Arquivo: conta.js

class Conta {

    titular = "";

    saldo = 0;

    constructor(titular, saldo) {

        this.titular = titular;

        this.saldo = saldo;

    }

    validarTitulo(titulo) {

        if(typeof(titulo) !== 'string' || !titulo) {

            throw new Error("Informe um um título!");

        }

    }

    validar(valor) {

        if(typeof(valor) !== 'number' || !valor || valor < 0) {

            throw new Error("Informe um valor numérico!");

        }

    }

    validarSaque(valor) {

        if(valor < 5) {

            throw new Error("Informe um valor numérico igual ou acima de cinco!");

        }

    }

    depositar(valor) {

        this.validar(valor);

        this.saldo += valor;

    }

    sacar(valor) {

        this.validar(valor);

        this.validarSaque(valor);

        this.saldo -= valor;

    }

    imprimir() {

        return `${this.titular.toUpperCase()} possui saldo de R$ ${this.saldo.toFixed(2)}.`;

    }

}

module.exports = Conta;

Arquivo: conta-poupanca.js

const Conta = require('./conta');

class ContaPoupanca extends Conta {

    rendimento = 0;

    constructor(titulo, saldo, rendimento) {

        super(titulo, saldo);

        this.rendimento = rendimento;

    }

    aplicarRendimento() {

        this.saldo += this.rendimento;

    }

    imprimir() {

        return `${this.titular.toUpperCase()} possui saldo de R$ ${this.saldo.toFixed(2)} com rendimento de R$ ${this.rendimento.toFixed(2)}.`;

    }

}

module.exports = ContaPoupanca;

Arquivo: conta-corrente.js

const Conta = require('./conta');

class ContaCorrente extends Conta {

    juros = 0;

    constructor(titulo, saldo, juros) {

        super(titulo, saldo);

        this.juros = juros;

    }

    aplicarJuros() {

        this.saldo += this.juros \* 0.1;

    }

    imprimir() {

        return `${this.titular.toUpperCase()} possui saldo de R$ ${this.saldo.toFixed(2)} com juros de ${this.juros.toFixed(1)}%.`;

    }

}

module.exports = ContaCorrente;

Arquivo: index.js

Importação do prompt:

const prompt = require('prompt-sync')();

Importação de Classes:

const Conta = require('./model/conta');

const ContaPoupanca = require('./model/conta-poupanca');

const ContaCorrente = require('./model/conta-corrente');

Pegar Título:

function getTitulo() {

    const titulo = prompt("Informe o nome do titular: ");

    return titulo;

}

Pegar Valor:

function getValor(operacao) {

    let valor = 0;

    if(operacao === "saque") {

        valor = +prompt(`Informe o valor de ${operacao}: `);

    }

    else if(operacao === "depósito") {

        valor = +prompt(`Informe o valor de ${operacao}: `);

    }

    return valor;

}

Pegar Rendimento:

function getRendimento() {

    const rendimento = +prompt("Informe o rendimento: ");

    return rendimento;

}

Pegar Juros:

function getJuros() {

    const juros = +prompt("Informe o juros: ");

    return juros;

}

Cadastrar Conta Genérica:

function cadastrarContaGenerica() {

    console.log("\nConta:");

    let conta = new Conta(getTitulo(), 0);

    conta.depositar(getValor('depósito'));

    conta.sacar(getValor('saque'));

    imprimirDados(conta);

}

Cadastrar Conta Poupança:

function cadastarContaPoupanca() {

    console.log("\nConta Poupança:");

    let contaPoupanca = new ContaPoupanca(getTitulo(), 0, 0);

    contaPoupanca.depositar(getValor('depósito'));

    contaPoupanca.sacar(getValor('saque'));

    contaPoupanca.rendimento = getRendimento();

    contaPoupanca.aplicarRendimento();

    imprimirDados(contaPoupanca);

}

Cadastrar Conta Corrente:

function cadastrarContaCorrente() {

    console.log("\nConta Corrente:");

    let contaCorrente = new ContaCorrente(getTitulo(), 0, 0);

    contaCorrente.depositar(getValor('depósito'));

    contaCorrente.sacar(getValor('saque'));

    contaCorrente.juros = getJuros();

    contaCorrente.aplicarJuros();

    imprimirDados(contaCorrente);

}

Imprimir Dados:

function imprimirDados(conta) {

    console.log("\nImpressão:");

    console.log(conta.imprimir());

    console.log("");

}

Validar

function validar(valor) {

    if(typeof(valor) !== 'number' || !valor || valor <= 0 || Number.isNaN(valor) || !Number.isInteger(valor)) {

        throw new Error("Informe números inteiros positivos diferente de zero!");

    }

}

console.log("");

Informa a quantidade de contas e executa aplicação:

try {

    var qtdContas = +prompt("Informe a quantidade de contas: ");

    validar(qtdContas);

    for(let i=0; i<qtdContas; i++) {

        let opcao = +prompt("Informe se deseja: [1]-Conta | [2]-Poupança | [3]-Corrente: ");

        validar(opcao);

        switch(opcao) {

            case 1: {

                cadastrarContaGenerica();

                console.log("");

                break;

            }

            case 2: {

                cadastarContaPoupanca();

                console.log("");

                break;

            }

            case 3: {

                cadastrarContaCorrente();

                console.log("");

                break;

            }

            default: {

                throw new Error("Opção inválida! Tente novamente.");

            }

        }

    }

}

catch(error) {

    console.log(`\n${error.message}\n`);

}