

Lớp File trong Java

Lớp File trong Java biểu diễn các **pathname** của các file và thư mục theo phương thức trừu tượng. Lớp này được sử dụng để tạo các file và thư mục, để tìm kiếm các file, xóa các file, ...

Đối tượng File biểu diễn file/thư mục thực sự trên đĩa. Có các constructor sau để tạo một đối tượng File trong Java:

Cú pháp sau tạo một đối tượng File mới từ một pathname trừu tượng cha và một chuỗi pathname con.

```
File(File parent, String child);
```

Cú pháp sau tạo một đối tượng File mới bởi việc biến đổi chuỗi pathname đã cho thành một pathname trừu tượng.

```
File(String pathname)
```

Cú pháp sau tạo một đối tượng File mới từ một chuỗi pathname cha và một chuỗi pathname con.

```
File(String parent, String child)
```

Cú pháp sau tạo một đối tượng File mới bởi việc biến đổi URI đã cho thành một pathname trừu tượng.

```
File(URI uri)
```

Khi bạn có đối tượng **File**, thì có các phương thức sau giúp bạn thao tác các file trong Java:

STT	Phương thức và Miêu tả
1	public String getName() Trả về tên của file hoặc thư mục đã được biểu thị bởi pathname trừu tượng này
2	public String getParent() Trả về chuỗi pathname của pathname trừu tượng cha, hoặc null nếu pathname này không là tên của một thư mục cha

3	public File getParentFile() Trả về pathname trừu tượng của pathname trừu tượng cha, hoặc null nếu pathname này không là tên của một thư mục cha
4	public String getPath() Biến đổi pathname trừu tượng này thành một chuỗi pathname
5	public boolean isAbsolute() Kiểm tra có hay không pathname trừu tượng này là absolute. Trả về true nếu là absolute, nếu không là false
6	public String getAbsolutePath() Trả về chuỗi pathname tuyệt đối của pathname trừu tượng này
7	public boolean canRead() Kiểm tra có hay không ứng dụng có thể đọc file, được biểu thị bởi pathname trừu tượng này. Trả về true nếu và chỉ nếu file đã xác định bởi pathname trừu tượng này là tồn tại và có thể được đọc từ ứng dụng; nếu không là false
8	public boolean canWrite() Kiểm tra có hay không ứng dụng có thể sửa đổi file, được biểu thị bởi pathname trừu tượng này. Trả về true nếu và chỉ nếu hệ thống file thực sự chứa một file được biểu thị bằng pathname trừu tượng này và ứng dụng được phép write tới file đó; nếu không là false
9	public boolean exists() Kiểm tra có hay không file hoặc thư mục được biểu thị bởi pathname trừu tượng này là tồn tại. Trả về true nếu và chỉ nếu file hoặc thư mục được biểu thị bởi pathname trừu tượng này là tồn tại; nếu không là false

10	public boolean isDirectory() Kiểm tra có hay không file được biểu thị bởi pathname trừu tượng này là một thư mục. Trả về true nếu và chỉ nếu file được biểu thị bởi pathname trừu tượng này là tồn tại và là một thư mục; nếu không là false
11	public boolean isFile() Kiểm tra có hay không file được biểu thị bởi pathname trừu tượng này là một normal file. Một file là normal nếu nó không là một thư mục và ngoài ra, thỏa mãn các tiêu chuẩn khác phụ thuộc vào hệ thống. Bất kỳ một non-directory file được tạo bởi một ứng dụng Java được bảo đảm là một normal file. Trả về true nếu và chỉ nếu file được biểu thị bởi pathname trừu tượng này là một normal file; nếu không là false .
12	public long lastModified() Trả về time mà file được biểu thị bởi pathname trừu tượng này được sửa đổi lần cuối cùng. Trả về giá trị long biểu diễn thời gian mà file được sửa đổi lần cuối cùng, được tính toán bằng mili giây từ 00:00:00 GMT, 1/1/1970, hoặc 0L nếu file không tồn tại hoặc nếu một I/O error xuất hiện
13	public long length() Trả về độ dài của file được biểu thị bởi pathname trừu tượng này. Giá trị trả về là không rõ ràng nếu pathname này biểu thị một thư mục
14	public boolean createNewFile() throws IOException Tạo một file mới, trống với tên pathname trừu tượng này nếu và chỉ nếu một file với tên này chưa tồn tại. Trả về true nếu và chỉ nếu file với tên này không tồn tại và được tạo thành công; nếu file với tên này đã tồn tại, thì trả về false
15	public boolean delete() Xóa file hoặc thư mục được biểu thị bởi pathname trừu tượng này. Nếu pathname này biểu thị một thư mục, thì thư mục phải là trống để bị xóa. Trả về true nếu và chỉ nếu

	file hoặc thư mục bị xóa thành công; nếu không là false
16	public void deleteOnExit() Yêu cầu rằng file hoặc thư mục được biểu thị bởi pathname trừu tượng này để bị xóa khi thiết bị ảo kết thúc
17	public String[] list() Trả về một mảng các chuỗi chỉ các file và thư mục trong thư mục được biểu thị bởi pathname trừu tượng này
18	public String[] list(FilenameFilter filter) Trả về một mảng các chuỗi chỉ các file và thư mục trong thư mục được biểu thị bởi pathname trừu tượng này mà thỏa mãn filter đã cho
20	public File[] listFiles() Trả về một mảng các pathname trừu tượng biểu thị các file trong thư mục được biểu thị bởi pathname trừu tượng này
21	public File[] listFiles(FileFilter filter) Trả về một mảng các pathname trừu tượng biểu thị các file và thư mục trong thư mục được biểu thị bởi pathname trừu tượng này mà thỏa mãn filter đã cho
22	public boolean mkdir() Tạo thư mục được chỉ bởi pathname trừu tượng này. Trả về true nếu và chỉ nếu thư mục được tạo; nếu không là false
23	public boolean mkdirs() Tạo thư mục được chỉ bởi pathname trừu tượng này, bao gồm bất kỳ thư mục cha nào cần thiết nhưng không tồn tại. Trả về true nếu và chỉ nếu thư mục được tạo, cùng với tất cả thư mục cha cần thiết; nếu không là false

24	public boolean renameTo(File dest) Đổi tên file được biểu thị bởi pathname trừu tượng này. Trả về true nếu và chỉ nếu việc đổi tên thành công; nếu không là false
25	public boolean setLastModified(long time) Thiết lập thời gian chỉnh sửa cuối cùng của file hoặc thư mục được chỉ bởi pathname trừu tượng này. Trả về true nếu và chỉ nếu hoạt động này thành công; nếu không là false
26	public boolean setReadOnly() Đánh dấu file hoặc thư mục được chỉ bởi pathname trừu tượng này để mà các hoạt động read-only được cho phép. Trả về true nếu hoạt động này thành công; nếu không là false
27	public static File createTempFile(String prefix, String suffix, File directory) throws IOException Tạo một file trống trong thư mục đã cho, bởi sử dụng các chuỗi tiền tố và hậu tố đã cung cấp để tạo tên của nó. Trả về pathname trừu tượng biểu thị một file trống mới được tạo
28	public static File createTempFile(String prefix, String suffix) throws IOException Tạo một file trống trong thư mục temporary-file, bởi sử dụng tiền tố và hậu tố đã cung cấp để tạo tên của nó. Triệu hồi phương thức này tương đương với gọi phương thức createTempFile(prefix, suffix, null). Trả về pathname trừu tượng biểu thị một file trống mới được tạo
29	public int compareTo(File pathname) So sánh 2 pathname trừu tượng theo từ điển. Trả về 0 nếu tham số là cân bằng với pathname trừu tượng này, một giá trị nhỏ hơn 0 nếu pathname trừu tượng này nhỏ hơn tham số theo từ điển, hoặc một giá trị lớn hơn 0 nếu pathname trừu tượng này lớn hơn tham số theo từ điển

30	public int compareTo(Object o) So sánh pathname trừu tượng này với đối tượng khác. Trả về 0 nếu tham số là cân bằng với pathname trừu tượng này, một giá trị nhỏ hơn 0 nếu pathname trừu tượng nhỏ hơn tham số theo từ điển, hoặc một giá trị lớn hơn 0 nếu pathname trừu tượng này lớn hơn tham số theo từ điển
31	public boolean equals(Object obj) Kiểm tra pathname trừu tượng này có cân bằng với đối tượng đã cung cấp không. Trả về true nếu và chỉ nếu tham số là không null và là một pathname trừu tượng mà biểu thị file hoặc thư mục giống như với pathname trừu tượng này
32	public String toString() Trả về chuỗi pathname của pathname trừu tượng này. Đây là chuỗi được trả về bởi phương thức getPath()

Ví dụ

Sau đây là ví dụ minh họa đối tượng File trong Java:

```
package com.tutorialspoint;

import java.io.File;

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

        File f = null;
        String[] strs = {"test1.txt", "test2.txt"};
        try{
            // for each string in string array
            for(String s:strs )
            {
                // create new file
            }
        }
    }
}
```

```
f= new File(s);

// true if the file is executable
boolean bool = f.canExecute();

// find the absolute path
String a = f.getAbsolutePath();

// prints absolute path
System.out.print(a);

// prints
System.out.println(" is executable: "+ bool);
}
}catch(Exception e){
    // if any I/O error occurs
    e.printStackTrace();
}
}
```

Giả sử có một file có thể thực thi là test1.txt và file khác là test2.txt là không thể thực thi trong thư mục hiện tại. Bây giờ chúng ta biên dịch và chạy chương trình trên, nó sẽ cho kết quả sau:

```
test1.txt is executable: true
test2.txt is executable: false
```