

	UNIVERSIDADE ESTADUAL DA PARAÍBA CENTRO DE CIÊNCIAS E TECNOLOGIA CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO	
ATIVIDADE DE ESTUDO V ^{ANO} 2021.1		
CURSO	Ciência da Computação	
DISCIPLINA	Linguagem de Programação II	
PROFESSOR TITULAÇÃO		
Jucelio Soares dos Santos Mestrado		
NOME	Lucas de Lucena Siqueira	
MATRÍCULA	201080354 CONCEITO	
DATA		

DESCRIÇÃO DA ATIVIDADE

1. Defina uma estrutura de dados para agrupar e representar os dados de um automóvel (marca do fabricante, modelo, ano de fabricação e quantidade máxima de passageiros).

```

1 package Lista05;
2
3 /*
4  * Defina uma estrutura de dados para agrupar e representar os dados de um automóvel
5  * (marca do fabricante, modelo, ano de fabricação e quantidade máxima de passageiros).*/
6  */
7 public class Carro {
8     private String fabricante;
9     private String modelo;
10    private int anoFabricacao;
11    private int maxPassageiros;
12 }

```

Obs.: Deixamos os atributos privados pois criamos os getters e setters logo a seguir também.

2. Considerando que toda pessoa possua um carro, que alteração no registro "Pessoa" mostrado na aula nós precisamos fazer para representar esse relacionamento?

```

1 package lista05;
2
3 public class Pessoa {
4     private String nome;
5     private String telefone;
6     private Endereco enderecoResidencia;
7     private Carro carroPessoal;
8 }

```

3. Escreva um programa que armazene os dados relacionados a um filme (nome, autor, ano, preço) em uma estrutura de dados do tipo registro. Crie rotinas para ler e para imprimir os dados desse tipo de registro. Crie uma rotina main para ler e imprimir os dados de 3 diferentes filmes.

```

1 package lista05;
2
3 import java.util.Scanner;
4
5 /**
6  * Escreva um programa que armazene os dados relacionados a um filme (nome, autor, ano, preço)
7  * e em uma estrutura de dados do tipo registro. Crie rotinas para ler e para imprimir os dados desse tipo de registro.
8  * Crie uma rotina main para ler e imprimir os dados de 3 diferentes filmes.*/
9 public class Filme {
10     private String nome;
11     private String autor;
12     private int ano;
13     private double preco;
14
15     public Filme(String nome, String autor, int ano, double preco) {
16         this.nome = nome;
17         this.autor = autor;
18         this.ano = ano;
19         this.preco = preco;
20     }
21
22     public String getNome() {
23         return nome;
24     }
25
26     public void setNome(String nome) {
27         this.nome = nome;
28     }
29
30     public String getAutor() {
31         return autor;
32     }
33
34     public void setAutor(String autor) {
35         this.autor = autor;
36     }
37
38     public int getAno() {
39         return ano;
40     }

```

```

42     public void setAno(int ano) {
43         this.ano = ano;
44     }
45
46     public double getPreço() {
47         return preco;
48     }
49
50     public void setPreço(double preco) {
51         this.preco = preco;
52     }
53 }
54

```

4. Considere que um programa precisa manipular os meses (janeiro a dezembro). Defina uma enumeração para representar esses valores.

```

1 package Lista05;
2
3 public enum DiasSemana {
4     Domingo, SegundaFeira, TercaFeira, QuartaFeira, QuintaFeira, SextaFeira, Sabado, Invalido;
5 }

```

```

1 package Lista05;
2 import java.util.*;
3 public class Main {
4     public static void main (String []args) {
5         Scanner entrada = new Scanner(System.in);
6         System.out.print("Digite um dia da semana com o seu respectivo numeral (1 a 7): ");
7         DiasSemana diasSemana = getDiaSemana(entrada.nextInt());
8         System.out.println("Dia da semana correspondente: " + transformarDiaSemana(diasSemana));
9     }
10
11     public static DiasSemana getDiaSemana(int diaSemanaF) {
12         return switch (diaSemanaF) {
13             case 1 -> DiasSemana.Domingo;
14             case 2 -> DiasSemana.SegundaFeira;
15             case 3 -> DiasSemana.TercaFeira;
16             case 4 -> DiasSemana.QuartaFeira;
17             case 5 -> DiasSemana.QuintaFeira;
18             case 6 -> DiasSemana.SextaFeira;
19             case 7 -> DiasSemana.Sabado;
20             default -> DiasSemana.Invalido;
21         };
22     }
23
24     public static String transformarDiaSemana(DiasSemana diasSemana) {
25         return switch (diasSemana) {
26             case Domingo -> "Domingo";
27             case SegundaFeira -> "Segunda-Feira";
28             case TercaFeira -> "Terça-Feira";
29             case QuartaFeira -> "Quarta-Feira";
30             case QuintaFeira -> "Quinta-Feira";
31             case SextaFeira -> "Sexta-Feira";
32             case Sabado -> "Sábado";
33             case Invalido -> "Número inválido";
34         };
35     }
36 }
37

```

5. Considerando a atividade anterior, crie agora funções para: ler valores dessa enumeração (meses do ano) a partir do teclado; escrever esses valores na tela.

```

1 package lista05;
2
3 public enum Meses {
4     Janeiro, Fevereiro, Abril, Maio, Marco, Junho, Julho, Agosto, Setembro, Outubro, Novembro, Dezembro;
5 }
6

```

```

1 package lista05;
2 import java.util.*;
3 public class Main {
4     public static void main(String []args) {
5         Scanner entrada = new Scanner(System.in);
6         Meses mes = setMeses();
7         System.out.println("Dia da semana correspondente: " + getMes(mes));
8     }
9     public static Meses setMeses() {
10         Scanner entrada = new Scanner(System.in);
11         int numeroMes;
12         Meses mes = null;
13         do {
14             System.out.print("Informe o numeral referente ao mês (1 a 12): ");
15             numeroMes = entrada.nextInt();
16             if (numeroMes > 12 || numeroMes < 1) {
17                 System.out.println("Número inválido!");
18             }
19         } while (numeroMes < 1 || numeroMes > 12);
20
21         switch (numeroMes) {
22             case 1 -> mes = Meses.Janeiro;
23             case 2 -> mes = Meses.Fevereiro;
24             case 3 -> mes = Meses.Marco;
25             case 4 -> mes = Meses.Abril;
26             case 5 -> mes = Meses.Maio;
27             case 6 -> mes = Meses.Junho;
28             case 7 -> mes = Meses.Julho;
29             case 8 -> mes = Meses.Agosto;
30             case 9 -> mes = Meses.Setembro;
31             case 10 -> mes = Meses.Outubro;
32             case 11 -> mes = Meses.Novembro;
33             case 12 -> mes = Meses.Dezembro;
34         }
35         return mes;
36     }
37     public static String getMes (Meses mes) {
38         return switch (mes) {
39             case Janeiro -> "Janeiro";
40

```

```

41             case Fevereiro -> "Fevereiro";
42             case Marco -> "Marco";
43             case Abril -> "Abril";
44             case Maio -> "Maio";
45             case Junho -> "Junho";
46             case Julho -> "Julho";
47             case Agosto -> "Agosto";
48             case Setembro -> "Setembro";
49             case Outubro -> "Outubro";
50             case Novembro -> "Novembro";
51             case Dezembro -> "Dezembro";
52         };
53     }
54 }

```

6. Explique o que são estruturas de dados do tipo registro, indicando em quais situações elas são úteis.

R/ Uma estrutura do tipo registro é composta por um conjunto de variáveis de diversos tipos, que terão uma relação lógica entre elas. Podem ser úteis para criar estruturas de dados que armazenam as informações de um filme por exemplo, que pode conter um nome (String), ano de lançamento (Inteiro) e preço (Double).

7. Explique o que são enumerações, indicando em quais situações elas são úteis na criação de programas de computador.

R/ As estruturas do tipo enum são utilizados na linguagem Java para a criação de estruturas de dados organizados, podendo agrupar valores que tenham o mesmo sentido para dentro dessa estrutura. Podem ser úteis para fazer conversões entre os meses do ano, por exemplo, que poderão ser informados pelo usuário a partir do seu respectivo numeral (1 - 12) e o programa fará a conversão para o seu respectivo nome por extenso.

8. Indique:

- a. Quais são os possíveis benefícios de utilizarmos registros.

R/ A construção de uma estrutura do tipo registro pode facilitar o preenchimento de um novo usuário, por exemplo, que através da referência de uma única variável do dia registro pode ser feito. Além de que também se torna possível enviar tais estruturas como parâmetros para funções.

- b. Quais são os possíveis benefícios de utilizarmos enumerações.

R/ Confere uma melhor semântica além de deixar o código mais legível e auxiliado pelo compilador.