

```
public class Pagamento {
    private String nomeDoPagador;
    private String cpf;
    private double valorASerPago;

    public Pagamento() {
        this("", "", 0.0);
    }

    public Pagamento(String nomeDoPagador, String cpf, double valorASerPago) {
        this.nomeDoPagador = nomeDoPagador;
        this.cpf = cpf;
        this.valorASerPago = valorASerPago;
    }

    public String getNomeDoPagador() {
        return nomeDoPagador;
    }

    public void setNomeDoPagador(String nomeDoPagador) {
        this.nomeDoPagador = nomeDoPagador;
    }

    @Override
    public String toString() {
        return "Pagamento{" +
            "nomeDoPagador='" + nomeDoPagador + '\'' +
            ", cpf='" + cpf + '\'' +
            ", valorASerPago=" + valorASerPago +
            '}';
    }
}

public class CartaoDeCredito extends Pagamento {
    private String numeroCartao;
    private String nomeTitular;
    private String dataValidade;

    public CartaoDeCredito(double valor, String numeroCartao, String nomeTitular, String dataValidade) {
        super(valor);
        this.numeroCartao = numeroCartao;
        this.nomeTitular = nomeTitular;
        this.dataValidade = dataValidade;
    }
}
```

```
}
```

```
public void setNumeroDoCheque(String numeroDoCheque) {  
    if (numeroDoCheque == null || numeroDoCheque.isEmpty()) {  
        throw new IllegalArgumentException("Número do cheque inválido");  
    }  
    this.numeroDoCheque = numeroDoCheque;  
}
```

```
@Override  
public String toString() {  
    return super.toString() + ", numeroDoCheque=" + numeroDoCheque + ", banco=" + banco + "\";  
}  
}
```

```
@Override  
public String toString() {  
    return super.toString() + ", numeroDoCartao=" + numeroDoCartao + "\";  
}  
}
```

```
public class Boleto extends Pagamento {  
    private String numeroDoBoleto;  
    private LocalDate dataVencimento;
```

```
public void setDataVencimento(LocalDate dataVencimento) {  
    if (dataVencimento == null || dataVencimento.isBefore(LocalDate.now())) {  
        throw new IllegalArgumentException("Data de vencimento inválida");  
    }  
    this.dataVencimento = dataVencimento;  
}
```

```
@Override  
public String toString() {  
    return super.toString() + ", numeroDoBoleto=" + numeroDoBoleto + ", dataVencimento=" +  
dataVencimento;  
}  
}
```

```
public class PessoaFisica {  
    private String nome;  
    private String sobrenome;  
    private String cpf;
```

```
public PessoaFisica() {
```

```

        this("Não informado", "Não informado", "000.000.000-00");
    }

    public PessoaFisica(String nome, String sobrenome, String cpf) {
        this.nome = nome;
        this.sobrenome = sobrenome;
        this.cpf = cpf;
    }

    public String dados() {
        return "Nome: " + nome + ", Sobrenome: " + sobrenome + ", CPF: " + cpf;
    }
}

public class Desempregado extends PessoaFisica {
    private double seguroDesemprego;

    public Desempregado() {
        super();
        this.seguroDesemprego = 0.0;
    }
    public Desempregado(String nome, String sobrenome, String cpf, double seguroDesemprego) {
        super(nome, sobrenome, cpf);
        this.seguroDesemprego = seguroDesemprego;
    }

    @Override
    public String dados() {
        return super.dados() + ", Seguro Desemprego: " + seguroDesemprego;
    }
}

public class Empregado extends PessoaFisica {
    private double salario;
    private String cargo;

    @Override
    public String dados() {
        return super.dados() + ", Salário: " + salario + ", Cargo: " + cargo;
    }
}

```

```
public class Main {  
    public static void main(String[] args) {  
        PessoaFisica pessoa = new Desempregado("João", "Silva", "12345678901", 1500.0);  
        System.out.println(pessoa.dados());  
  
        Empregado empregado = new Mensalista("Maria", "Santos", "98765432109", 5000.0, "Analista");  
        System.out.println(empregado.dados());  
    }  
}
```