

# Java IO

cuu duong than cong . com

Nguyễn Văn Khiết

cuu duong than cong . com

# Nội dung

- Giới thiệu
- Làm việc với byte stream
- Làm việc với character stream

# Giới thiệu

- Các lớp chính liên quan đến việc xử lý IO đều nằm trong gói java.io
- Các lớp xử lý IO trong java được chia thành 2 loại chính:
  - Loại xử lý liên quan đến byte và số, thuộc họ các lớp InputStream và OutputStream.
  - Loại xử lý liên quan đến ký tự và text, thuộc họ các lớp Reader và Writer.

# Stream

- Stream là một dãy tuần tự các byte với chiều dài không xác định
- **Input Stream** là các stream thực hiện việc di chuyển đưa dãy các byte vào trong chương trình Java từ một nguồn bên ngoài.
- **Output Stream** đưa dãy các byte từ chương trình Java đến các nơi bên ngoài

# Sử dụng các byte stream

- Các byte stream được chia làm 2 loại: byte stream cấp thấp và byte stream cấp cao.
- Byte stream cấp thấp thực hiện việc đọc/ghi trực tiếp dữ liệu.
- Byte stream cấp cao sử dụng byte stream cấp thấp và cung cấp thêm các tiện ích trên dữ liệu

# Sử dụng các byte stream

- Byte stream cấp thấp:
  - **InputStream** và **OutputStream** là các abstract class.

cuu duong than cong . com

cuu duong than cong . com

# InputStream

- **InputStream** định nghĩa các phương thức sau
  - `int available( )`
  - `void close( )`
  - `void mark(int numBytes)`
  - `boolean markSupported( )`
  - `int read( )`
  - `int read(byte buffer[ ])`
  - `int read(byte buffer[ ], int offset, int numBytes)`
  - `void reset( )`
  - `long skip(long numBytes)`

# InputStream

- Một số lớp con tiêu biểu của **InputStream**:
  - FileInputStream
  - FilterInputStream
  - ByteArrayInputStream
  - StringBufferInputStream
  - SequenceInputStream
  - PipedInputStream
  - ...

# OutputStream

- **OutputStream** định nghĩa các phương thức sau:
  - void close( )
  - void flush( )
  - void write(int b)
  - void write(byte buffer[ ])
  - void write(byte buffer[ ], int offset, int numBytes)

# OutputStream

- **Một số lớp con của OutputStream:**
  - FileOutputStream
  - FilterOutputStream
  - ByteArrayOutputStream
  - PipedOutputStream
  - ...

# Các đối tượng có sẵn của Java

- System.in
- System.out
- System.err

cuu duong than cong . com

cuu duong than cong . com

# System.in

- Một InputStream đọc dữ liệu trực tiếp từ bàn phím.

cuu duong than cong . com

cuu duong than cong . com

# System.out

- Ghi dữ liệu ra console
- Là một PrintStream
  - Kế thừa OutputStream
  - Có thêm các phương thức print và println
  - Kể từ J2SE 1.5, PrintStream được bổ sung thêm 2 phương thức mới là format và printf

# System.err

- Ghi lỗi ra console

cuu duong than cong . com

cuu duong than cong . com

# Chuyển hướng System.in, System.out, System.err

- `public static void setIn(InputStream in)`
- `public static void setOut(PrintStream out)`
- `public static void setErr(PrintStream err)`

cuu duong than cong . com

# Đọc và ghi file

- Đọc và ghi file dưới dạng byte stream được thực hiện thông qua 2 lớp `FileInputStream` và `FileOutputStream`

cuu duong than cong . com

cuu duong than cong . com

# Đọc file

- Tạo đối tượng FileInputStream, khởi tạo đối tượng với tên file cần đọc
- Dùng các hàm read() để đọc dữ liệu
- gọi close() để đóng stream lại

cuu duong than cong . com

# Ghi file

- Tạo đối tượng `FileOutputStream`, khởi tạo đối tượng với tên file cần ghi.
- dùng các hàm `write()` để ghi dữ liệu
- dùng `flush()` để đưa dữ liệu còn trong buffer vào file
- gọi `close()` để đóng stream lại

# Các InputStream khác

- ByteArrayInputStream
- ObjectInputStream
- PipedInputStream
- SequenceInputStream
- StringBufferInputStream

# Các OutputStream khác

- ByteArrayOutputStream
- ObjectOutputStream
- PipedOutputStream

# Làm việc với các byte stream cấp cao

- Các stream cấp cao là các stream cung cấp các phương thức giúp người dùng làm việc tiện lợi hơn
- Các stream cấp cao
  - sử dụng các stream cấp thấp để thực hiện việc truy xuất dữ liệu trực tiếp
  - bản thân stream thực hiện việc biến đổi và cung cấp các tiện ích trên dữ liệu thô nhận được từ các stream cấp thấp

- hàm khởi tạo (constructor) của các stream cấp cao yêu cầu phải cung cấp một stream khác.
- Các stream đọc cấp cao kế thừa từ một stream cấp thấp là `FilterInputStream`
- Các stream ghi cấp cao kế thừa từ một stream cấp thấp đó là `FilterOutputStream`

# Đọc và ghi dữ liệu nhị phân

- Giải quyết nhu cầu đọc và ghi dữ liệu dưới dạng số nguyên, số thực, ...
- Sử dụng lớp `DataInputStream` và `DataOutputStream`

cuu duong than cong . com

- DataInputStream implement DataInput interface
  - boolean readBoolean( )
  - byte readByte( )
  - char readChar( )
  - double readDouble( )
  - float readFloat( )
  - int readInt( )
  - long readLong( )
  - short readShort( )
  - int readUnsignedByte( )
  - int readUnsignedShort( )
  - String readUTF( )

- DataOutputStream implement DataOutput interface
  - void writeBoolean(boolean val )
  - void writeByte(int val )
  - void writeChar(int val )
  - void writeDouble(double val )
  - void writeFloat(float val )
  - void writeInt(int val )
  - void writeLong(long val )
  - void writeShort(int val )
  - void writeUTF(String s)

# Sử dụng BufferedInputStream và BufferedOutputStream

- BufferedInputStream
  - override các phương thức đã được cung cấp trong InputStream
  - đọc từng phần dữ liệu lớn rồi đóng gói dữ liệu này vào một buffer , đạt được hiệu quả hơn so với việc đọc từng byte tại nhiều thời điểm
  - cơ chế đọc dữ liệu hoàn toàn trong suốt đối với lập trình viên

- **BufferedInputStream**

- khởi tạo một BufferedInputStream

- public BufferedInputStream(InputStream in)
    - public BufferedInputStream(InputStream in, int size)

- **BufferedOutputStream**

- có các đặc tính tương tự BufferedInputStream nhưng đây là một stream dùng để ghi.

# Sử dụng PrintStream

- Dùng để ghi trực tiếp text hoặc các kiểu dữ liệu cơ sở.
- Ta có thể khởi tạo PrintStream bằng các cách sau:

```
PrintStream ps=new PrintStream(OutputStream out);  
PrintStream ps=new PrintStream(OutputStream out,  
    boolean autoflush);
```

# Tập tin truy cập ngẫu nhiên

- RandomAccessFile không kế thừa từ InputStream, OutputStream
- implements các interface DataInput, DataOutput
- cung cấp phương thức giúp người dùng định vị đến một vị trí của con trỏ file

- Các phương thức của RandomAccessFile:

- void seek(long newPos)
- boolean readBoolean( )
- byte readByte( )
- char readChar( )
- double readDouble( )
- float readFloat( )
- int readInt( )
- long readLong( )
- short readShort( )
- void writeBoolean(boolean val )
- void writeByte(int val )
- void writeChar(int val )
- void writeDouble(double val )
- void writeFloat(float val )
- void writeInt(int val )
- void writeLong(long val )
- void writeShort(int val )

# Đọc/Ghi dữ liệu sử dụng các character stream

- Để xử lý việc đọc ghi dữ liệu dưới dạng ký tự hoặc text, ta nên sử dụng các lớp đọc ghi dựa trên ký tự do Java cung cấp.
- Các character stream làm việc với dữ liệu Unicode
- Các character stream gồm: bộ đọc và bộ ghi.
  - các bộ đọc đều kế thừa từ Reader
  - các bộ ghi thì kế thừa từ Writer

# Reader

- Reader là một lớp abstract.
  - abstract void close( )
  - void mark(int numChars)
  - boolean markSupported( )
  - int read( )
  - int read(char buffer[ ])
  - abstract int read(char buffer[ ], int offset, int numChars)
  - int read(CharBuffer buffer)
  - boolean ready( )
  - void reset( )
  - long skip(long numChars)

# Reader

- CharArrayReader
- FilterReader
- InputStreamReader
- PipedReader
- StringReader
- FileReader
- ...

# Writer

- Writer cũng là một lớp abstract
- Các phương thức Writer định nghĩa:
  - `Writer append(char ch)`
  - `Writer append(CharSequence chars)`
  - `Writer append(CharSequence chars, int begin, int end)`
  - `abstract void close( )`
  - `abstract void flush( )`
  - `void write(int ch)`
  - `void write(char buffer[ ])`
  - `abstract void write(char buffer[ ], int offset, int numChars)`
  - `void write(String str)`
  - `void write(String str, int offset, int numChars)`

# Writer

- CharArrayWriter
- OutputStreamWriter
- PipedWriter
- StringWriter
- PrintWriter
- FileWriter
- ...

# Sử dụng FileWriter

- writer chuyên biệt dùng để ghi file, khởi tạo, cần đưa vào tên file muốn ghi

cuu duong than cong . com

cuu duong than cong . com

# FileReader

- chuyên biệt cho việc đọc chuỗi dữ liệu, ký tự Unicode từ file
- Khi khởi tạo đối tượng FileReader, ta cần cung cấp tên file cần đọc, nếu file này chưa tồn tại thì sẽ sinh ra exception `FileNotFoundException`.

# InputStreamReader và OutputStreamWriter

- InputStreamReader là một reader đọc dữ liệu dưới dạng byte từ input stream, sau đó, chuyển dữ liệu đó sang dạng character. [cuu duong than cong . com](http://cuuduongthancong.com)
- InputStreamReader rất phù hợp trong trường hợp ta muốn đọc dữ liệu character từ các nguồn (source) chỉ cung cấp thông tin dưới dạng byte, như System.in

# PrintWriter

- Đây là lớp được khuyến khích sử dụng khi cần đưa dữ liệu Unicode ra ngoài.

cuu duong than cong . com

cuu duong than cong . com

# BufferedReader

- BufferedReader là bộ đọc cấp cao, đọc dữ liệu từ một bộ đọc (Reader) khác.
- Các phương thức cung cấp bởi BufferedReader
  - `int read( )`
  - `int read(char data[])`
  - `int read(char data[], int start, int max)`
  - `String readLine()`

# Bài tập luyện tập – Bài 1

- Viết chương trình quản lý học sinh (lưu danh sách học sinh thành file nhị phân), cho phép thực hiện các thao tác:
  - Thêm học sinh. (MHS, TenHS, Diem, Hình ảnh, Địa chỉ, GhiChu)
  - Cập nhật thông tin học sinh
  - Xóa học sinh
  - Xem danh sách học sinh:
    - MHS tăng dần, giảm dần
    - Diem tăng dần, giảm dần
  - Import/Export danh sách học sinh ra file text (csv)

# Bài tập luyện tập – Bài 2

- Viết chương trình quản lý file (Total Commander, Windows Explorer), các chức năng tối thiểu:
  - Tạo, xóa, đổi tên, copy thư mục
  - Tạo, xóa, đổi tên, copy file
  - Nén/Giải nén thư mục, file.
  - View/Edit 1 file (text)
  - Cắt/Cộng file

(giao diện đồ họa).

Hạn nộp: 23h ngày 25/02/2018