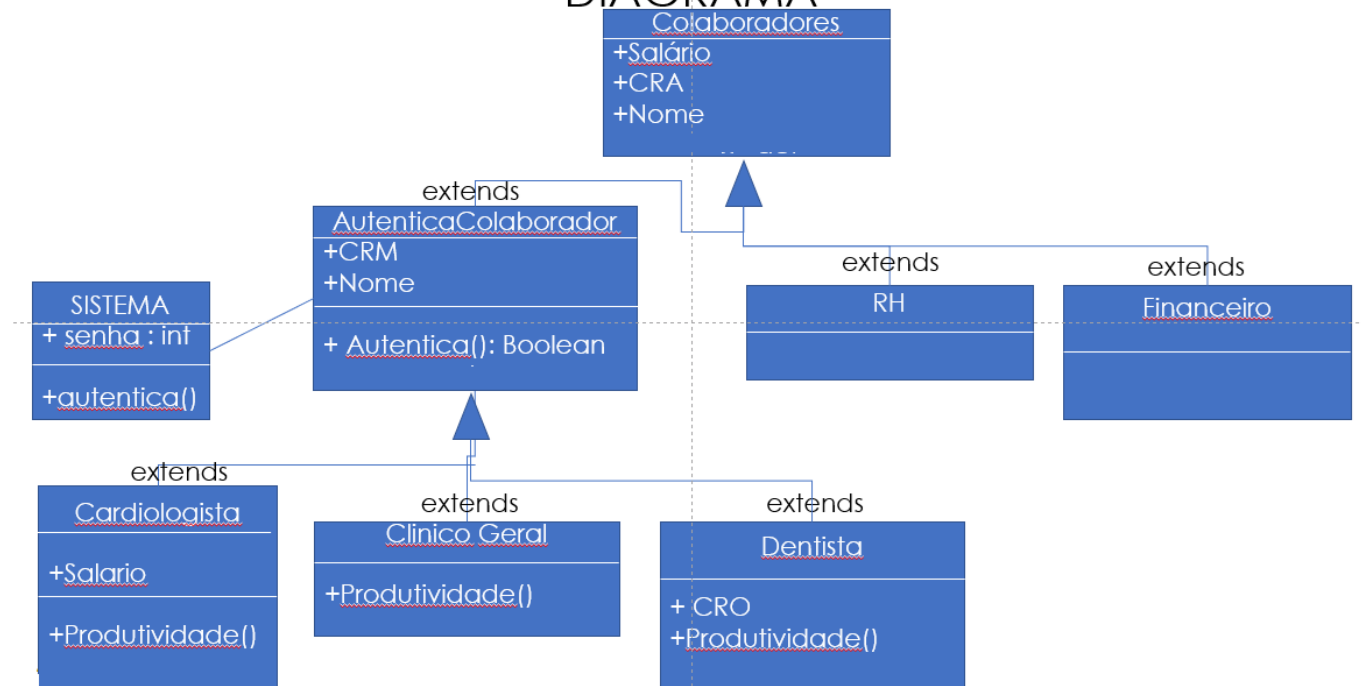


INFORMAÇÕES IMPORTANTES:

- 1ª) A avaliação somente poderá ser entregue depois de decorrido **50** minutos de seu início. A entrada após esse tempo só será permitida se nenhum aluno tiver saído da sala.
- 2ª) Não é permitido o uso de telefones celulares nem de equipamentos eletrônicos. O uso de calculadoras individuais deverá ser previamente autorizado pelo professor titular da disciplina,
- 3ª) Utilize caneta esferográfica azul ou preta. Provas entregues escritas a lápis **NÃO** serão corrigidas.
- 4ª) Será atribuída nota zero ao aluno que devolver sua prova em branco, independentemente de ter assinado ou não a Ata de Prova.
- 5ª) Ao aluno flagrado em **"FILA"** ou que facilite outro a **"FILAR"**, mesmo que já esteja do lado de fora da sala de aula, será atribuído nota zero e, mediante representação do professor, responderá a Procedimento Administrativo Disciplinar, com base no Código de Ética.
- 6ª) A revisão da prova será realizada APENAS na próxima aula. O aluno faltoso à revisão coletiva em sala poderá requerer a segunda revisão de prova, em até 72h, após o dia de entrega. No caso da Avaliação final, após a postagem da nota no Portal Acadêmico, em até 72h, o pedido de revisão dar-se-á mediante o pagamento de taxa no CRA. **Não há entrega e revisão coletiva de prova FINAL.**
- 7ª) Não é permitida consulta de qualquer natureza, EXCETO quando autorizada pelo professor titular da disciplina.
- 8ª) Não será permitido o empréstimo de qualquer material entre os alunos, mesmo daqueles que já tenham terminado as provas.
- 9ª) Não será aceito resposta em papel não autorizado pela Instituição de Ensino.
- 10ª) Não é permitido o uso de corretivo. Escreva com letra legível. No caso de erro, risque, com um traço simples, a palavra, a frase, o trecho ou o sinal gráfico e escreva o respectivo substitutivo. **Lembre-se:** parênteses não podem ser usados para tal finalidade.

QUESTÃO: Utilizando conteúdo apresentado em sala de aula, desenvolva uma aplicativo (utilizando a ferramenta Eclipse) para atender o diagrama apresentado na questão, implementando conceitos de Polimorfismo e Herança oriundos da Programação Orientada a Objetos.

DIAGRAMA



As seguintes restrições precisam ser implementadas no sistema a ser desenvolvido:

- O sistema só deve autenticar apenas os colaboradores: Cardiologista, Clinico Geral e Dentista

- O colaborador Cardiologista possui produtividade (a produtividade é calculada em 30% de seu salário)
- O colaborador Clínico Geral possui produtividade (a produtividade é calculada em 20% de seu salário)
- O colaborador Dentista possui produtividade (a produtividade é calculada em 10% de seu salário)
- Deve ser adicionado ao modelo um método para calcular o total de salários a ser pago para os colaboradores: Cardiologia, Clínico Geral e Dentista. Como também um método para calcular o total de salários que serão pagos para os funcionários do RH e Financeiro.
- Deve ser criado classes para testar os métodos implementados.

```
public class Colaboradores {
    public double salario;
    public String cra;
    public String name;

    public double getSalario() {
        return salario;
    }
    public void setSalario(double salario) {
        this.salario = salario;
    }
    public String getCra() {
        return cra;
    }
    public void setCra(String cra) {
        this.cra = cra;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}

}
```

```
public class AutenticaColaborador extends Colaboradores {

    public int senha;
    public int crm;
    public boolean autenticado;
    public double produtividade;

    public int getSenha() {
        return senha;
    }

    public void setSenha(int senha) {
        this.senha = senha;
    }

    public int getCrm() {
        return crm;
    }
}
```

```

    public void setCrm(int crm) {
        this.crm = crm;
    }

    public boolean isAutenticado() {
        return autenticado;
    }

    public void setAutenticado(boolean autenticado) {
        this.autenticado = autenticado;
    }

    public void Produtividade() {
    }
}

```

```

public class Cardiologista extends AutenticaColaborador{
    int senha= 333;
    public double getSalario() {
        return this.salario * 0.3;
    }

    public void Autentica(int senha){
        if(this.senha == senha) {
            this.autenticado = true;
        }else {
            this.autenticado = false;
        }
    }

    public static void main (String[] args) {
        Cardiologista c1 = new Cardiologista();
        c1.setSalario(3000);
        c1.setName("Jubileu");
        c1.setCra("12345");
        c1.setSenha(333);

        c1.Autentica(333);

        System.out.println(c1.autenticado);
    }
}

```

```

public class Dentista extends AutenticaColaborador{
    int senha= 111;
    public double getSalario() {
        return this.salario * 0.1;
    }

    public void Autentica(int senha){
        if(this.senha == senha) {
            this.autenticado = true;
        }else {
            this.autenticado = false;
        }
    }
}

```

```

    }
}
public static void main (String[] args) {
    Dentista d1 = new Dentista();
    d1.setSalario(3000);
    d1.setName("Creuza");
    d1.setCra("12345");
    d1.setSenha(111);

    d1.Autentica(111);

    System.out.println(d1.autenticado);
}
}

```

```

public class ClinicoGeral extends AutenticaColaborador{
    int senha= 222;
    public double getSalario() {
        return this.salario * 0.2;
    }

    public void Autentica(int senha){
        if(this.senha == senha) {
            this.autenticado = true;
        }else {
            this.autenticado = false;
        }
    }
}
public static void main (String[] args) {
    ClinicoGeral cg1 = new ClinicoGeral();
    cg1.setSalario(3000);
    cg1.setName("Creuza");
    cg1.setCra("12345");
    cg1.setSenha(222);

    cg1.Autentica(222);

    System.out.println(cg1.autenticado);
}
}

```

```

public class Financeiro extends Colaboradores{

}

```

```

public class RH extends Colaboradores{

}

```