

LÓGICA DE PROGRAMAÇÃO

“

Registros são estruturas de dados heterogêneas. Vetores e matrizes são estruturas de dados homogêneas, ou seja, armazenam somente um tipo de dados. Registros, por sua vez, podem armazenar um ou mais tipos de dados, inclusive outros registros.

REGISTROS

- ⦿ Existem vários tipos de estruturas de dados;
- ⦿ A princípio, quando se estuda matrizes e vetores, tem-se como objeto variáveis unidimensionais e multidimensionais homogêneas, ou seja, são estruturas preparadas para receber valores de somente um tipo de dados, no momento em que foram declaradas;
- ⦿ As variáveis possuem a função, além de armazenarem tipos de dados, de representar entidades identificadas no problema que será resolvido pelo programa que está sendo criado

REGISTROS

- Surge então a necessidade de armazenar, em uma mesma variável, diferentes tipos de dados. Daí surgem os **registros**;
- Graças aos registros é possível mapear com maior precisão o que se chama de **domínio** dentro de uma solução de software, visto que agora conceitos como cliente, cidade e fornecedor passam a possuir uma estrutura de dados que os representa adequadamente
- Compreender este conceito é o primeiro passo para iniciar a compreensão da estrutura de um banco de dados, por exemplo.

REGISTROS

“

Registros são estruturas que conseguem **agregar vários tipos de dados**, não se limitando a utilizar os tipos de dados primitivos fornecidos pelas linguagens de programação. Cada dado contido em um registro é chamado de **campo**. Os campos podem ser de diferentes tipos primitivos, sendo, por esta razão, os registros serem conhecidos como **variáveis compostas heterogêneas** (ASCENCIO; CAMPOS, 2012)

REGISTROS

- Com base no que foi apresentado, é possível por exemplo entender uma ficha financeira com os seguintes componentes: nome do cliente, telefone, saldo devedor, valor movimentado, maior compra, cidade e cep;
- Sendo assim, é correto afirmar que a ficha financeira é um **registro**, e cada atributo já citado possui o tipo correspondente, como decimal, literal e assim sucessivamente;
- O uso de vetores ou matrizes já deixa de ser indicado, visto que essas estruturas de dados armazenam dados **de mesmo tipo** apenas;

REGISTROS E VETORES

- ⦿ Registros e vetores podem se compostos a fim de resolver problemas mais complexos;
- ⦿ Considerando que um vetor é um conjunto de elementos do mesmo tipo, é natural que, quando precisarmos agrupar vários registros (por exemplo, várias Fichas Financeiras), utilizaremos vetores (conjuntos) de registros;
- ⦿ Eventualmente, pode haver necessidade de que um registro tenha como um dos seus campos um vetor (como o de campos para data de recebimento);

REGISTROS E VETORES

- Os vetores de registros visam a armazenar conjuntos de elementos complexos. Ao invés de armazenar apenas as notas dos alunos de uma turma, um vetor de registro pode armazenar as informações cadastrais de todos os alunos da turma;
- Ou seja, as estruturas de dados homogêneas e heterogêneas não se anulam, muito pelo contrário, se complementam para que dessa forma seja possível construir programas cada vez mais complexos;
- A API de coleções da linguagem Java é um belo exemplo de como os conceitos podem ajudar e muito no dia a dia do programador;

REGISTROS

```
DECLARE nome_registro REGISTRO (nome_campo1 TIPO_CAMPO1,  
nome_campo2 TIPO_CAMPO2..., nome_campoN TIPO_CAMPOn )
```

```
DECLARE conta REGISTRO (num, saldo NUMERICO, nome CARACTERE)
```

Variável conta

num	
saldo	
nome	

```
DECLARE conta[3] REGISTRO (num, saldo NUMERICO, nome CARACTERE)
```

Variável conta

num	
saldo	
nome	

Variável conta

num	
saldo	
nome	

Variável conta

num	
saldo	
nome	

REGISTROS

1	DECLARE conta REGISTRO (numero, saldo NUMERICO, nome
2	CARACTERE)
3	ESCREVA "Entre com o número da conta"
4	LEIA conta.numero
5	ESCREVA "Entre com o saldo"
6	LEIA conta.saldo
7	ESCREVA "Entre com o nome do titular da conta"
8	LEIA conta.nome

1	DECLARE conta[3] REGISTRO (numero, saldo, i NUMERICO, nome
2	CARACTERE)
3	PARA i de 1 até 3 FAÇA
4	ESCREVA "Entre com o número da conta", i, ":"
5	LEIA conta[i].numero
6	ESCREVA "Entre com o saldo"
7	LEIA conta[i].saldo
8	ESCREVA "Entre com o nome do titular da conta"
9	LEIA conta[i].nome
10	FIM

REGISTROS EM JAVA

- ⦿ A criação de registros em Java pode ser simulada, sendo assim o que será apresentado é uma técnica de declaração de forma muito simular a forma construída em pseudocódigo;
- ⦿ Isso ocorrerá dessa forma para que os conceitos de OO não se misturem ao conceito de registros, sendo que é por meio dessa separação, que mais adiante no curso será compreendida melhor a diferença entre classes x objetos x registros;

REGISTROS EM JAVA

- ⦿ A ordem de declaração de um registro em Java seguirá a mesma sequência que proposta em pseudocódigo, porém se a ordem for alterada, por uma questão da linguagem, o código será compilado sem problemas;
- ⦿ Vale lembrar que a forma como será apresentada a construção desses registros em Java, incluindo a sequência de criação do mesmo será observada em trabalhos e exercícios, buscando assim a aderência do aluno ao que está sendo proposto;

REGISTROS EM JAVA

```
1 import javax.swing.JOptionPane;
2
3 public class GerenciadorDeRegistros {
4
5     static class Conta {
6         String numero;
7         double saldo;
8     }
9
10    public static void main(String[] args) {
11
12        Conta conta = new Conta();
13        conta.numero = "1030-5";
14        conta.saldo = 1560.90;
15
16        JOptionPane.showMessageDialog(null, "Número: "
17            + conta.numero + " Saldo: R$" + conta.saldo);
18    }
19 }
20
21 }
```

REGISTROS EM JAVA

- ⦿ A definição (declaração) do registro segue o padrão:

- ⦿ `static class` <nome do registro> {

- <tipo> <nome do campo 1>

- <tipo> <nome do campo 1>

- <tipo> <nome do campo N>

- }

REGISTROS EM JAVA

- ⦿ Uma vez declarado, é preciso inicializar ele da mesma forma como ocorre com vetores e matrizes:
 - ⦿ `<nome do registro> <nome da variavel> = new <nome do registro>()`
- ⦿ Para acessar ou alterar o valor dos campos do registro (numero e conta) usa-se o operador **ponto (.)**:
 - ⦿ `<nome da variavel>.<campo1>`
 - ⦿ `<nome da variavel>.<campo1> = <valor>`

REGISTROS EM JAVA

```
1 import javax.swing.JOptionPane;
2
3 public class GerenciadorDeRegistros {
4     static Conta[] contas;
5     static Conta novaConta;
6     static int i;
7     static class Conta {
8         String numero;
9         double saldo;
10    }
11    public static void main(String[] args) {
12        contas = new Conta[10];
13        for (int i = 0; i < contas.length; i++) {
14            novaConta = new Conta();
15            novaConta.numero = JOptionPane.showInputDialog(
16                "Digite o número da conta");
17            novaConta.saldo = Double.parseDouble(
18                JOptionPane.showInputDialog("Digite o saldo: "));
19            contas[i] = novaConta;
20        }
21        for (int i = 0; i < contas.length; i++) {
22            JOptionPane.showMessageDialog(null,
23                "Número: " + contas[i].numero +
24                " Saldo: R$" + contas[i].saldo);
25        }
26    }
27 }
```


REGISTROS EM JAVA

- ⦿ Ao se trabalhar com vetores de registros, todas as regras já vistas em outros tópicos estão valendo exatamente do mesmo jeito. O mesmo vale para matrizes;
- ⦿ É possível declarar ainda, variáveis globais do registro definido, confoem a linha 5 do último algoritmo apresentado;
- ⦿ Nesse caso, por meio do uso de matrizes é possível reproduzir, mesmo que de forma rudimentar, o comportamento de um banco de dados, o que abre muito o leque de situações de uso e codificação;

PSEUDOCÓDIGO X JAVA

1	DECLARE conta REGISTRO (numero, saldo NUMERICO, nome
2	CARACTERE)
3	ESCREVA "Entre com o número da conta"
4	LEIA conta.numero
5	ESCREVA "Entre com o saldo"
6	LEIA conta.saldo
7	ESCREVA "Entre com o nome do titular da conta"
8	LEIA conta.nome

```
1 import javax.swing.JOptionPane;
2
3 public class GerenciadorDeRegistros {
4
5     static class Conta {
6         String numero;
7         double saldo;
8     }
9
10    public static void main(String[] args) {
11
12        Conta conta = new Conta();
13        conta.numero = "1030-5";
14        conta.saldo = 1560.90;
15
16        JOptionPane.showMessageDialog(null, "Número: "
17            + conta.numero + " Saldo: R$" + conta.saldo);
18
19    }
20
21 }
```

PSEUDOCÓDIGO X JAVA

1	DECLARE conta[3] REGISTRO (numero, saldo, i NUMERICO, nome
2	CARACTERE)
3	PARA i de 1 até 3 FAÇA
4	ESCREVA "Entre com o número da conta", i, ":"
5	LEIA conta[i].numero
6	ESCREVA "Entre com o saldo"
7	LEIA conta[i].saldo
8	ESCREVA "Entre com o nome do titular da conta"
9	LEIA conta[i].nome
10	FIM

```
1 import javax.swing.JOptionPane;
2
3 public class GerenciadorDeRegistros {
4     static Conta[] contas;
5     static Conta novaConta;
6     static int i;
7     static class Conta {
8         String numero;
9         double saldo;
10    }
11    public static void main(String[] args) {
12        contas = new Conta[10];
13        for (int i = 0; i < contas.length; i++) {
14            novaConta = new Conta();
15            novaConta.numero = JOptionPane.showInputDialog(
16                "Digite o número da conta");
17            novaConta.saldo = Double.parseDouble(
18                JOptionPane.showInputDialog("Digite o saldo: "));
19            contas[i] = novaConta;
20        }
21        for (int i = 0; i < contas.length; i++) {
22            JOptionPane.showMessageDialog(null,
23                "Número: " + contas[i].saldo
24                + " Saldo: R$" + contas[i].saldo);
25        }
26    }
27 }
```

MÃO NA MASSA

Caso 1

Construa um algoritmo que armazene em um **registro** os dados de um diário de classe, sendo eles: disciplina, data de início e término, qtde de alunos matriculados. Ao final mostre os dados do diário criado.

Caso 2

Desenvolva um programa que armazene em um **registro** as provas de um aluno, sendo eles: data de realização e nota. Crie também outro registro, só que armazenando os dados do aluno, sendo eles: nome, idade, primeira prova, segunda prova e terceira prova. Ao final apresente a média simples do aluno.

Uma **função** deve ser utilizada para o cálculo da média.

MÃO NA MASSA

Caso 3

Dados os seguintes campos de um **registro**: nome, telefone, dia de aniversário e mês de aniversário, desenvolver um programa que mostre em um dado mês do ano, quem são as pessoas que fazem aniversário, exibir também o dia. Considere um conjunto de 50 pessoas.

Caso 4

Uma pessoa cadastrou um conjunto de 15 registros contendo o nome da loja, telefone e preço de um eletrodoméstico. Desenvolver um algoritmo que permita exibir qual foi a média dos preços cadastrados e uma relação contendo o nome e o telefone das lojas cujo preço estava abaixo da média.