

# PROGRAMAÇÃO ORIENTADA A OBJETOS

---

Prof. Delano M. Beder

---

## Roteiro 02a

### Paradigma Estruturado - Conta Corrente (número e saldo)

---

Conta Corrente (número e saldo) e 4 operações:

- retirada de um valor em uma conta corrente (verificar se o saldo é suficiente para a retirada)
  - depósito de um valor em uma conta corrente
  - transferência de um valor entre 2 contas (retirada da 1a conta e depósito na 2a conta)
  - imprime informações (número e saldo) de uma conta corrente
- 

1. Crie um projeto (Aplicação C) denominado **ContaCorrente-C**

2. Implementação da abstração **Conta Corrente**

Crie um arquivo de cabeçalho C denominado **ContaCorrente.h**

```
#ifndef CONTACORRENTE_H
#define CONTACORRENTE_H

#define FALSE 0
#define TRUE 1

typedef struct ContaCorrente {
    int numero;
    double saldo;
} ContaCorrente;

int retirada(ContaCorrente* conta, double valor);

void deposito(ContaCorrente* conta, double valor);

int transferencia(ContaCorrente* c1, ContaCorrente* c2, double valor);

void imprime(ContaCorrente conta);

#endif /* CONTACORRENTE_H */
```

Crie um arquivo de código-fonte C denominado **ContaCorrente.c**

```
#include <stdio.h>
#include "ContaCorrente.h"

int retirada(ContaCorrente* conta, double valor) {
    if (conta->saldo - valor >= 0) {
        conta->saldo -= valor;
        return TRUE;
    }
```

```

    }
    return FALSE;
}

void deposito(ContaCorrente* conta, double valor) {
    conta->saldo += valor;
}

int transferencia(ContaCorrente* c1, ContaCorrente* c2, double valor) {
    int ok = retirada(c1, valor);
    if (ok) {
        deposito(c2, valor);
    }
    return ok;
}

void imprime(ContaCorrente conta) {
    printf("Numero: %d, saldo: %5.2f\n", conta.numero, conta.saldo);
}

```

### 3. Atualize o arquivo de código-fonte C denominado **main.c** (programa principal)

```

#include <stdio.h>
#include "ContaCorrente.h"

void main() {

    ContaCorrente c1;
    c1.numero = 1000;
    c1.saldo = 1200.50;

    ContaCorrente c2;
    c2.numero = 2000;
    c2.saldo = 0;

    printf("Antes da Transferência\n\n");

    imprime(c1);
    imprime(c2);

    transferencia(&c1, &c2, 500);

    printf("\nDepois da Transferência\n\n");

    imprime(c1);
    imprime(c2);
}

```

4. Crie o arquivo **Makefile** (opcional)

```
CXX      := gcc
CXX_FLAGS :=

LIBRARIES :=
EXECUTABLE := ContaCorrente-C

all: run
    @rm -f $(EXECUTABLE)

run: $(EXECUTABLE)
    @./$(EXECUTABLE)

$(EXECUTABLE): *.c
    $(CXX) $(CXX_FLAGS) -I. $^ -o $@ $(LIBRARIES)
```

5. Compile, execute e verifique a saída impressa

6. Fim