

JAVA BEGINNER

FILE IO



Abstract of New Technology

Website <http://www.antkh.com>

Tel 010 / 016 66 66 53

Prepared By Tum Sakal

Tel 087 36 31 30

Mail tumsakal.ts@gmail.com

Facebook [sak kal](#)

FILE

- File គឺជា Class ដែលណាងអោយ File រឺ Folder

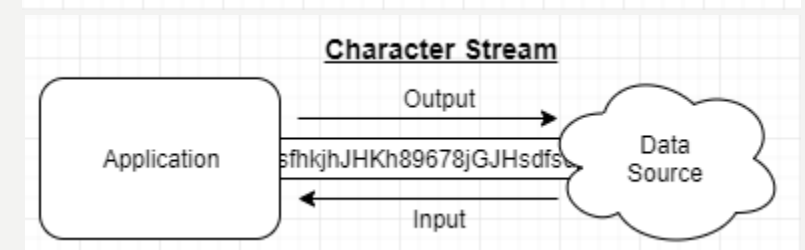
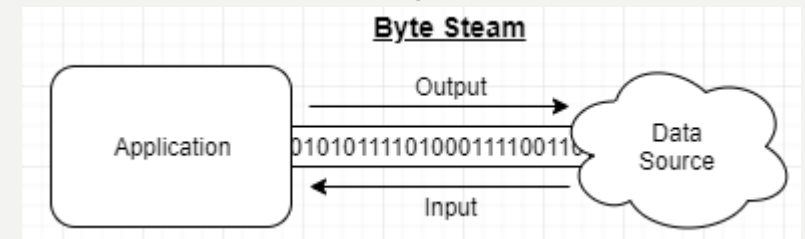
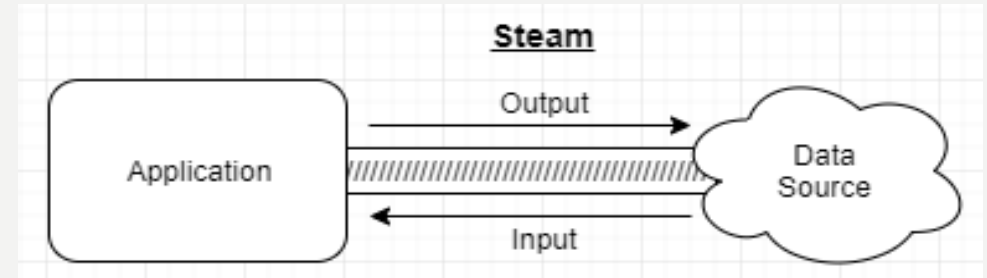
```
File folder = new File("D:\\MyFolder");
```

```
File file = new File("D:\\MyFolder\\myfile.txt");
```

- File Class អាចយកមកប្រើដើម្បី
 - បង្កើត
 - លុប
 - ស្វែងរក
 - ទាញយកបញ្ជីនៃ Files និង Folders ដែលមាននៅក្នុង Folder មួយ
 - និង ទាញយកព័ត៌មានមួយចំនួនរបស់ File រឺ Folder មួយជាដើម

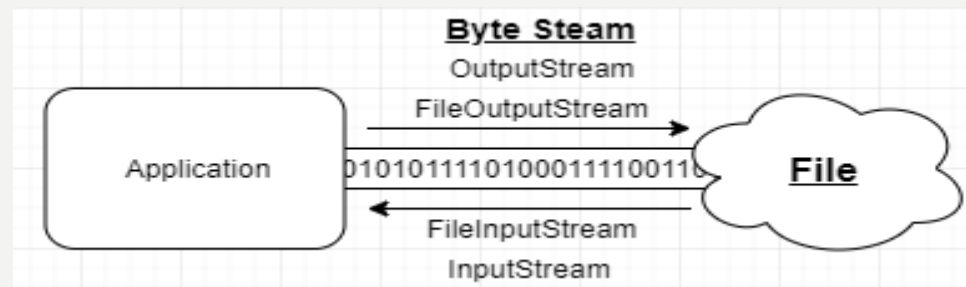
STREAM

- Stream សំដៅទៅលើផ្លូវលំហូរនៃទិន្នន័យ ទៅវិញទៅមក ពីកម្មវិធី ទៅកាន់ប្រភពទិន្នន័យណាមួយ។
- Stream ជំនឿការព័រប្រភេទគឺ
 - Output : លំហូរទិន្នន័យពីកម្មវិធី ទៅកាន់ប្រភពទិន្នន័យណាមួយ
 - Input: លំហូរទិន្នន័យពីប្រភពទិន្នន័យណាមួយ មកកាន់កម្មវិធី
- Stream អាចបែងចែកជាព័រប្រភេទគឺ
 - Byte Stream
 - Character Stream



BYTE STREAM

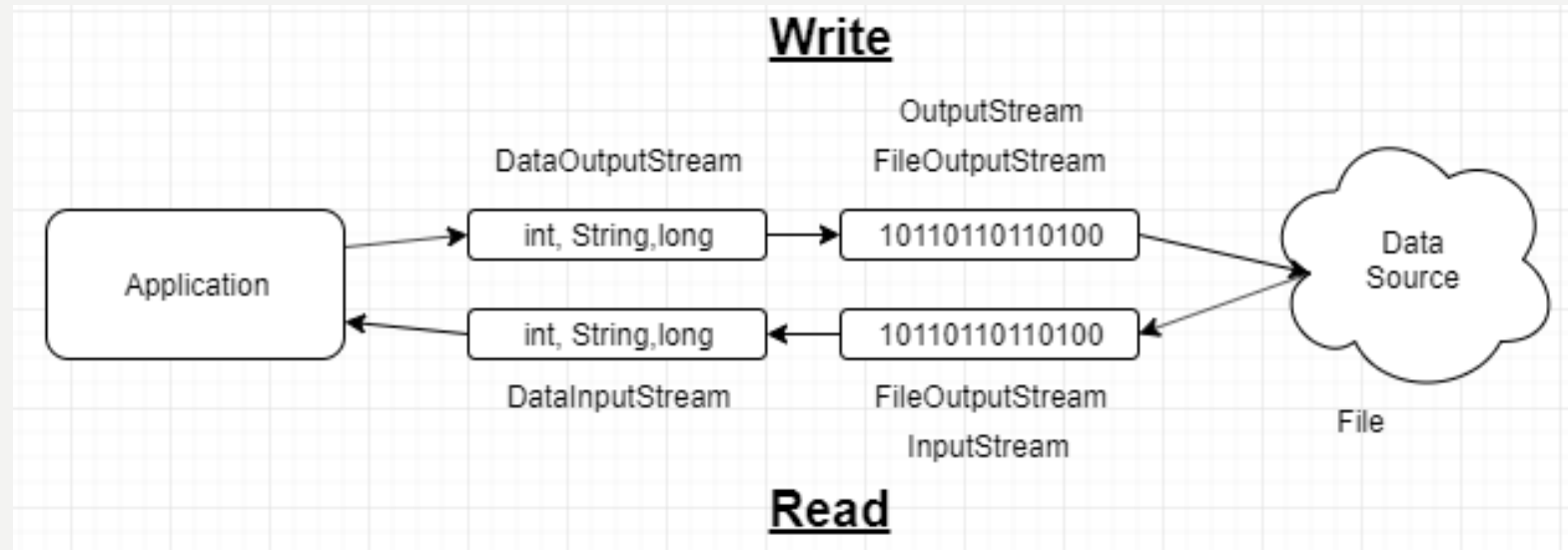
- **OutputStream**: ជាប្រភេទ Stream ដែលប្រើសំរាប់ Output ទិន្នន័យ ក្នុងទំរង់ byte ទៅប្រភពទិន្នន័យណាមួយ
- **FileOutputStream**: ជាប្រភេទ OutputStream ទៅកាន់ File
- **InputStream**: ជាប្រភេទ Stream ដែលប្រើសំរាប់ Input ទិន្នន័យពី ប្រភពណាមួយក្នុងទំរង់ byte ចូលមកកម្មវិធី
- **FileInputStream**: ជាប្រភេទ InputStream ទៅកាន់ File



PRIMITIVE DATA

- *DataOutput* គឺជា Interface ដែលផ្តល់មុខងារបំប្លែងទិន្នន័យ Primitive ទៅកាន់ Binary និងរុញចូលទៅកាន់ Byte Stream
- DataOutputStream ជា class ដែល Implements *DataOutput* អនុញ្ញាតអោយកម្មវិធី Write Primitive ទៅកាន់ OutputStream
- *DataInput* គឺជា Interface ដែលផ្តល់មុខងារទាញទិន្នន័យ Binary ពី Byte Stream បំប្លែងទិន្នន័យទៅកាន់ Primitive វិញ
- DataInputStream ជា class ដែល Implements *DataInput* អនុញ្ញាតអោយកម្មវិធី Read Primitive Data ចេញពី InputStream

- `File file = new File("path_to_file");`
- `OutputStream outStream = new FileOutputStream(file);`
- `DataOutput writer = new DataOutputStream(outStream);`



- `File file = new File("path_to_file");`
- `InputStream inStream = new FileInputStream(file);`
- `DataInput reader = new DataInputStream(inStream);`

DATAOUTPUTSTREAM

Return

Method

void	writeByte(byte data)	void	flush()
void	writeShort(short data)	void	close()
void	writeInt(int data)		
void	writeLong(long data)		
void	writeFloat(float data)		
void	writeDouble(double data)		
void	writeChar(char data)		
void	writeUTF(String data)		
void	writeBoolean(boolean data)		

DATAOUTPUTSTREAM

Return

Method

byte

readByte()

void

close()

short

readShort(data)

int

readInt()

long

readLong()

float

readFloat()

double

readDouble()

char

readChar()

String

readUTF()

boolean

readBoolean()

TRY-WITH-RESOURCE

- ការសរសេរកូដ Read និង Write ទិន្នន័យ ជាទូទៅអ្នកសរសេរកូដ តែងតែជួបបញ្ហាមួយជាដើមៗ គឺភ្លេចសរសេរកូដហៅ method “close()” ដើម្បីបិទ Stream
- **Try-catch-with-resource** សំរួលអ្នកសរសេរកូដ ដោយជួយហៅ method “close()” និងបិទ Stream ដោយស្វ័យប្រវត្តិ
- try-with-resource អាចប្រើបានជាមួយ Class ដែល implements interface “java.lang.*AutoCloseable*”

```
try(resource_1; resource_2; ...){ ... }
```

TRY-WITH-RESOURCE

```
File file = new File("path_to_file");  
try (OutputStream outStream = new FileOutputStream(file);  
     DataOutput writer = new DataOutputStream(outStream)){  
    //write statements...  
}catch (IOException ignored){    }
```

```
try (InputStream inStream = new FileInputStream(file);  
     DataInput reader = new DataInputStream(inStream)) {  
    //read statements...  
}catch (IOException ignored){    }
```