

Atividade de Laboratório III

Programação Orientada à Objeto

Professor Me. Eng. Gerson Neto.

Java SE.

Programas utilizados:

- IDE Eclipse.
- Plataforma Java SE.

1. Desenvolva 3 aplicativos que funcionem como uma agenda eletrônica. Siga como referência os códigos das Figuras 1, 2, 3 e 4. O objetivo é armazenar informações de contatos como: Nome, Email e Telefone. O primeiro aplicativo deve conter a persistência das informações armazenada em objetos do tipo Array, o segundo deve ter a persistência armazenada em objetos do tipo Vecto<E> e o terceiro deve usar objetos do tipo ArrayList<E> para armazenar as informações. Todos os aplicativos devem permitir armazenar e excluir contatos.

```
public class Pessoa {
    private String nome;
    private String telefone;
    private String email;

    public Pessoa(String nome, String telefone, String email){
        this.nome = nome;
        this.telefone = telefone;
        this.email = email;
    }

    public Pessoa(){}

    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getTelefone() {
        return telefone;
    }
    public void setTelefone(String telefone) {
        this.telefone = telefone;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
}
```

Figura 1. Classe Pessoa.

```

public class AgendaTelefonica {
    private Pessoa[] pessoas;
    private Scanner scanner;

    public AgendaTelefonica(){
        pessoas = new Pessoa[5];
        scanner = new Scanner(System.in);
    }

    public void cadastrarPessoa(){
        for(int i = 0; i < pessoas.length; i++){
            if(pessoas[i] == null){
                System.out.println("\nInsira um nome para a pessoa: ");
                String nome = scanner.next();
                System.out.println("\nInsira um telefone para a pessoa: ");
                String telefone = scanner.next();
                System.out.println("\nInsira um email para a pessoa: ");
                String email = scanner.next();
                Pessoa nova = new Pessoa(nome,telefone,email);
                pessoas[i] = nova;
                System.out.println("\nNovo contato adicionado\n");
                break;
            }
        }
    }
}

```

Figura 2. Classe agenda telefônica

```

    public void imprimirOpcoes(){
        System.out.println("\nEscoha uma das opções: \n");
        System.out.println("1 - Listar contatos. \n");
        System.out.println("2 - Incluir contatos. \n");
        System.out.println("3 - Sair \n");
    }

    public void imprimirPessoas(){
        for(Pessoa pessoa:pessoas){
            if(pessoa != null){
                System.out.println("\nNome: "+pessoa.getNome()+"\nTelefone: "
                    +pessoa.getTelefone()+"\nEmail: "+pessoa.getEmail());
            }
        }
    }

    public Pessoa[] getPessoas() {
        return pessoas;
    }

    public void setPessoas(Pessoa[] pessoas) {
        this.pessoas = pessoas;
    }
}

```

Figura 3. Classe agenda telefônica continuação.

```

public class Aplicacao {
    private static Scanner scanner;
    public static void main(String args[]){
        scanner = new Scanner(System.in);
        AgendaTelefonica agenda = new AgendaTelefonica();
        int sair = 0;
        int opcao = 0;
        while(sair == 0){
            switch(opcao){
                case 0:{
                    agenda.imprimirOpcoes();
                    opcao = scanner.nextInt();
                    break;
                }
                case 1:{
                    agenda.imprimirPessoas();
                    opcao = 0;
                    break;
                }
                case 2:{
                    agenda.cadastrarPessoa();
                    opcao = 0;
                    break;
                }
                case 3:{
                    sair = 1;
                    System.out.println("\nAté mais!\n");
                    break;
                }
                default:{
                    agenda.imprimirOpcoes();
                    opcao = scanner.nextInt();
                    break;
                }
            }
        }
    }
}

```

Figura 4. Classe aplicacao.

2. Uma empresa que faz negócios por reembolso postal vende cinco produtos cujos preços de varejo são como segue: P1 R\$1,00 – P2 R\$2,98 – P3 R\$9,98 – P4 R\$4,49 e P5 R\$6,87. Escreva um aplicativo que leia uma série de pares de números como segue: código do produto, quantidade vendida. Seu programa deve determinar o valor total de produtos vendidos, discriminando a porcentagem vendida em dinheiro de cada produto.
3. Um estacionamento cobra uma taxa mínima de R\$2,00 para estacionar por até três horas. Um adicional de R\$0,50 por hora não necessariamente inteira é cobrado após as três primeiras horas. Escreva um aplicativo que calcule e exiba as taxas de estacionamento para um cliente que estacionou. O aplicativo deve receber o intervalo de tempo que o veículo ficou estacionado e exibir a cobrança para o cliente.
4. Escreva um método múltiplo() que determina um par de inteiros se o segundo é múltiplo do primeiro. O método deve retornar o valor booleano true se o segundo número for múltiplo do primeiro.
5. Escreva um aplicativo que utiliza um objeto do tipo Array de 1000 elementos para exibir os números primos entre 2 a 999.
6. Uma loja contratou para desenvolver um aplicativo para gerenciamento de relacionamento com clientes e mineração de dados. O objetivo do é armazenar informações sobre cada cliente e o total de gasto por cada cliente ao longo de 12 meses. Dessa forma escreva um aplicativo que armazene informações de clientes como: Nome, Email, Telefone e Endereço e de dinheiro gasto para cada um dos últimos 12 meses. O Aplicativo deve informar o montante gasto por todos os clientes para cada mês e para o total do período de 12 meses, além de informar qual foi o cliente que mais gastou dinheiro por mês e no período total de 12 meses. O aplicativo também deve permitir cadastro e exclusão de clientes e também a possibilidade de editar dados de clientes como telefone ou endereço.