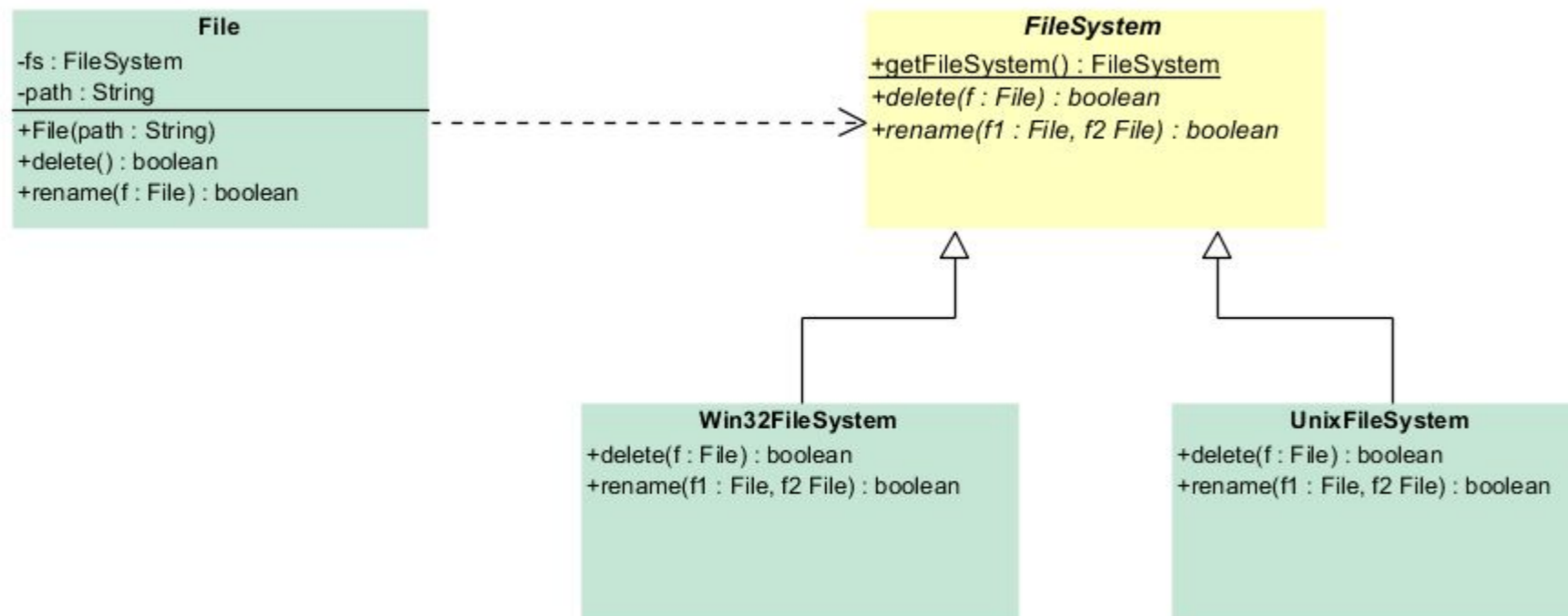




# V. Ввод - вывод

## 1. Файлы



```
package java.io;

public class File {

    private String path;
    static private FileSystem fs = FileSystem.getFileSystem();

    public File(String pathname) {
        if (pathname == null) {
            throw new NullPointerException();
        }
        this.path = fs.normalize(pathname);
        this.prefixLength = fs.prefixLength(this.path);
    }

    public File(String parent, String child)
    public File(File parent, String child)

    public boolean exists()

    public String getPath() {
        return path;
    }

    ...
}
```



Класс File из пакета java.io представляет путь к директории или файлу и содержит методы для получения информации о файловой системе и внесения изменений в файловую систему. Конструкторы позволяют задавать путь к файлу или директории. Директория или файл чей путь хранится в File может не существовать.

```
public class FileNameDemo {  
  
    public static void main(String[] args) {  
  
        File file = new File("Hello World!");  
        System.out.println("File \""+ file.getPath() +"\" exists? " + file.exists());  
    }  
}
```

```
File "Hello World!" exists? false
```

# Получение информации о пути к файлу и об имени файла

```
public class File {  
  
    ...  
  
    public String getName()  
  
    public File getParentFile()  
    public String getParent()  
  
    public File getAbsolutePath()  
    public String getAbsolutePath()  
  
}
```



Используя методы класса File можно запросить информацию об имени файла или директории и о пути к файлу или директории.

```
public class PathInfoDemo {  
  
    public static void main(String[] args) {  
  
        File file = new File("I:\\FileIO\\somefile.txt");  
  
        System.out.println("File name: " + file.getName());  
        System.out.println("File directory: " + file.getParent());  
        System.out.println("File full path: " + file.getAbsolutePath());  
  
    }  
}
```

```
File name: somefile.txt  
File directory: I:\\FileIO  
File full path: I:\\FileIO\\somefile.txt
```



# Создание, удаление и переименование файлов

```
package java.io;

public class File {

    ...
    public boolean createNewFile()
    public boolean delete()
    public boolean renameTo(File dest)
}
```



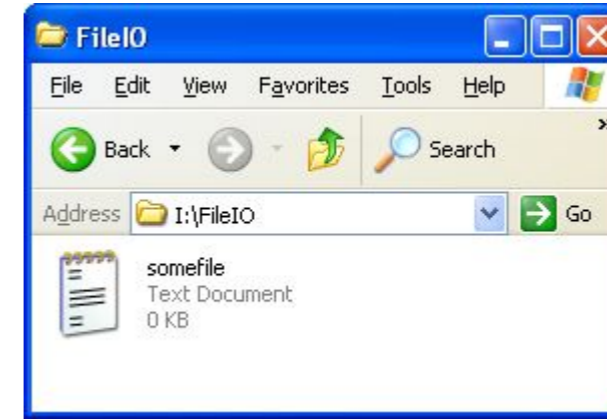
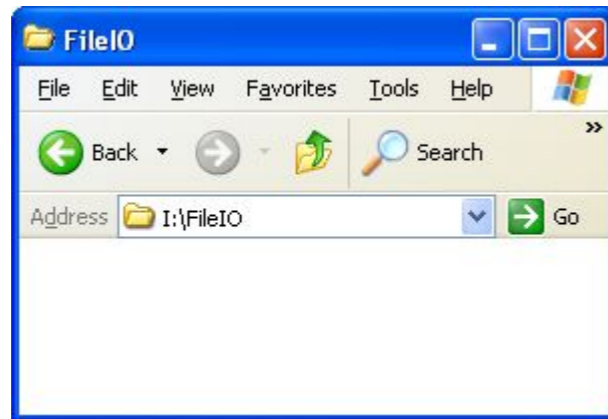
Создать новый файл можно с помощью метода `createNewFile`. Попытка создать файл в несуществующей директории приведёт к выбрасыванию исключения `IOException`. Удалить файл можно вызвав метод `delete`. Если файл успешно удалён `delete` вернёт `true` если удаляемый файл не существует `delete` вернёт `false`. Переименовать файл можно с помощью метода `renameTo`. В случае успешного переименования он вернёт `true` в случае если переименуемый файл не существует `renameTo` вернёт `false`.

```
public class CreateFileDemo {  
  
    public static void main(String[] args) {  
  
        try {  
  
            File file = new File("I:\\FileIO\\somefile.txt");  
  
            if (file.createNewFile()) {  
                System.out.println("File " + file.getName() + " was created");  
            } else {  
                System.out.println("Create operation failed.");  
            }  
  
        } catch (IOException e) {  
  
            e.printStackTrace();  
        }  
  
    }  
  
}
```



Обычно файлы не создаются явно с помощью `createNewFile`. Они создаются при помощи классов потомков `OutputStream` или `Writer`.

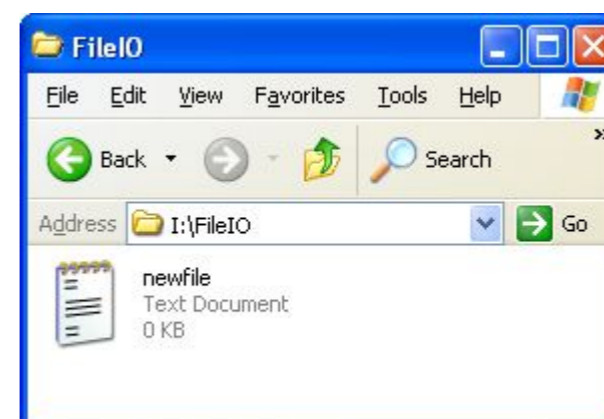
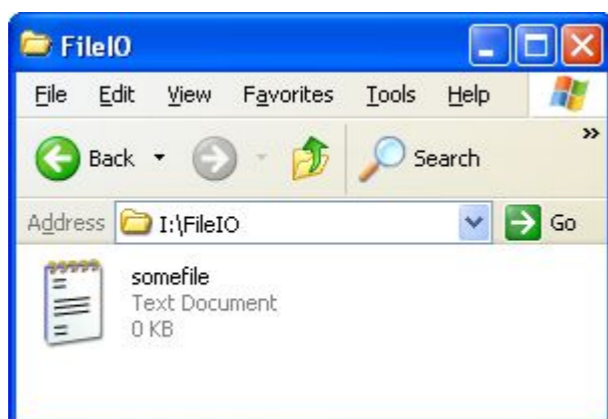
# Создание файла



File somefile.txt was created

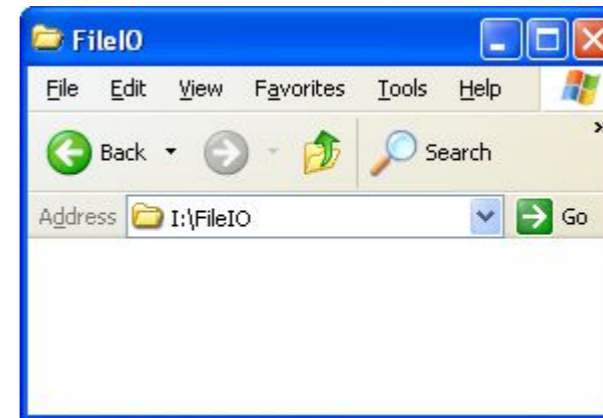
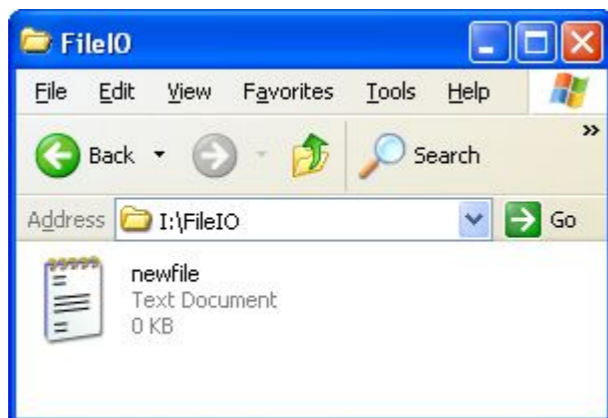
```
public class RenameFileDemo {  
  
    public static void main(String[] args) {  
  
        File oldfile = new File("I:\\FileIO\\somefile.txt");  
        File newfile = new File("I:\\FileIO\\newfile.txt");  
  
        if (oldfile.renameTo(newfile)) {  
            System.out.println("Rename succesful " + oldfile.getName() + " was renamed to " +  
newfile.getName());  
        } else {  
            System.out.println("Rename failed");  
        }  
    }  
}
```

# Переименование файла



Rename succesful somefile.txt was renamed to newfile.txt

```
public class DeleteFileDemo {  
  
    public static void main(String[] args) {  
  
        File file = new File("I:\\FileIO\\somefile.txt");  
  
        if (file.delete()) {  
            System.out.println(file.getName() + " was deleted!");  
        } else {  
            System.out.println("Delete operation failed.");  
        }  
    }  
}
```



`newfile.txt was deleted!`



# Запрашивание свойств

```
package java.io;

public class File {

    ...
    public boolean isDirectory()
    public boolean isFile()

    public boolean isHidden()
    public boolean canRead ()
    public boolean canWrite ()
    public boolean canExecute ()

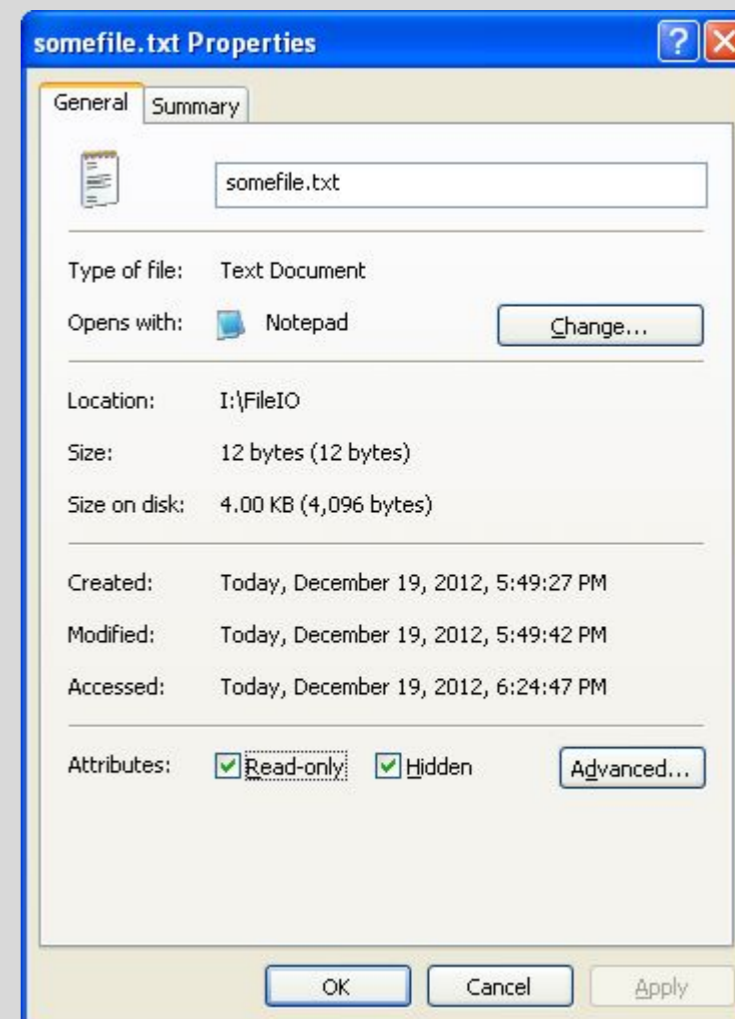
    public long lastModified()
    public long length()
}
```



Получить информацию о том является ли файл скрытым и изменяемым можно с помощью методов `isHidden` и `canWrite` соответственно. Время последней модификации можно узнать с помощью метода `lastModified`. Длину файла можно узнать с помощью метода `length`.

```
public class ListAttributesDemo {  
  
    public static void main(String[] args) {  
  
        File file = new File("I:\\FileIO\\somefile.txt");  
  
        if (file.isHidden()) {  
            System.out.println("This file is hidden");  
        } else {  
            System.out.println("This file is not hidden");  
        }  
  
        if (file.canRead()) {  
            System.out.println("This file is readable");  
        } else {  
            System.out.println("This file is not readable");  
        }  
  
        if (file.canWrite()) {  
            System.out.println("This file is writable");  
        } else {  
            System.out.println("This file is not writable");  
        }  
  
        if (file.canExecute()) {  
            System.out.println("This file is executable");  
        } else {  
            System.out.println("This file is not executable");  
        }  
    }  
}
```

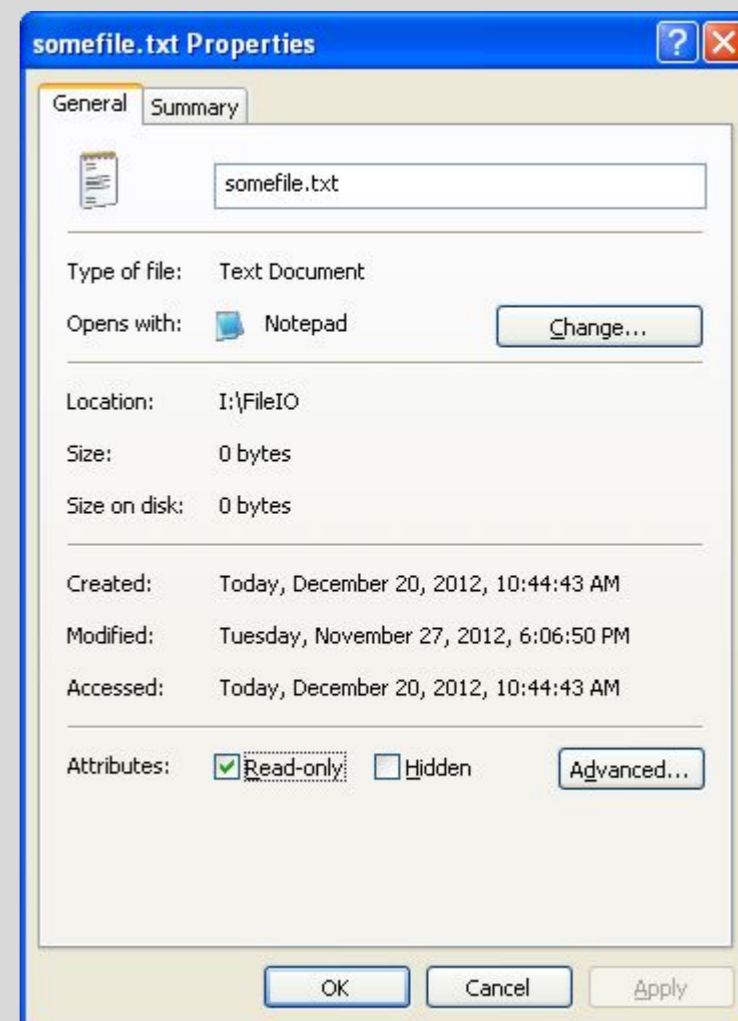
This file is hidden  
This file is readable  
This file is not writable  
This file is executable



```
public class TimeDemo {  
  
    public static void main(String[] args) {  
  
        File file = new File("F:\\FileIO\\somefile.txt");  
  
        System.out.println("Before Format : " + file.lastModified());  
  
        SimpleDateFormat sdf = new SimpleDateFormat("MM/dd/yyyy HH:mm:ss");  
  
        System.out.println("After Format : " + sdf.format(file.lastModified()));  
    }  
}
```

# Запрашивание времени последнего изменения

Before Format : 1354014410437  
After Format : 11/27/2012 18:06:50



# Изменение свойств

```
public class File {  
  
    ...  
    public boolean setReadOnly()  
    public boolean setWritable(boolean writable)  
    public boolean setExecutable(boolean executable)  
  
    public boolean setLastModified(long time)  
  
}
```



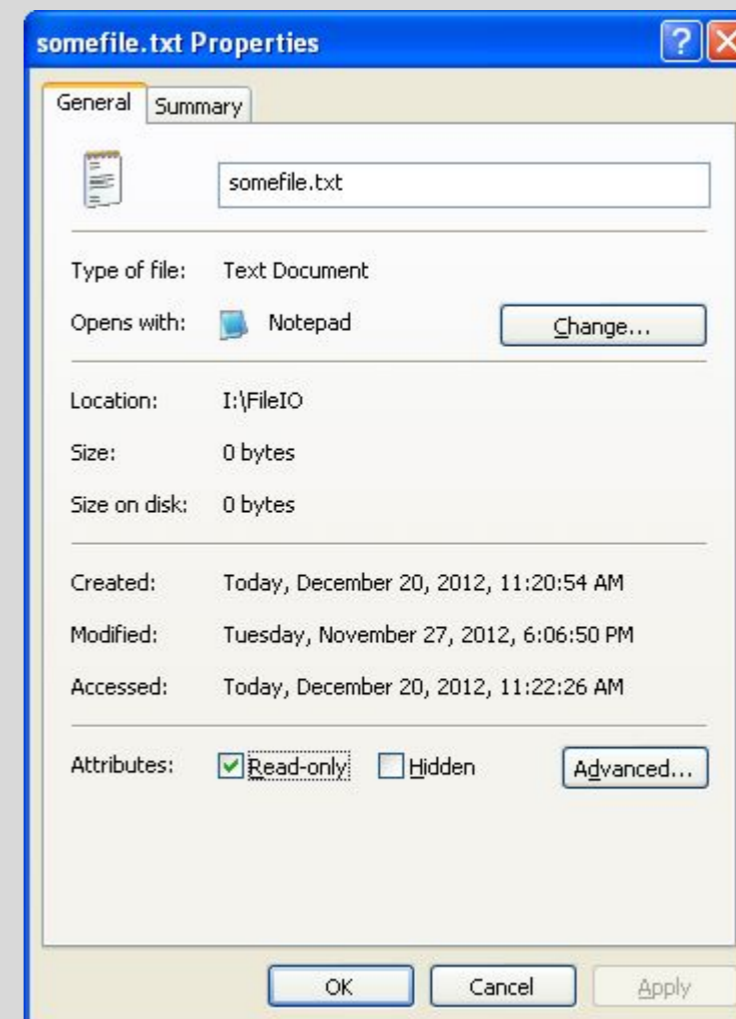
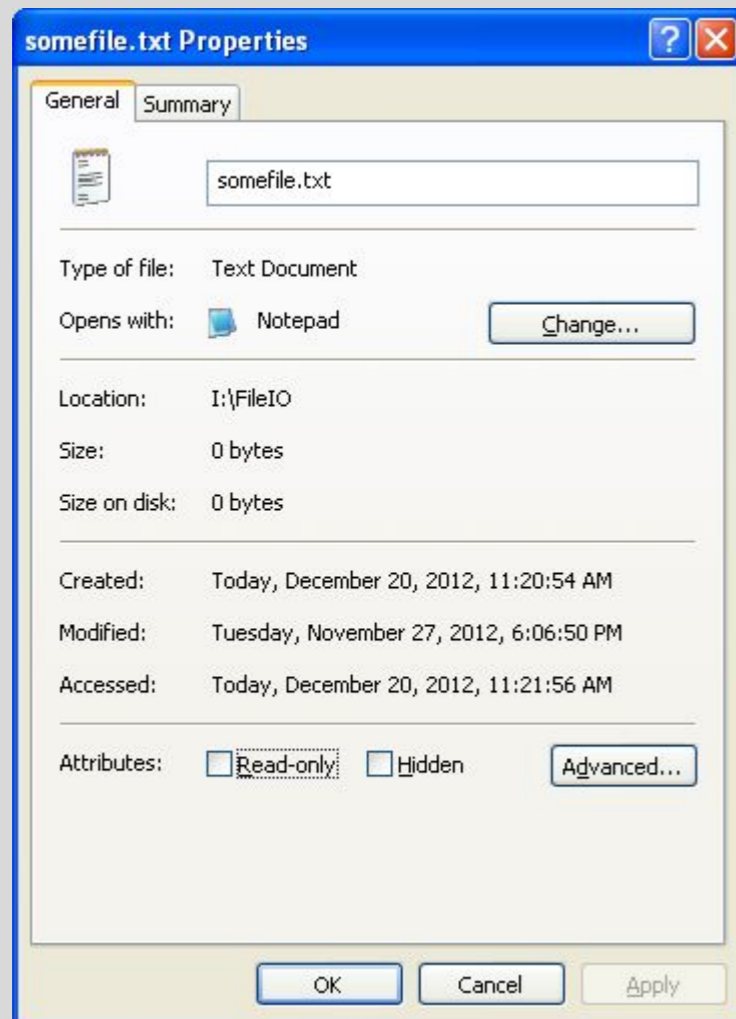
Файл можно сделать изменяемым или неизменяемым с помощью методов `setReadOnly` и `setWritable`. Время последней модификации можно задать с помощью `setLastModified`. Сделать файл скрытым или не скрытым нельзя.



```
public class SetAttributesDemo {  
  
    public static void main(String[] args) {  
  
        File file = new File("I:\\FileIO\\somefile.txt");  
  
        file.setWritable(false);  
        //file.setReadOnly();  
  
        if (file.canRead()) {  
            System.out.println("This file is readable");  
        } else {  
            System.out.println("This file is not readable");  
        }  
  
        if (file.canWrite()) {  
            System.out.println("This file is writable");  
        } else {  
            System.out.println("This file is not writable");  
        }  
    }  
}
```

# Изменение атрибутов

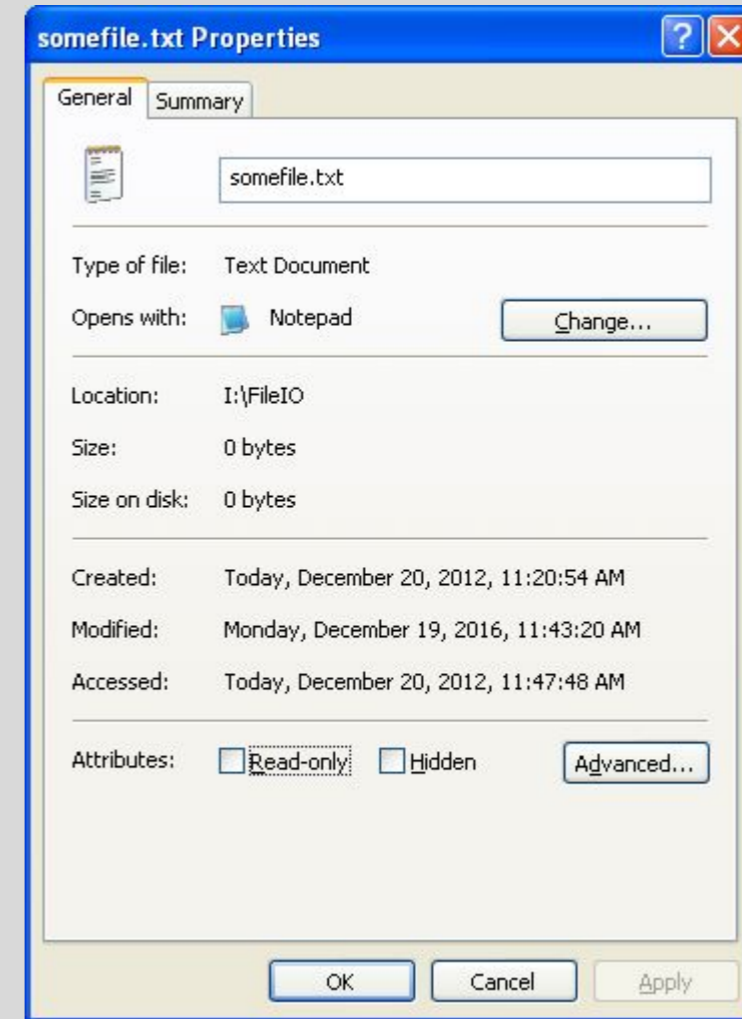
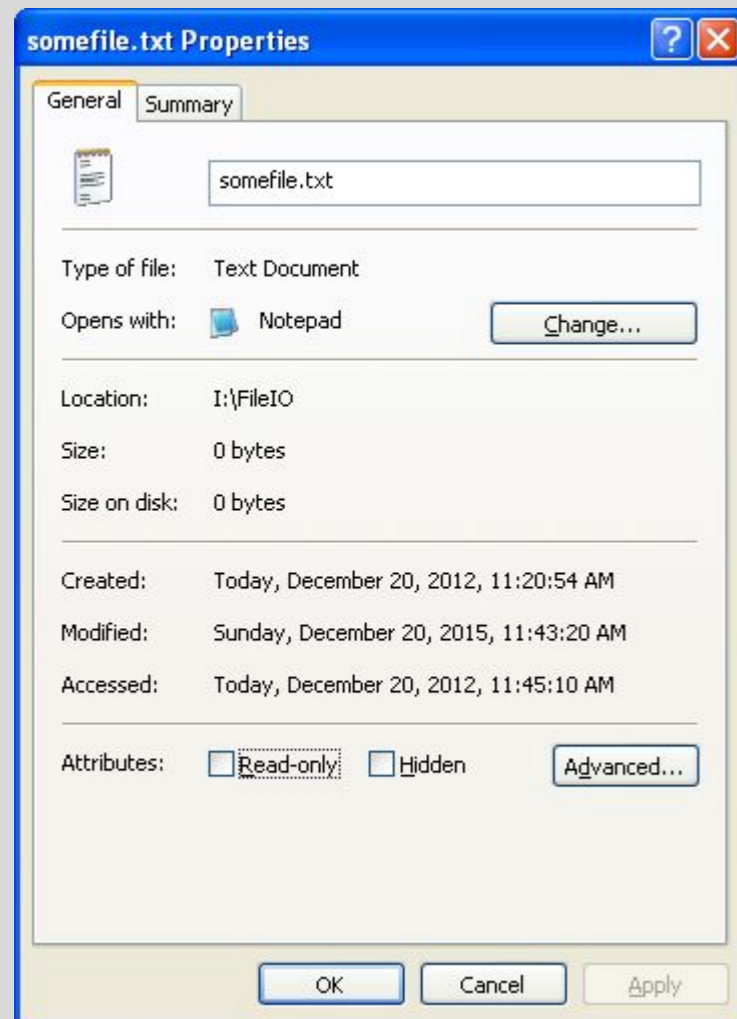
This file is readable  
This file is not writable



```
public class SetTimeDemo {  
  
    public static void main(String[] args) {  
  
        File file = new File("I:\\FileIO\\somefile.txt");  
  
        SimpleDateFormat sdf = new SimpleDateFormat("MM/dd/yyyy HH:mm:ss");  
        System.out.println("Last modified : " + sdf.format(file.lastModified()));  
  
        long mils = file.lastModified();  
        mils += 1000L * 60 * 60 * 24 * 365;  
  
        if (file.setLastModified(mils)) {  
            System.out.println("File " + file.getName()  
                + " last modified time was changed");  
        } else {  
            System.out.println("Last modified change operation failed.");  
        }  
        System.out.println("Last modified : " + sdf.format(file.lastModified()));  
    }  
}
```

# Изменение времени последней модификации

Last modified : 12/20/2015 11:43:20  
File somefile.txt last modified time was changed  
Last modified : 12/19/2016 11:43:20



# Работа с директориями

```
public class File {  
  
    ...  
    public boolean mkdir()  
    public boolean mkdirs()  
    public boolean isDirectory()  
    public File[] listFiles()  
    public boolean delete()  
    public boolean renameTo(File dest)  
}
```



Директорию можно создать с помощью метода `mkdir`. Несколько вложенных директорий можно создать с помощью метода `mkdirs`. Получить список файлов и директорий можно с помощью метода `listFiles`. Удаление и переименование директорий возможно с помощью методов `delete` и `renameTo`.

```
public class ListDirDemo {  
  
    public static void main(String args[]) {  
  
        String dirname = "F:\\eclipse";  
        File file = new File(dirname);  
  
        if (file.isDirectory()) {  
            System.out.println("Directory of " + dirname);  
            File[] files = file.listFiles();  
            for (int i = 0; i < files.length; i++) {  
  
                if (files[i].isDirectory()) {  
                    System.out.println(files[i].getAbsolutePath()  
                        + " is a directory");  
                } else {  
                    System.out.print(files[i] + " is a file ");  
                    System.out.println(" is hidden? : " + files[i].isHidden());  
                }  
            }  
        } else {  
            System.out.println(dirname + " is not a directory");  
        }  
    }  
}
```

# Получение списка файлов и директорий

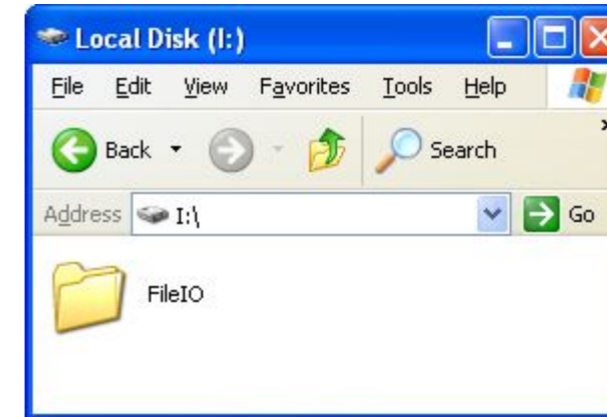
```
Directory of F:\eclipse
F:\eclipse\.eclipseproduct is a file is hidden? : false
F:\eclipse\artifacts.xml is a file is hidden? : false
F:\eclipse\configuration is a directory
F:\eclipse\dropins is a directory
F:\eclipse\eclipse.exe is a file is hidden? : false
F:\eclipse\eclipse.ini is a file is hidden? : false
F:\eclipse\eclipsesec.exe is a file is hidden? : false
F:\eclipse\epl-v10.html is a file is hidden? : false
F:\eclipse\features is a directory
F:\eclipse\hs_err_pid3840.log is a file is hidden? : false
F:\eclipse\notice.html is a file is hidden? : false
F:\eclipse\p2 is a directory
F:\eclipse\plugins is a directory
F:\eclipse\readme is a directory
```





```
public class CreateDirDemo {  
  
    public static void main(String args[]) {  
  
        String dirname = "I:\\FileIO";  
        File dir = new File(dirname);  
  
        if (dir.exists()) {  
            System.out.println("Directory of " + dirname + " exists");  
        }  
        else {  
            System.out.println("Directory of " + dirname + " does not exist");  
        }  
  
        if (dir.mkdir()) {  
            System.out.println("Directory " + dir.getName() + " was created");  
        } else {  
            System.out.println("Create operation failed.");  
        }  
    }  
}
```

# Создание директорий



```
Directory of I:\FileIO does not exist  
Directory FileIO was created
```

**Спасибо**

**Россия, 127018,  
Москва, ул. Полковая 3, стр. 14  
Тел.: +7(495) 780 7575, 789 9339  
Факс: +7(495) 780 7576, 789 9338  
[info@diasoft.ru](mailto:info@diasoft.ru), [www.diasoft.ru](http://www.diasoft.ru)**