

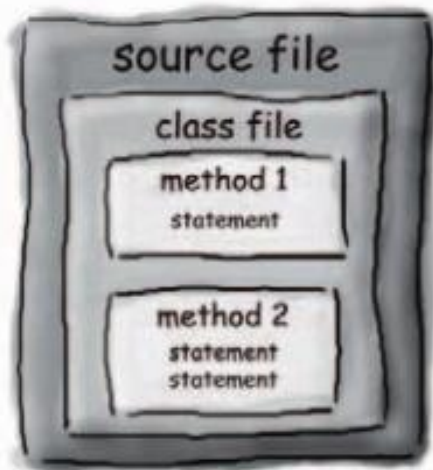


# Java Quick Review

# Code Structure



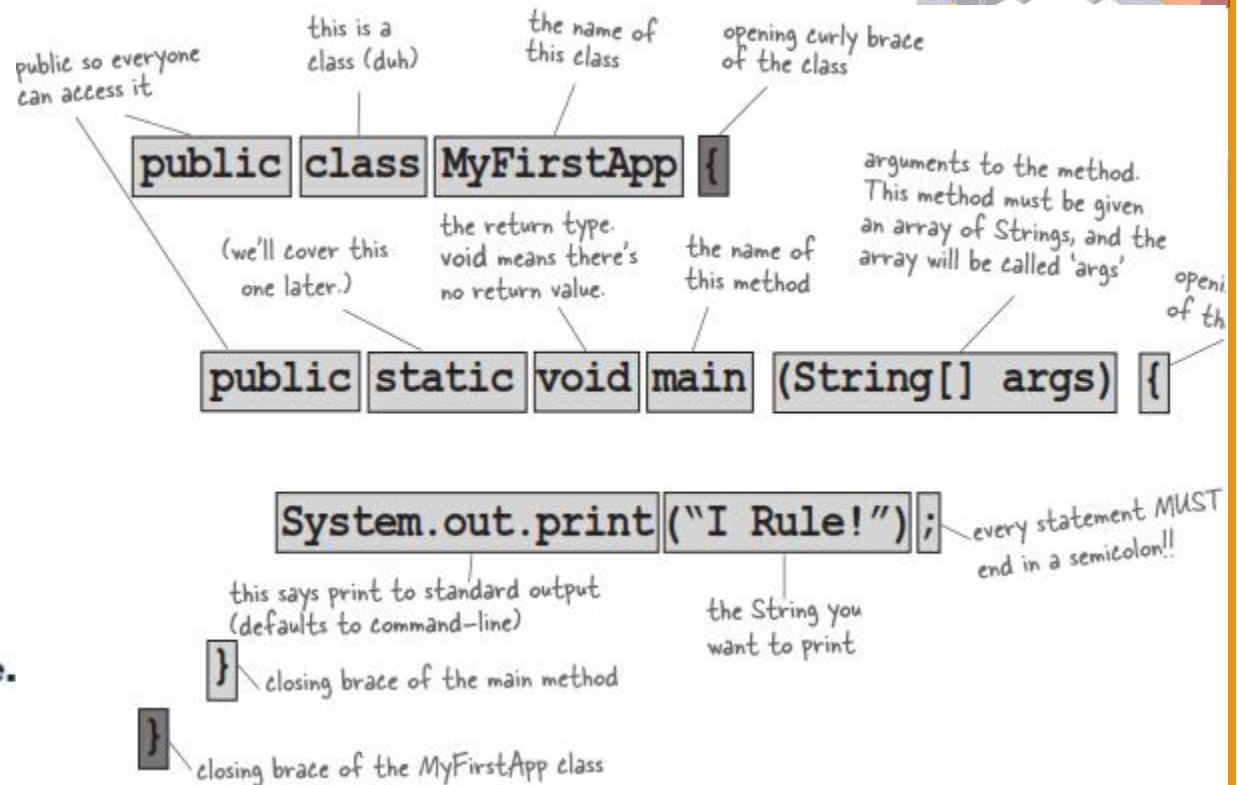
## Code structure in Java



**Put a class in a source file.**

**Put methods in a class.**

**Put statements in a method.**

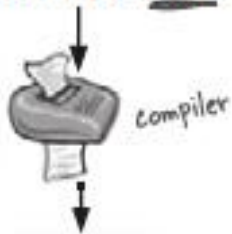


# First App



```
public class MyFirstApp {  
    public static void main  
    (String[] args) {  
        System.out.println("I Rule!");  
    }  
}
```

**MyFirstApp.java**



```
Method(Party) 0 value: 0 1  
invokeSpecial() Method  
invokeStatic() Method  
4 return  
Method void  
main(java.lang.String[])  
0 public static void main
```

**MyFirstApp.class**

```
public class MyFirstApp {  
  
    public static void main (String[] args) {  
        System.out.println("I Rule!");  
        System.out.println("The World");  
    }  
}
```

## 1 Save

MyFirstApp.java

## 2 Compile

javac MyFirstApp.java

## 3 Run

```
File Edit Window Help Screen  
  
% java MyFirstApp  
  
I Rule!  
  
The World
```

# Statements, Looping, Branching



## 1 do something

**Statements:** declarations, assignments, method calls, etc.

```
int x = 3;  
String name = "Dirk";  
x = x * 17;  
System.out.print("x is " + x);  
double d = Math.random();  
// this is a comment
```

## 2 do something again and again

**Loops:** *for* and *while*

```
while (x > 12) {  
    x = x - 1;  
}  
  
for (int x = 0; x < 10; x = x + 1) {  
    System.out.print("x is now " + x);  
}
```

## 3 do something under this condition

**Branching:** *if/else* tests

```
if (x == 10) {  
    System.out.print("x must be 10");  
} else {  
    System.out.print("x isn't 10");  
}  
  
if ((x < 3) & (name.equals("Dirk"))) {  
    System.out.println("Gently");  
}  
  
System.out.print("this line runs no matter what");
```