

IO

Platformy Technologiczne

Michał Piotrowski
Michał Wójcik

Katedra Architektury Systemów Komputerowych
Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańska

4 marca 2011

Strumienie:

- Byte Streams — pisanie/czytanie bajtami:
 - InputStream,
 - OutputStream,
 - FileInputStream i FileOutputStream,
 - baza dla wszystkich rodzaj strumienie, odradza się używania;
- Character Streams — pisanie/czytanie znakami:
 - Reader,
 - Writer,
 - FileReader i FileWriter,
 - BufferedReader i PrintWriter — dekoratory umożliwiające czytanie/pisanie liniami,
 - wykorzystuje Unicode do przechowywania znaków,
 - automatyczna translacja znaków pomiędzy Unicode a lokalnym zbiorem znaków.

Uwagi:

- zawsze zamykać strumienie!

Strumienie buforowane — zmniejszenie liczby odwołań do natywnego API:

- Byte Streams:
 - `BufferedInputStream`,
 - `BufferedOutputStream`;
- Character Streams:
 - `BufferedReader`,
 - `BufferedWriter`;
- w formie dekoratorów,
- `flush` dla wypisania buforu przed jego zapełnieniem.

Standardowe strumienie:

- `System.in` (`InputStream`),
- `System.out` (`PrintStream`),
- `System.err` (`PrintStream`).

Konsola:

- `System.console` — pobranie konsoli, zwraca `NULL` w przypadku gdy jest to nie możliwe,
- `char[] readPassword` – pobranie hasła z konsoli.

Serializacja — konwersja obiektu do strumienia bajtów:

- implementacja interfejsu `Serializable`,
- strumienie:
 - `ObjectInputStream`,
 - `ObjectOutputStream`;
- `transient` — pola nie podlegające serializacji,
- `static final long serialVersionUID` — wersjonowanie klas.

Klasa File:

- `canRead`, `canWrite`, `canExecute`,
- `createNewFile`, `delete`, `mkdir`, `makedirs`,
- `isFile`, `isHidden`, `isDirectory`,
- `listFiles`.