一，FileWritter写入文件

FileWritter, 字符流写入字符到文件。默认情况下，它会使用新的内容取代所有现有的内容，然而，当指定一个true （布尔）值作为FileWritter构造函数的第二个参数，它会保留现有的内容，并追加新内容在文件的末尾。

1. 替换所有现有的内容与新的内容。

new FileWriter(file);2. 保留现有的内容和附加在该文件的末尾的新内容。

复制代码代码如下:

new FileWriter(file,true);

追加文件示例   
一个文本文件，命名为“javaio-appendfile.txt”，并包含以下内容。

ABC Hello追加新内容 new FileWriter(file,true)

复制代码代码如下:

package com.yiibai.file;  
  
import java.io.File;  
import java.io.FileWriter;  
import java.io.BufferedWriter;  
import java.io.IOException;  
  
public class AppendToFileExample   
{  
    public static void main( String[] args )  
    {   
     try{  
      String data = " This content will append to the end of the file";  
  
      File file =new File("javaio-appendfile.txt");  
  
      //if file doesnt exists, then create it  
      if(!file.exists()){  
       file.createNewFile();  
      }  
  
      //true = append file  
      FileWriter fileWritter = new FileWriter(file.getName(),true);  
             BufferedWriter bufferWritter = new BufferedWriter(fileWritter);  
             bufferWritter.write(data);  
             bufferWritter.close();  
  
         System.out.println("Done");  
  
     }catch(IOException e){  
      e.printStackTrace();  
     }  
    }  
}

结果   
现在，文本文件“javaio-appendfile.txt”内容更新如下：

ABC Hello This content will append to the end of the file

二，BufferedWriter写入文件

缓冲字符（BufferedWriter ）是一个字符流类来处理字符数据。不同于字节流（数据转换成字节），你可以直接写字符串，数组或字符数据保存到文件。

复制代码代码如下:

package com.yiibai.iofile;  
  
import java.io.BufferedWriter;  
import java.io.File;  
import java.io.FileWriter;  
import java.io.IOException;  
  
public class WriteToFileExample {  
 public static void main(String[] args) {  
  try {  
  
   String content = "This is the content to write into file";  
  
   File file = new File("/users/mkyong/filename.txt");  
  
   // if file doesnt exists, then create it  
   if (!file.exists()) {  
    file.createNewFile();  
   }  
  
   FileWriter fw = new FileWriter(file.getAbsoluteFile());  
   BufferedWriter bw = new BufferedWriter(fw);  
   bw.write(content);  
   bw.close();  
  
   System.out.println("Done");  
  
  } catch (IOException e) {  
   e.printStackTrace();  
  }  
 }  
}

三，FileOutputStream写入文件

文件输出流是一种用于处理原始二进制数据的字节流类。为了将数据写入到文件中，必须将数据转换为字节，并保存到文件。请参阅下面的完整的例子。

复制代码代码如下:

package com.yiibai.io;  
  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
  
public class WriteFileExample {  
 public static void main(String[] args) {  
  
  FileOutputStream fop = null;  
  File file;  
  String content = "This is the text content";  
  
  try {  
  
   file = new File("c:/newfile.txt");  
   fop = new FileOutputStream(file);  
  
   // if file doesnt exists, then create it  
   if (!file.exists()) {  
    file.createNewFile();  
   }  
  
   // get the content in bytes  
   byte[] contentInBytes = content.getBytes();  
  
   fop.write(contentInBytes);  
   fop.flush();  
   fop.close();  
  
   System.out.println("Done");  
  
  } catch (IOException e) {  
   e.printStackTrace();  
  } finally {  
   try {  
    if (fop != null) {  
     fop.close();  
    }  
   } catch (IOException e) {  
    e.printStackTrace();  
   }  
  }  
 }  
}

//更新的JDK7例如，使用新的“尝试资源关闭”的方法来轻松处理文件。

package com.yiibai.io;  
  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
  
public class WriteFileExample {  
 public static void main(String[] args) {  
  
  File file = new File("c:/newfile.txt");  
  String content = "This is the text content";  
  
  try (FileOutputStream fop = new FileOutputStream(file)) {  
  
   // if file doesn't exists, then create it  
   if (!file.exists()) {  
    file.createNewFile();  
   }  
  
   // get the content in bytes  
   byte[] contentInBytes = content.getBytes();  
  
   fop.write(contentInBytes);  
   fop.flush();  
   fop.close();  
  
   System.out.println("Done");  
  
  } catch (IOException e) {  
   e.printStackTrace();  
  }  
 }  
}