
SENAI

Aula 08

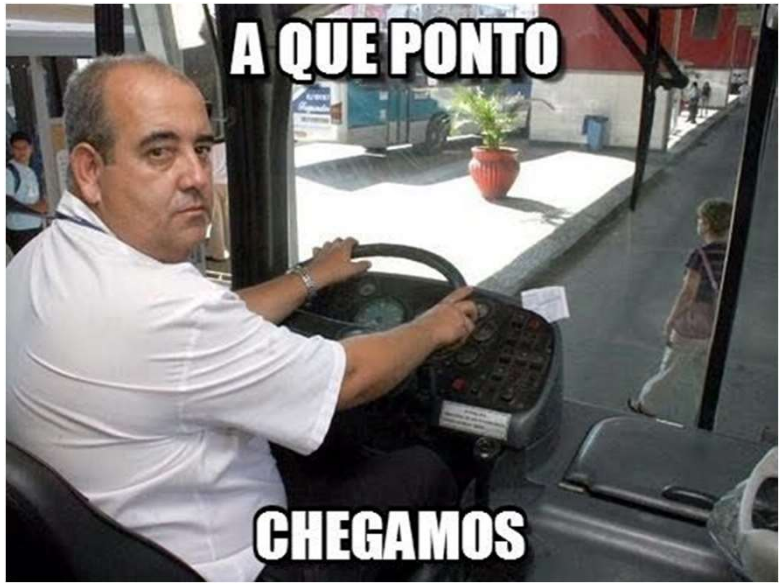
Classes para tratamento de dados

SENAI

Orientação a Objetos

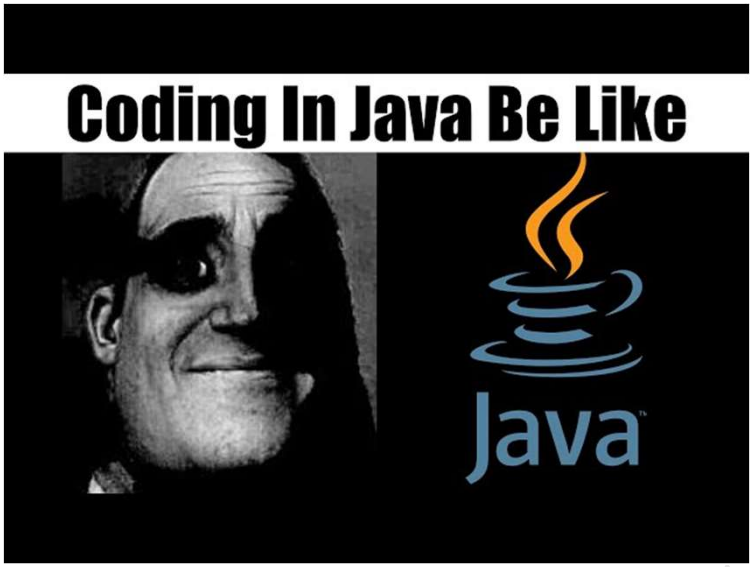
Na aula passada...

- Funções em Java: definição, síntaxe e usos práticos;
- Método construtor: definição, tipos e usos práticos;
- Revisão da aula 1 de Orientação a Objetos.



Na aula de hoje

- Explorar explorações matemáticas com a classe ‘Math’;
- Geração de números aleatórios com a classe ‘Random’;
- Exemplos de código das classes;
- Conversão e formatação de data, hora e números com as classes Date, DateFormat, Calendar, Formatter, NumberFormat;
- Enumerações (enum);
- Exemplos de código;
- Exercícios práticos;



Classes Math e Random
Math

Conceito

- A classe Math faz parte do pacote java.lang e oferece métodos estáticos para operações matemáticas.

- Não é necessário instalar a classe, pois faz parte da biblioteca padrão do Java.

Instalação

A classe Math não requer instalação manual, pois já vem embutida na linguagem Java. Basta importá-la em seu código:

```
import java.lang.Math;
```

Métodos Comuns

Funções Trigonométricas
`Math.sin(), Math.cos(), Math.tan()`

Exponenciação e Logaritmos
`Math.exp(), Math.log()`

Raízes e Potências
`Math.sqrt(), Math.pow()`

Arredondamentos
`Math.ceil(), Math.floor(), Math.round()`

Classes Math e Random
Math

Exemplos de Código

```
public class MathExamples {

    public static void main(String[] args) {

        double angle = Math.toRadians(45);
        System.out.println("Seno de 45°: " +
Math.sin(angle));
        System.out.println("Exponencial de 2: "
+ Math.exp(2));
        System.out.println("Raiz quadrada de 16:
" + Math.sqrt(16));
        System.out.println("2 elevado a 3: " +
Math.pow(2, 3));
    }
}
```

Usos práticos

- Cálculo de potências e raízes quadradas
- Funções trigonométricas (seno, cosseno, tangente)
- Arredondamento e formatação de números
- Operações com números complexos

Classes Math e Random
Math

Principais Síntaxes

Math.sqrt(x): Calcula a raiz quadrada de x.
Math.pow(x, y): Calcula x elevado a y.
Math.sin(x): Calcula o seno de x (em radianos).
Math.cos(x): Calcula o cosseno de x (em radianos).
Math.tan(x): Calcula a tangente de x (em radianos).
Math.round(x): Arredonda x para o inteiro mais próximo.
Math.ceil(x): Arredonda x para o inteiro mais próximo maior que x.
Math.floor(x): Arredonda x para o inteiro mais próximo menor que x.
Math.abs(x): Calcula o valor absoluto de x.
Math.max(x, y): Retorna o maior valor entre x e y.
Math.min(x, y): Retorna o menor valor entre x e y.
Math.PI: Constante que representa o valor de Pi.
Math.E: Constante que representa o valor de Euler.

Classes Math e Random
Math

Exemplos de Código

```
public class MathExamples {

    public static void main(String[] args) {

        double angle = Math.toRadians(45);
        System.out.println("Seno de 45°: " +
Math.sin(angle));
        System.out.println("Exponencial de 2: "
+ Math.exp(2));
        System.out.println("Raiz quadrada de 16:
" + Math.sqrt(16));
        System.out.println("2 elevado a 3: " +
Math.pow(2, 3));
    }
}
```

```
double latitudeRestaurante = -23.5489;
double longitudeRestaurante = -46.6325;
double latitudeCliente = -23.6031;
double longitudeCliente = -46.7109;

double distanciaEmLinhaReta = Math.sqrt(
    Math.pow(latitudeCliente -
latitudeRestaurante, 2) +
    Math.pow(longitudeCliente -
longitudeRestaurante, 2)
);

System.out.println("Distância em linha
reta: " + distanciaEmLinhaReta + " km");
```

Classes Math e Random
Math

Exemplos de Código

```
Random random = new Random();
double anguloAleatorio =
Math.toRadians(random.nextDouble() * 360);

System.out.println("Ângulo aleatório: " +
Math.toDegrees(anguloAleatorio) + " graus");
```

A sintaxe da classe Math varia de acordo com o método utilizado. Consulte a documentação oficial para detalhes específicos de cada um.

Classes Math e Random
Random

Conceito

- A classe Random faz parte do pacote java.util e gera números pseudo-aleatórios.

- Não é necessário instalar a classe, pois faz parte da biblioteca padrão do Java.

A classe Random permite gerar números aleatórios com diversas distribuições de probabilidade.

Instalação

Assim como a classe Math, a Random não exige instalação manual, pois já vem embutida na linguagem Java. Basta importá-la em seu código:

```
import java.util.Random;
```

Métodos Comuns

Geração de Inteiros
`nextInt()`

Geração de Doubles
`nextDouble()`

Geração de Booleans
`nextBoolean()`

Semente
`setSeed()`

Classes Math e Random
Random

Principais Síntaxes

`Random random = new Random();` : Cria um novo objeto Random com um gerador de números aleatórios baseado no tempo.

`random.nextInt(n):` Gera um inteiro aleatório entre 0 (inclusive) e n (exclusivo).

`random.nextDouble():` Gera um número aleatório entre 0 (inclusive) e 1 (exclusivo).

`random.nextFloat():` Gera um número aleatório entre 0 (inclusive) e 1 (exclusivo), como float.

`random.nextLong():` Gera um número aleatório long.

`random.nextBoolean():` Gera um valor booleano aleatório.

Classes Math e Random
Random

Exemplos de Código

```
// Gerando um inteiro aleatório entre 0 e 10 (exclusivo)
Random random = new Random();
int numeroAleatorio = random.nextInt(10);
System.out.println("Número aleatório: " + numeroAleatorio);

// Gerando um número aleatório entre 0 e 1
double numeroAleatorio2 = random.nextDouble();
System.out.println("Número aleatório (double): " + numeroAleatorio2);

// Simulando um lançamento de moeda
boolean caraOuCoroa = random.nextBoolean();
if (caraOuCoroa) {
    System.out.println("Cara!");
} else {
    System.out.println("Coroa!");
}
```

```
import java.util.Random;

public class ExemplosNumerosAleatorios {
    public static void main(String[] args)
{
        Random rand = new Random();
        System.out.println("Número inteiro aleatório: " + rand.nextInt());
        System.out.println("Número inteiro entre 0 e 100: " + rand.nextInt(100));
        System.out.println("Número double aleatório: " + rand.nextDouble());
        System.out.println("Valor booleano aleatório: " + rand.nextBoolean());
    }
}
```

Classes Math e Random
Random

Exemplos de Código

```
import java.util.Random;

public class RandomExamples {
    public static void main(String[] args) {
        Random rand = new Random();
        System.out.println("Número inteiro
aleatório: " + rand.nextInt());
        System.out.println("Número inteiro entre
0 e 100: " + rand.nextInt(100));
        System.out.println("Número double
aleatório: " + rand.nextDouble());
        System.out.println("Valor booleano
aleatório: " + rand.nextBoolean());
    }
}
```

Usos práticos

- Geração de dados de teste
- Simulações estatísticas
- Jogos e sorteios

Classes Math e Random
Exercícios

Classe Math

- Calcular o seno, cosseno e tangente de um ângulo fornecido pelo usuário.

Classe Random

- Gerar 10 números inteiros aleatórios entre 1 e 50.

Conversão, Formatação e enum
Classe Date

Conceito
A classe java.util.Date representa um ponto específico no tempo, medido em milissegundos desde 1° de janeiro de 1970, 00:00:00 GMT.

Criação de Objetos Date
- Date dataAtual = new Date(); // Cria um objeto com a data e hora atuais
- Date dataEspecifica = new Date(long milisegundos); // Cria um objeto com a data e hora correspondentes aos milissegundos.

Métodos úteis
- long getTime(): Retorna os milissegundos desde 1° de janeiro de 1970.
- void setTime(long milisegundos): Define a data e hora com base nos milissegundos.

Para importar a classe

```
import java.util.Date;
```

Exemplo de código

```
import java.util.Date;

public class DateExample {
    public static void main(String[] args)
{
        Date now = new Date();
        System.out.println("Data e Hora
atuais: " + now);
    }
}
```

Conversão, Formatação e enum
Classe Calendar

Conceito
A classe java.util.Calendar fornece métodos para manipular campos específicos de datas, como ano, mês, dia, hora, minuto e segundo.

Criação de Objetos Calendar
- `Calendar calendario = Calendar.getInstance();` // Obtém um Calendar com a data e hora atuais.

Manipulando campos:
- `int ano = calendario.get(Calendar.YEAR);`
- `calendario.set(Calendar.MONTH, Calendar.DECEMBER);`
- `calendario.add(Calendar.DAY_OF_MONTH, 5);`

//Adiciona 5 dias à data atual.

Para importar a classe

```
import java.util.Calendar;
```

Exemplo de código

```
import java.util.Calendar;

public class ExemploCalendar {
    {
        Calendar calendar =
Calendar.getInstance();
        System.out.println("Ano atual: " +
calendar.get(Calendar.YEAR));
        System.out.println("Mês atual: " +
calendar.get(Calendar.MONTH));
        System.out.println("Dia do mês: " +
calendar.get(Calendar.DAY_OF_MONTH));
    }
```

Conversão, Formatação e enum
Exercícios

Classe Date

1. Crie um programa que obtenha a data e hora atual e exiba em um formato legível, por exemplo, "Data e Hora atuais: 20 de maio de 2024, 14:35:10".

Classe Calendar

1. Crie um programa que peça ao usuário sua data de nascimento (dia, mês e ano separadamente) e, em seguida, calcule e exiba a idade da pessoa.

Conversão, Formatação e enum
Classe DateFormat

SENAI

Conceito
A classe abstrata java.text.DateFormat fornece métodos para formatar e analisar datas e horas em diferentes formatos.

Formatando Datas
- `DateFormat formatoData = new SimpleDateFormat("dd/MM/yyyy");`

- `String dataFormatada = formatoData.format(dataAtual);`

Analisando Datas
```
Date data = formatoData.parse("26/10/2023");
```

Para importar a classe
```
import java.util.DateFormat;
```

Exemplo de código
```
import java.text.SimpleDateFormat;
import java.util.Date;

public class FormatacaoData {

    public static void main(String[] args)
{
        Date dataAtual = new Date();
        SimpleDateFormat formato = new
SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
        String dataFormatada =
formato.format(dataAtual);

        System.out.println("Data e hora
formatadas: " + dataFormatada);
    }
}
```

Conversão, Formatação e enum
Classe DateFormat

SENAI

Exemplo de código

```
import java.text.SimpleDateFormat;
import java.util.Date;

public class FormatacaoData {

    public static void main(String[] args) {
        Date dataAtual = new Date();
        SimpleDateFormat formato = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
        String dataFormatada = formato.format(dataAtual);

        System.out.println("Data e hora formatadas: " + dataFormatada);
    }
}
```

Conversão, Formatação e enum
Exercícios

Classe DateFormat

1. Crie um programa que receba uma data e hora no formato "dd/MM/yyyy HH:mm:ss" e converta para o formato "mm,d, yyyy h:mm".

Conversão, Formatação e enum
Classe NumberFormat

Conceito
A classe abstrata java.text.NumberFormat fornece métodos para formatar e analisar números em diferentes formatos, incluindo moeda, porcentagem e números decimais.

Obtenção da instância

```
NumberFormat formatoMoeda =
NumberFormat.getCurrencyInstance();

NumberFormat formatoPorcentagem =
NumberFormat.getPercentInstance();

NumberFormat formatoDecimal =
NumberFormat.getNumberInstance();
```

Formatando números

```
String valorFormatado =
formatoMoeda.format(1234.56);
// Retorna "R$ 1.234,56" (dependendo da localização)
```

Para importar a classe

```
import java.text.NumberFormat;
```

Conversão, Formatação e enum
Classe NumberFormat

SENAI

Exemplo de código - Formatando números

```
import java.text.NumberFormat;
import java.util.Locale;

public class FormatandoMoeda {
    public static void main(String[] args) {
        double valor = 1234.56;

        NumberFormat formatoMoedaBrasil = NumberFormat.getCurrencyInstance(new
Locale("pt", "BR"));
        NumberFormat formatoMoedaEUA = NumberFormat.getCurrencyInstance(Locale.US);

        System.out.println("Valor (Brasil): " + formatoMoedaBrasil.format(valor));
        System.out.println("Valor (EUA): " + formatoMoedaEUA.format(valor));
    }
}
```

Conversão, Formatação e enum
Exercícios

SENAI

Classe NumberFormat

1. Crie um programa que solicite ao usuário a inserção de um valor em reais. Converta esse valor para ienes e euros, de acordo com a cotação atual. Utilizando a classe NumberFormat, exiba os valores convertidos de acordo com as moedas locais.

Conversão, Formatação e enum
Classe Formatter

Conceito
A classe Formatter é usada para gerar strings formatadas usando especificações de formato.

Definição do formato
`%[arg_index$][flags][width][.precision]conversion`

%: Sinalizador de formato.
arg_index: Índice do argumento (opcional).
flags: Opções (ex: - para alinhar à esquerda).
width: Largura mínima.
.precision: Casas decimais.
conversion: Tipo de conversão (ex: d para inteiro, f para float).

Para importar a classe

```
import java.util.Formatter;
```

Link para documentação:
https://docs.oracle.com/javase/8/docs/api/java/util/Formatter.html#skip.navbar.top

Conversão, Formatação e enum
Classe Formatter

Exemplo de código

```
import java.util.Formatter;

public class FormatacaoAvancada {
    public static void main(String[] args) {
        int idade = 30;
        double saldo = 1234.5678;

        // Formatação de string usando Formatter
        String saida = String.format("Nome: %-10s Idade: %2d Saldo: R$
%7.2f", "João", idade, saldo);

        System.out.println(saida);
    }
}
```

Conversão, Formatação e enum
Enumerações (enum)

Conceito

Enumerações (enum) são um tipo especial de classe que representa um grupo de constantes (variáveis imutáveis).

Enums (enumerações) são como conjuntos especiais de constantes nomeadas. Em vez de números espalhados pelo código, você usa nomes descritivos, tornando tudo mais claro.

Sintaxe

```
public enum DiaDaSemana {
    DOMINGO, SEGUNDA, TERCA, QUARTA, QUINTA,
SEXTA, SABADO
}
```

Exemplo de código

```
public class EnumExample {
    public enum DiaDaSemana {
        SEGUNDA, TERCA, QUARTA, QUINTA,
SEXTA, SABADO, DOMINGO
    }

    public static void main(String[] args)
{
        DiaDaSemana hoje =
DiaDaSemana.SEGUNDA;
        System.out.println("Hoje é: " +
hoje);

        for (DiaDaSemana dia :
DiaDaSemana.values()) {
            System.out.println(dia);
        }   }    }
```

Conversão, Formatação e enum
Enumerações (enum)

SENAI

Exemplo de código

```
public class UsandoEnums {
    public static void main(String[] args) {
        DiaDaSemana hoje = DiaDaSemana.QUINTA;

        if (hoje == DiaDaSemana.SABADO || hoje ==
DiaDaSemana.DOMINGO) {
            System.out.println("Dia de C#");
        } else {
            System.out.println("Dia de Java!");
        }
    }
}
```

Conversão, Formatação e enum
Enumerações (enum)

SENAI

Exemplo de código

```
public class Pedido {
    enum StatusPedido {
        RECEBIDO, PROCESSANDO, ENVIADO, ENTREGUE, CANCELADO;

        public boolean estaFinalizado() {
            return this == ENTREGUE || this == CANCELADO;
        }
    }

    StatusPedido status;

    // ... outros métodos do pedido
}
```

Conversão, Formatação e enum
Exercícios

Classe NumberFormat

1. Crie um programa que receba uma string representando um número com vírgulas e pontos e converta esse valor para um número decimal.

Classe Formatter

1. Crie um programa que formate e exiba os detalhes de um produto (nome, preço, quantidade em estoque) em uma tabela de texto.

Enumerações

1. Defina uma enumeração para os dias da semana e crie um programa que receba um número (1-7) e exiba o dia correspondente.