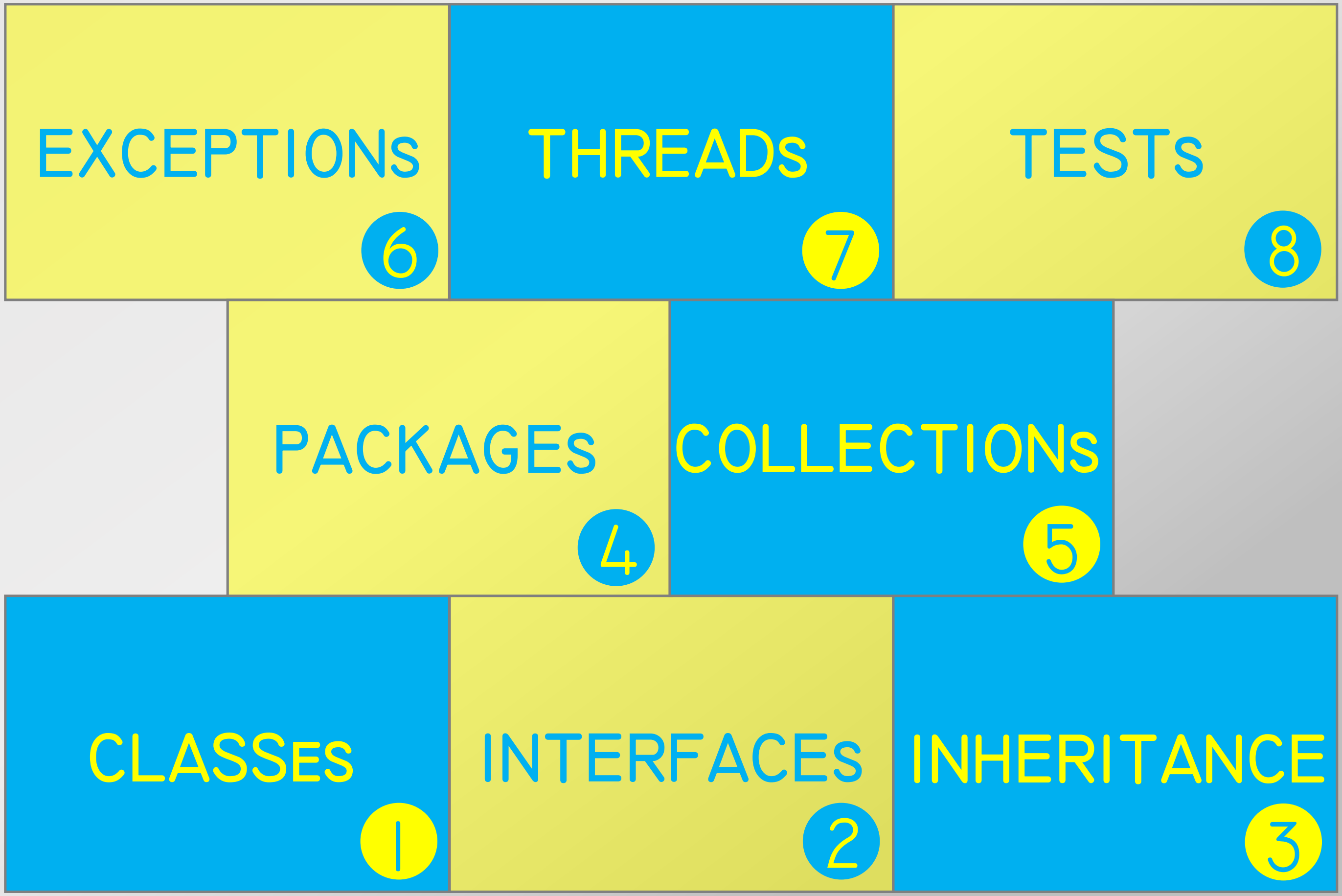
BASICS of Java



EXCEPTIONs 6

THREADs 7

TESTs 8

PACKAGEs 4

COLLECTIONs 5

CLASSEs 1

INTERFACEs 2

INHERITANCE 3

BASICS of Java

EXCEPTIONs

THREADs

TESTs

PACKAGEs

COLLECTIONs

CLASSEs

INTERFACEs

INHERITANCE

CLASSes

- How to create a class
- How to add a constructor
- How to add a method
- How to create an object
- How to call a method
- How to display in the console
- How to deal with Strings

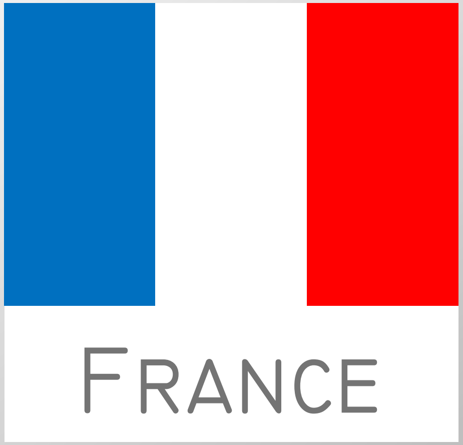
COUNTRY

Class

a template for a country

Object

a concrete country

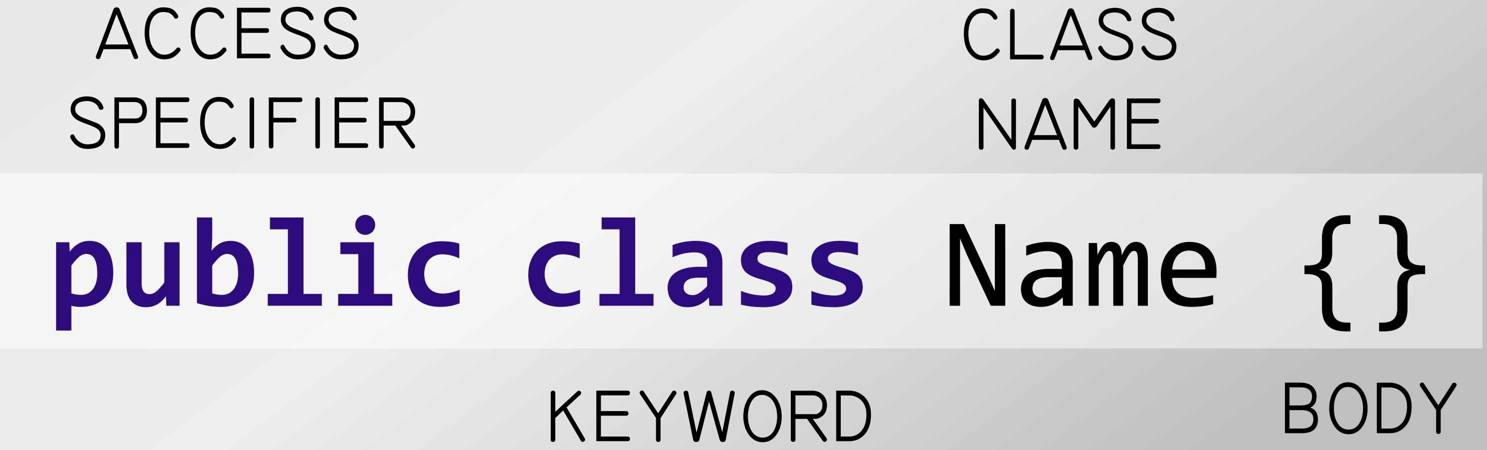


Object

a concrete country

Object

a concrete country



ACCESS SPECIFIER

CLASS NAME

```
public class Name {}
```

KEYWORD

BODY

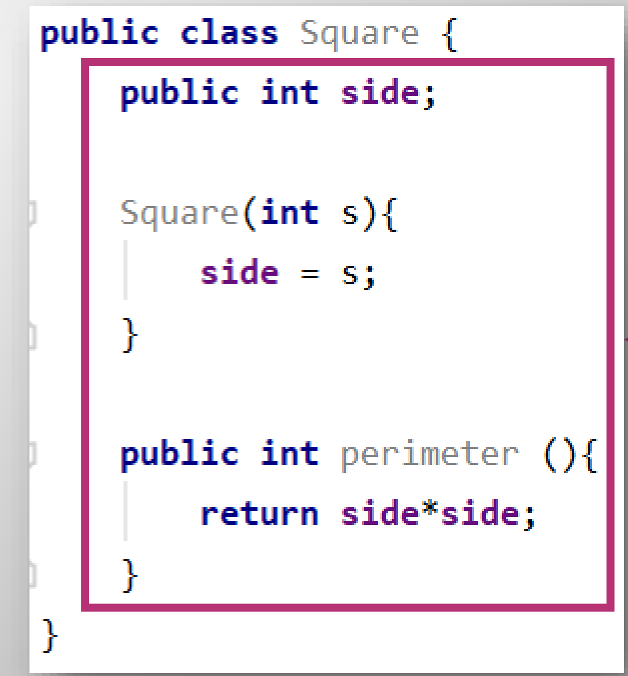
ACCESS SPECIFIER

CLASS NAME

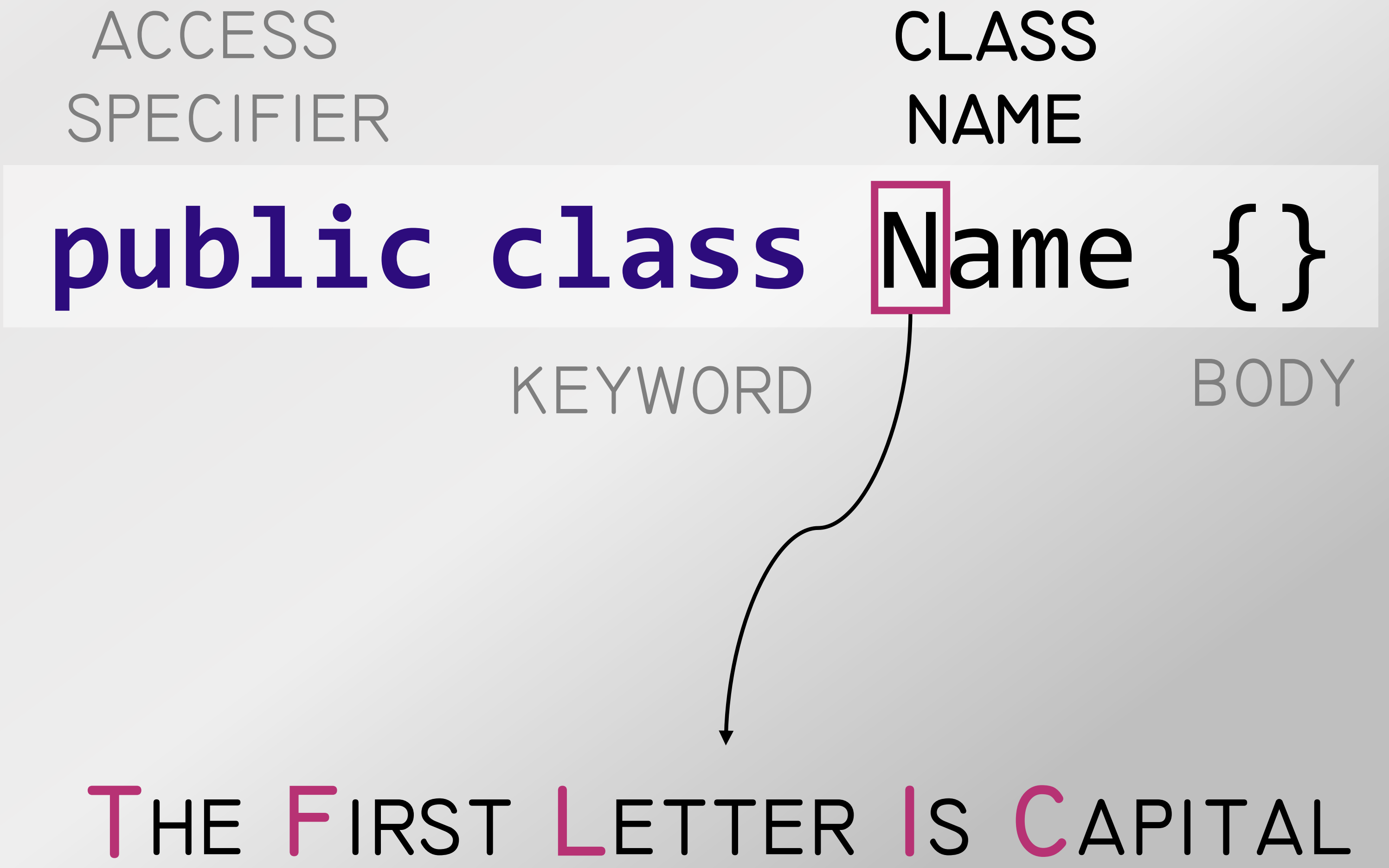
```
public class Name {}
```

KEYWORD

BODY



```
public class Square {
    public int side;

    Square(int s){
        side = s;
    }

    public int perimeter (){
        return side*side;
    }
}
```



ACCESS SPECIFIER

CLASS NAME

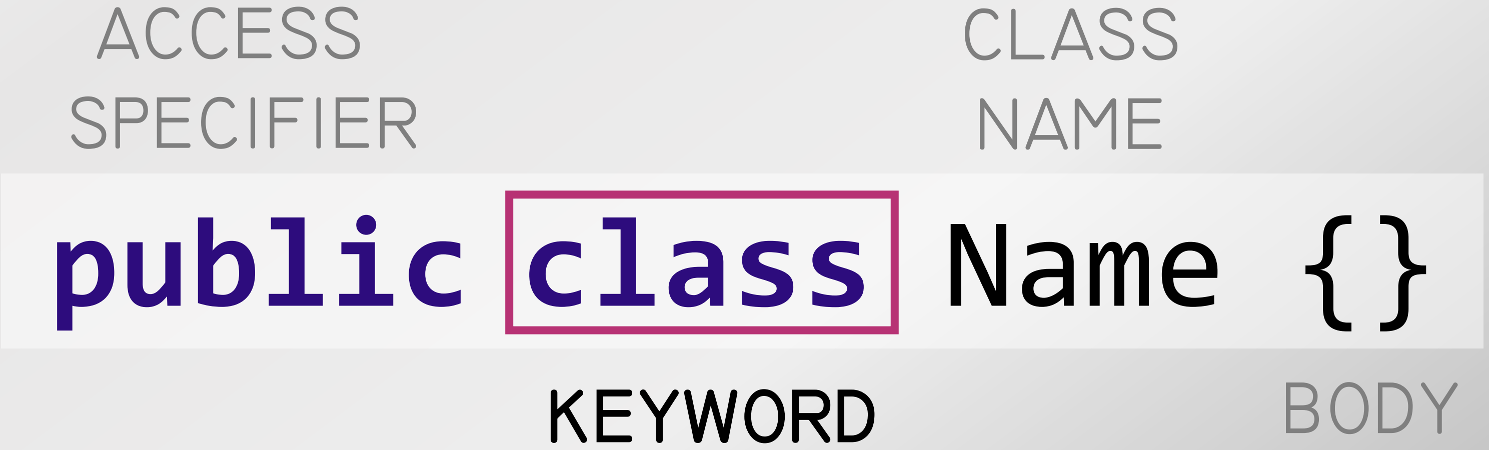
```
public class Name {}
```

KEYWORD

BODY

THE FIRST LETTER IS CAPITAL

JAVA NAMING CONVENTIONS: HTTPS://WWW.ORACLE.COM/JAVA/TECHNOLOGIES/JAVASE/CODECONVENTIONS-NAMINGCONVENTIONS.HTML



ACCESS SPECIFIER

CLASS NAME

```
public class Name {}
```

KEYWORD

BODY

ACCESS SPECIFIER

CLASS NAME

```
public class Name {}
```

KEYWORD

BODY

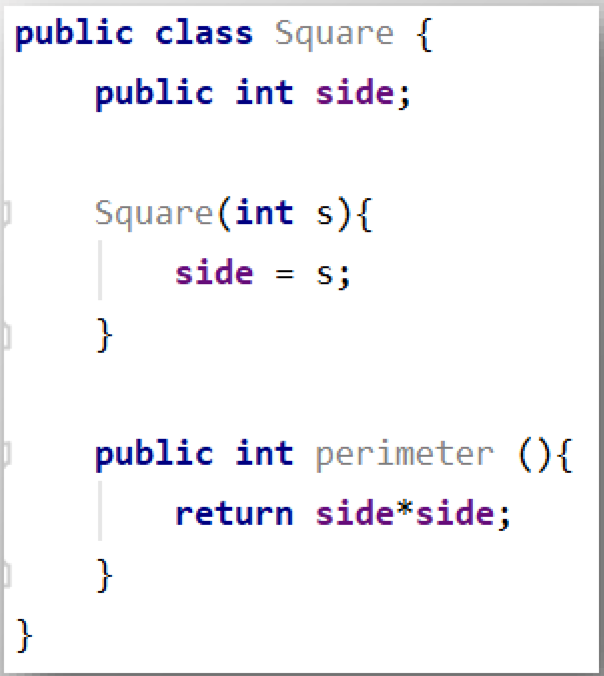
ACCESS SPECIFIER

CLASS NAME

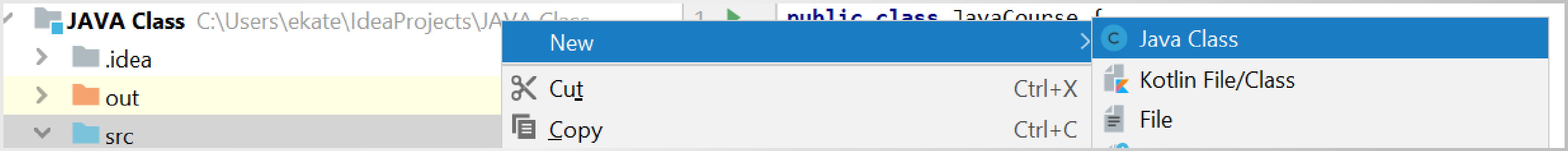
```
public class Name {}
```

KEYWORD

BODY



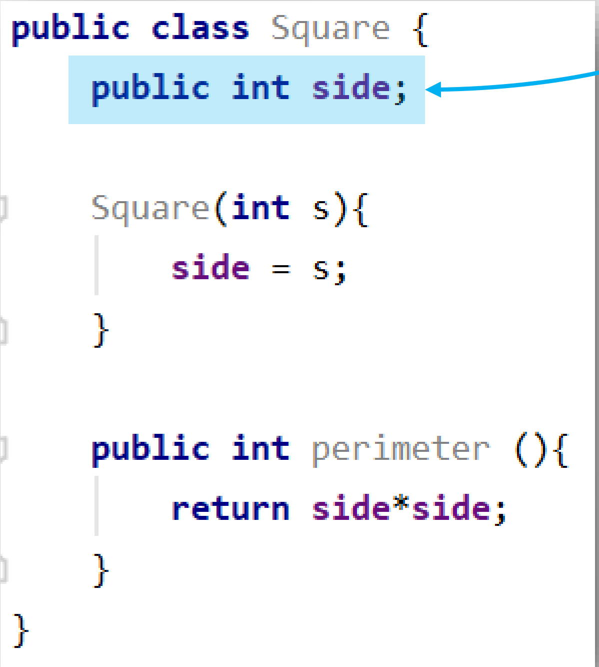
```java
public class Square {
    public int side;

    Square(int s){
        side = s;
    }

    public int perimeter (){
        return side*side;
    }
}
```


JAVA Class C:\Users\ekate\IdeaProjects\JAVA Class
.idea
out
src
public class JavaCourse {
New
Cut Ctrl+X
Copy Ctrl+C
Java Class
Kotlin File/Class
File

RIGHT-CLICK ON THE SRC FOLDER -> NEW -> JAVA CLASS

□ CREATE A (PUBLIC) CLASS CALLED <Circle>

1



```
public class Square {
    public int side;

    Square(int s){
        side = s;
    }

    public int perimeter (){
        return side*side;
    }
}
```

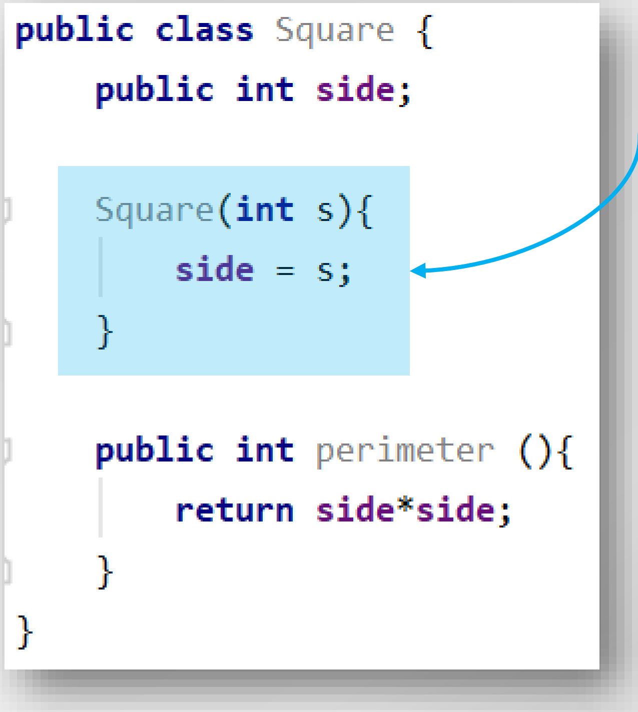
FIELD

= A **VARIABLE** THAT IS DECLARED IN A CLASS

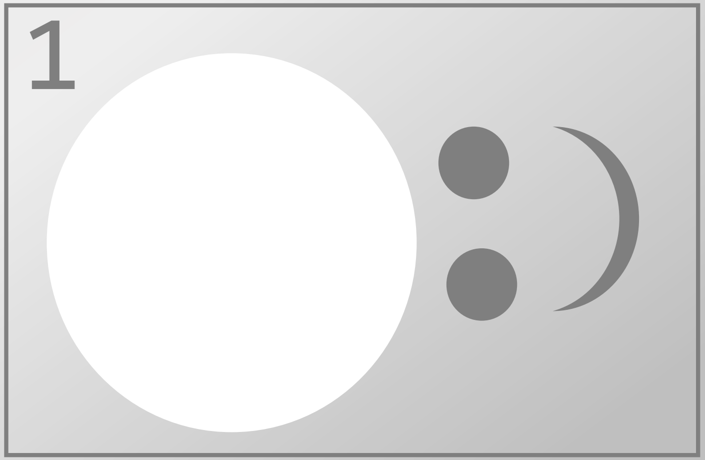
- □ CREATE A (PUBLIC) CLASS CALLED <CIRCLE>
- □ ADD A FIELD(= A VARIABLE) OF THE DATATYPE INTEGER, CALLED <RADIUS>

2

CONSTRUCTOR



```
public class Square {
    public int side;

    Square(int s){
        side = s;
    }

    public int perimeter (){
        return side*side;
    }
}
```

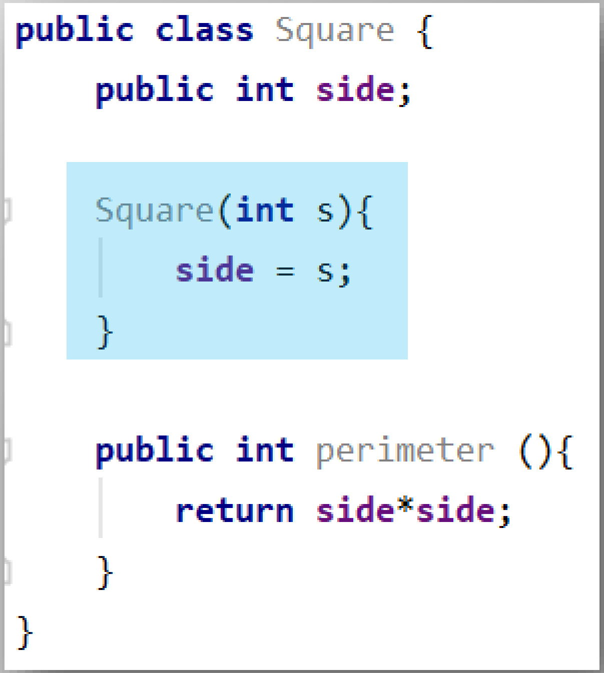


CLASS NAME

BODY

```
Name(){}
```

PARAMETERS



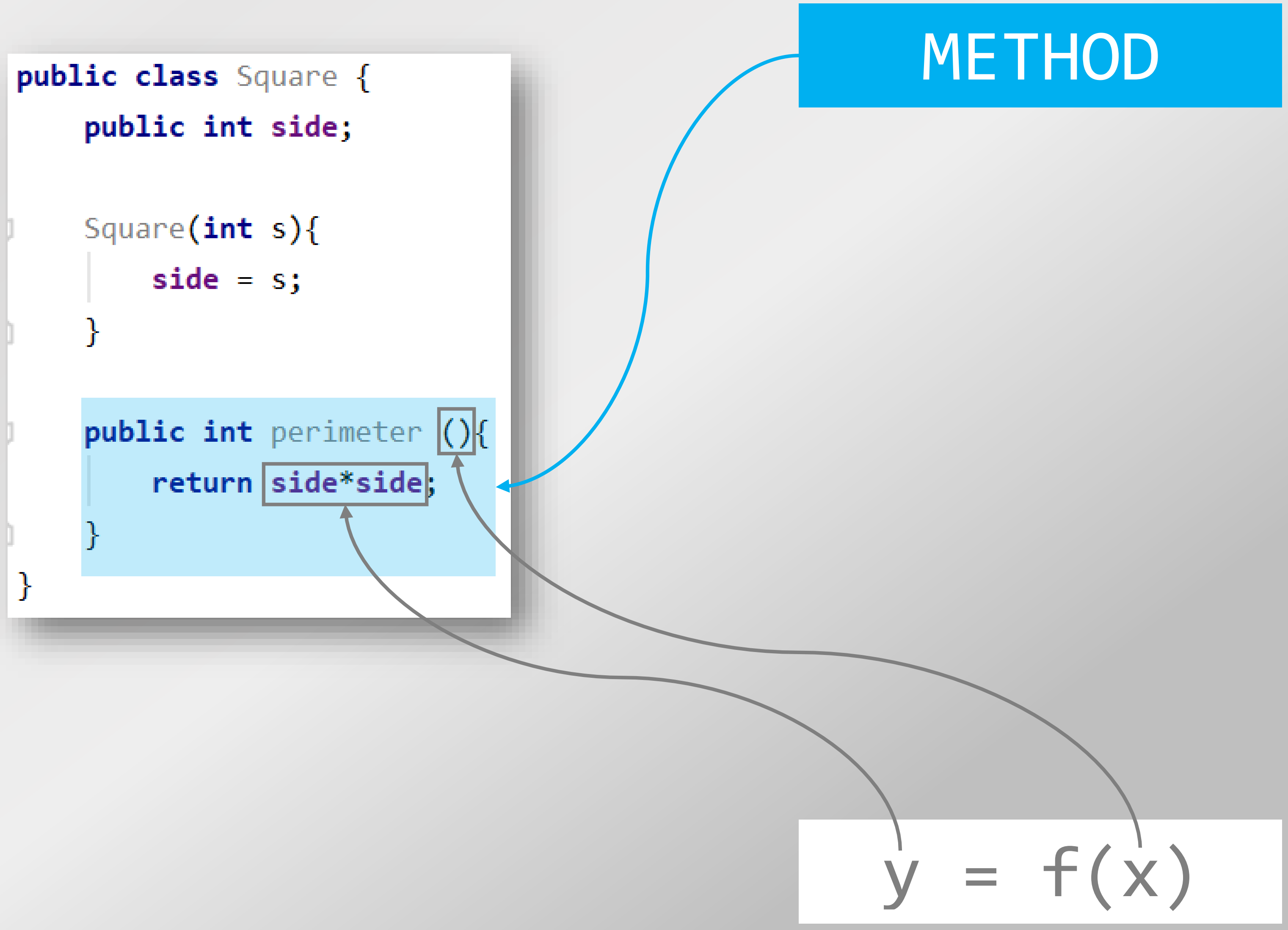
```
public class Square {
    public int side;

    Square(int s){
        side = s;
    }

    public int perimeter (){
        return side*side;
    }
}
```

□ CREATE A (PUBLIC) CLASS CALLED <CIRCLE>

□ ADD A FIELD(= A VARIABLE) OF THE DATATYPE INTEGER, CALLED <RADIUS>

3

□ ADD A CONSTRUCTOR THAT TAKES AS AN ARGUMENT (= PARAMETER) AN INTEGER VALUE AND ASSIGNS THIS VALUE TO THE VARIABLE <RADIUS>.

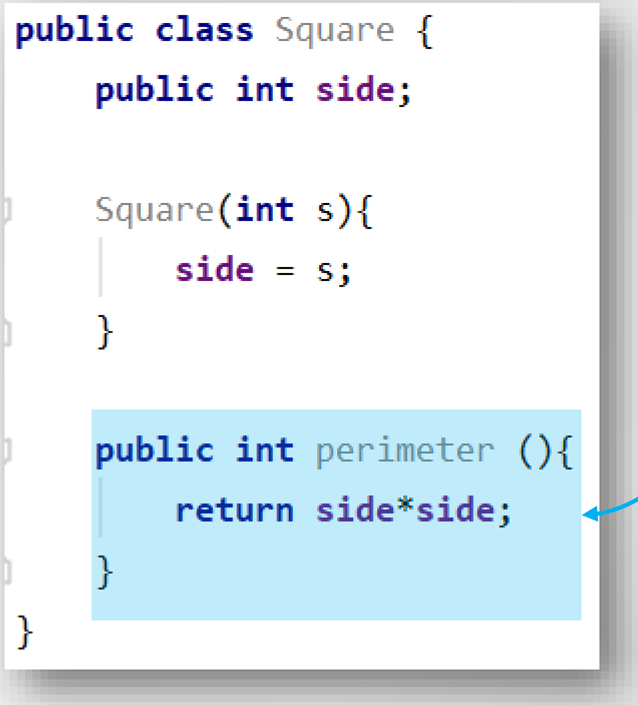


```
public class Square {
    public int side;

    Square(int s){
        side = s;
    }

    public int perimeter (){
        return side*side;
    }
}
```

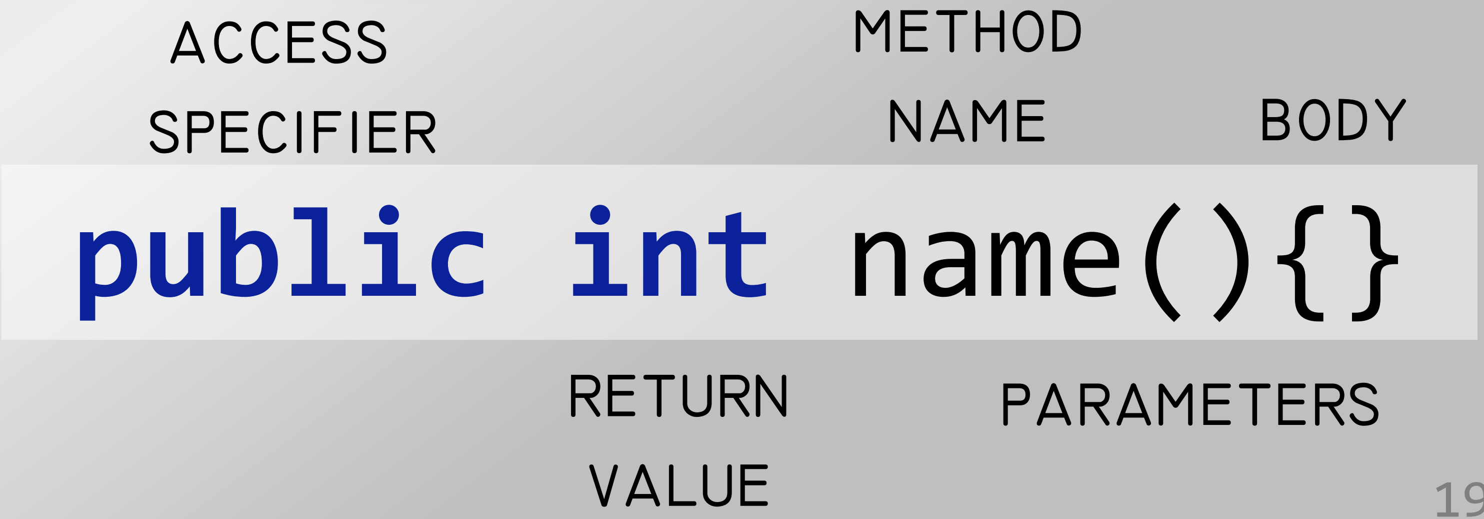
METHOD

$y = f(x)$

METHOD



```
public class Square {
    public int side;

    Square(int s){
        side = s;
    }

    public int perimeter (){
        return side*side;
    }
}
```



ACCESS SPECIFIER

METHOD NAME

BODY

```
public int name(){}
```

RETURN VALUE

PARAMETERS

4

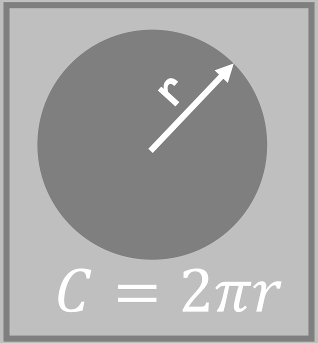
□ CREATE A (PUBLIC) CLASS CALLED <CIRCLE>

□ ADD A FIELD(= A VARIABLE) OF THE DATATYPE FLOAT, CALLED <RADIUS>

□ ADD A CONSTRUCTOR THAT TAKES AS AN ARGUMENT (= PARAMETER) A FLOAT VALUE AND ASSIGNS THIS VALUE TO THE VARIABLE <RADIUS>.

□ ADD A METHOD CALLED <CIRCUMFERENCE> THAT RETURNS A FLOAT VALUE AND TAKES NO ARGUMENTS. THE METHOD CALCULATES THE CIRCUMFERENCE OF THE CIRCLE.

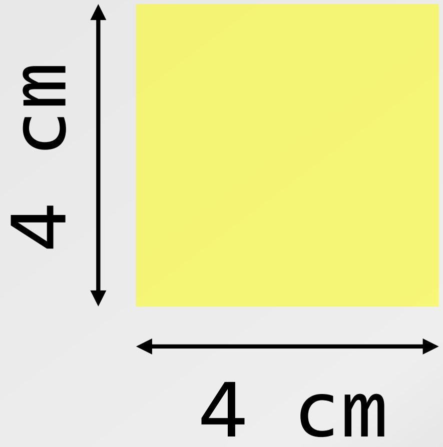
NOTE: IN JAVA π = **MATH.PI** (ACCORDING TO JAVA NAMING CONVENTIONS, NAME OF CONSTANTS ARE WRITTENT IN ALL CAPITAL LETTERS).



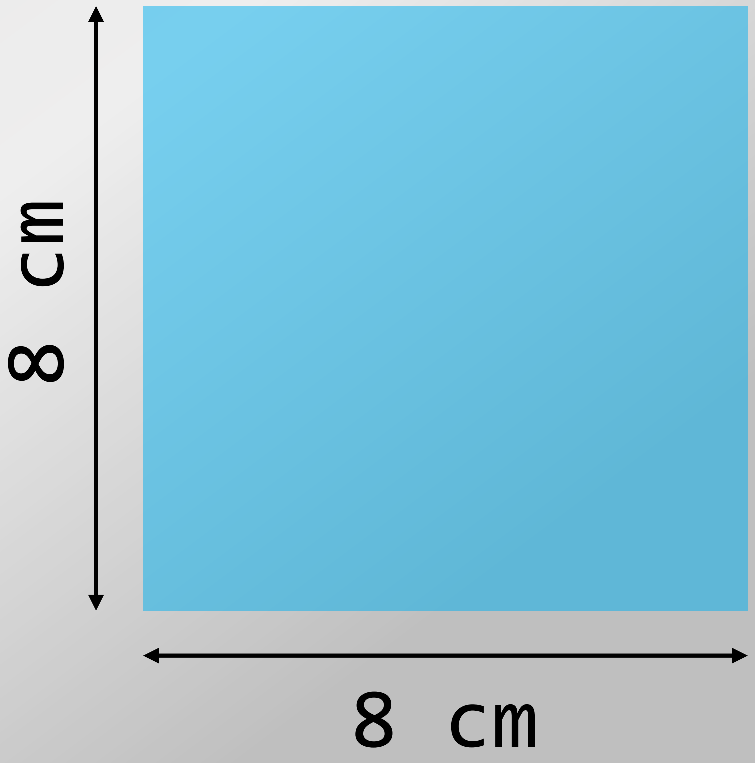
ABOUT CONSTANTS IN JAVA:

HTTPS://WWW.TUTORIALSPOINT.COM/WHAT-IS-A-CONSTANT-AND-HOW-TO-DEFINE-CONSTANTS-IN-JAVA#:~:TEXT=A%20CONSTANT%20IS%20A%20VARIABLE,READ%20AND%20UNDERSTOOD%20BY%20OTHERS.&TEXT=TO%20DEFINE%20A%20VARIABLE%20AS,FRONT%20OF%20THE%20VARIABLE%20DECLARATION.

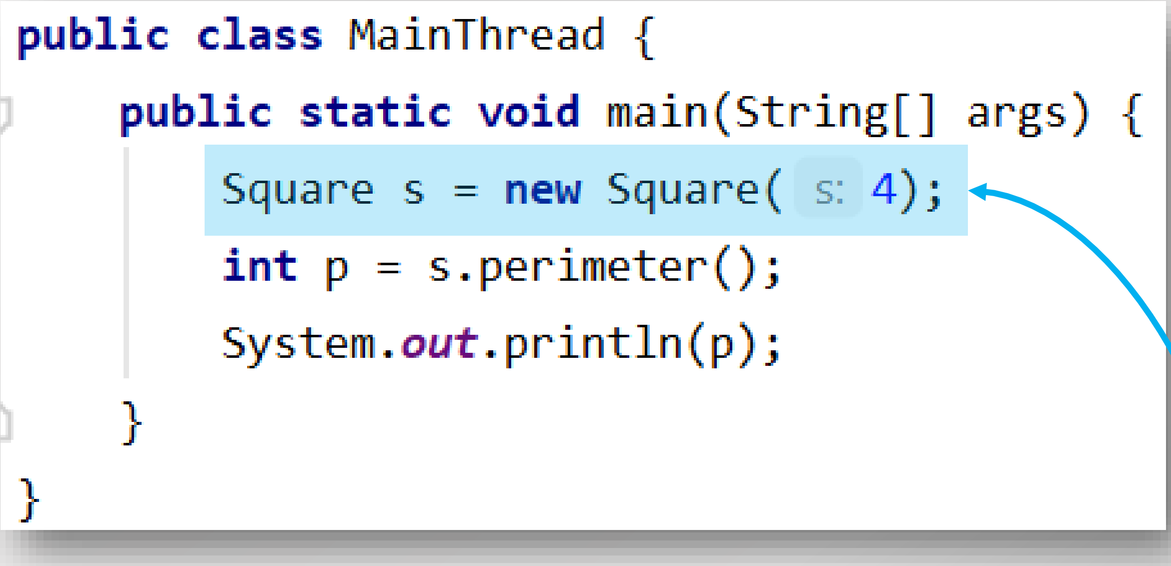
```
class Square {}
```



a concrete OBJECT of the <Square> class



a concrete OBJECT of the <Square> class



```
public class MainThread {
    public static void main(String[] args) {
        Square s = new Square( s: 4);
        int p = s.perimeter();
        System.out.println(p);
    }
}
```

we create an Object of the <Square> class

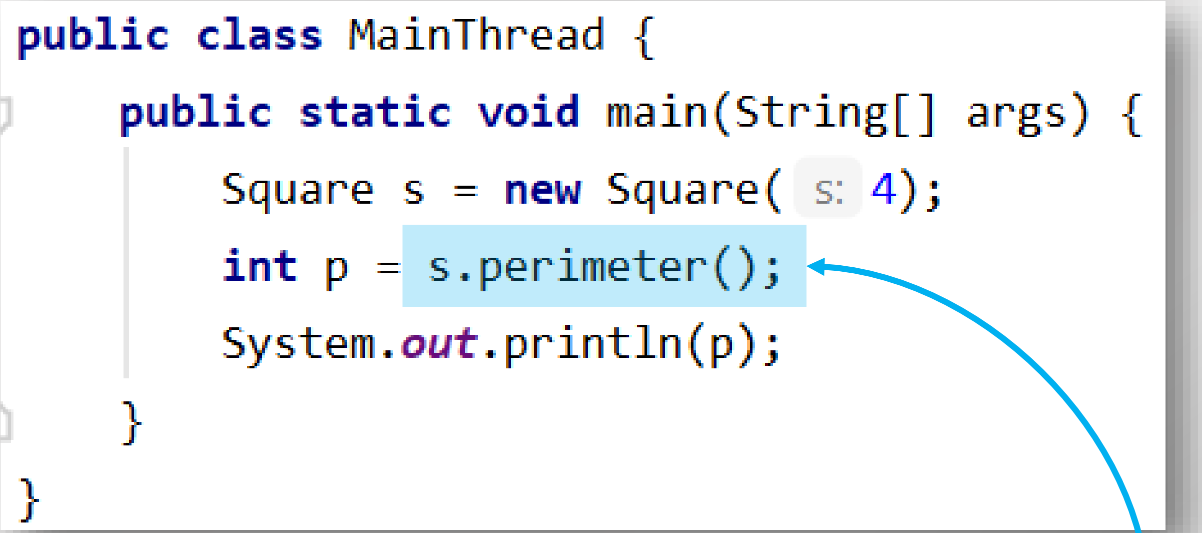
KEYWORD

`<ClassName> <objectName> = new <ClassName>();`

CONSTRUCTOR

5

- In the main program create an object of class <Circle> called <c> and assign to the <radius> variable the value 4.

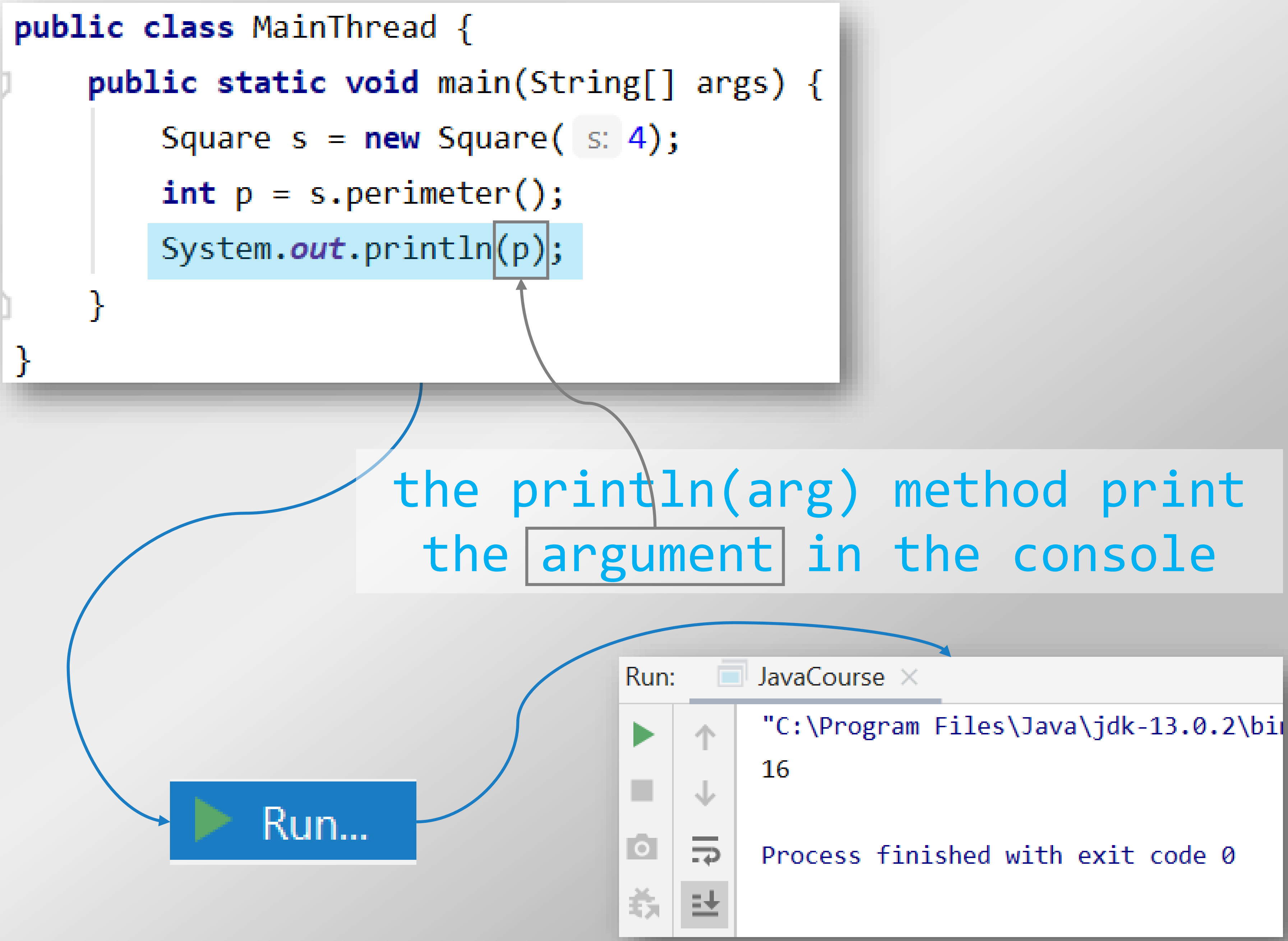


```java
public class MainThread {
    public static void main(String[] args) {
        Square s = new Square( s: 4);
        int p = s.perimeter();
        System.out.println(p);
    }
}
```

we call the <perimeter> method

PARAMETERS

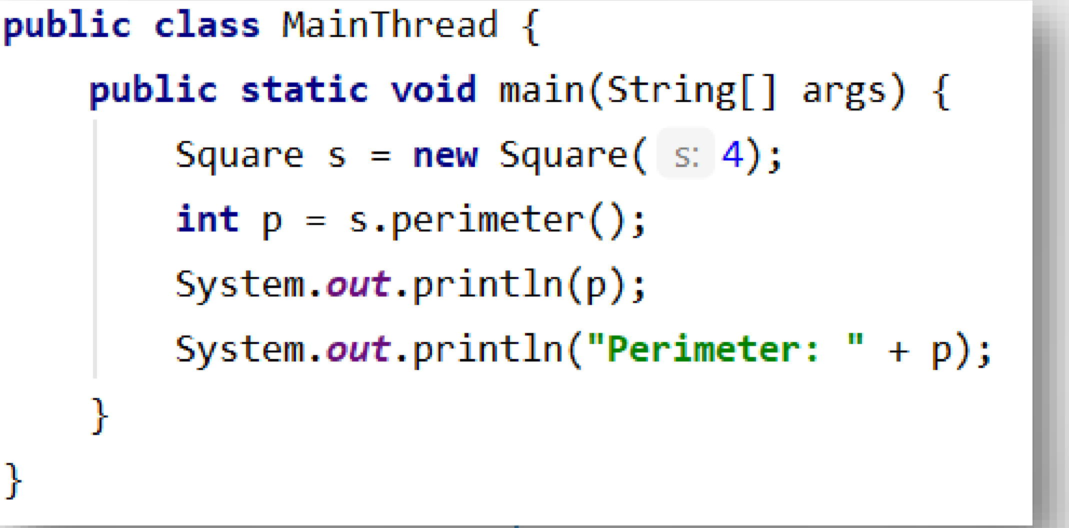
```
<objectname>.<methodname>();
```



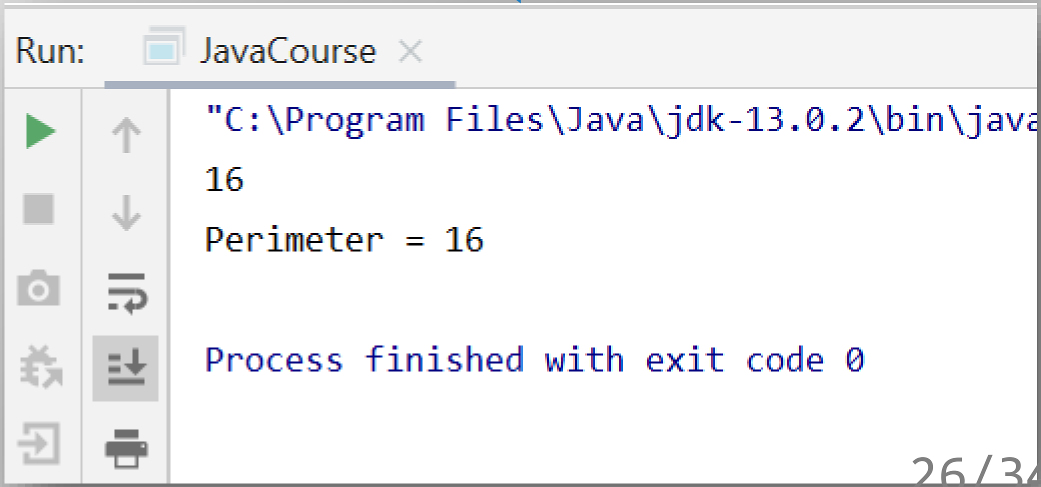
```
public class MainThread {
    public static void main(String[] args) {
        Square s = new Square( s: 4);
        int p = s.perimeter();
        System.out.println(p);
    }
}
```

the println(arg) method print the argument in the console

```
"C:\Program Files\Java\jdk-13.0.2\bi
16

Process finished with exit code 0
```



```
public class MainThread {
    public static void main(String[] args) {
        Square s = new Square( s: 4);
        int p = s.perimeter();
        System.out.println(p);
        System.out.println("Perimeter: " + p);
    }
}
```



```
"C:\Program Files\Java\jdk-13.0.2\bin\java
16
Perimeter = 16

Process finished with exit code 0
```

6

□ In the main program create an object of class <Circle> called <c> and assign to the <radius> variable the value 4.

□ In the main program call the <circumference> method of the <c> object and display its return value in the console.

7-9

□ IN THE MAIN PROGRAM CREATE AN OBJECT OF CLASS <CIRCLE> CALLED <C> AND ASSIGN TO THE <RADIUS> VARIABLE THE VALUE 4.

□ IN THE MAIN PROGRAM CALL THE <CIRCUMFERENCE> METHOD OF THE <C> OBJECT AND DISPLAY ITS RETURN VALUE IN THE CONSOLE.

□ IN THE MAIN PROGRAM DISPLAY THE VALUE OF <RADIUS> IN THE CONSOLE: `System.out.println(c.radius)`.

□ IN THE <CIRCLE> CLASS CHANGE THE ACCESS MODIFIER OF THE <RADIUS> FIELD FROM **public** TO **private**.

□ ADD METHODS THAT ALLOW TO SET AND GET THE VALUE OF THE <RADIUS> FIELD.

NOTE: READ ABOUT GETTER AND SETTER METHODS

10

□ create a class called <Rectangle> that has two fields of the integer datatype called <l> (length) and <w> (width)

□ add a constructor that takes two arguments of the integer datatype and assign their values to the variables <l> and <w>

□ add a method called <perimeter> that calculates the perimeter of a rectangle

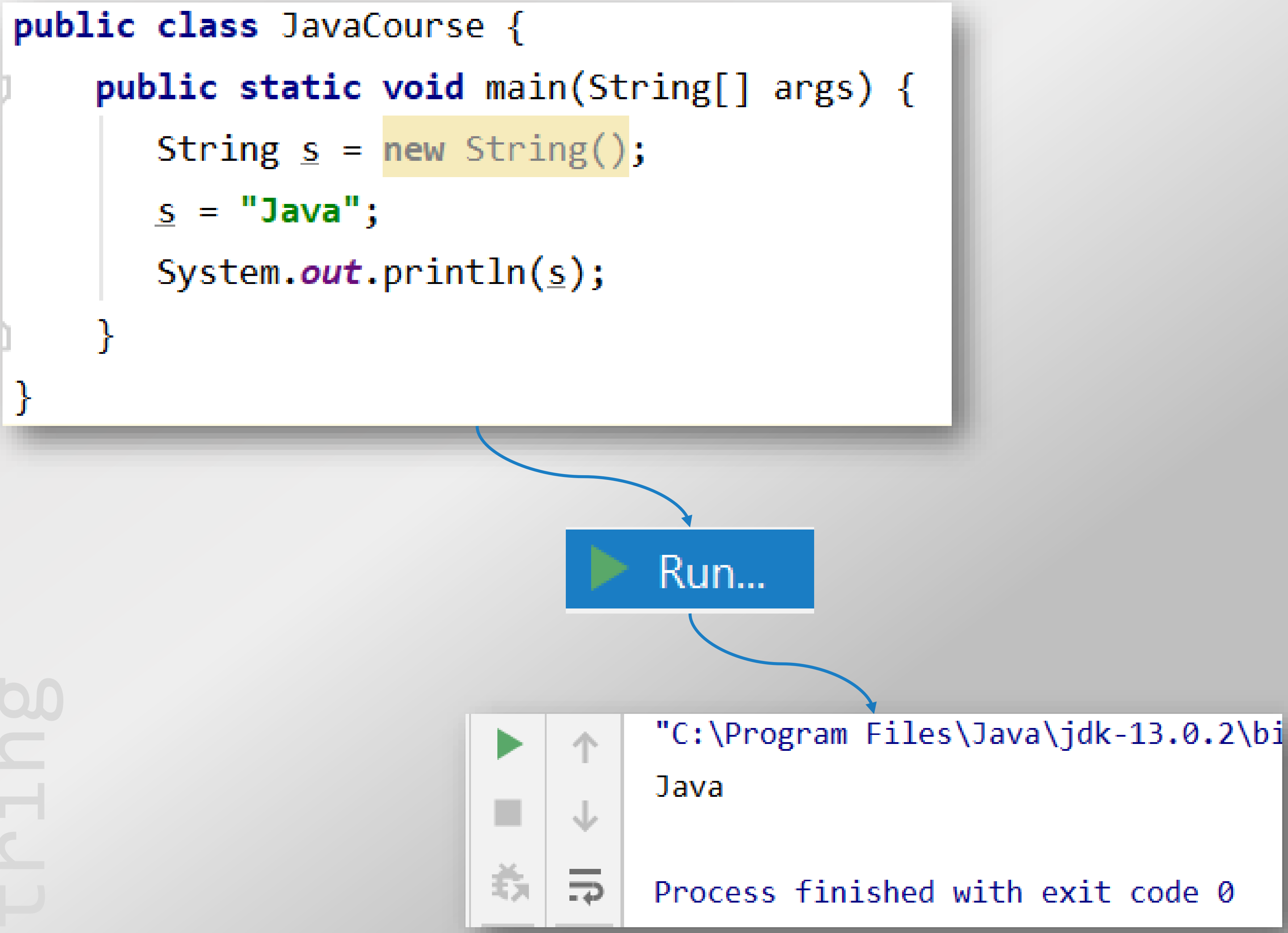
□ add a method <area>that calculates the area of a rectangle

□ in the main program create an object of the <Rectangle> class called <r>. Set its length <l> to 2 and its width <w> to 5

□ call the <perimeter> method and display its return value

□ call the <area> method and display its return value

String



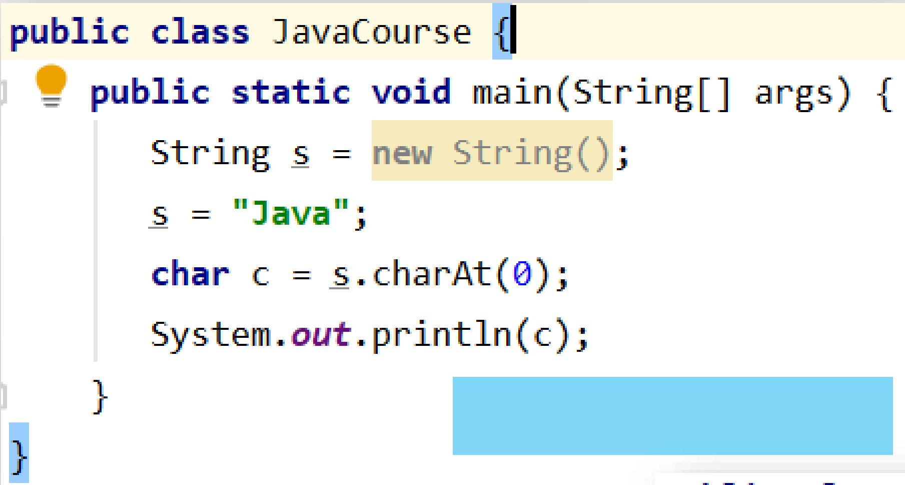
```
public class JavaCourse {
    public static void main(String[] args) {
        String s = new String();
        s = "Java";
        System.out.println(s);
    }
}
```

```
"C:\Program Files\Java\jdk-13.0.2\bi
Java

Process finished with exit code 0
```

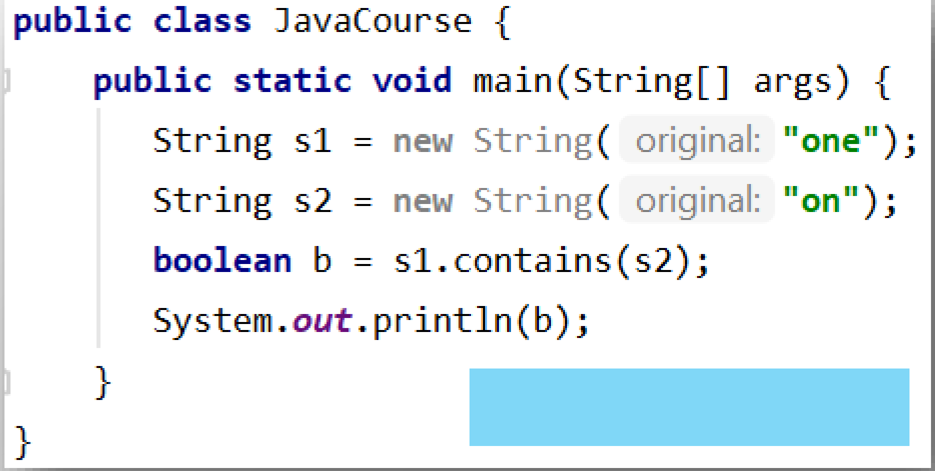
11-12

□ Create an object of the <String> class called <st> that contains the <Hello, World!> sentence and then display the content of this string in the console

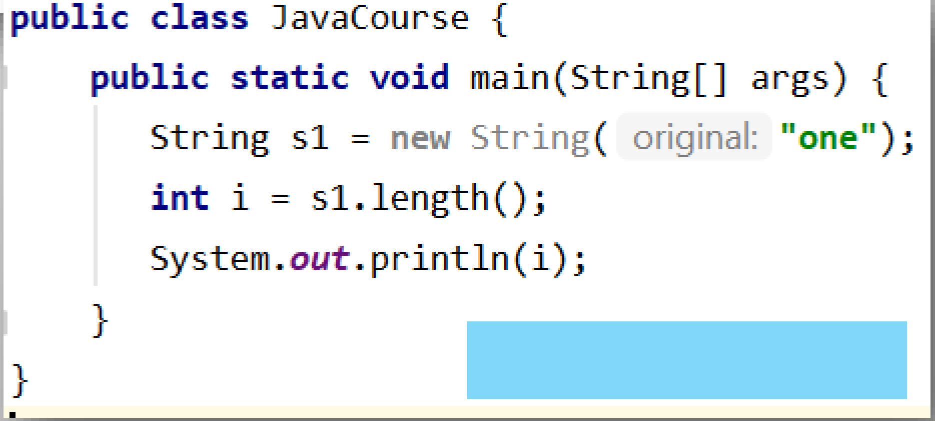
□ Follow this link: https://docs.oracle.com/javase/7/docs/api/java/lang/String.html Go to the <Method Summary> section. Find methods <charAt>, <contains>, and <length>. Read the descriptions and write what would be displayed in the console.



```java
public class JavaCourse {
    public static void main(String[] args) {
        String s = new String();
        s = "Java";
        char c = s.charAt(0);
        System.out.println(c);
    }
}
```



```java
public class JavaCourse {
    public static void main(String[] args) {
        String s1 = new String( original: "one");
        String s2 = new String( original: "on");
        boolean b = s1.contains(s2);
        System.out.println(b);
    }
}
```



```java
public class JavaCourse {
    public static void main(String[] args) {
        String s1 = new String( original: "one");
        int i = s1.length();
        System.out.println(i);
    }
}
```

13-15

□ CREATE A STRING THAT CONTAINS THE WORD ”Happy”. USING THE APPROPRIATE METHOD PRINT OUT THE INDEX OF THE LETTER ”P”.

□ CREATE A STRING THAT CONTAINS THE WORD ”FUNNY”. USING THE APPROPRIATE METHOD CONVERT ALL THE LETTERS FROM CAPITAL TO SMALL AND PRINT THE RESULTANT STRING IN THE CONSOLE.

□ CREATE A STRING CALLED <SI> THAT CONTAINS THE WORD ”UNHAPPY”. USING THE APPROPRIATE METHOD CREATE A STRING <S2> THAT CONTAINS THE LETTERS OF <SI> FROM INDEX 2 TO 7 AND PRINT <S2> IN THE CONSOLE.

16

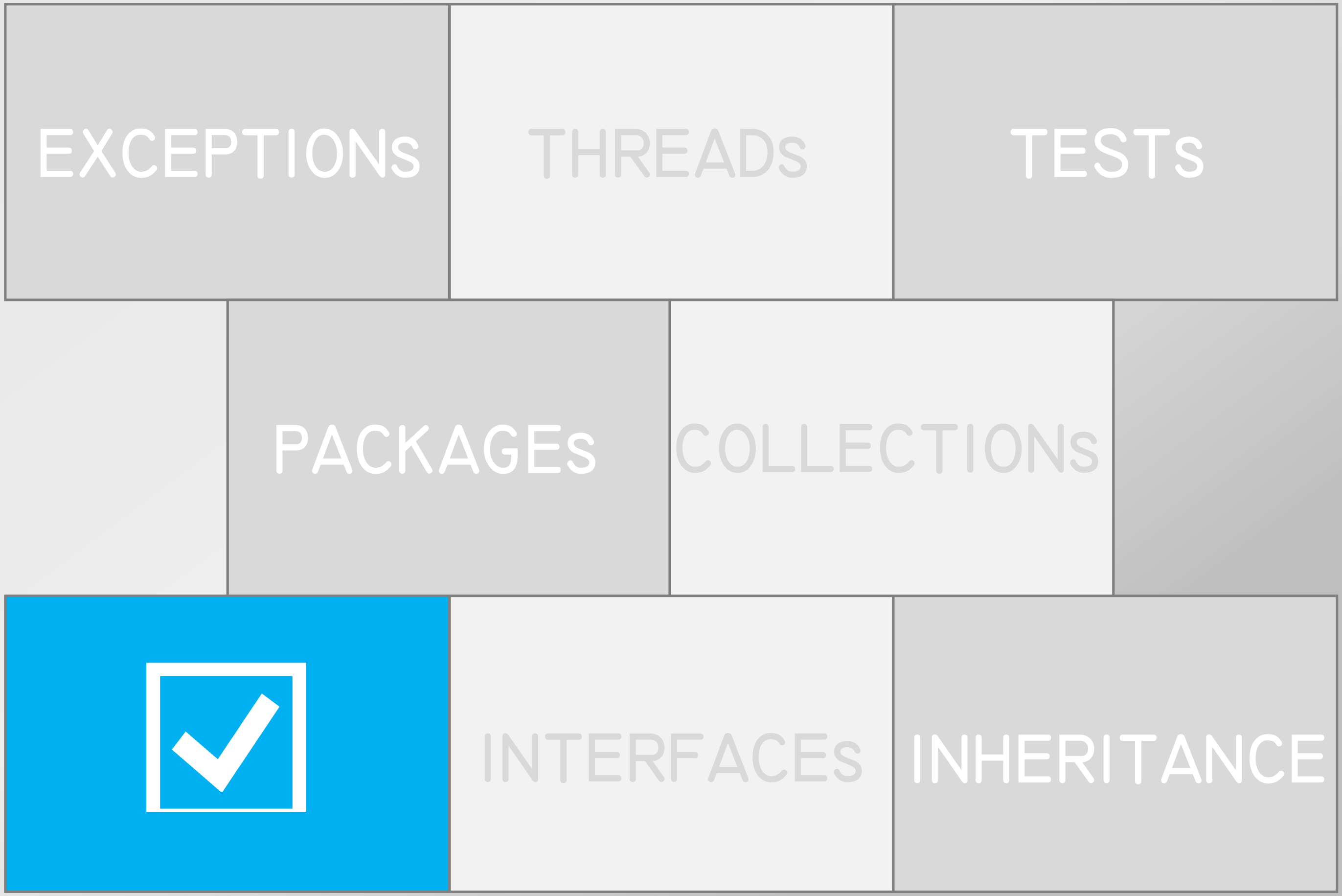
- □ Create a class called <Book>. Add fields of the String class <author> and <Title>, and a field of the integer datatype <noOfPages>.

- □ Add a constructor that takes as arguments the author's name (String), the book title (String), and the number of pages (Integer). Then assigns their content to the corresponding fields of the <Book> class.

- □ Add a method called <titleLength> that returns the length of the title.

- □ In the main program create two objects of the <Book> class: Leo Tolstoy, War and Piece, 1000 pages; René Goscinny, Le Petit Nicola, 100 pages. Call the <titleLength> method for both objects and display the lengthes in the console.

BASICS OF *Java*



EXCEPTIONs

THREADs

TESTs

PACKAGEs

COLLECTIONs

☑

INTERFACEs

INHERITANCE