

Beginners Book: Java Basics

Lesson 5: Data Types In Java

In Java there are eight primitive data types:

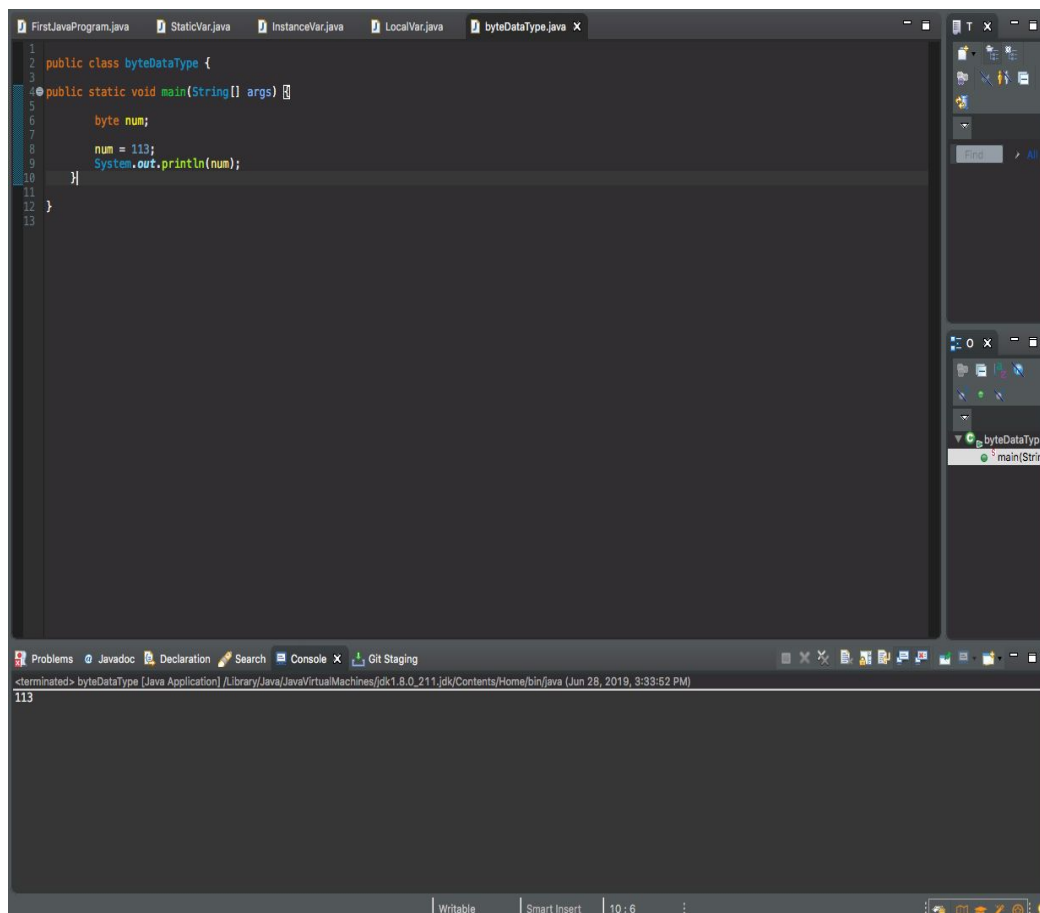
byte:

A byte can hold whole number between -128 and 127. They are mostly used to save memory and when you are certain that the numbers would be in the limit specified by byte data type.

Default Size: 1 byte

Default Value: 0

Example:



```
1 public class byteDataType {
2
3
4 public static void main(String[] args) {
5
6     byte num;
7
8     num = 113;
9     System.out.println(num);
10
11 }
12
13 }
```

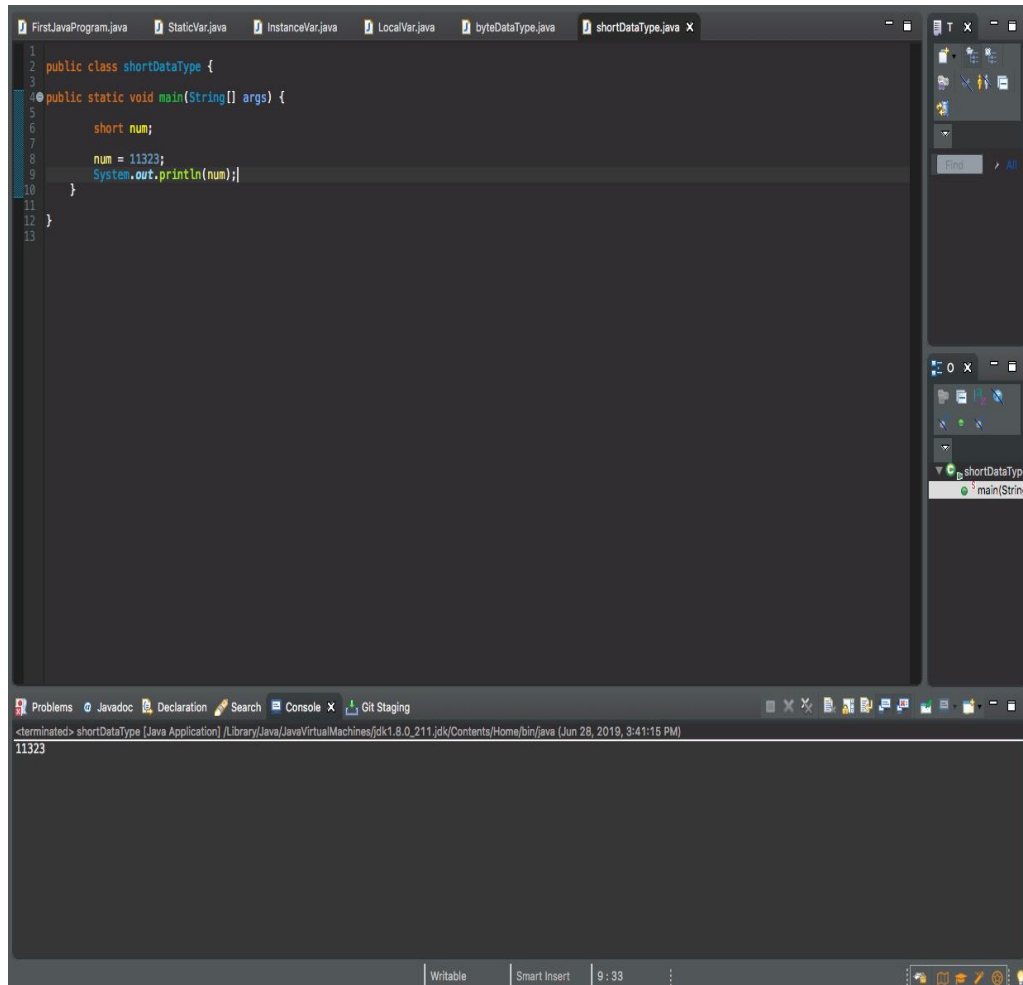
The screenshot shows an IDE with a Java file named `byteDataType.java`. The code defines a public class `byteDataType` with a `main` method. Inside the `main` method, a `byte` variable `num` is declared and assigned the value 113. The value of `num` is then printed to the console using `System.out.println(num);`. The IDE's output window at the bottom shows the result of the program execution: `113`.

short:

A short is greater in size than byte and less than integer. Its range is -32,768 to 32767.

Default Size: 2 byte

Example:



The screenshot shows an IDE with a Java file named `shortDataType.java`. The code defines a class `shortDataType` with a `main` method. Inside the `main` method, a `short` variable `num` is declared and assigned the value `11323`, which is then printed to the console. The IDE's output window at the bottom shows the result of the program execution, which is `11323`.

```
1 public class shortDataType {  
2  
3  
4 public static void main(String[] args) {  
5  
6     short num;  
7  
8     num = 11323;  
9     System.out.println(num);  
10  
11 }  
12  
13 }
```

Output: 11323

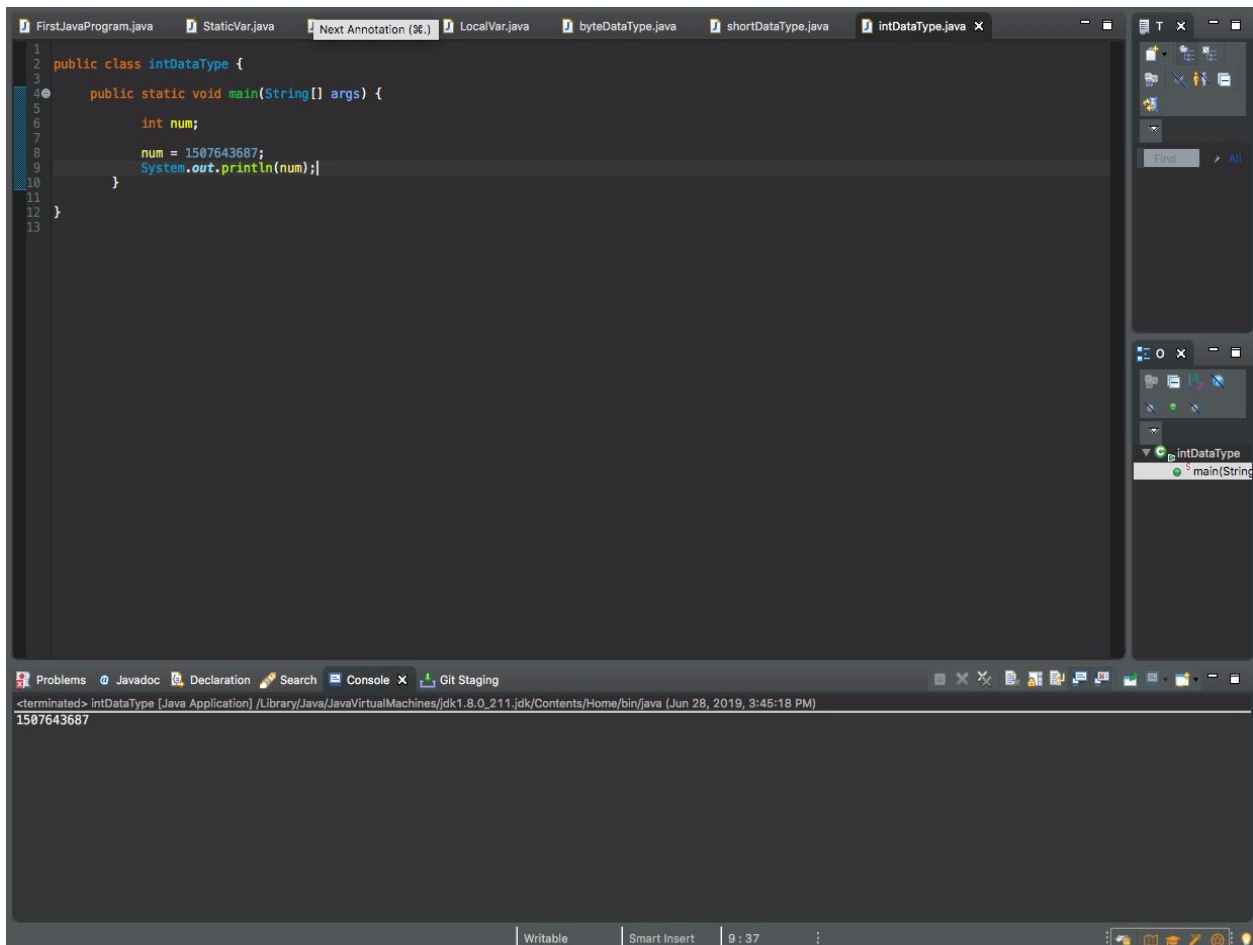
int:

An int is used when short is not large enough to hold the number as it has a wider range: -2,147,483,648 to 2,147,483,647

Default Size: 4 byte

Default Value: 0

Example:



The screenshot shows an IDE with several tabs open. The active tab is `intDataType.java`, which contains the following code:

```
1 public class intDataType {
2
3
4     public static void main(String[] args) {
5
6         int num;
7
8         num = 1507643687;
9         System.out.println(num);
10    }
11 }
12
13
```

The console output at the bottom shows the program terminated successfully and printed the value `1507643687`.

```
<terminated> intDataType [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/bin/java (Jun 26, 2019, 3:45:18 PM)
1507643687
```

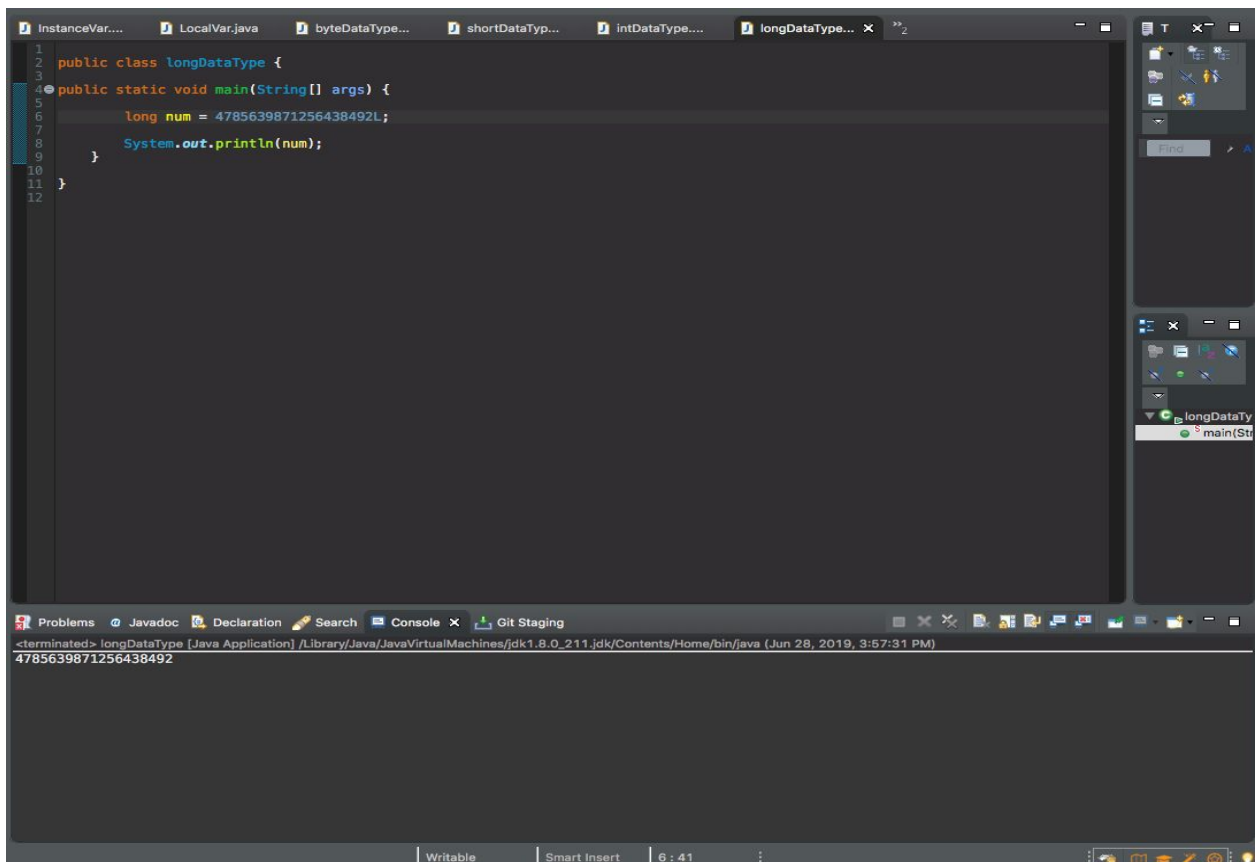
long:

A long is used when int is not large enough to hold the value as it has wider range than int data type, ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807.

Default Size: 8 byte

Default Value: 0

Example:



The screenshot shows an IDE with a Java file named `longDataType.java`. The code defines a class `longDataType` with a `main` method. Inside `main`, a `long` variable `num` is initialized with the value `4785639871256438492L`, and then printed using `System.out.println(num)`. The IDE's console at the bottom shows the output of the program: `4785639871256438492`. The IDE interface includes a top toolbar, a left sidebar with a file explorer, a right sidebar with a run and debug console, and a bottom status bar.

```
1 public class longDataType {  
2  
3  
4 public static void main(String[] args) {  
5  
6     long num = 4785639871256438492L;  
7  
8     System.out.println(num);  
9  
10 }  
11  
12 }
```

Problems Javadoc Declaration Search Console X Git Staging
<terminated> longDataType [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/bin/java (Jun 28, 2019, 3:57:31 PM)
4785639871256438492

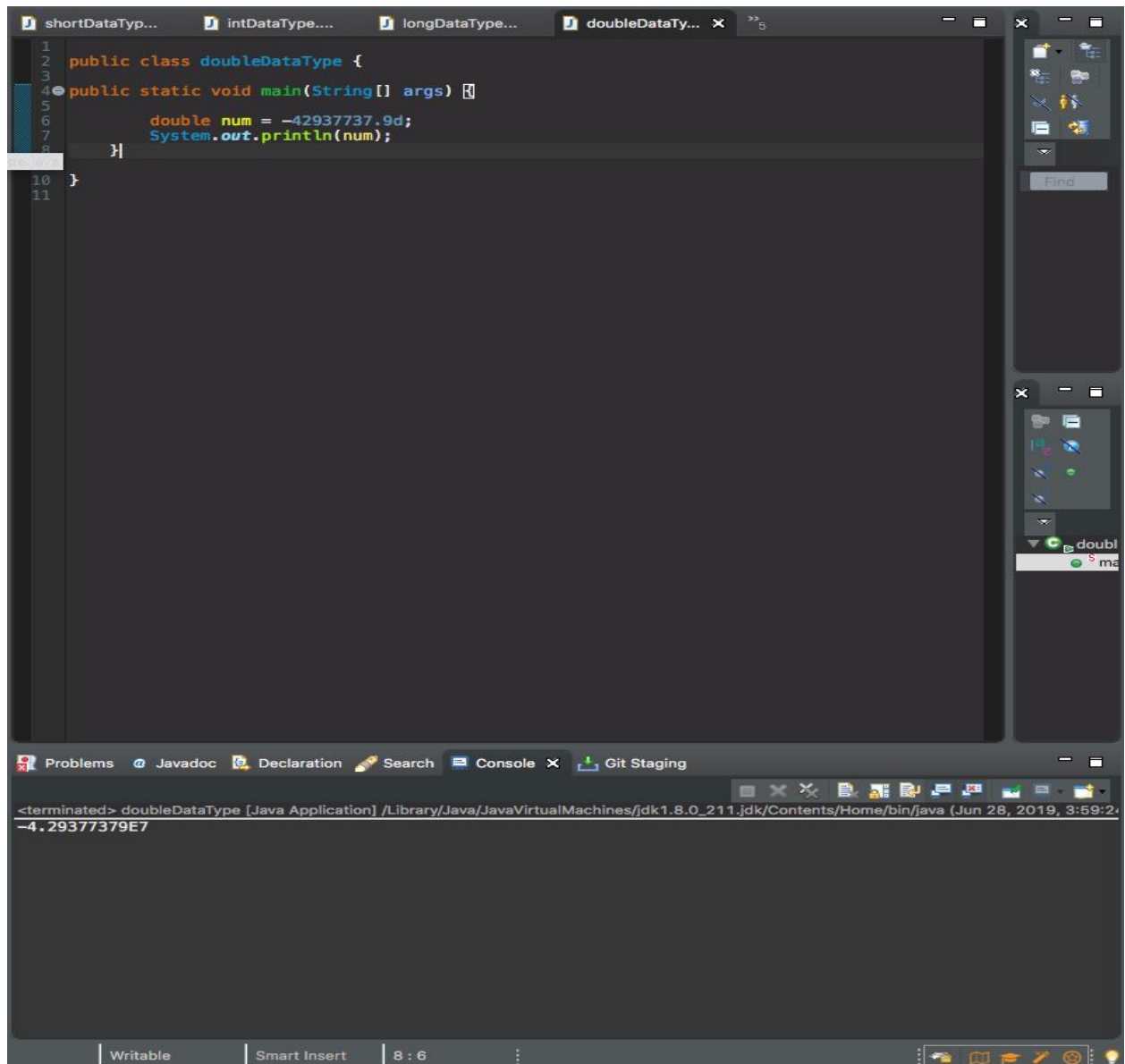
Writable Smart Insert 6 : 41

double:

A double is sufficient for holding 15 decimal digits

Default Size: 8 byte

Example:



The screenshot shows an IDE with a Java file named `doubleDataType.java`. The code defines a public class `doubleDataType` with a `main` method. Inside the `main` method, a `double` variable `num` is initialized with the value `-42937737.9d`, and its value is printed to the console using `System.out.println(num)`. The IDE's output console at the bottom shows the program's execution, displaying the output `-4.29377379E7` after the program has terminated. The IDE interface includes a sidebar with project files, a search bar, and a bottom status bar with options like 'Writable' and 'Smart Insert'.

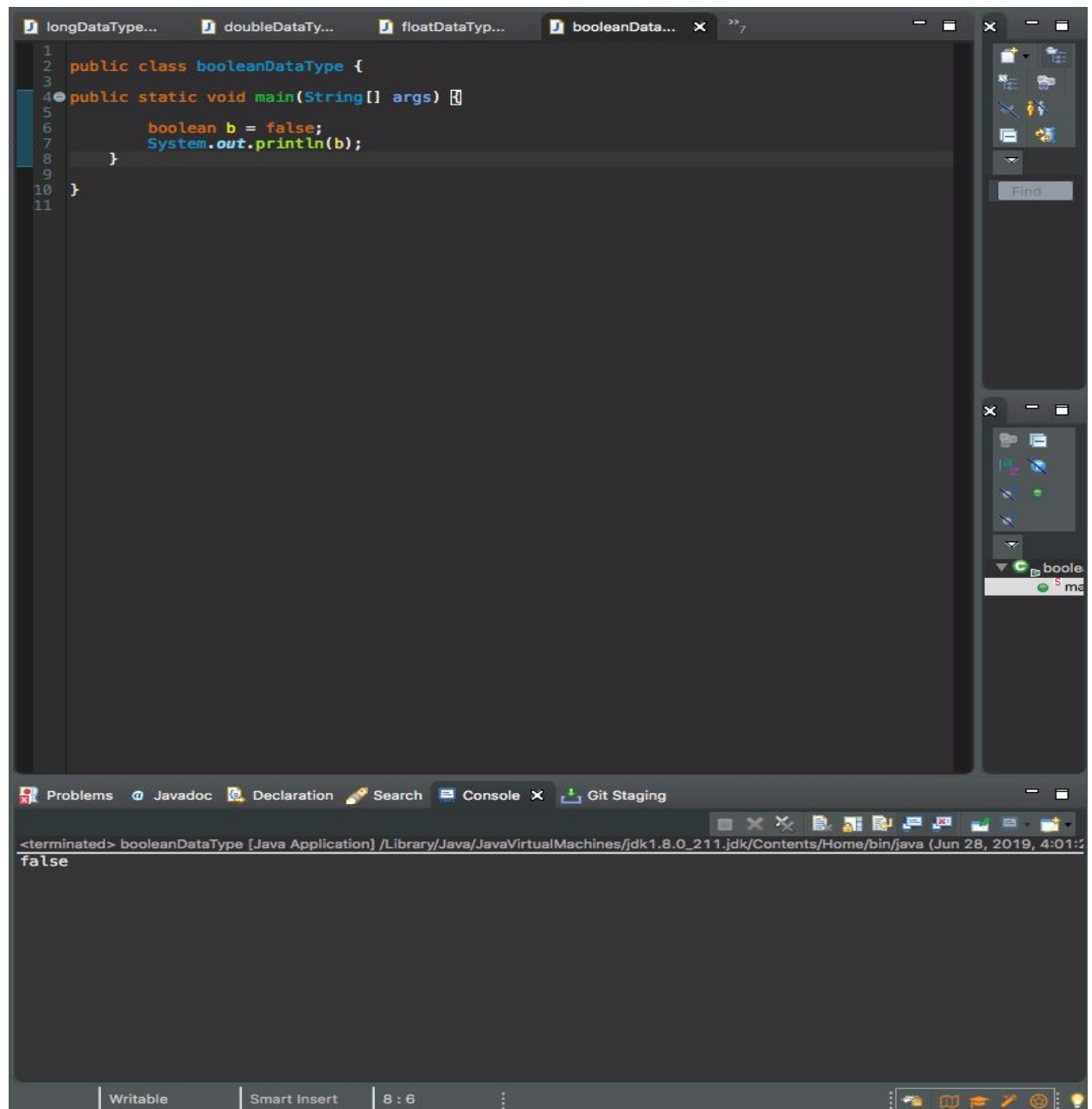
```
1 public class doubleDataType {  
2  
3  
4 public static void main(String[] args) {  
5  
6     double num = -42937737.9d;  
7     System.out.println(num);  
8  
9 }  
10  
11 }
```

`<terminated> doubleDataType [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/bin/java (Jun 28, 2019, 3:59:24)
-4.29377379E7`

boolean:

A boolean holds either true or false.

Example:



The screenshot shows an IDE with a dark theme. The main editor window displays a Java class named `booleanDataType` with a `main` method. The code is as follows:

```
1 public class booleanDataType {  
2  
3  
4 public static void main(String[] args) {  
5  
6     boolean b = false;  
7     System.out.println(b);  
8  
9 }  
10  
11 }
```

The right sidebar contains a 'Find' search bar and a file explorer showing the project structure. The bottom status bar indicates the file is 'Writable' and 'Smart Insert' is active, with a cursor at line 8, column 6. The console at the bottom shows the output of the program:

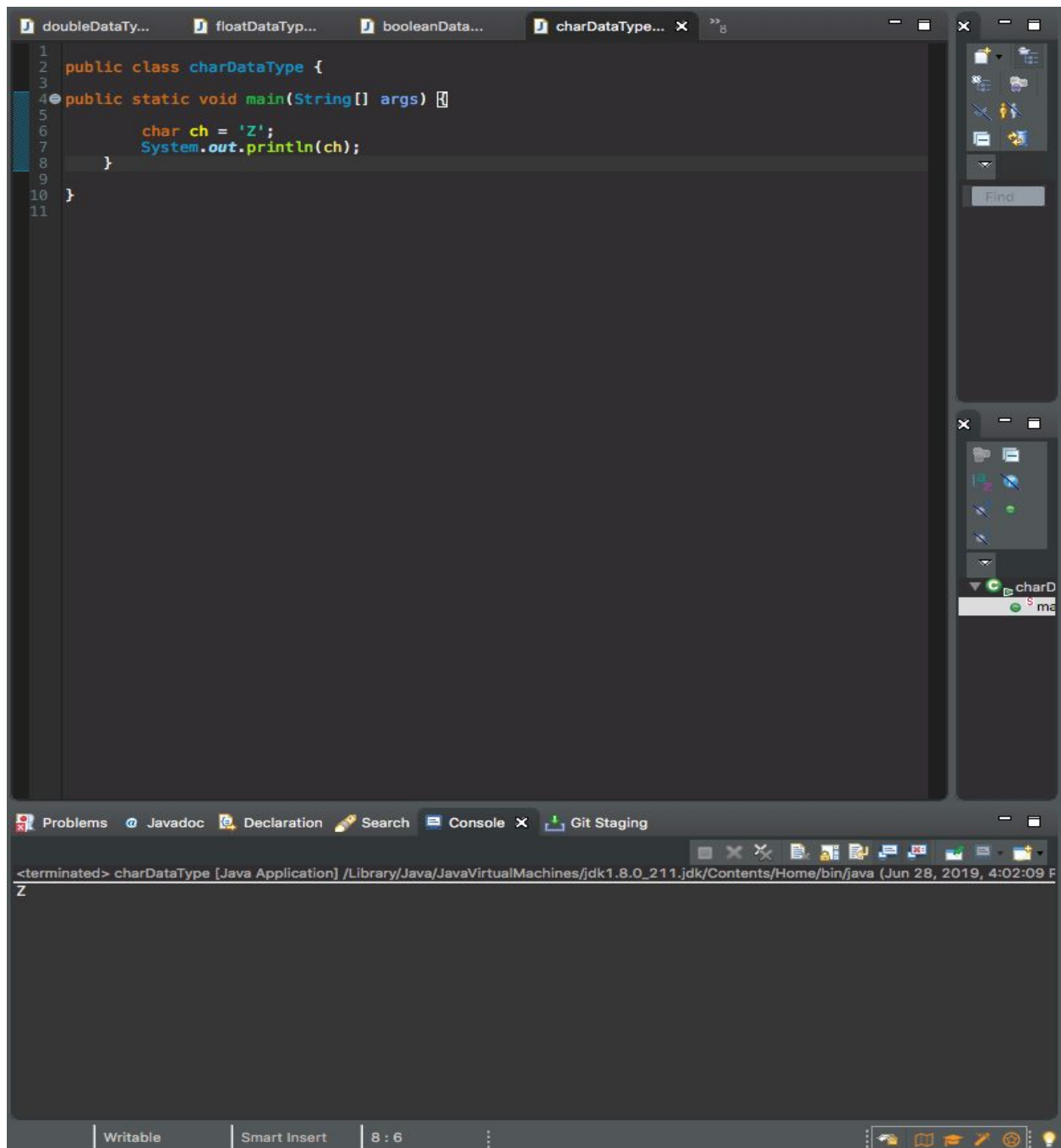
```
<terminated> booleanDataType [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/bin/java (Jun 28, 2019, 4:01:2  
false
```

char:

A char holds characters.

Default Size: 2 byte

Example:



The screenshot shows an IDE with a Java file named `charDataType.java`. The code defines a class `charDataType` with a `main` method that declares a `char` variable `ch` and assigns it the value `'Z'`, then prints it using `System.out.println(ch)`. The IDE's output console at the bottom shows the execution result: `<terminated> charDataType [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/bin/java (Jun 28, 2019, 4:02:09 PM) Z`. The status bar at the bottom indicates the file is `Writable`, `Smart Insert` is active, and the cursor is at line 8, column 6.

```
1 public class charDataType {
2
3
4 public static void main(String[] args) {
5
6     char ch = 'Z';
7     System.out.println(ch);
8 }
9
10 }
11
```

`<terminated> charDataType [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/bin/java (Jun 28, 2019, 4:02:09 PM) Z`

