

PROGRAMAÇÃO ORIENTADA A OBJETOS

JAVA

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java

1995



ORIENTADA
A OBJETO

ROBUSTA

PORTATIL

OPERAÇÃO
EM REDE

SEGURANÇA

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java

- *O Java é uma linguagem de programação de propósito geral, concorrente, baseada em classes e orientada a objetos. Projetada para ser simples o bastante para que a maioria dos programadores se torne fluente na linguagem. Java tem relação com C e C++, porém é organizada de forma diferente, com vários aspectos de C e C++ omitidos e algumas ideias de outras linguagens incluídas.*

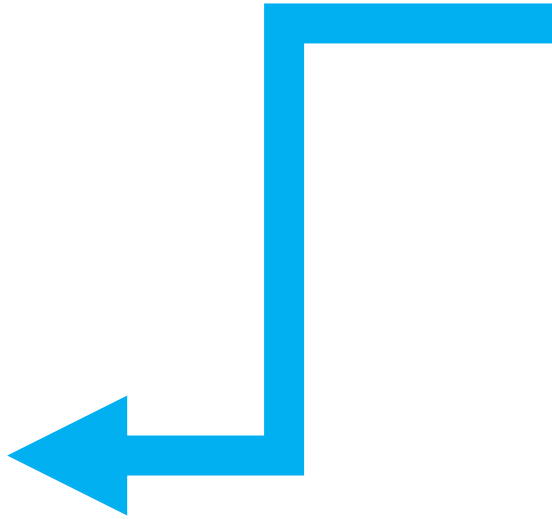
PROGRAMAÇÃO ORIENTADA A OBJETOS

Java

1991



GREEN PROJECT



PROGRAMAÇÃO ORIENTADA A OBJETOS

Java



OAK



GREEN
OS

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java

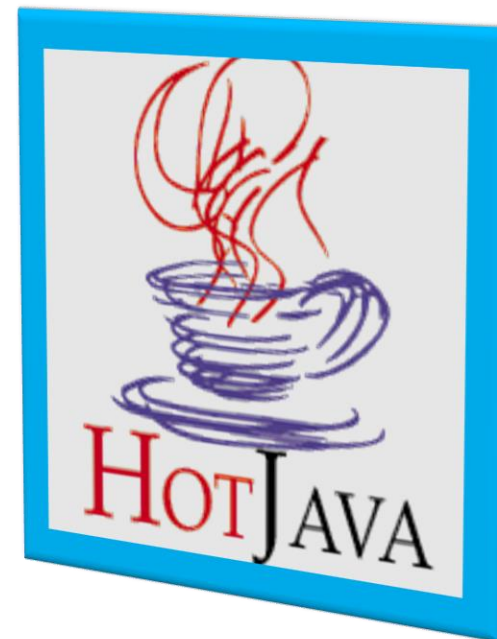


PROGRAMAÇÃO ORIENTADA A OBJETOS

Java



OAK



PROGRAMAÇÃO ORIENTADA A OBJETOS

Java



PROGRAMAÇÃO ORIENTADA A OBJETOS

Java

JDK (JAVA DEVELOPMENT KIT)

JVM (JAVA VIRTUAL MACHINE)

JAVA SE (JAVA STANDARD EDITION)

JAVA EE (JAVA ENTERPRISE EDITION)

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java

ORACLE

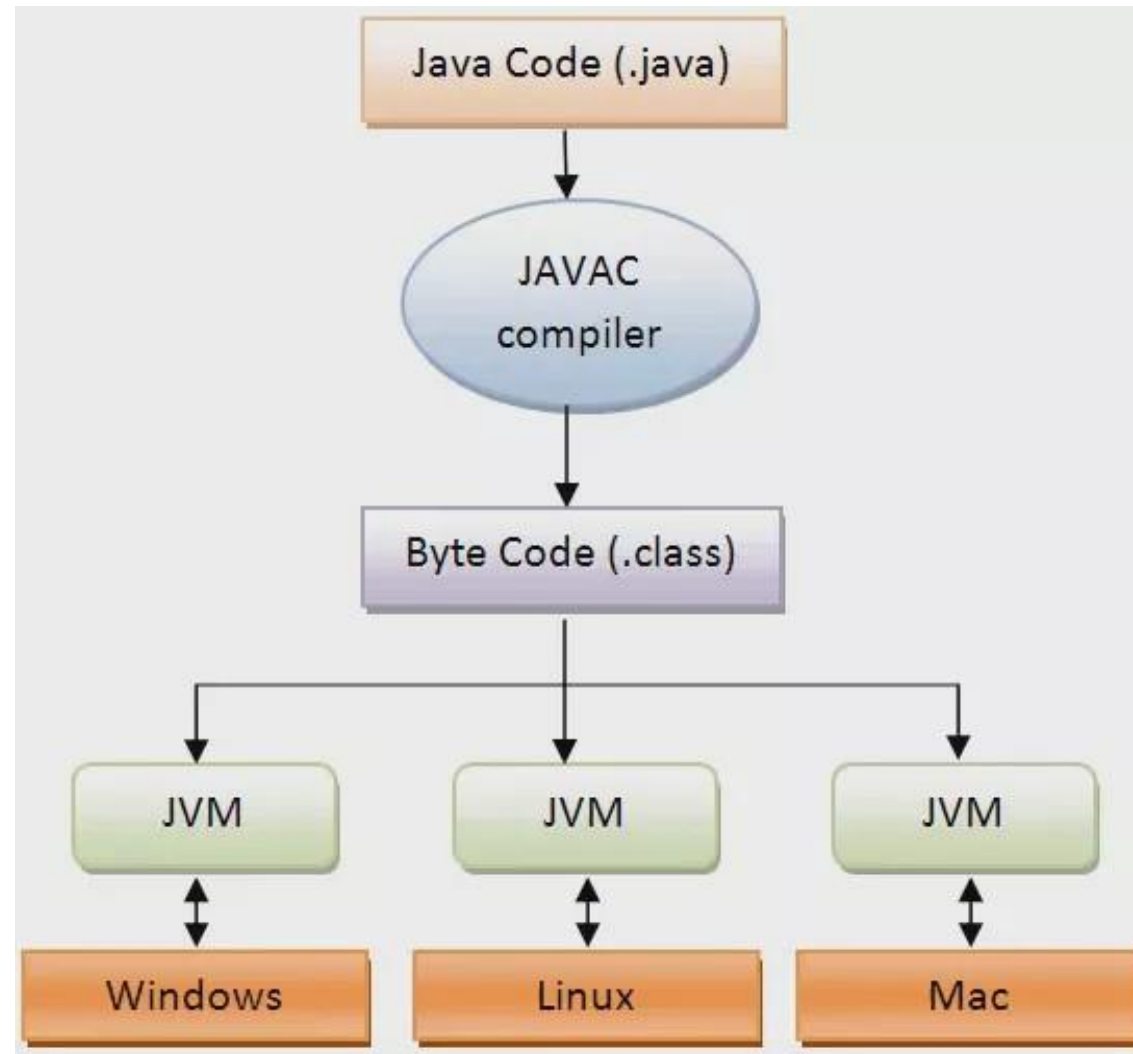


2009



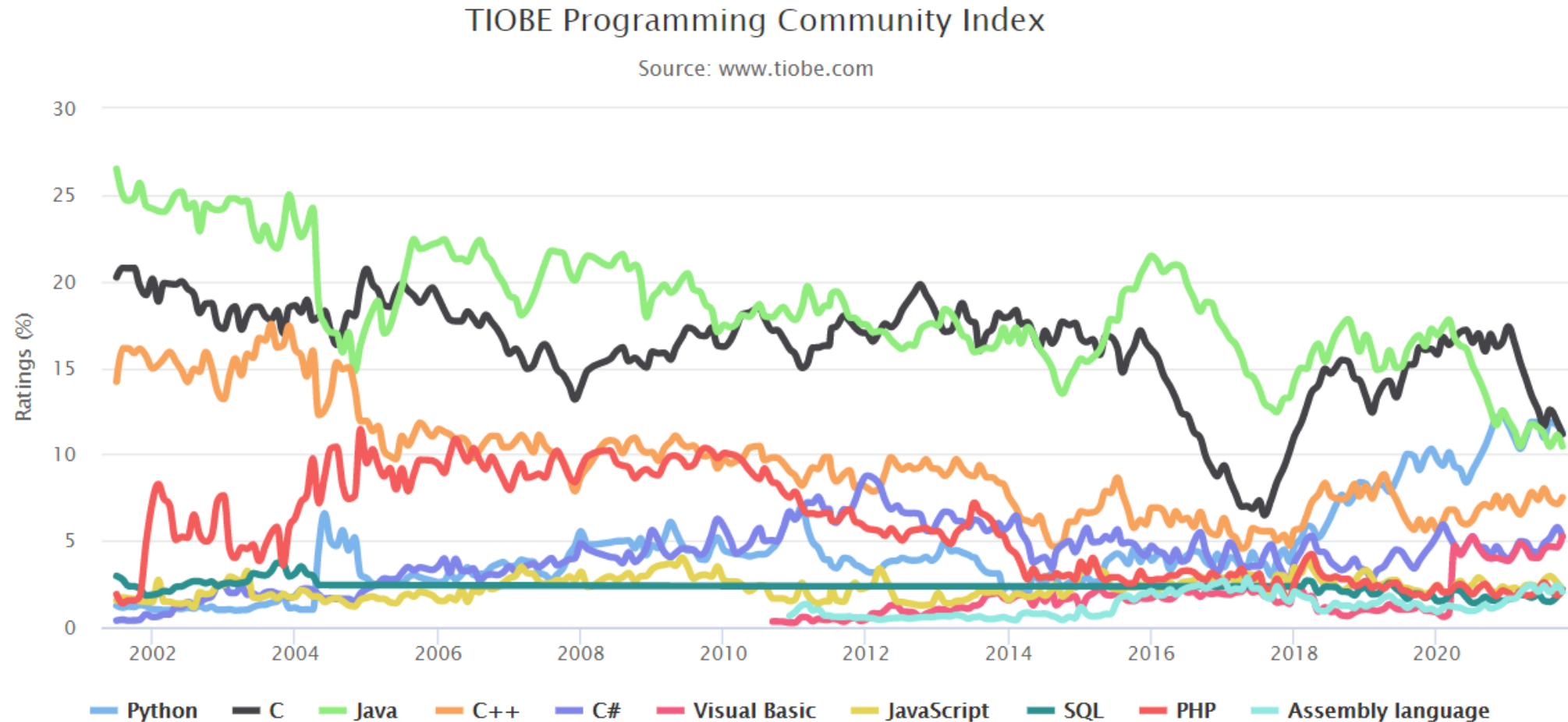
PROGRAMAÇÃO ORIENTADA A OBJETOS

Java - JVM













PROGRAMAÇÃO ORIENTADA A OBJETOS

Linguagens de Programação mais populares



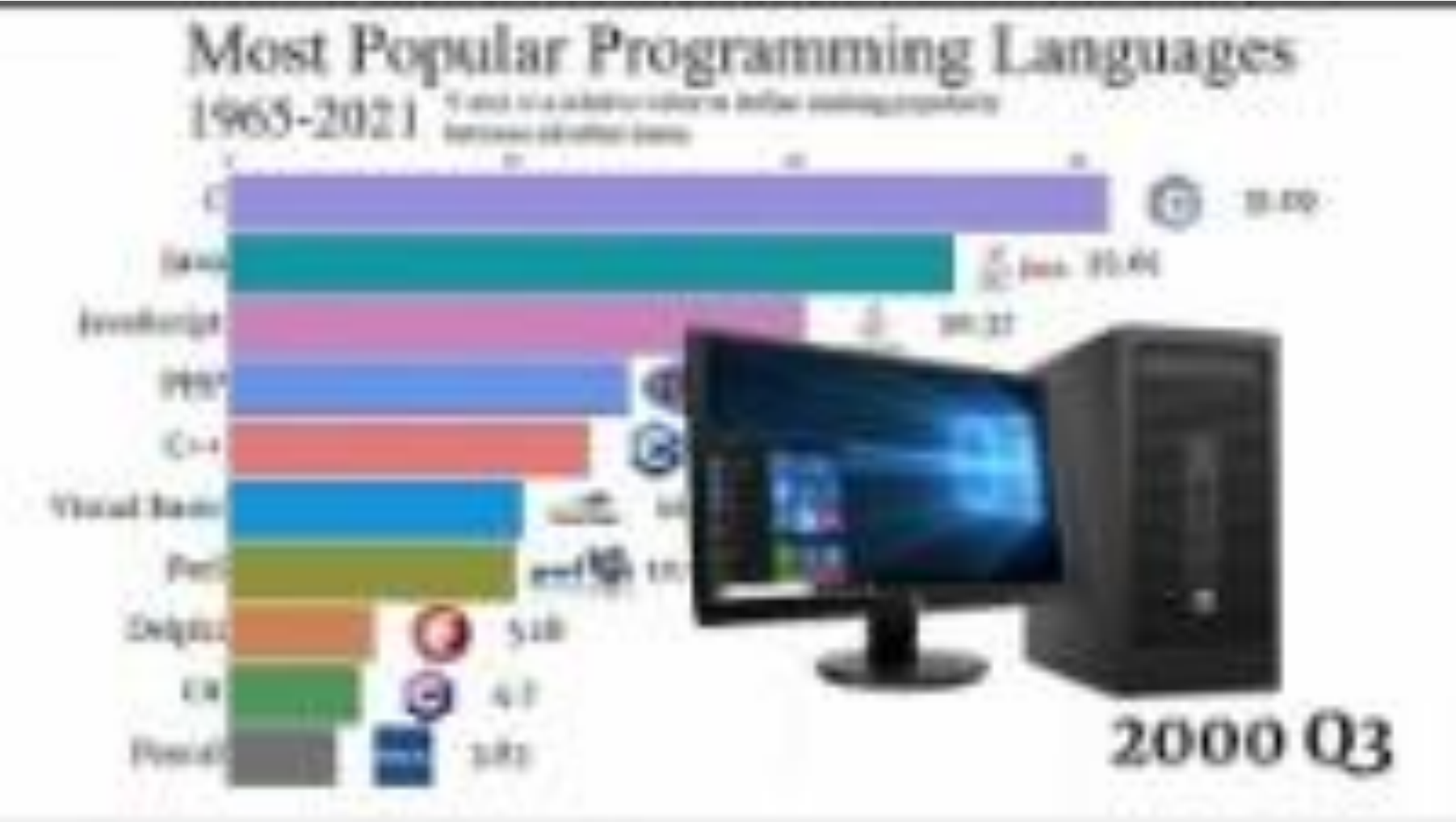
PROGRAMAÇÃO ORIENTADA A OBJETOS

Linguagens de Programação mais populares

Oct 2021	Oct 2020	Change	Programming Language		Ratings
1	3	↑		Python	11.27%
2	1	↓		C	11.16%
3	2	↓		Java	10.46%
4	4			C++	7.50%
5	5			C#	5.26%
6	6			Visual Basic	5.24%
7	7			JavaScript	2.19%
8	10	↑		SQL	2.17%
9	8	↓		PHP	2.10%
10	17	↑↑		Assembly language	2.06%

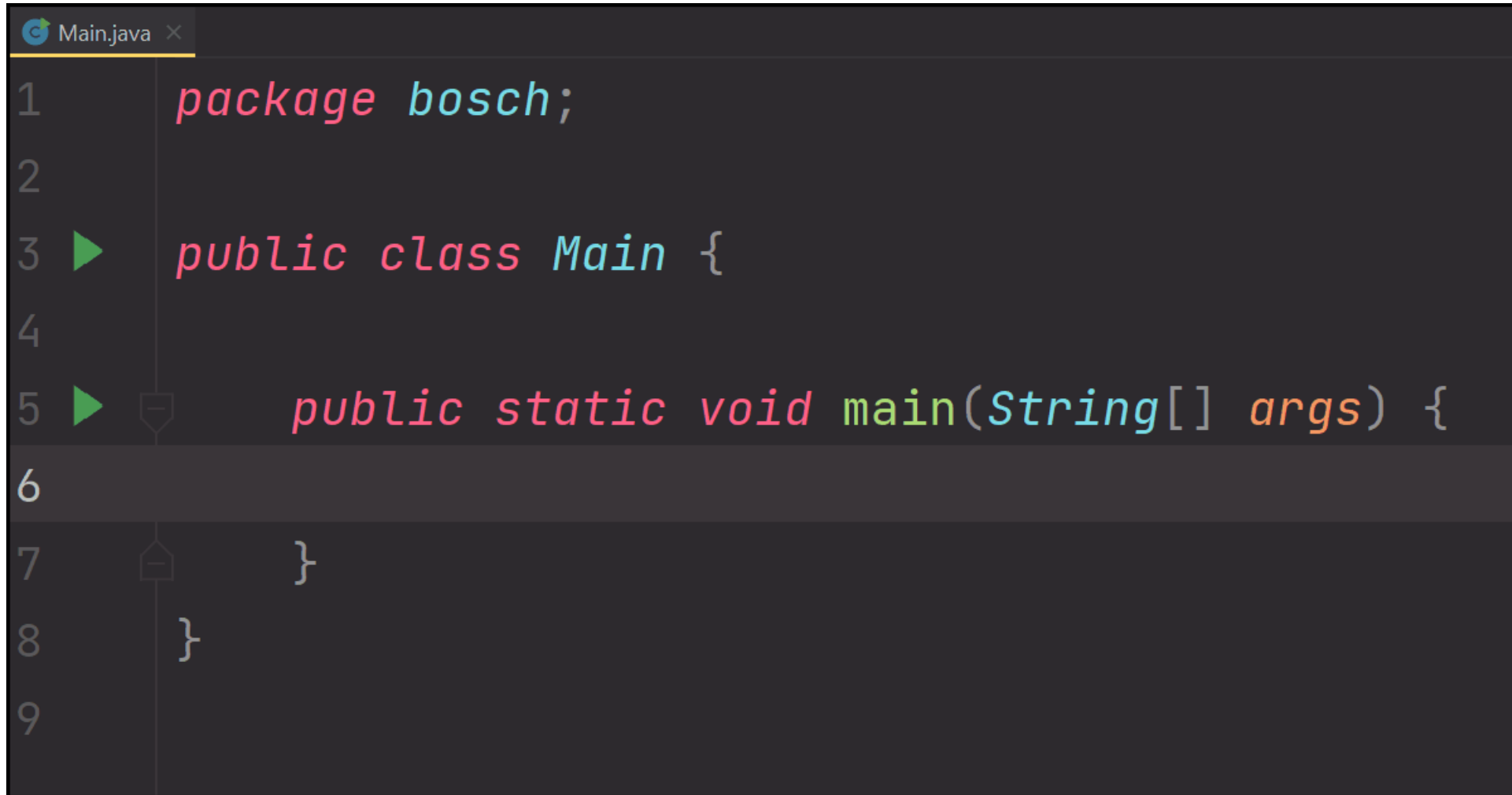
PROGRAMAÇÃO ORIENTADA A OBJETOS

Linguagens de Programação mais populares



PROGRAMAÇÃO ORIENTADA A OBJETOS

JAVA – Comentário de uma linha

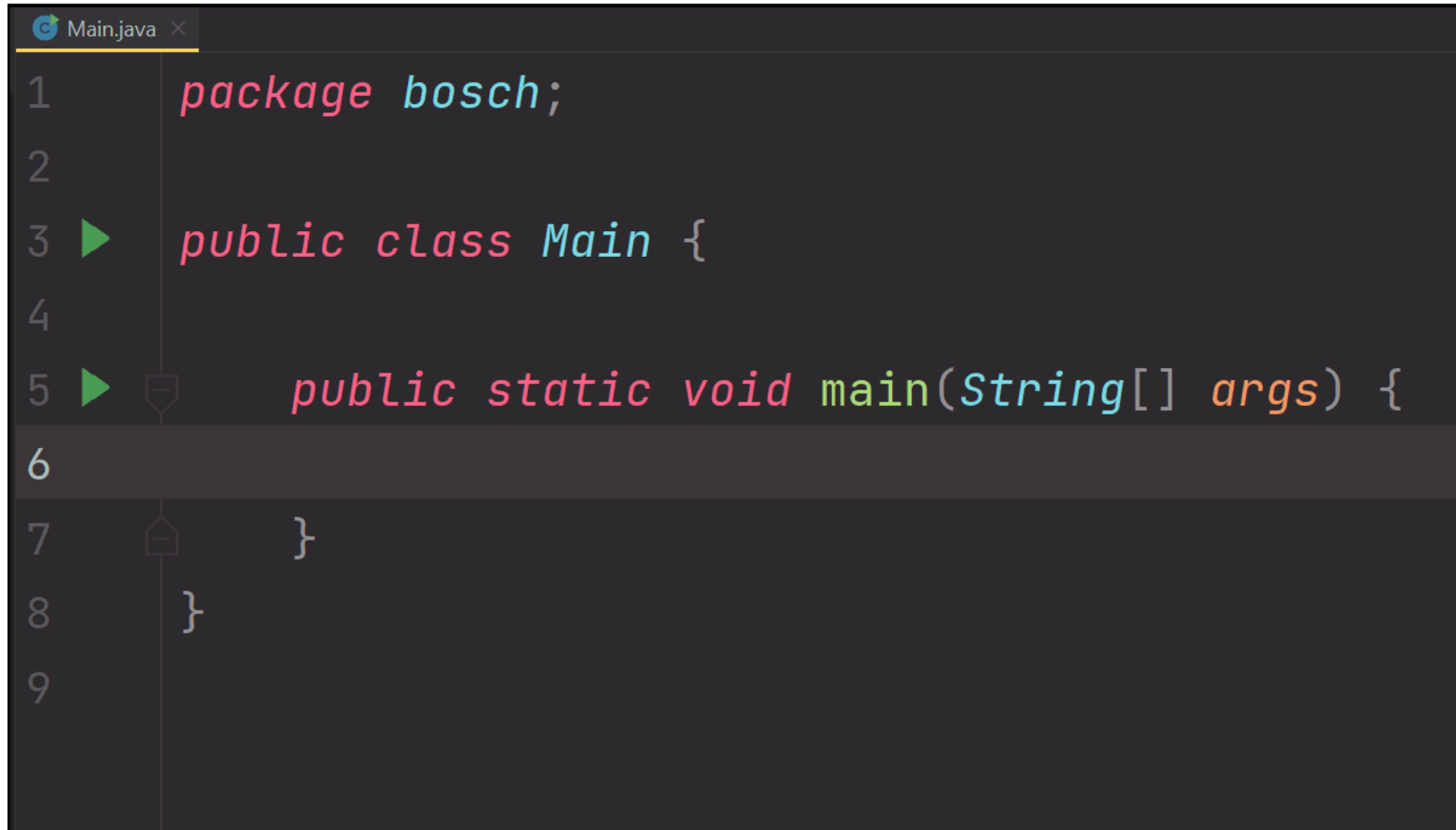


```
1  package bosch;  
2  
3  ▶ public class Main {  
4  
5  ▶  └─ public static void main(String[] args) {  
6  
7  └─ }  
8  }  
9
```

The image shows a code editor window titled 'Main.java'. The code is written in Java and includes a package declaration, a class declaration, and a main method. A single-line comment is placed at the beginning of the main method. The code is as follows:

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Comentário de Múltiplas linhas



The screenshot shows a code editor window titled 'Main.java'. The code is as follows:

```
1 package bosch;  
2  
3 public class Main {  
4  
5     public static void main(String[] args) {  
6  
7     }  
8 }  
9
```

Lines 5 and 6 are highlighted with a green background. A green arrow points to line 5, and a green arrow points to line 6. A vertical line connects the two arrows, indicating a multi-line comment.

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Saída não formatada



The image shows a screenshot of a Java IDE. The top part displays a code editor with a file named 'Main.java'. The code is as follows:

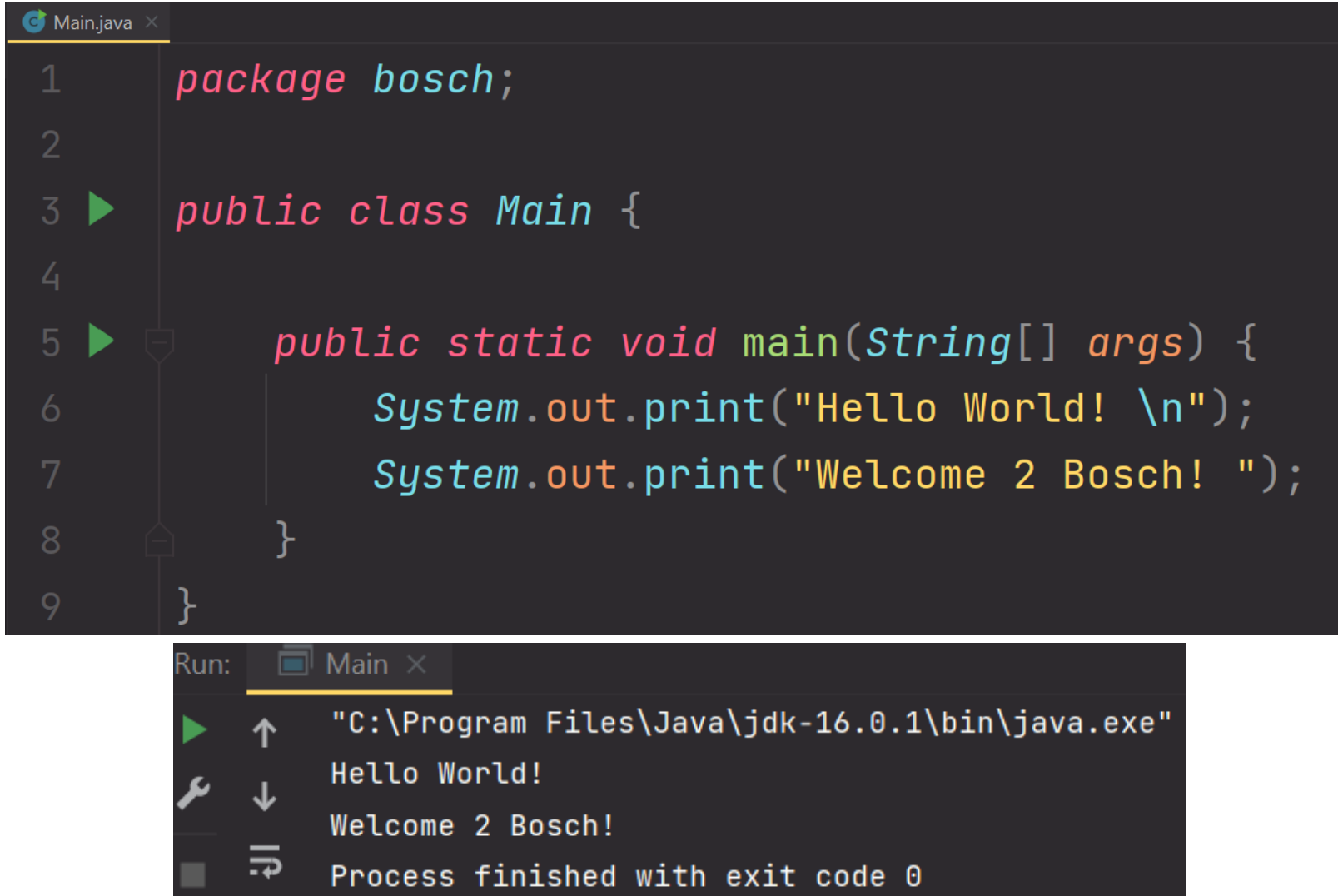
```
1 package bosch;  
2  
3 public class Main {  
4  
5     public static void main(String[] args) {  
6         System.out.print("Hello World!");  
7         System.out.print("Welcome 2 Bosch! ");  
8     }  
9 }
```

The bottom part of the screenshot shows the 'Run' console. It displays the command used to run the program, the output, and the exit code:

```
Run: Main  
↑ "C:\Program Files\Java\jdk-16.0.1\bin\java.exe"  
Hello World! Welcome 2 Bosch!  
↓  
Process finished with exit code 0
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Saída não formatada



The image shows a screenshot of an IDE with a dark theme. The top part displays a Java file named 'Main.java' with the following code:

```
1 package bosch;  
2  
3 public class Main {  
4  
5     public static void main(String[] args) {  
6         System.out.print("Hello World! \n");  
7         System.out.print("Welcome 2 Bosch! ");  
8     }  
9 }
```

The bottom part shows the 'Run' output window for the 'Main' class. It displays the command used to run the program, the output of the program, and the exit code.

```
Run: Main ×  
↑ "C:\Program Files\Java\jdk-16.0.1\bin\java.exe"  
Hello World!  
↓  
Welcome 2 Bosch!  
Process finished with exit code 0
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Saída não formatada

```
Main.java x
1  package bosch;
2
3  ▶ public class Main {
4
5  ▶   public static void main(String[] args) {
6      System.out.println("Hello World! ");
7      System.out.print("Welcome 2 Bosch! ");
8   }
9 }
```

```
Run: Main x
▶ ↑ "C:\Program Files\Java\jdk-16.0.1\bin\java.exe"
  ↓ Hello World!
  ⚙ ↓ Welcome 2 Bosch!
  ■ ⚙ Process finished with exit code 0
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Saída formatada

```
Main.java x
1  package bosch;
2
3  ▶ public class Main {
4
5  ▶  - public static void main(String[] args) {
6      // Area do circulo
7      int raio = 5;
8      final double pi = 3.141;
9      double areaCirculo=pi*raio*raio;
10     System.out.printf(" A área do criculo é %.2f",areaCirculo);
11     }
12 }
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Tipos de Dados Primitivos

Tipo Inteiro

byte

short

int

long

Tipo Real

float

double

Caractere

char

Lógico

boolean

PROGRAMAÇÃO ORIENTADA A OBJETOS

Categoria	Tipo	Bytes	bits	Faixa de Valores
Inteiro	Byte	1	8	De -128 a +127
	Short	2	16	De -32.768 a 32.767
	Int	4	32	De -2.147.483.648 a +2.147.483.647
	Long	8	64	De -223.372.036.854.775.808 a +9.223.372.036.854.775.807
Real	float	4	32	Valores Positivos: +1.40129846432481707e-45 a 3.402823466385288860e+38 Valores Negativos: -3.402823466385288860e+38 a +1.40129846432481707e-45
	double	8	64	Valores Positivos: +4.94065645841246544e-324 a 1.7976933486231570e+108 Valores Negativos: - 1.7976933486231570e+108 a -4.94065645841246544e-324
Caractere	char	2	16	De u\0000 a u\FFFF
Lógico	boolean	1	8	false e true

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Tipos de Dados Primitivos

```
Main.java x
1      package bosch;
2
3      public class Main {
4
5      public static void main(String[] args) {
6          // Tipos de Variaveis Inteiras
7          byte meuByte=127;
8          System.out.printf("Tamanho do Byte: %d\n",meuByte);
9          short meuShort=32767;
10         System.out.printf("Tamanho do Short: %d\n", meuShort);
11         int meuInt = 2_147_483_647;
12         System.out.printf("Tamanho do Int: %d\n", meuInt);
13         long meuLong = 9_223_372_036_854_775_807L;
14         System.out.printf("Tamanho do Long: %d\n",meuLong);
15     }
16 }
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Tipos de Dados Primitivos

```
Main.java x
1  package bosch;
2
3  ▶ public class Main {
4
5  ▶   public static void main(String[] args) {
6       //precisão de 6 a 7 dígitos depois da vírgula
7       float meuFloat = 3.4e+38F;
8       System.out.printf("Valor de Float: %f\n", meuFloat);
9       // precisão de 15 a 16 dígitos depois da vírgula
10      double meuDouble = 1.7e+308;
11      System.out.printf("Valor de Double: %f\n", meuDouble);
12  }
13 }
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Tipos de Dados Primitivos

```
Main.java x
1      package bosch;
2
3      ▶ public class Main {
4
5      ▶   public static void main(String[] args) {
6          char meuCaracter = 'A';
7          System.out.printf("Meu caracter é: %c\n", meuCaracter);
8          boolean meuBoolean = true;
9          System.out.printf("Meu booleano é: %b\n ", meuBoolean);
10
11      }
12  }
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – String

```
Main.java ×
1  package bosch;
2
3  ▶ public class Main {
4
5  ▶  - public static void main(String[] args) {
6      String minhaString=" Hello World !";
7      System.out.println(minhaString);
8      - }
9  }
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Tipos de Dados Primitivos

```
1 package bosch;
2
3 import java.util.Scanner;
4
5 public class Main {
6
7     public static void main(String[] args) {
8         int numero;
9         Scanner entrada = new Scanner(System.in);
10        System.out.print("Digite um número: ");
11        numero=entrada.nextInt();
12        System.out.print("O número que você digitou foi: "+numero);
13        entrada.close();
14    }
15 }
16
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Entrada Formatada

```
1 package bosch;
2
3 import java.util.Scanner;
4
5 public class Main {
6
7     public static void main(String[] args) {
8         String palavra;
9         Scanner entrada = new Scanner(System.in);
10        System.out.print("Digite uma palavra: ");
11        palavra=entrada.nextLine();
12        System.out.print("A palavra que você digitou foi: "+palavra);
13        entrada.close();
14    }
15 }
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Entrada Formatada

```
5 ▶ public class Main {  
6  
7 ▶   public static void main(String[] args) {  
8       Scanner entrada = new Scanner(System.in);  
9       String nome,sobrenome;  
10      int idade;  
11      nome=entrada.nextLine();  
12      System.out.println(nome);  
13      idade=entrada.nextInt();  
14      System.out.println(idade);  
15      entrada.nextLine();  
16      sobrenome=entrada.nextLine();  
17      System.out.println(sobrenome);  
18   }  
19 }
```


PROGRAMAÇÃO ORIENTADA A OBJETOS

Java - Inferência de Tipos

```
Main.java x
1  package bosch;
2
3  ▶ public class Main {
4
5  ▶   ▶ public static void main(String[] args) {
6      var a=10;
7      System.out.println(a);
8      var b=3.14;
9      System.out.println(b);
10     var c = 'x';
11     System.out.println(c);
12     var d = false;
13     System.out.println(d);
14     var e = "Hello World!";
15     System.out.println(e);
16   }
17 }
```


PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Convertendo para Objeto e Verificando o Tipo

```
Main.java x
1  package bosch;
2
3  ▶ public class Main {
4
5  ▶   public static void main(String[] args) {
6      var a = 10;
7      System.out.println(a+ " "+
8      ((Object)a).getClass().getSimpleName());
9  }
10 }
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Classe Math

```
1  package bosch;  
2  
3  
4  ▶ public class Main {  
5  
6  ▶   public static void main(String[] args) {  
7      double raiz;  
8      raiz=Math.sqrt(4);  
9      System.out.println(raiz);  
10  
11     double potencia;  
12     potencia= Math.pow(2,3);  
13     System.out.println(potencia);  
14  
15     }  
16 }
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Operadores Aritméticos

```
1  package bosch;  
2  
3  ▶ public class Main {  
4  
5  ▶   public static void main(String[] args) {  
6      int num1=9;  
7      int num2=2;  
8      System.out.println(num1+num2); // Adição  
9      System.out.println(num1-num2); //Subtração  
10     System.out.println(num1*num2); //Multiplicação  
11     System.out.println(num1/num2); //Divisão  
12     System.out.println(num1%num2); //Resto da divisão  
13   }  
14 }
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Operadores Aritméticos

```
Main.java x
1  package bosch;
2
3  ▶ public class Main {
4
5  ▶   public static void main(String[] args) {
6       double num1=11;
7       double num2=4;
8       System.out.println(num1+num2); // Adição
9       System.out.println(num1-num2); //Subtração
10      System.out.println(num1*num2); //Multiplicação
11      System.out.println(num1/num2); //Divisão
12      System.out.println(num1%num2); //Resto da divisão
13  }
14 }
```

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Exercícios Básicos

1. Criar um programa que leia a temperatura em Fahrenheit e converta para Celsius.
2. Criar um programa que leia a temperatura em Celsius e converta para Fahrenheit.
3. Criar um programa que leia o peso e a altura do usuário e imprima no console o IMC.
4. Criar um programa que leia um valor e apresente os resultados ao quadrado e ao cubo do valor.
5. Criar um programa que leia o valor da base e da altura de um triângulo e calcule a área.
6. Criar um programa que resolve equações do segundo grau ($ax^2 + bx + c = 0$) utilizando a fórmula de Bhaskara. Use como exemplo $a = 1$, $b = 12$ e $c = -13$. Encontre o delta

PROGRAMAÇÃO ORIENTADA A OBJETOS

Java – Exercícios Básicos

$$1) \left[\frac{3 \cdot \left(\frac{-3}{4} \right)^{-2} + 6 \cdot \left(\frac{3^{-1}}{4} \right) - 4}{7 \cdot \left(\frac{-3}{4} \right)^{-1} + 2} \right]^{-1} + 4 =$$