

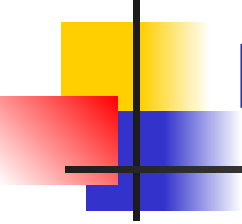
**Потоки введення-
виведення. Робота з
файлами.**

Потоки введення-виведення: ОСНОВНІ ПОНЯТТЯ



- Поняття потоку лежить в основі всього введення-виведення в Java
- ***Потік введення-виведення*** - абстракція, яка описує передачу даних від джерела до приймача.
- Більш точно: звідки можна читати - джерело, а куди можна записувати - приймач.

Потоки введення-виведення: продовження



- Фізично потоки можуть бути приєднані до будь-яких джерел та приймачів: консоль, файли, мережа та ін.
- Для окремих типів джерел і приймачів існують методи, які дозволяють створити той чи інший потік.



Основні пакети

- **java.io** - базові можливості;
- **java.nio** - новий інтерфейс введення-виведення з важливими удосконаленнями; вважається більш ефективним для роботи в мережі та клієнт-серверних застосувань.

Загальна класифікація ПОТОКІВ



- Потоків дуже багато.
- Потоки: **байтові** (орієнтовані на читання/запис байтів) і **СИМВОЛЬНІ** (читання/запис символів).
- **В 1.0 все введення-виведення було байтовим. Зараз у ряді випадків використання байтових потоків вважається deprecated.**



Базові потоки

- На вершині ієрархії: байтові потоки **InputStream** та **OutputStream**; символні потоки **Reader** та **Writer**.
- Стандартні потоки (байтові): **System.in**; **System.out**; **System.err**.



Метод read()

- Абстрактний метод, визначений в `InputStream`. Зчитує один байт.
- В кінці потоку повертає `-1`.
- Інші методи викликають цей метод.
- Перевизначається в підкласах.
- Аналогічний метод - в `Reader`: читає один символ.

Проблема блокування потоку



- Виникає, якщо нема чого зчитувати.
- Метод *available()* повертає кількість байтів, які можуть бути зчитані.



Виведення

- Метод *write(byte b)* з класу **OutputStream** дозволяє вивести байт у потік.
- Аналогічний метод класу **Writer** дозволяє вивести в потік символ.
- Особливість: стандартний потік **System.out** є екземпляром класу **PrintStream**; для нього визначені узагальнені методи *print* та *println*.



Клас `BufferedReader`

- Важливий клас, який має корисні методи для читання.
- Зокрема, ми часто будемо використовувати `readLine()` - читання рядка.



Базові файлові потоки

- Байтові: **FileInputStream;**
FileOutputStream.
- Символьні: **FileReader; FileWriter.**
- Корисна форма конструктора:
відкриття файлу в режимі append:
FileWriter(filename, true);



“Перехідні” потоки

- **InputStreamReader** - бере дані з байтового потоки і переводить в символи Unicode;
- **OutputStreamWriter** - записує символи в байтовий потік.

- Типовий приклад:

```
BufferedReader br=new BufferedReader(new  
InputStreamReader (System.in));
```



Інші важливі можливості

- запис до рядків і читання з рядків;
- потоки, пов'язані з zip-файлами;
- канали: використання потоків для обміну даними між процесами;
- серіалізація;
- ...



Клас File

- Методи для операцій з файлами на рівні файлової системи.