Programação de Computadores - I

Profª Beatriz

Profº Israel

Arrays

Arrays - Principais Características

- Arrays são objetos que armazenam diversas variáveis do mesmo tipo.
- Podem conter variáveis de referência primitivas ou de objeto.

Array - Sintaxe

- O nome de um array segue os mesmos padrões das demais variáveis em Java seguido de um par de colchetes [].
- Declarando um Array:
  - String vetNome[] = new String[5];
  - Cliente vetCliente[] = new Cliente[12];
  - int idades[] = { 25,39,45,58};
  - String casais[][]=new String[3] [2];

Array – Atribuição de Valores

- vetNome[0]=“Paula”;
- vetCliente[8]=cli;
- Idades[] = {1,5,8,12};
- casais[1][1]=“Maria”;
- casais[][]={{“José”,”Maria”},{“Barth”,”Indira”},{“Aquira”,”Paola”}};
- vetCliente[1].setNome(“João”);

Array – Leitura de Valores

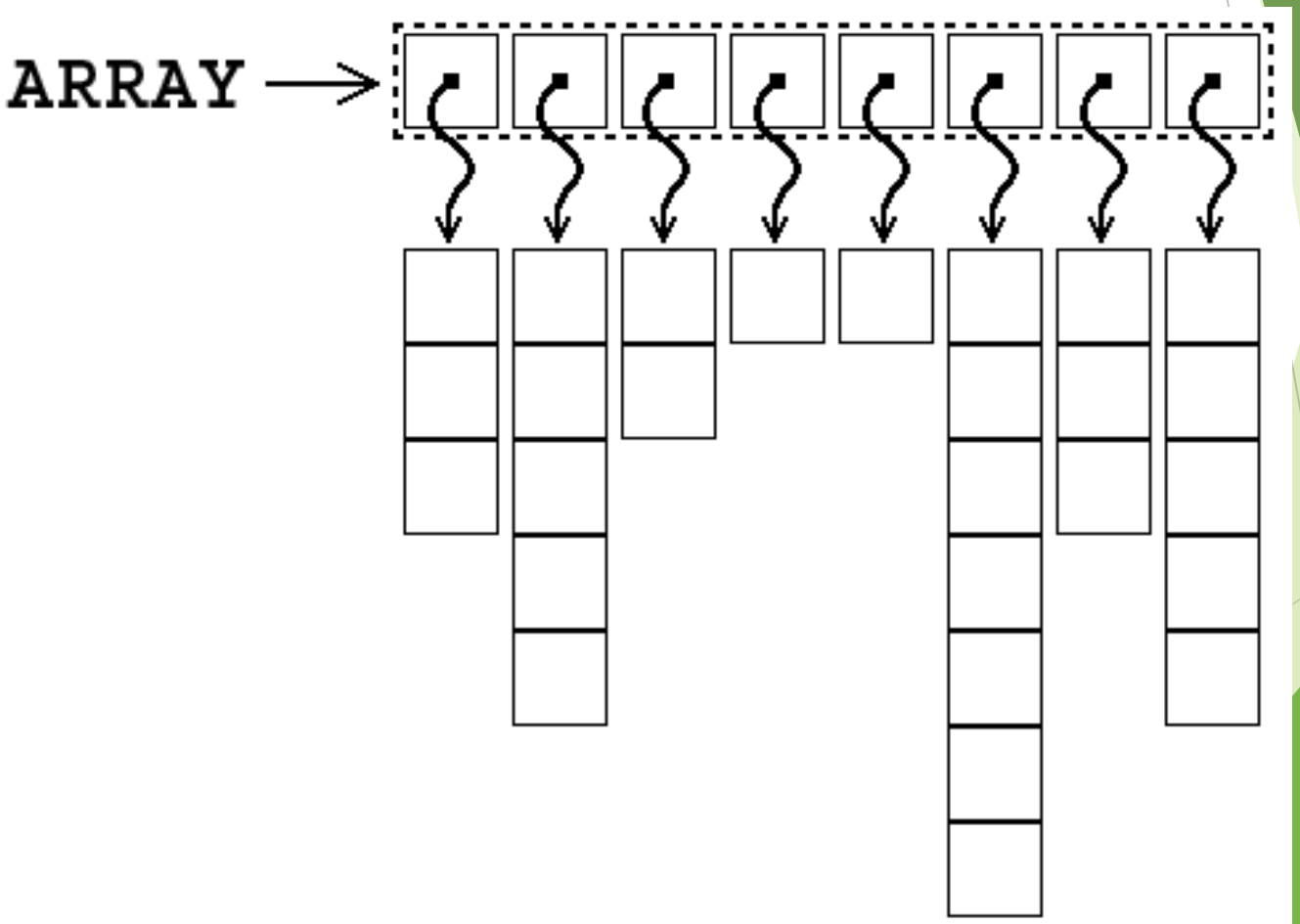
- System.out.println(vetCliente[1].getNome());

Array - Percorrimento

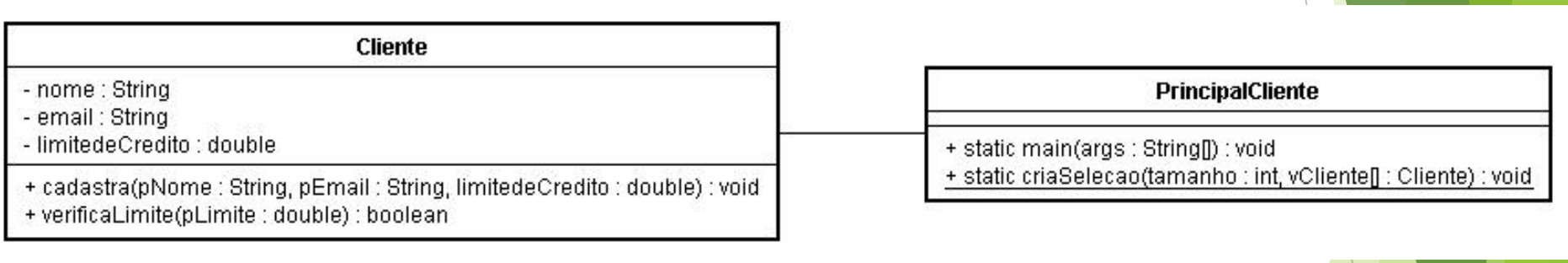
```
for(int i=0;i<9;i++){
    System.out.println(vetNome[i]);
}

for(int i=0; i <= vetNome.length - 1; i++) {
    System.out.println(vetNome[i]);
}
```

Arrays multidimensionais (Matrizes)

- Matrizes em java são, na verdade, arrays de arrays



Analise o código que implementa o problema a seguir:



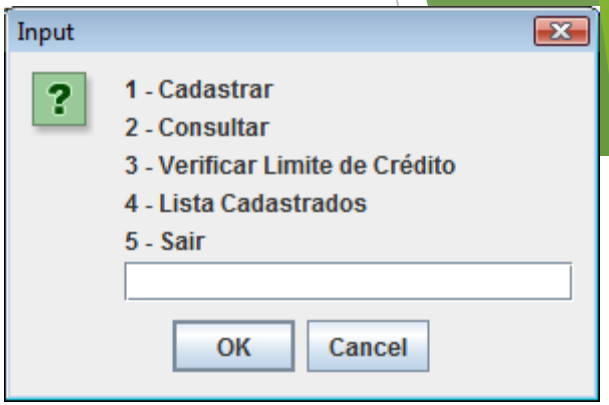
Classe Cliente

```
public class Cliente {
    private String nome;
    private String email;
    private double limitedeCredito;
    public Cliente(){
        this("","",0.0);
    }
    public Cliente(String nome, String email, double limitedeCredito) {
        this.nome = nome;
        this.email = email;
        this.limitedeCredito = limitedeCredito;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public double getLimitedeCredito() {
        return limitedeCredito;
    }
    public void setLimitedeCredito(double limitedeCredito) {
        this.limitedeCredito = limitedeCredito;
    }
    public void cadastrarCliente(String pNome,String pEmail,double pLimitedeCredito)
    {
        this.setNome(pNome);
        this.setEmail(pEmail);
        this.setLimitedeCredito(pLimitedeCredito);
    }
```

Classe Cliente

```
public boolean verificaLimite(double pLimite){
    boolean retorno=false;
    if (this.getLimitedeCredito() >= pLimite)
    {
        retorno=true;
    }
    return retorno;
}
```

Classe PrincipalCliente




```
import javax.swing.JOptionPane;
public class PrincipalCliente {
    public static void main(String[] args)
    {
        //Inicialização das variáveis
        Cliente vetCliente[] = new Cliente[10];
        int op=0,indice=0,vIndiceCliente=0;
        String vNome="",vEmail="";
        double vLimitedeCredito=0.0;
        do {
            //Instancia um objeto da Classe Cliente
            Cliente cli = new Cliente();
            //Exibe o menu de opções e aguarda a digitacao da opção escolhida
            op = Integer.parseInt(JOptionPane.showInputDialog("1 - Cadastrar\n2 - Consultar \n3 - Verificar Limite d
            switch(op){
```

Classe PrincipalCliente

- Implementação do cadastro de Cliente

```
case 1:
    // Lê o nome do novo cliente
    vNome=JOptionPane.showInputDialog("Digite o nome do Cliente");
    //Lê o e-mail do novo cliente
    vEmail=JOptionPane.showInputDialog("Digite o e-mail do Cliente");
    //Lê o limite de Crédito do novo Cliente
    vLimitedeCredito=Double.parseDouble(JOptionPane.showInputDialog("Digite o Limitem de Credito do
    //Chama o método cadastraCliente
    cli.cadastrarCliente(vNome, vEmail, vLimitedeCredito);
    //Coloca o objeto no vetor de objetos
    vetCliente[indice]=cli;
    //Incrementa o índice de Clientes Cadastrados
    indice++;
    //Destroi o objeto cliente
    cli = null;
    break;
```

Classe PrincipalCliente

- Implementa a consulta de clientes:

```
case 2:
    //Le o nome procurado
    vNome=JOptionPane.showInputDialog("Digite o nome do Cliente procurado");
    //Inicializa o Flag que mostará se foi encontrado o cliente ou não.
    boolean fAchou=false;
    // Faz o loop para a busca do cliente
    for(int i = 0;i < indice; i++){
        // Comprara o nome digitado com o nome procurado
        if (vetCliente[i].getNome().equals(vNome)){
            //Se achou exibe os dados do cliente
            JOptionPane.showMessageDialog(null, vetCliente[i].getNome() + " - "+ vetCliente[i].getEn
            //Altera o valor do flag para true
            fAchou=true;
        }
    }
    // Testa se não achou o cliente
    if (fAchou == false){
        //Se não achou exibe mensagem
        JOptionPane.showMessageDialog(null, "Cliente não Cadastrado.");
    }
    break;
```

Classe PrincipalCliente

- Implementa a consulta de limite de crédito.

```
case 3:
    vIndiceCliente = Integer.parseInt(JOptionPane.showInputDialog(null, criaRelacao(indice,vetClient
    vLimitedeCredito=Double.parseDouble(JOptionPane.showInputDialog("Digite o valor da Compra do Cli
    if (vetCliente[vIndiceCliente].verificaLimite(vLimitedeCredito)) {
        JOptionPane.showMessageDialog(null, "Valor Liberado para o Cliente");
    }
    else{
        JOptionPane.showMessageDialog(null, "Valor Excede o Limite de Crédito do Cliente");
    }
    break;
```

Classe PrincipalCliente

- Implementa a Listagem de clientes.

```
            case 4:
                ;
                JOptionPane.showMessageDialog(null, criaRelacao(indice,vetCliente)+"\n");
                break;
            case 5:
                break;
            default:
                JOptionPane.showMessageDialog(null,"Digite uma opção válida");
                break;
        }
    }while(op != 5);
}
public static String criaRelacao(int tamanho,Cliente[] vCliente){
    String relacao="Relação de Clientes Cadastrados\n";
    for(int i = 0;i < tamanho; i++){
        relacao = relacao +"\n"+ String.valueOf(i)+" - " + vCliente[i].getNome() + " - " +
    }
    return relacao;
}
```