

```
package pctEmpleado;

public abstract class Empregado {

    private String nome;
    private String nrDocto;
    private double rendimento;

    public Empregado(String nome, String nrDocto, double rendimento) {
        this.nome = nome;
        this.nrDocto = nrDocto;
        this.rendimento = rendimento;
    }

    public String getNome() {
        return nome;
    }

    public String getNrDocto() {
        return nrDocto;
    }

    public double getRendimento() {
        return rendimento;
    }

    public void setRendimento(double rendimento) {
        this.rendimento = rendimento;
    }

    public abstract double rendimentos();
}
```

```

package pctEmpregado;

public class Salarior extends Empregador {

    private double salarioMensal;

    public Salarior(String nome, String nrDocto, double rendimento,
double salarioMensal) {
        super(nome, nrDocto, rendimento);
        this.salarioMensal = salarioMensal;

        if (salarioMensal < 0.0)
        {
            throw new IllegalArgumentException("Salário Mensal deve
                ser maior que ZERO!!!");
        }
        this.salarioMensal = salarioMensal;
    }

    public double getSalarioMensal() {
        return salarioMensal;
    }

    public void setSalarioMensal(double salarioMensal) {
        if (salarioMensal < 0.0)
        {
            throw new IllegalArgumentException("Salário Mensal deve
                ser maior que ZERO!!!");
        }
        this.salarioMensal = salarioMensal;
    }

    @Override
    public double rendimentos() {
        setRendimento(getSalarioMensal());
        return getSalarioMensal();
    }
}

```

```
package pctEmpregado;

public class Horario extends Empregado {

    private double salario;
    private double horas;

    public Horario(String nome, String nrDocto, double rendimento,
double salario, double horas) {
        super(nome, nrDocto, rendimento);

        if (salario < 0.0)
        {
            throw new IllegalArgumentException("Salário por hora
deve ser maior que ZERO!!!");
        }

        if (horas < 0.0)
        {
            throw new IllegalArgumentException("Número de horas
devem ser maior que ZERO!!!");
        }

        this.salario = salario;
        this.horas = horas;
    }

    public double getSalario() {
        return salario;
    }

    public void setSalario(double salario) {

        if (salario < 0.0)
        {
            throw new IllegalArgumentException("Salário por hora
deve ser maior que ZERO!!!");
        }

        this.salario = salario;
    }

    public double getHoras() {
        return horas;
    }

    public void setHoras(double horas) {

        if (horas < 0.0)
```

```

        {
            throw new IllegalArgumentException("Número de horas
                devem ser maior que ZERO!!!");
        }

        this.horas = horas;
    }

    @Override
    public double rendimentos() {
        if (getHoras() <= 160)
            setRendimento(getSalario() * getHoras());
        else
            setRendimento( 160 * getSalario() + (getHoras() - 160) *
                getSalario() * 1.5);
        return getRendimento();
    }

}

```

```

package pctEmpregado;

public class Comissao extends Empregado {

    private double vendaBruta;
    private double taxaComissao;

    public Comissao(String nome, String nrDocto, double rendimento,
double vendaBruta, double taxaComissao) {
        super(nome, nrDocto, rendimento);

        if (taxaComissao <= 0.0 || taxaComissao >= 1.0)
        {
            throw new IllegalArgumentException("Taxa de comissão
            deve ser entre 0 e 1.0!!!");
        }

        if (vendaBruta < 0.0)
        {
            throw new IllegalArgumentException("Vendas devem ser
            maior que ZERO!!!");
        }
        this.vendaBruta = vendaBruta;
        this.taxaComissao = taxaComissao;
    }

    public double getVendaBruta() {
        return vendaBruta;
    }

    public void setVendaBruta(double vendaBruta) {
        if (vendaBruta < 0.0)
        {
            throw new IllegalArgumentException("Vendas devem ser
            maior que ZERO!!!");
        }
        this.vendaBruta = vendaBruta;
    }

    public double getTaxaComissao() {
        return taxaComissao;
    }

    public void setTaxaComissao(double taxaComissao) {
        if (taxaComissao <= 0.0 || taxaComissao >= 1.0)
        {
            throw new IllegalArgumentException("Taxa de comissão
            deve ser entre 0 e 1.0!!!");
        }
    }
}

```

```
        this.taxaComissao = taxaComissao;
    }

    @Override
    public double rendimentos() {
        setRendimento(getTaxaComissao() * getVendaBruta());
        return getRendimento();
    }
}
```

```
package pctEmpregado;

public class TestePagamento {

    public static void main(String[] args) {
        //Criando objetos

        Salario sal = new Salario("João", "123456789",0, 2500.00);

        System.out.println("***Salário Mensal do Empregado***");

        System.out.println("Nome :"+sal.getNome());
        System.out.println("Documento :"+sal.getNrDocto());
        System.out.println("Salário Mensal:"+sal.getSalarioMensal());

        System.out.println();
        sal.rendimentos();
        System.out.println("Rendimentos :"+sal.getRendimento());
        System.out.println();
        System.out.println();

        Horario hor = new Horario("João", "123456789",0,
                                   30.00, 12.00);

        System.out.println("***Salário por Hora do Empregado***");

        System.out.println("Nome
                            :"+hor.getNome());
        System.out.println("Documento
                            :"+hor.getNrDocto());
        System.out.println("Número de Horas Trabalhadas
                            :"+hor.getHoras());
        System.out.println("Valor hora
                            :"+hor.getSalario());

        System.out.println();
        hor.rendimentos();
        System.out.println("Rendimentos :"+hor.getRendimento());
        System.out.println();
        System.out.println();

        Comissao com = new Comissao("João", "123456789",
                                      0, 2000.00, 0.45);

        System.out.println("***Vendas Brutas & Comissões***");

        System.out.println("Nome
                            :"+com.getNome());
```

```
System.out.println("Documento  
: "+com.getNrDocto());  
System.out.println("Taxa de Comissão de Vendas  
: "+com.getTaxaComissao());  
System.out.println("Vendas Brutas  
: "+com.getVendaBruta());  
  
System.out.println();  
  
com.rendimentos();  
System.out.println("Rendimentos : "+com.getRendimento());  
  
System.out.println();  
System.out.println();  
  
}  
  
}
```