**Java ByteArrayOutputStream Class**

Java ByteArrayOutputStream class is used to **write common data** into multiple files. In this stream, the data is written into a byte array which can be written to multiple streams later.

The ByteArrayOutputStream holds a copy of data and forwards it to multiple streams.

The buffer of ByteArrayOutputStream automatically grows according to data.

**Java ByteArrayOutputStream class declaration**

Let's see the declaration for Java.io.ByteArrayOutputStream class:

1. **public** **class** ByteArrayOutputStream **extends** OutputStream

**Java ByteArrayOutputStream class constructors**

|  |  |
| --- | --- |
| **Constructor** | **Description** |
| ByteArrayOutputStream() | Creates a new byte array output stream with the initial capacity of 32 bytes, though its size increases if necessary. |
| ByteArrayOutputStream(int size) | Creates a new byte array output stream, with a buffer capacity of the specified size, in bytes. |

**Java ByteArrayOutputStream class methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| int size() | It is used to returns the current size of a buffer. |
| byte[] toByteArray() | It is used to create a newly allocated byte array. |
| String toString() | It is used for converting the content into a string decoding bytes using a platform default character set. |
| String toString(String charsetName) | It is used for converting the content into a string decoding bytes using a specified charsetName. |
| void write(int b) | It is used for writing the byte specified to the byte array output stream. |
| void write(byte[] b, int off, int len | It is used for writing **len** bytes from specified byte array starting from the offset **off** to the byte array output stream. |
| void writeTo(OutputStream out) | It is used for writing the complete content of a byte array output stream to the specified output stream. |
| void reset() | It is used to reset the count field of a byte array output stream to zero value. |
| void close() | It is used to close the ByteArrayOutputStream. |

**EXAMPLE**

|  |  |
| --- | --- |
| **package** com.company;  **import** java.io.\*;  **public class** Main {   **public static void** main(String[] args) {  **try** {  FileOutputStream fileOutputStream = **new** FileOutputStream(**"D:\\Cat.txt"**);  FileOutputStream fileOutputStream1 = **new** FileOutputStream(**"D:\\Cat2.txt"**);   ByteArrayOutputStream byteArrayOutputStream = **new** ByteArrayOutputStream();  **byte**[] b = **new byte**[6];  b[0] = 65;  b[1] = 66;  b[2] = 70;  b[3] = 67;  b[4] = 68;  b[5] = 69;    byteArrayOutputStream.write(b);  byteArrayOutputStream.writeTo(fileOutputStream);  byteArrayOutputStream.writeTo(fileOutputStream1);  byteArrayOutputStream.flush();  byteArrayOutputStream.close();  fileOutputStream1.close();  fileOutputStream.close();  } **catch** (Exception ex){  System.***out***.println(**"Message : "** + ex);  }  } } | **ABFCDE**  **ABFCDE** |

**byte[] toByteArray()**

|  |  |
| --- | --- |
| **package** com.company;  **import** java.io.\*;  **public class** Main {   **public static void** main(String[] args) **throws** Exception{  **byte**[] bs = {65, 66, 67, 68, 69};  ByteArrayOutputStream baos = **null**;   **try** {   *// create new ByteArrayOutputStream* baos = **new** ByteArrayOutputStream();   *// write byte array to the output stream* baos.write(bs);   *// for each byte in the buffer* **for** (**byte** b : baos.toByteArray()) {   *// write byte in to output stream* baos.write(b);   *// print every byte* System.***out***.println(b);  }  } **catch** (Exception ex){  System.***out***.println(**"Message : "** + ex);  }  } } | **65**  **66**  **67**  **68**  **69** |