Arithmetic Operators and Comments

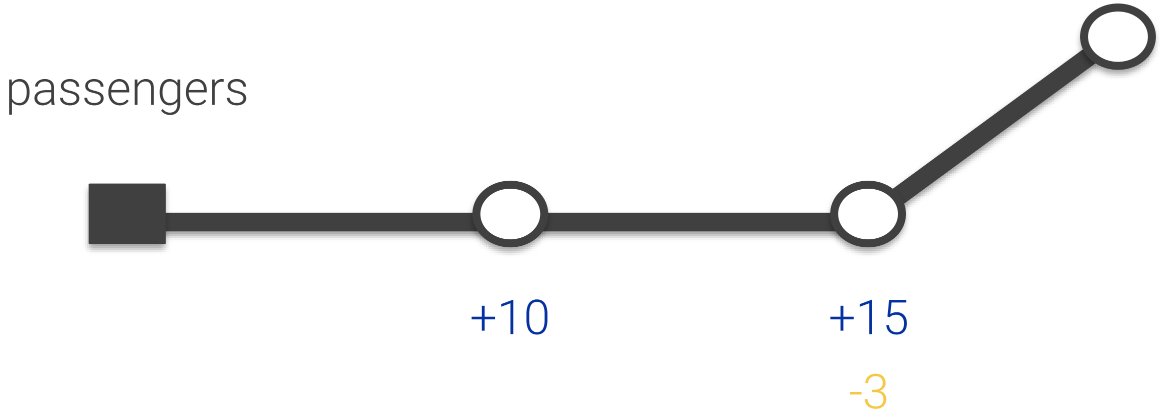
1. Arithmetic operators
2. String concatenation
3. Comments

Arithmetic operators

1. Arithmetic operators

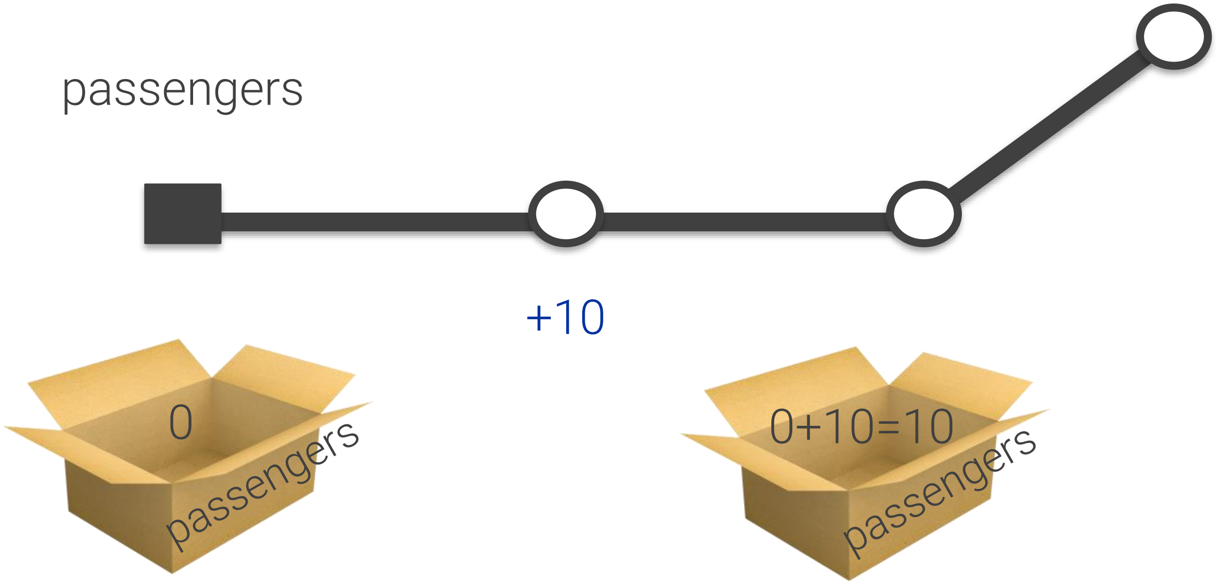
Suma	+
	++
Resta	-
	--
Multiplicación	*
División	/
Resto División	%

1. Arithmetic operators



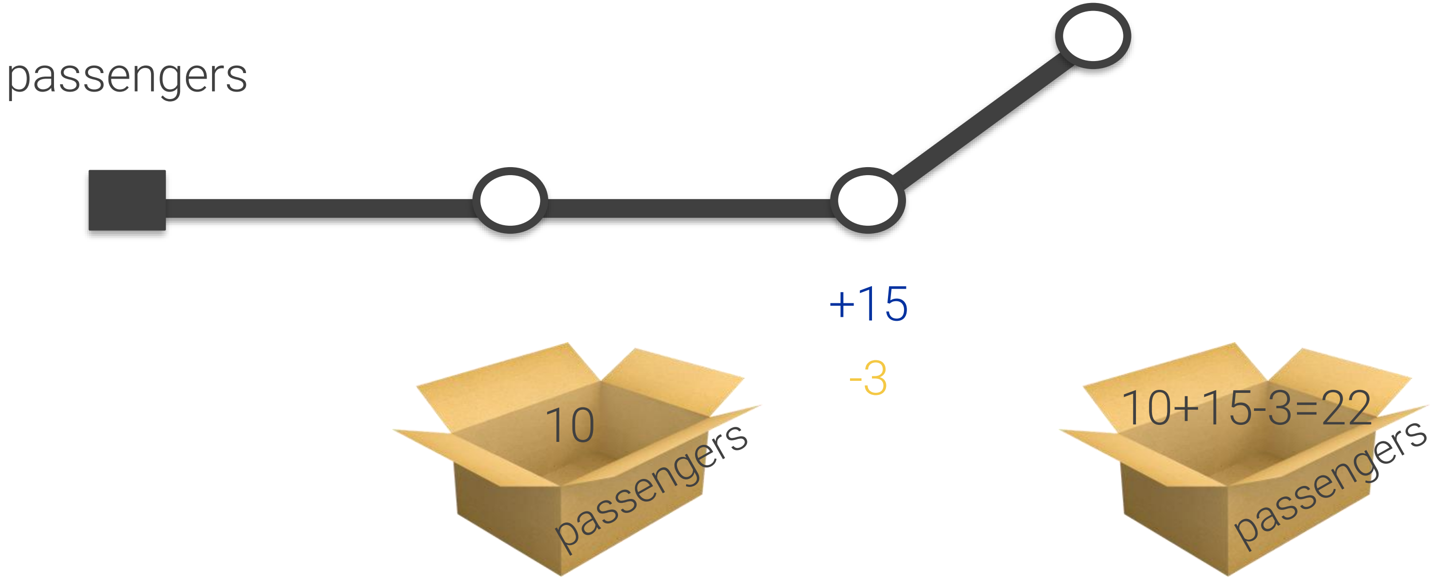
1. Arithmetic operators

```
int passengers = 0;
passengers = passengers + 10;        //first stop
```



1. Arithmetic operators

```
int passengers = 0;
passengers = passengers + 10;         //first stop
passengers = passengers + 15 – 3;   //second stop
```



1. Arithmetic operators

Supongamos que en la siguiente parada del metro se suben 5 pasajeros y se bajan 10. ¿Qué instrucción usarías para actualizar el valor de la variable passengers?

a. passengers = passengers +10 -5;

b. passengers = 5 – 10;

c. passengers = passengers +5 -10;

d. passengers = 10;

1. Arithmetic operators

Supongamos que en la siguiente parada del metro se suben 5 pasajeros y se bajan 10. ¿Qué instrucción usarías para actualizar el valor de la variable passengers?

a. passengers = passengers +10 -5;

b. passengers = 5 – 10;

c. **passengers = passengers +5 -10;**

d. passengers = 10;

1. Arithmetic operators

```
int add= 1 + 2;    //=3
```

1. Arithmetic operators

```
int minus = 1 - 4;    //=-3
```

1. Arithmetic operators

```
int value= 2;
value++;  //=3
```

value = value +1
(postfix notation)

1. Arithmetic operators

```
int value= 2;
++value; //=3
```

value = value +1
(prefix notation)

1. Arithmetic operators

```
int a= 10;
System.out.println(a);
```

Print output

```
10
```

1. Arithmetic operators

The a value will increment after this current value is used

```
int a= 10;
System.out.println(a);
System.out.println(a++);
System.out.println(a);
```

Print output

```
10
10
11
```

1. Arithmetic operators

The a value will increment before this current value is used

```
int a= 10;
System.out.println(a);
System.out.println(++a);
System.out.println(a);
```

Print output

```
10
11
11
```

1. Arithmetic operators

```
int value= 2;
value--;  //=1
--value; //=0
```

value = value -1
(postfix notation)

value = value -1
(prefix notation)

1. Arithmetic operators

```
int multiplication = 3 * 4;    //=12
```

1. Arithmetic operators

```
int div= 5 / 2;
```

1. Arithmetic operators

int div= 5 / 2;   //~~=2.5?~~

= 2

1. Arithmetic operators

double div= 5 / 2; //~~=2.5?~~

= 2.0

1. Arithmetic operators

```
double div= 5 / 2;    //=2.5?
```

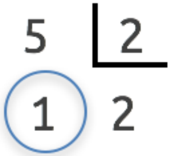
Truncation:
Cutting of the digits to the right. of a decimal point

1. Arithmetic operators

```
double div= 5 / 2.0;    //=2.5
```

1. Arithmetic operators

```
int aux= 5 % 2;    //=1
```



1. Arithmetic operators

```
int x = 1 + 2;    //=3
int y = 4 - 5;    //=-1
int z = x * y;    //=-3
```

1. Arithmetic operators

```
public class Main{
        public static void main(String[] args) {
              double precioMenu= 23.5;
          double pagado= 30;
          double propina= (pagado - precioMenu) * 0.10;  //0.65
        }
}
```

1. Arithmetic operators

```
public class Main{
     public static void main(String[] args) {
          double precioMenu= 23.5;
       double pagado= 30;
       double propina= (pagado - precioMenu) * 0.10;  //0.65
     }
}
```

Parentheses:
Grouping numbers for order of operations

1. Arithmetic operators

```
public class Main{
        public static void main(String[] args) {
              double precioMenu= 23.5;
         double pagado= 30;
         double propina= (pagado - precioMenu) * 0.10;  //0.65
        }
}
```

6.5

1. Arithmetic operators

```
int value1 = (4+6) * 7;   //70
              10
```

Order of operations

1. Parentheses
2. Multiplication and division (from left to right)
3. Addition and substraction (from left to right)

1. Arithmetic operators

```
int value2 = 16 – 3*4 ;  //4
```

12

Order of operations

1. Parentheses
2. Multiplication and division (from left to right)
3. Addition and substraction (from left to right)

1. Arithmetic operators

```
public class Main{
        public static void main(String[] args) {
              double precioMenu= 23.5;
          double pagado= 30;
          double propina= pagado – precioMenu * 0.10;
        }
}
```

2.35

Order of operations

1. Parentheses
2. Multiplication and division (from left to right)
3. Addition and substraction (from left to right)

String concatenation

2. String concatenation

```
String studentFirstName = “John”;
String studentLastName = “Kenedy”;
String studentFullName = studentFirstName + studentLastName;
```

String concatenation

2. String concatenation

```
String studentFirstName = “John”;
String studentLastName = “Kenedy”;
String studentFullName = studentFirstName + studentLastName;
System.out.println(studentFullName);
```

Print output

```
JohnKenedy
```

2. String concatenation

```
String studentFirstName = “John”;
String studentLastName = “Kenedy”;
String studentFullName = studentFirstName+“  ”+studentLastName;
System.out.println(studentFullName);
```

Print output

```
John  Kenedy
```

2. String concatenation

```
public class Main{
  public static void main(String[] args) {
    int stops = 0;
    int passengers = 0;
    stops++;
    passengers = passengers+10;
    System.out.println(“The subway has ”+passengers+” passengers after ”+stops+” stops”);
  }
}
```

Print output

```
The subway has 10 passengers after 1 stops
```

2. String concatenation

```
public class Main{
  public static void main(String[] args) {
    int stops = 0;
    int passengers = 0;
    stops++;
    passengers = passengers+10;
    System.out.println(“The subway has ”+passengers+” passengers after ”+stops+” stops”);
  }
}
```

String literal

Print output

The subway has 10 passengers after 1 stops

2. String concatenation

```
public class Main{
  public static void main(String[] args) {
    int stops = 0;
    int passengers = 0;
    stops++;
    passengers = passengers+10;
    System.out.println(“The subway has ”+passengers+” passengers after ”+stops+” stops”);
  }
}
```

variable

Print output

```
The subway has 10 passengers after 1 stops
```

2. String concatenation

```
public class Main{
  public static void main(String[] args) {
    int stops = 0;
    int passengers = 0;
    stops++;
    passengers = passengers+10;
    System.out.println(“The subway has ”+passengers+” passengers after ”+stops+” stops”);
  }
}
```

String literal

Print output

The subway has 10 **passengers after** 1 stops

2. String concatenation

```
public class Main{
  public static void main(String[] args) {
    int stops = 0;
    int passengers = 0;
    stops++;
    passengers = passengers+10;
    System.out.println(“The subway has ”+passengers+” passengers after ”+stops+” stops”);
  }
}
```

variable

Print output

```
The subway has 10 passengers after 1 stops
```

2. String concatenation

```
public class Main{
  public static void main(String[] args) {
    int stops = 0;
    int passengers = 0;
    stops++;
    passengers = passengers+10;
    System.out.println(“The subway has ”+passengers+” passengers after ”+stops+” stops”);
  }
}
```

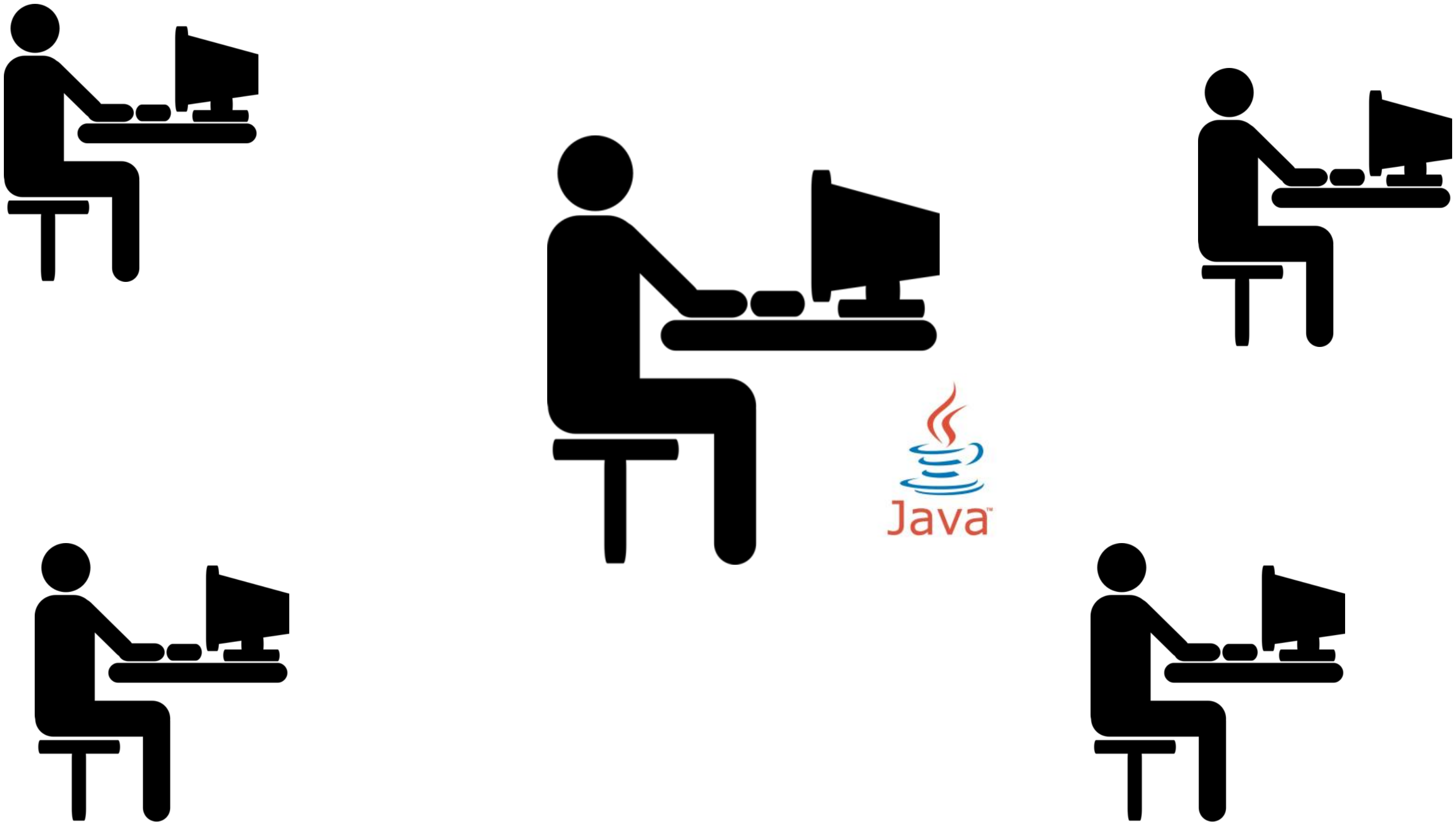
String literal

Print output

```
The subway has 10 passengers after 1 stops
```

Comments

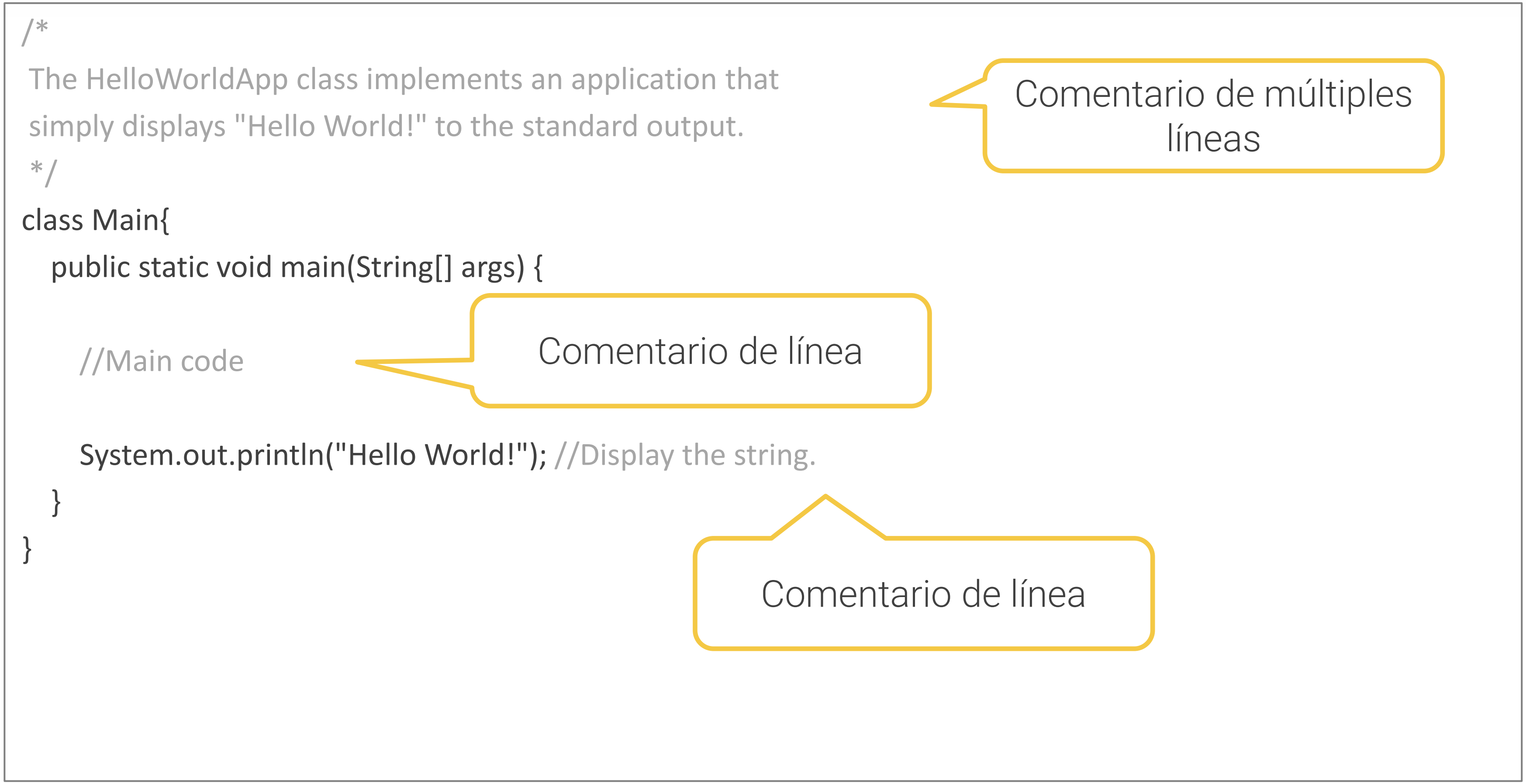
3. Comments



3. Comments

```
class Main{
   public static void main(String[] args) {
      System.out.println("Hello World!");
   }
}
```

3. Comments



```
/*
 The HelloWorldApp class implements an application that
 simply displays "Hello World!" to the standard output.
 */
class Main{
   public static void main(String[] args) {

      //Main code

      System.out.println("Hello World!"); //Display the string.
   }
}
```

Comentario de múltiples líneas

Comentario de línea

Comentario de línea

“Dime y lo olvido, enséñame y lo recuerdo,
involúcrame y lo aprendo.“

BENJAMIN FRANKLIN.