіть програму, яка вводить три імені директорій (A, B і C) та об'єднує дві директорії A і B в одну директорію C. Якщо директорія C вже існує, то виведіть попередження. Якщо директорії A і B містять файли з однаковими іменами, то в C потрібно помістити тільки одну, найновішу версію цього файла.
4*. (*Для програмістів*) Напишіть програму, яка сканує всі файли вихідних кодів Java в заданій директорії і визначає, скільки відсотків коду складають коментарі. Необхідно враховувати і однорядкові, і багаторядкові коментарі. Програма повинна обробляти всі вкладені директорії.

Варіант 11

3. Напишіть програму, яка копіює з першої заданої директорії до другої всі файли, змінені за останні 3 дні. Виведіть імена файлів, що копіюються, на екран і в текстовий файл.
4*. (*Для програмістів*) Напишіть програму, яка порівнює вміст двох заданих директорій і виводить список файлів, які розрізняються. Для порівняння

файлів використовуйте ім’я, розмір і дату останньої зміни файла. Програма повинна обробляти всі вкладені директорії.

Варіант 12

3. Задані ім’я файла *name* і ціле число *N*. Напишіть програму, яка розбиває заданий файл на фрагменти довжиною не більше *N* байт. Кожен фрагмент потрібно зберегти в окремому файлі з ім’ям *name-xx*, де *xx* – порядковий номер фрагмента. Виведіть імена створюваних файлів і їх розміри на екран, а також у текстовий файл.

4*. (*Для програмістів*) Напишіть програму, яка за заданим ім’ям файла-фрагмента *name-xx* шукає інші фрагменти та об’єднує їх в один файл з ім’ям *name*. Якщо якийсь із фрагментів відсутній, видайте повідомлення про помилку.

Варіант 13

3. Напишіть програму, яка знаходить кількість рядків у кожному .txt-файлі заданої директорії. Ім’я директорії потрібно вводити з клавіатури. Результати виведіть на екран і в текстовий файл.

4*. (*Для програмістів*) Напишіть програму, яка вводить імена двох директорій і регулярний вираз, а потім копіює з першої директорії до другої всі файли, імена яких відповідають регулярному виразу. Програма повинна обробляти всі вкладені директорії. При копіюванні структура вкладених директорій має зберігатися.

Варіант 14

3. Напишіть програму, яка вводить три імені директорій (A, B і C) і об’єднує дві директорії A і B в одну директорію C. Якщо директорія C вже існує, то виведіть попередження. Якщо директорії A і B містять файли з однаковими іменами, то в C потрібно помістити тільки одну, найновішу версію цього файла.

4*. (*Для програмістів*) Напишіть програму, яка для заданої директорії визначає, яку частину загального обсягу (у відсотках) займають файли кожного типу. Тип файла визначається його розширенням. Для зберігання даних використовуйте колекцію HashMap.

Варіант 15

3. Напишіть програму, яка для заданої директорії визначає кількість файлів з урахуванням вкладених директорій. Виведіть імена всіх цих файлів на екран, а також у текстовий файл.

4*. (*Для програмістів*) Напишіть програму, яка шукає адреси електронної пошти в усіх текстових файлах заданої директорії і виводить усі знайдені адреси на екран. При виведенні адреси відсортуйте за ім'ям домену.

Вказівка. Визначити, чи є файл текстовим, можна за його розширенням; список допустимих розширень текстових файлів (наприклад, txt, htm, html, csv, java, ...) програма повинна читати з окремого файла.

Додаткові завдання

У Java для читання і запису zip-архівів застосовуються класи ZipInputStream і ZipOutputStream. Використовуючи ці класи, напишіть програму, яка може архівувати і розпаковувати набір файлів або директорій. Вхідні дані (тобто набір імен файлів або директорій) програма повинна отримувати у вигляді параметрів командного рядка.

Налаштуйте конфігурацію запуску проекту так, щоб отримати файл .jar, який можна запустити в ОС Windows подвійним клацанням по ньому. Протестуйте його роботу, викликаючи його з командного рядка і передаючи в командному рядку імена файлів (директорій).

Додаток А. Архітектура Java IO

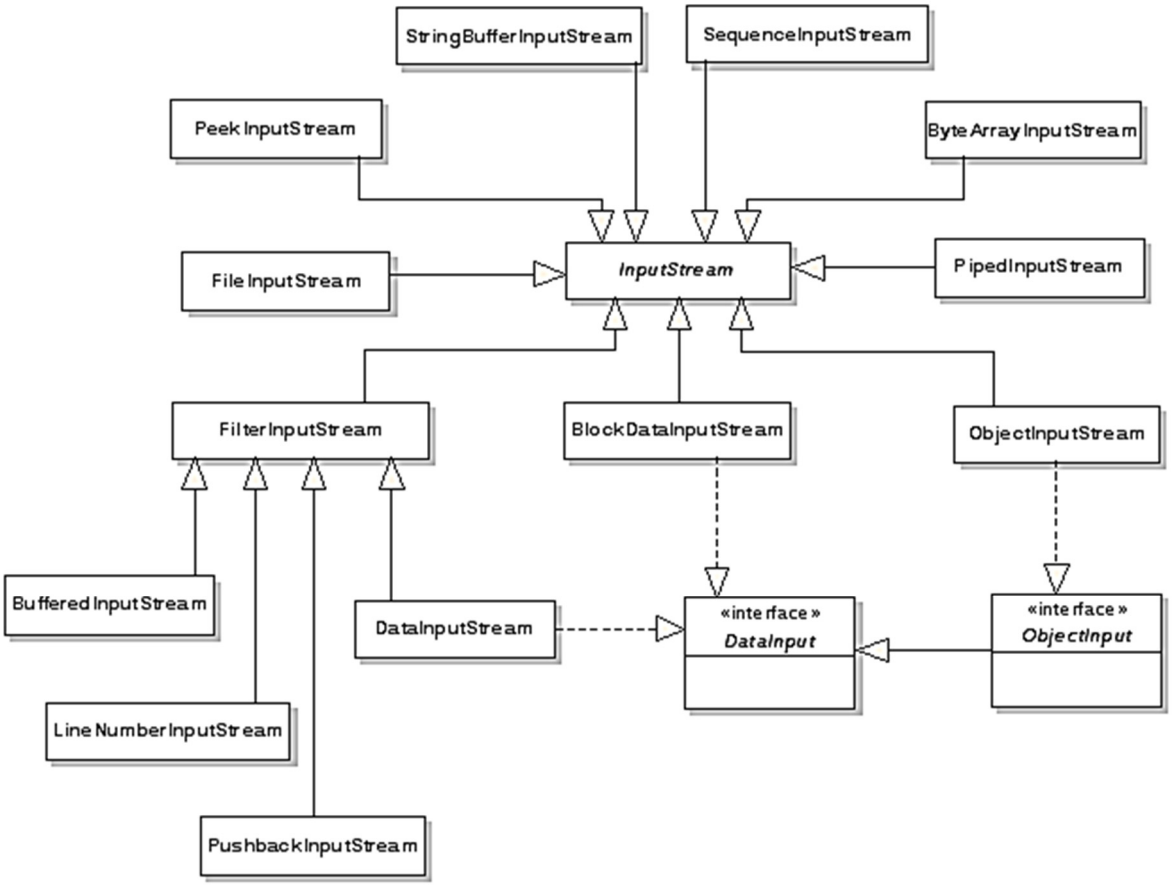

Рис. 4.1 – Ієрархія байтових потоків введення InputStream

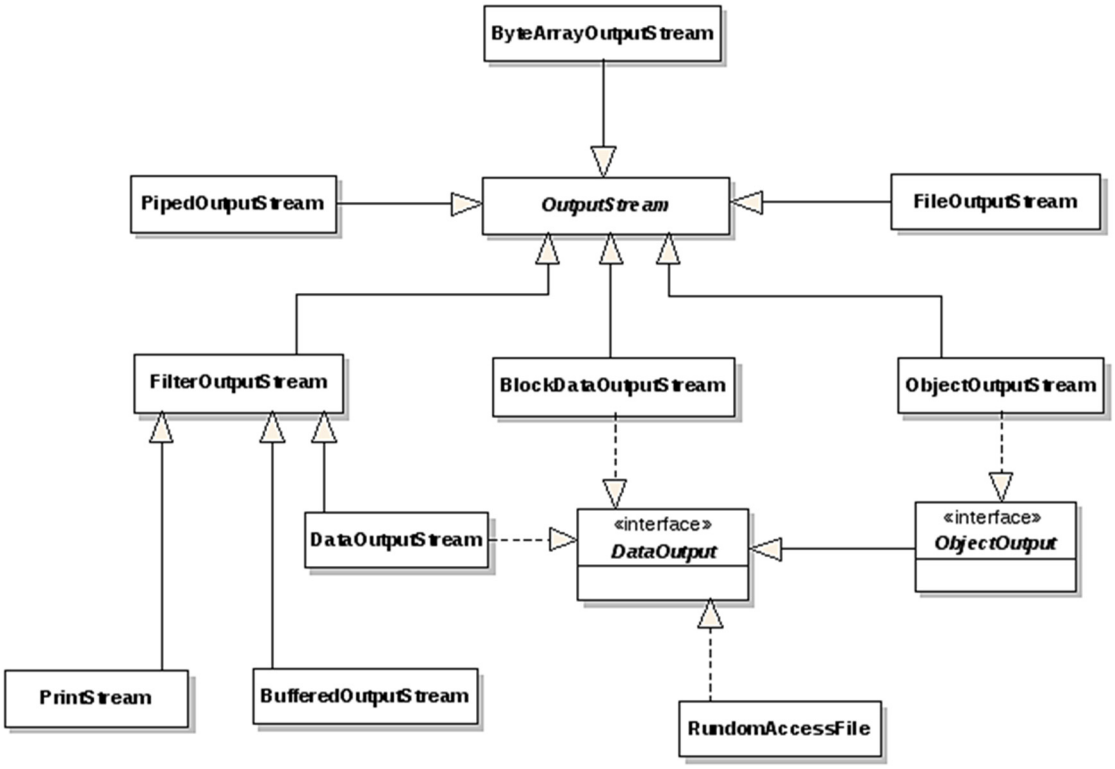

Рис. 4.2 – Ієрархія байтових потоків виведення OutputStream

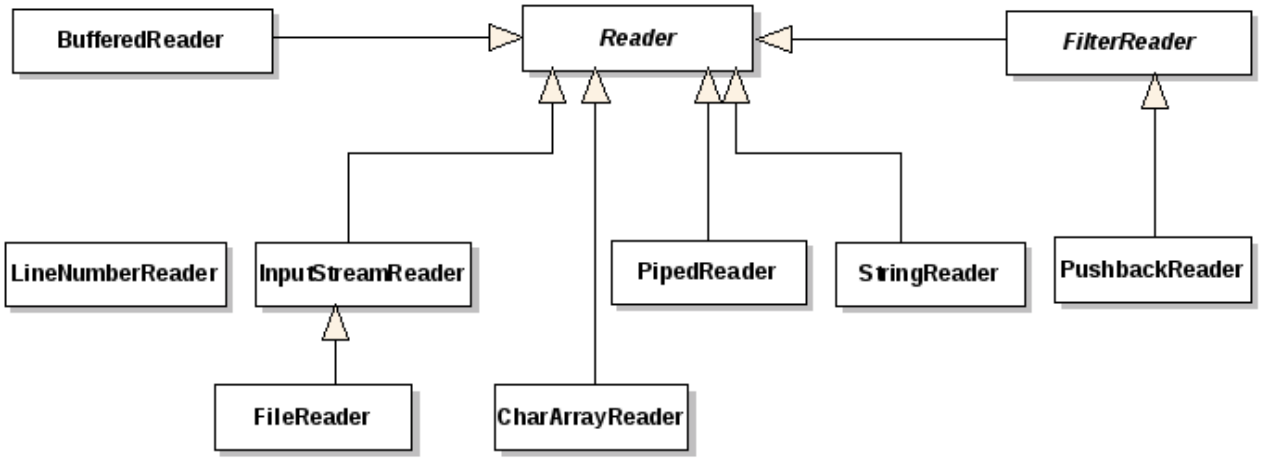

Рис. 4.3 – Ієрархія текстових потоків введення Reader

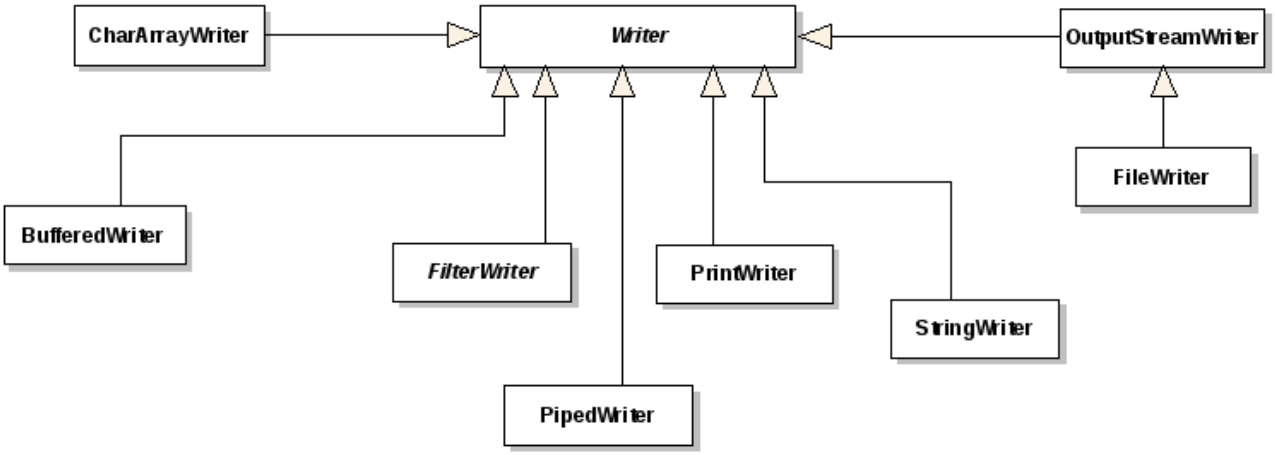

Рис. 4.4 – Ієрархія текстових потоків виведення Writer