

Atividade de Laboratório V

Programação Orientada à Objeto

Professor Me. Eng. Gerson Neto.

Java SE.

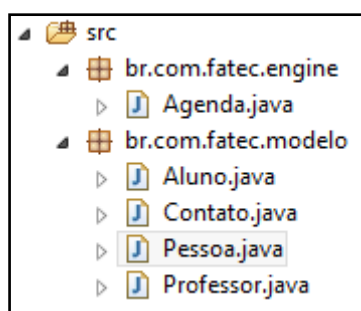
Programas utilizados:

- IDE Eclipse.
- Plataforma Java SE.

1. Desenvolva um aplicativo que funcione como uma agenda eletrônica para a faculdade. O objetivo é armazenar informações de contatos como: Nome, Email e Telefone. O aplicativo deve separar os contatos por conjuntos de acordo com sua função na faculdade entre professor e aluno. A cada vez que o aplicativo listar os contatos de um aluno, junto deve ser exibido o número de matrícula. A cada vez que o aplicativo listar os contatos de um professor junto deve ser exibido o número de registro do professor e a quantidade de horas/aulas que ele possui na Faculdade. O aplicativo deverá permitir inserção, edição e remoção de contatos.

O aplicativo deverá permitir a opção de salvar a agenda em disco e também carregar informações de uma agenda salva em disco, dessa forma criam-se as idéias de portabilidade e persistência das informações.

Utilize as Figuras de 1 à 8 como modelo para o seu desenvolvimento. Você deve utilizar a interface Contato e a classe abstrata Pessoa na construção de sua agenda.



**Figura 1. Estrutura de pacotes.**

```
public interface Contato {  
    public String dadosFormatados();  
}
```

Figura 2. Interface contato.

```
public abstract class Pessoa implements Contato{  
    private String nome;  
    private String telefone;  
    private String email;  
  
    @Override  
    public String dadosFormatados(){  
        String dadosFormatados;  
        dadosFormatados = "\nNome: "+this.getNome();  
        dadosFormatados = dadosFormatados+"\nTelefone: "+  
            this.getTelefone();  
        dadosFormatados = dadosFormatados+"\nEmail: "+  
            this.getEmail();  
        return dadosFormatados;  
    }  
    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;  
    }  
}
```

Figura 3. Classe pessoa.

```
    public String getEmail() {  
        return email;  
    }  
    public void setEmail(String email) {  
        this.email = email;  
    }  
}
```

Figura 4. Classe pessoa continuação.

```
public class Aluno extends Pessoa{
    private String numeroMatricula;
    @Override
    public String dadosFormatados() {
        String dadosFormatados = super.dadosFormatados();
        dadosFormatados = dadosFormatados+
            "\nMatricula: "+this.getNumeroMatricula();
        return dadosFormatados;
    }

    public String getNumeroMatricula() {
        return numeroMatricula;
    }
    public void setNumeroMatricula(String numeroMatricula) {
        this.numeroMatricula = numeroMatricula;
    }
}
```

Figura 5. Classe aluno.

```

public class Professor extends Pessoa{
    private String numeroRegistro;
    private String quantidadeHoraAula;

    @Override
    public String dadosFormatados() {
        String dadosFormatados = super.dadosFormatados();
        dadosFormatados = dadosFormatados+
            "\nRegistro: "+this.getNumeroRegistro();
        dadosFormatados = dadosFormatados+
            "\nHora/Aula: "+this.getQuantidadeHoraAula();
        return dadosFormatados();
    }

    public String getNumeroRegistro() {
        return numeroRegistro;
    }
    public void setNumeroRegistro(String numeroRegistro) {
        this.numeroRegistro = numeroRegistro;
    }
    public String getQuantidadeHoraAula() {
        return quantidadeHoraAula;
    }
    public void setQuantidadeHoraAula(String quantidadeHoraAula) {
        this.quantidadeHoraAula = quantidadeHoraAula;
    }
}

```

Figura 6. Classe professor.

```

public class Agenda {
    private List<Aluno> alunos;
    private List<Professor> professores;

    /*Crie os métodos para inserir objetos do tipo professor e aluno
    *Cada objeto deve ser inserido na sua respectiva lista.
    */

    public void listarTodos(){
        System.out.println("Lista de contatos de alunos: ");
        for(Aluno aluno:alunos){
            System.out.println("Nome: "+aluno.getNome());
            System.out.println("Telefone: "+aluno.getTelefone());
            System.out.println("Email: "+aluno.getEmail());
            System.out.println("Nº Matricula: "+aluno.getNumeroMatricula()+"\n");
        }
        System.out.println("Lista de contatos de professores: ");
        for(Professor professor:professores){
            System.out.println("Nome: "+professor.getNome());
            System.out.println("Telefone: "+professor.getTelefone());
            System.out.println("Email: "+professor.getEmail());
            System.out.println("Registro: "+professor.getNumeroRegistro());
            System.out.println("Quantidade Hora: "+professor.getNumeroRegistro()+"\n");
        }
    }
}

```

Figura 7. Classe agenda.

```

    public List<Aluno> getAlunos() {
        return alunos;
    }
    public void setAlunos(List<Aluno> alunos) {
        this.alunos = alunos;
    }
    public List<Professor> getProfessores() {
        return professores;
    }
    public void setProfessores(List<Professor> professores) {
        this.professores = professores;
    }
}

```

Figura 8. Classe agenda continuação.

2. Escreva um aplicativo que solicite ao usuário inserir o tamanho do lado de um quadrado e então exiba um quadrado vazio desse tamanho com asteriscos. Seu programa deve trabalhar com quadrados de todos os comprimentos de lado possíveis entre 1 e 20.
3. Escreva um aplicativo que leia três valores, diferentes de zero, inseridos pelo usuário e determine e imprima se eles poderiam representar os lados de um triângulo retângulo.