Exceções
Lab 9

Neste laboratório faremos uso das estruturas de tratamento de **“erros”** usando **Exceções**. A linguagem de programação Java possui uma estrutura de manipulação de exceção muito bem elaborada, que ajuda os desenvolvedores a separar a lógica de tratamento de exceções de sua lógica de negócios.

Duração prevista: 60 minutos

Exercícios

Exercício 1: Criando exceções

Exercício 2: Usando exceções

Criando exceções

1 - Quando sacamos e transferimos dinheiro de uma instância da classe **ContaService**, **Laboratório 5 exercício 3, item 1,** espera-se que o saldo seja suficiente para cobrir a operação, porém pode ocorrer uma situação de exceção, ou seja, situação inesperada. Neste caso podemos usar os mecanismos de controle de exceção em Java para contornar a situação. Veja o código abaixo, nele criamos uma nova classe de exceção **SaldoInsuficienteException**, para indicarmos que não há saldo suficiente para que uma operação saque/transferência ocorra.

```
public class SaldoInsuficienteException extends Exception {

	public SaldoInsuficienteException() {

		super("Saldo insuficiente.");
	}

	public SaldoInsuficienteException( String mensagem ) {

		super(mensagem);
	}
}
```

2 - Modifique o método **transferir()** da classe **ContaService** conforme exemplo abaixo.
A partir de agora quando uma transferência for executada, será necessário verificar se a conta possui saldo, se o saldo for insuficiente à aplicação lançará uma exceção.

```
@Override
public boolean transferir(Conta contaSaque, double valor,
		Conta contaDestino, String descr)  throws SaldoInsuficienteException {

	if (contaSaque.getSaldo() - valor >= 0) {
		this.debito(contaSaque, valor);
		this.credito(contaDestino, valor);
		this.historicoTransacao(contaSaque, contaDestino, valor, descr,
				EnumTipoTransacao.TRANSFERENCIA);

		return true;

	} else {
		throw new SaldoInsuficienteException();
	}
```

```
}
```

3 - A compilação da classe **ContaService** irá gerar um erro, isto se deve ao fato do método **transferir()** usado na sobrecarga não estar capturando e tratando a exceção **SaldoInsuficienteException**. Modifique o método sobrecarregado **transferir()** para desviar a exceção usando a cláusula **throws** na assinatura do método.

4 - Faça o mesmo tratamento de exceção para o método **sacar().**

Usando exceções

1 - Crie a classe **MovimentoContaCorrente** conforme exemplo abaixo. Iremos testar e manipular a exceção que acabou de ser criada e adicionada ao método **transferir().**

```
public class MovimentoContaCorrente {

	public static void main(String[] args) {

		// Cria uma instância de ContaService onde está presente as operações para Objeto Conta
		ContaService operacoesConta = new ContaService();

		Conta correntista1 = new Conta("Aluno", 1001);
		Conta correntista2 = new Conta("Aluna", 21);

		// faz deposito
		operacoesConta.depositar(correntista1, 100);

		// faz Transferencia proibida
		try {
			operacoesConta.transferir(correntista1, 600, correntista2);
		} catch (Exception e) {
			System.out.println(e.getMessage());
		}

		// faz Transferencia autorizada
		try {
			operacoesConta.transferir(correntista1, 99.00, correntista2);
		} catch (Exception e) {
			System.out.println(e.getMessage());
		}

		ExtratoTXT movimento = new ExtratoTXT(correntista1);
		System.out.println(movimento.formatar());
	}

}
```