

2021/12/10 上午2:40	TestBufferedStreams.java	2021/12/10 上午2:40	TestDataStreams.java
1 package ex7;	1 package ex7;	1 package ex7;	1 package ex7;
2	2	2	2
3 import java.io.*;	3 import java.io.*;	3 import java.io.*;	3 import java.io.*;
4	4	4	4
5 public class TestBufferedStreams {	5 public class TestBufferedStreams {	5 public class TestDataStreams {	5 public class TestDataStreams {
6	6	6	6
7 /** Main method	7 /** Main method	7 public static void main(String[] args) throws IOException {	7 public static void main(String[] args) throws IOException {
8 @param args[0] for sourcefile	8 @param args[0] for sourcefile	8 // Create an output stream for file temp.dat	8 // Create an output stream for file temp.dat
9 @param args[1] for target file	9 @param args[1] for target file	9 DataOutputStream output = new DataOutputStream(new	9 DataOutputStream output = new DataOutputStream(new
10 */	10 */	9 FileOutputStream("temp.dat"));	9 FileOutputStream("temp.dat"));
11 public static void main(String[] args) throws IOException {	11 public static void main(String[] args) throws IOException {	10	10
12 // Check command-line parameter usage	12 // Check command-line parameter usage	11 // Write student test scores to the file	11 // Write student test scores to the file
13 if (args.length != 2) {	13 if (args.length != 2) {	12 output.writeUTF("John");	12 output.writeUTF("John");
14 System.out.println(	14 System.out.println(	13 output.writeDouble(85.5);	13 output.writeDouble(85.5);
15 "Usage: java CopyFile sourceFile targetFile");	15 "Usage: java CopyFile sourceFile targetFile");	14 output.writeUTF("Jim");	14 output.writeUTF("Jim");
16 System.exit(0);	16 System.exit(0);	15 output.writeDouble(185.5);	15 output.writeDouble(185.5);
17 }	17 }	16 output.writeUTF("George");	16 output.writeUTF("George");
18	18	17 output.writeDouble(105.25);	17 output.writeDouble(105.25);
19 // Check if source file exists	19 // Check if source file exists	18	18
20 File sourceFile = new File(args[0]);	20 File sourceFile = new File(args[0]);	19 // Close output stream	19 // Close output stream
21 if (!sourceFile.exists()) {	21 if (!sourceFile.exists()) {	20 output.close();	20 output.close();
22 System.out.println("Source file " + args[0] + " not exist");	22 System.out.println("Source file " + args[0] + " not exist");	21	21
23 System.exit(0);	23 System.exit(0);	22 // Create an input stream for file temp.dat	22 // Create an input stream for file temp.dat
24 }	24 }	23 DataInputStream input = new DataInputStream(new FileInputStream("temp.dat"));	23 DataInputStream input = new DataInputStream(new FileInputStream("temp.dat"));
25	25	24	24
26 // Check if target file exists	26 // Check if target file exists	25 // Read student test scores from the file	25 // Read student test scores from the file
27 File targetFile = new File(args[1]);	27 File targetFile = new File(args[1]);	26 System.out.println(input.readUTF() + " " + input.readDouble());	26 System.out.println(input.readUTF() + " " + input.readDouble());
28 if (targetFile.exists()) {	28 if (targetFile.exists()) {	27 System.out.println(input.readUTF() + " " + input.readDouble());	27 System.out.println(input.readUTF() + " " + input.readDouble());
29 System.out.println("Target file " + args[1] + " already exists");	29 System.out.println("Target file " + args[1] + " already exists");	28 System.out.println(input.readUTF() + " " + input.readDouble());	28 System.out.println(input.readUTF() + " " + input.readDouble());
30 System.exit(0);	30 System.exit(0);	29 }	29 }
31 }	31 }	30 }	30 }
32	32	31	31
33 // Create an input stream	33 // Create an input stream		
34 BufferedInputStream input = new BufferedInputStream(new	34 BufferedInputStream(input = new BufferedInputStream(new		
35 FileInputStream(sourceFile));	35 FileInputStream(sourceFile));		
36	36		
37 // Create an output stream	37 // Create an output stream		
38 BufferedOutputStream output = new BufferedOutputStream(new	38 BufferedOutputStream output = new BufferedOutputStream(new		
39 FileOutputStream(targetFile));	39 FileOutputStream(targetFile));		
40	40		
41 // Display the file size	41 // Display the file size		
42 System.out.println("The file " + args[0] + " has " + input.available() + "	42 System.out.println("The file " + args[0] + " has " + input.available() + "		
43 bytes");	43 bytes");		
44	44		
45 // Continuously read a byte from input and write it to output	45 // Continuously read a byte from input and write it to output		
46 int r;	46 int r;		
47 while ((r = input.read()) != -1) {	47 while ((r = input.read()) != -1) {		
48 output.write(r);	48 output.write(r);		
49 }	49 }		
50 // Close streams	50 // Close streams		
51 input.close();	51 input.close();		
52 output.close();	52 output.close();		
53	53		
54 System.out.println("Copy done!");	54 System.out.println("Copy done!");		
55 }	55 }		

```

1 package ex7;
2
3 import java.io.File;
4
5 public class TestFileClass {
6
7     public static void main(String[] args) {
8         File file = new File("image/us.gif");
9         System.out.println("Absolute path is " + file.getAbsolutePath());
10        System.out.println("Relative path is " + file.getPath());
11        System.out.println("Last modified on " + new
12            java.util.Date(file.lastModified()));
13        System.out.println("Does it exist? " + file.exists());
14        System.out.println("Can it be read? " + file.canRead());
15        System.out.println("Can it be written? " + file.canWrite());
16        System.out.println("Is it a directory? " + file.isDirectory());
17        System.out.println("Is it a file? " + file.isFile());
18        System.out.println("Is it hidden? " + file.isHidden());
19    }
20 }

```

```
2021/12/10 上午2:40
TestFileReaderWriter.java
60 }
61
62 // Close the output stream
63 input2.close();
64 }
65
66 }
67
```

```
2021/12/10 上午2:40
TestFileStreams.java
1 package ex7;
2
3 import java.io.*;
4
5 public class TestFileStreams {
6
7     public static void main(String[] args) throws IOException {
8         // Create an output stream to the file
9         FileOutputStream output = new FileOutputStream("temp.dat");
10
11         // Output values to the file
12         for (int i = 1; i <= 10; i++) {
13             output.write(i);
14         }
15
16         // Close the output stream
17         output.close();
18
19         // Create an input stream for the file
20         FileInputStream input = new FileInputStream("temp.dat");
21
22         // Read values from the file
23         int value;
24         while ((value = input.read()) != -1) {
25             System.out.print(value + " ");
26         }
27
28         // Close the output stream
29         input.close();
30     }
31 }
32
```

```
1 package ex7;
2
3 import java.io.*;
4 import java.util.Date;
5
6 public class TestObjectStreams {
7
8     public static void main(String[] args) throws ClassNotFoundException, IOException
9     {
10         ObjectOutputStream output = new ObjectOutputStream(new
11             FileOutputStream("object.dat"));
12
13         // Write a string, double value, and object to the file
14         output.writeUTF("John");
15         output.writeDouble(85.5);
16         output.writeObject(new Date());
17
18         // Close output stream
19         output.close();
20
21         // Create an input stream for file object.dat
22         ObjectInputStream input = new ObjectInputStream(new
23             FileInputStream("object.dat"));
24
25         // Write a string, double value, and object to the file
26         String name = input.readUTF();
27         double score = input.readDouble();
28         Date date = (Date) (input.readObject());
29         System.out.println(name + " " + score + " " + date);
30
31         // Close output stream
32         input.close();
33     }
34 }
```

```
1 package ex7;
2
3 import java.io.*;
4
5 public class TestRandomAccessFile {
6
7     public static void main(String[] args) throws IOException {
8         // Create a random access file
9         RandomAccessFile inout = new RandomAccessFile("inout.dat", "rw");
10
11         // Clear the file to destroy the old contents if exists
12         inout.setLength(0);
13
14         // Write new integers to the file
15         for (int i = 0; i < 200; i++) {
16             inout.writeInt(i);
17         }
18
19         // Display the current length of the file
20         System.out.println("Current file length is " + inout.length());
21
22         // Retrieve the first number
23         inout.seek(0); // Move the file pointer to the beginning
24         System.out.println("The first number is " + inout.readInt());
25
26         // Retrieve the second number
27         inout.seek(1 * 4); // Move the file pointer to the second number
28         System.out.println("The second number is " + inout.readInt());
29
30         // Retrieve the tenth number
31         inout.seek(9 * 4); // Move the file pointer to the tenth number
32         System.out.println("The tenth number is " + inout.readInt());
33
34         // Modify the eleventh number
35         inout.writeInt(555);
36
37         // Append a new number
38         inout.seek(inout.length()); // Move the file pointer to the end
39         inout.writeInt(999);
40
41         // Display the new length
42         System.out.println("The new length is " + inout.length());
43
44         // Retrieve the new eleventh number
45         inout.seek(10 * 4); // Move the file pointer to the eleventh number
46         System.out.println("The eleventh number is " + inout.readInt());
47
48         inout.close();
49     }
50 }
51 }
```