Bài 16
Thao tác với file text

Module: ADVANCED PROGRAMMING WITH JAVA

Mục tiêu

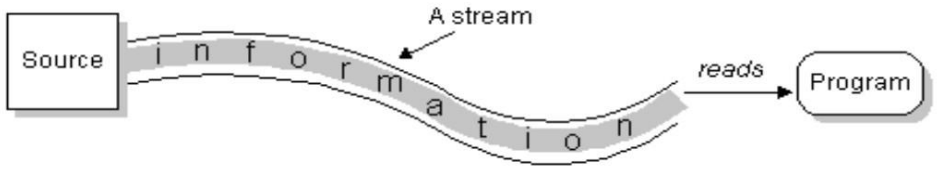
- Sử dụng được các Wrapper Class
- Trình bày được gói java.io
- Trình bày được Stream
- Thực hiện được thao tác cơ bản với file và thư mục
- Thực hiện được các thao tác đọc và ghi với file text

Thao tác với file text

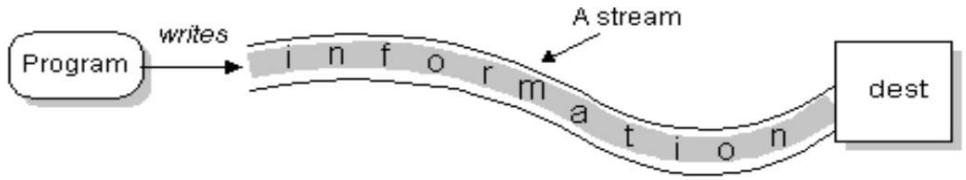
Giới thiệu gói java.io
Giới thiệu về Stream
Thao tác cơ bản với file và thư mục
Thao tác đọc và ghi với file text

Giới thiệu về Stream

- Các hoạt động nhập/xuất dữ liệu như nhập dữ liệu từ bàn phím, đọc dữ liệu từ file, ghi dữ liệu màn hình, ghi ra file, ghi ra đĩa, ghi ra máy in ... được gọi là stream (dòng).
- Dòng vào là luồng cho phép chương trình đọc dữ liệu từ một nguồn nào đó như bàn phím, file, máy scan ...



- Dòng ra là luồng cho phép chương trình ghi dữ liệu lên nó *để* chuyển *đến đích* nào *đó*: màn hình, file, máy in ...

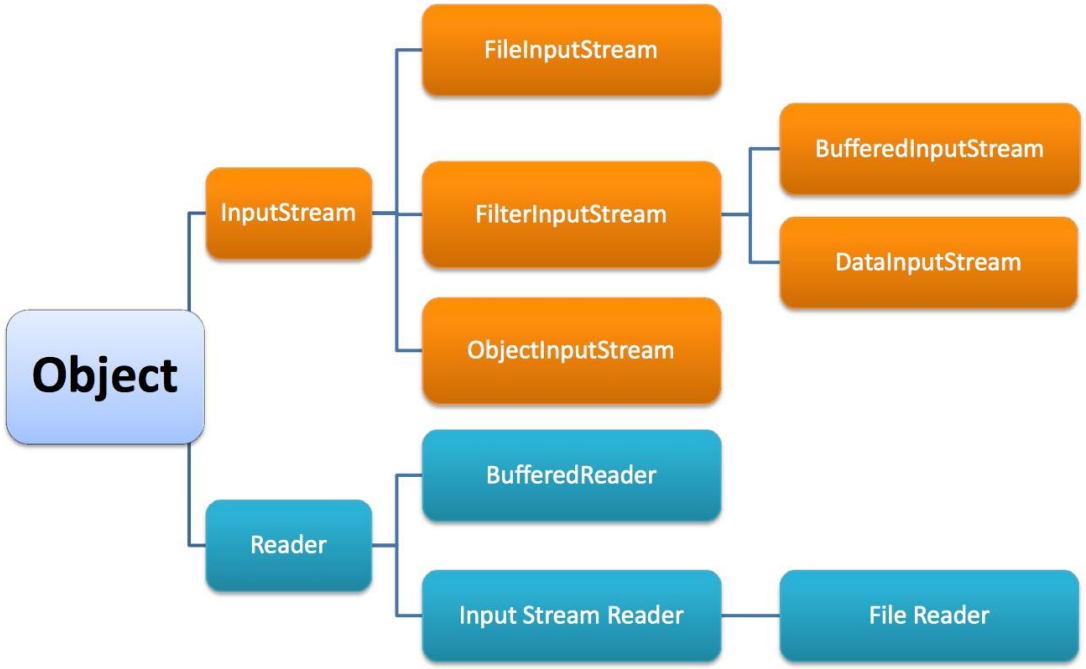


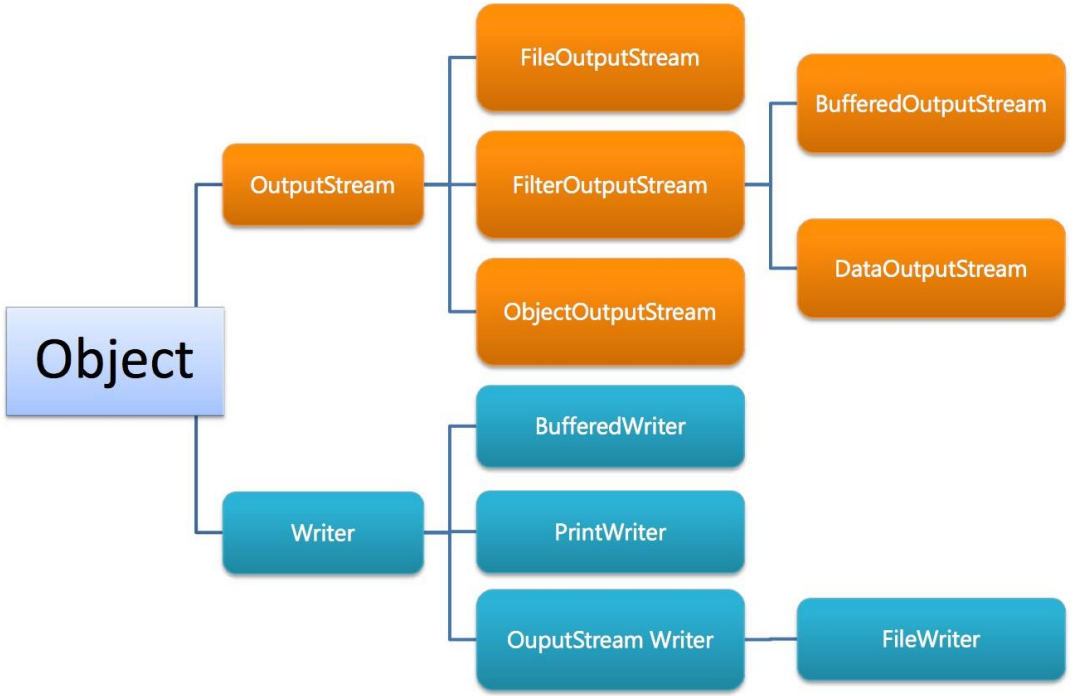
Các loại luồng dữ liệu

- Có 2 kiểu luồng trong Java: dòng bye (dòng nhị phân) và dòng character (dòng văn bản)
- Dòng byte (byte - based stream)
  - Hỗ trợ việc xuất nhập dữ liệu theo byte
  - Thường được dùng khi đọc ghi dữ liệu nhị phân
- Dòng character (character - based stream)
  - Luồng character được thiết kế hỗ trợ việc xuất nhập dữ liệu kiểu ký tự

Giới thiệu gói java.io

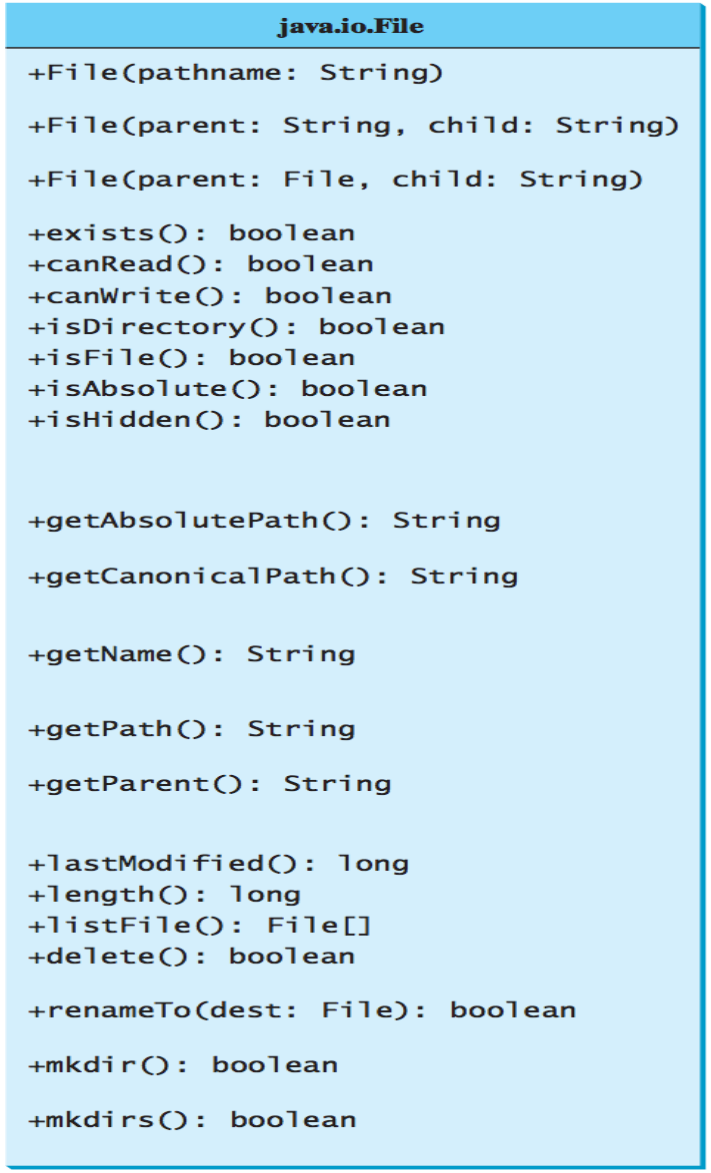
- Gói java.io chứa các interface, class để thao tác với dòng dữ liệu.





Lớp File trong Java

- Lớp File thuộc gói java.io đại diện cho một file hoặc một thư mục.
- Lớp này không có các tiện ích đọc/ghi file, nhưng nó là đại diện an toàn cho file hơn là chuỗi ký tự tên file.
- Hầu hết các lớp lấy tên file làm tham số cho phương thức khởi tạo.



Lớp File trong Java

- Tạo một đối tượng File đại diện cho một file đang tồn tại

  File f=new File("foo.txt");

- Tạo một thư mục mới

  File dir=new File("Books");

  dir.mkdir();

- Liệt kê nội dung của một thư mục

```
if (dir.isDirectory()) {
  String[] dirContents = dir.list();
  for (int i = 0; i < dirContents; i++)
    System.out.println(dirContents[i]);
}
```

Lớp File trong Java

- Lấy đường dẫn tuyệt đối của file hoặc thư mục

```
System.out.println(dir.getAbsolutePath());
```

- Xoá file hoặc thư mục

```
boolean isDeleted =f.delete();
```

Ví dụ sử dụng lớp File

```
public class TestFileClass {
  public static void main(String[] args) {
    java.io.File file = new java.io.File("image/us.gif");                    create a File
    System.out.println("Does it exist? " + file.exists());                   exists()
    System.out.println("The file has " + file.length() + " bytes");          length()
    System.out.println("Can it be read? " + file.canRead());                 canRead()
    System.out.println("Can it be written? " + file.canWrite());             canWrite()
    System.out.println("Is it a directory? " + file.isDirectory());          isDirectory()
    System.out.println("Is it a file? " + file.isFile());                    isFile()
    System.out.println("Is it absolute? " + file.isAbsolute());              isAbsolute()
    System.out.println("Is it hidden? " + file.isHidden());                  isHidden()
    System.out.println("Absolute path is " +
      file.getAbsolutePath());                                               getAbsolutePath()
    System.out.println("Last modified on " +
      new java.util.Date(file.lastModified()));                              lastModified()
  }
}
```

Thao tác ghi với file text

- Sử dụng lớp FileWriter thực hiện ghi chuỗi ký tự ra file text

```
import java.io.*;

public class WriteATextFile {
  public static void main (String[] args) {

    try {
      FileWriter writer = new FileWriter("Hello.txt");

      writer.write("Hello!");

      writer.close();

    }
    catch (IOException e) {
      e.printStackTrace();
    }
  }
}
```

import gói java.io để dùng FileWriter

mở file để ghi

các lệnh I/O đều có thể gây ngoại lệ nên phải có try/catch

write() ghi một đối tượng String ra file

đóng file khi xong việc

Bộ nhớ đệm

- Bộ nhớ đệm (buffer) cho một nơi lưu trữ tạm thời để tăng hiệu quả của thao tác đọc/ghi dữ liệu
- Bộ nhớ đệm trong java được hỗ trợ qua việc sử dụng lớp BufferWriter trong gói java.io
- Cách sử dụng BufferWriter:

```
BufferWriter writer =new BufferWriter(newFileWriter(aFile));
```

Thao tác đọc file text

- Sử dụng lớp FileReader thực hiện đọc file text và BufferReader để tăng hiệu quả đọc.

```
import java.io.*;

public class ReadATextFile {
  public static void main (String[] args) {

    try {
      File inFile = new File("Hello.txt");
      FileReader fileReader = new FileReader(inFile);

      BufferReader reader = new BufferReader(fileReader);

      String line = null;

      while ((line = reader.readLine()) != null) {
        System.out.println(line);
      }
      reader.close();

    }
    catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

nối FileReader với một file text

nối BufferReader với FileReader

đọc từng dòng cho đến khi không đọc được gì nữa (hết file)

Tổng kết

- Các hoạt động nhập/xuất dữ liệu như nhập dữ liệu từ bàn phím, đọc dữ liệu từ file, ghi dữ liệu màn hình, ghi ra file, ghi ra đĩa, ghi ra máy in ...được gọi là stream (dòng).
- Có 2 kiểu luồng trong Java: dòng bye (dòng nhị phân) và dòng character (dòng văn bản)
- Lớp File thuộc gói java.io đại diện cho một file hoặc một thư mục.
- Sử dụng lớp FileWriter thực hiện ghi chuỗi ký tư ra file text
- Sử dụng lớp FileReader thực hiện đọc file text và BufferReader để tăng hiệu quả đọc.

Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: Binary File and Serialization