MODULO 8
Exceções

Classe SaldoInsuficienteException

```
public class SaldoInsuficienteException extends Exception {
	private double quantidade;

	public SaldoInsuficienteException(double quantidade) {
		this.quantidade = quantidade;
	}

	public double getAmount() {
		return quantidade;
	}
}
```

Classe Conta

```
public class Conta {
	private double saldo;
	private int numeroConta;

	public Conta(int numeroConta) {
		this.numeroConta = numeroConta;
	}

	public void depositar(double valor) {
		saldo += valor;
	}

	public void sacar(double valor) throws SaldoInsuficienteException {
		if (valor <= saldo) {
			saldo -= valor;
		} else {
			double quantidade = valor - saldo;
			throw new SaldoInsuficienteException(quantidade);
		}
	}

	public double getSaldo() {  return saldo; }

	public int getNumeroConta() {  return numeroConta;    }
}
```

Classe BankDemo

```
public class BankDemo {

	public static void main(String[] args) {
		Conta c = new Conta(101);
		System.out.println("Depositando $500...");
		c.depositar(500.00);
		try {
			System.out.println("Saque de $100...");
			c.sacar(100.00);
```

```
                  System.out.println("Saque de $600...");
                  c.sacar(600.00);
            } catch (SaldoInsuficienteException e) {
                  System.out.println("Desculpe, porem voce esta com deficit de
$" + e.getAmount());
                  StackTraceElement[] traceElements = e.getStackTrace();
                  System.out.println("Class\t\tMethod\tLine number  ");
                  for(StackTraceElement element : traceElements){
                        System.out.printf("%s\t",element.getFileName());
                        System.out.printf("%s\t",element.getMethodName());
                        System.out.printf("%s\n",element.getLineNumber());

                  }
            }
            System.out.println("Transações possiveis realizadas com sucesso");
      }
}
```

Saída ao rodar a classe BankDemo:

```
Depositando $500...
Saque de $100...
Saque de $600...
Desculpe, porem voce esta com deficit de $200.0
Class           Method Line number
Conta.java      sacar  23
BankDemo.java   main   13
Transações possiveis realizadas com sucesso
```

Lista de Exercícios 8

1. Modifique a implementação do exemplo mostrado previamente para que uma exceção de tipo SaqueAcimaDoLimiteException seja lançada quando o valor a ser sacado exceda um limite dado.
   Para isso deverá criar a classe SaqueAcimaDoLimiteException contendo um atributo que armazene o valor da diferença entre o que se desejó sacar e o que era permitido sacar como máximo.
   Esse limite máximo deverá ser acrescentado como atributo na classe conta e deverá modificar o construtor para que receba dois parâmetros numeroConta e limite.
   Modifique BankDemo para testar essa funcionalidade.

2. Adicione uma exceção que seja lançada quando o valor do deposito for inferior a 5 reais ou negativo. Para isso você criará uma classe DepositoNaoPermitidoException que conterá como atributo o valor que tentou-se depositar.
   Modifique BankDemo para testar essa funcionalidade.

3. Desenvolva um método que recebe um String como parâmetro e verifica se o mesmo é composto apenas por caracteres maiúsculos .
   O método deve lançar 2 tipos de exceções específicas:
   a) uma para indicar se existe algum caracter que não é uma letra e
   b) outra para indicar se alguma das letras não é uma maiúscula.

   Lembre que um String é um array de caracteres. A classe String possui funções úteis como charAt(), toCharArray(), length(), etc.
   Para verificar o tipo dos caracteres use os métodos “isLetter” e “isUpperCase” da classe “Character” (ambos static).