

```
PACKAGE LANCHONETE
IMPORT LANCHONETE.PEDIDO
IMPORT JAVA.UTIL.ARRAYLIST;
PUBLIC CLASS CLIENTE {

    PRIVATE STRING NOME;
    PRIVATE STRING ENDERECO;
    PRIVATE STRING NUMERODETELEFONE;

    PRIVATE ARRAYLIST<PEDIDO> PEDIDOS = NEW ARRAYLIST<PEDIDOS>();

    PUBLIC VOID ADICIONARPEDIDO(PEDIDO p) {
        PEDIDOS.ADD(p);
    }

    PUBLIC VOID IMPRIMIRDADOS() {
        System.out.println("NOME: " + NOME);
        System.out.println("ENDERECO: " + ENDERECO);
        System.out.println("Nº DE TELEFONE: " + NUMERODETELEFONE);
    }

    PUBLIC SOMAR PEDIDOS() {
        FLOAT VALORTOTAL;
        FOR (INT i=0, i < PEDIDOS.SIZE(), i++) {
            VALORTOTAL += PEDIDOS.GET(i).GETVALOR();
        }
        System.out.println("VALOR TOTAL: " + VALORTOTAL);
    }

    Public CLIENTE(STRING NOME, STRING ENDERECO, STRING TELEFONE) {
        SUPER();
        THIS.NOME=NOME;
        THIS.ENDERECO=ENDERECO;
        THIS.NUMERODETELEFONE=TELEFONE
    }
}
```

```
PACKAGE LANCMONETE;
```

```
PUBLIC CLASS PEDIDO {
```

```
    PUBLIC VOID PEDIDO (INT NUMERODoPEDIDO, STRING DESCRIÇÃO, FLOAT VALOR  
    & SUPER);
```

```
    THIS.NUMERODoPEDIDO = NUMERODo PEDIDO;
```

```
    THIS.DESCRIÇÃO = DESCRIÇÃO;
```

```
    THIS.VALOR = VALOR;
```

```
}
```

```
    PUBLIC STRING DESCRIÇÃO;
```

```
    PUBLIC INT NUMERO Do PEDIDO;
```

```
    PUBLIC FLOAT VALOR;
```

```
    PUBLIC VOID IMPRIMIRDADOS () {
```

```
        System.out.println ("Número: " + NUMERO Do PEDIDO);
```

```
        System.out.println ("VALOR: " + VALOR);
```

```
        System.out.println ("DESCRÍCÃO: " + DESCRIÇÃO);
```

```
}
```

```

PACKAGE LANCHONETE;
IMPORT JAVA.UTIL.ARRAYLIST;
IMPORT LANCHONETE.PEDIDO;
IMPORT LANCHONETE.CLIENTE;
PUBLIC CLASS MAIN{
PUBLIC STATIC VOID MAIN (STRING[] args){
ARRAYLIST < CLIENTE > CLIENTES = NEW ARRAYLIST<CLIENTE>();
INT OPCAO;
SCANNER SC = NEW SCANNER (System.in)
WHILE (TRUE) {
System.out.println("1 - CADASTRAR CLIENTE");
System.out.println("2 - CADASTRAR PEDIDO");
System.out.println("3 - LISTAR PEDIDOS DE UM CLIENTE");
System.out.println("4 - IMPRIMIR CLIENTE");
System.out.println("5 - IMPRIMIR TODOS OS PEDIDOS");
System.out.println("6 - IMPRIMIR GASTOS DO CLIENTE");
OPCAO = SC.NEXTINT();
SWITCH (OPCAO) {
CASE 1:
System.out.print ("Nome: ");
STRING NOME = SC.NEXTLINE();
System.out.print ("Endereço: ");
STRING ENDERECO = SC.NEXTLINE();
System.out.print ("Número de Telefone: ");
STRING TELEFONE = SC.NEXTLINE();
CLIENTE C = NEW CLIENTE (NOME, ENDERECO, TELEFONE);
CLIENTES.ADD(C);
BREAK;
CASE 2:
System.out.print ("Cliente do Pedido: ");
String CLIENTEPEDIDO;
CLIENTE CI = NULL;
FOR (INT i=0; i < CLIENTES.SIZE(); i++){
IF (CLIENTES.GET(i).GETNOME == CLIENTEPEDIDO) {
CI = CLIENTES.GET(i);
}
}
}
}
}

```

```

IF (C1 == NULL) {
    System.out.println("O cliente não está cadastrado.");
    BREAK;
}

System.out.println("Descrição:");
String descrição = SC.NEXTLINE();
System.out.println("Número do pedido:");
INT NUMERO = SC.NEXTINT();
GETLINE();
System.out.println("Valor:");
FLOAT VALOR = SC.NEXTFLOAT();

PEDIDO* P1 = NEW PEDIDO (NUMERO, descrição, VALOR);

C1. ADICIONAR PEDIDO (P1);
BREAK;

CASE 3:
System.out.println("Nome:");
STRING NOME1 = SC.NEXTLINE(); // CLIENTE* P=NULL;
FOR (INT i=0; i < CLIENTES.SIZE(); i++) {
    IF (CLIENTES.GET(i).GETNOME() == NOME1) {
        P = CLIENTES.GET(i);
    }
}
IF (P == NULL) {
    System.out.println("O cliente desejado não está cadastrado.");
    BREAK;
}
FOR (INT i=0; i < P.PEDIDOS.SIZE(); i++) {
    P.PEDIDOS.GET(i).IMPRIMIR DADOS();
}
BREAK;

```

CASE 4:

```
IF(CLIENTES.ISEMPTY()){
    System.out.println("Não há cliente cadastrado");
    BREAK;
} ELSE {

    FOR (int i=0; i < CLIENTES.SIZE(); i++) {
        CLIENTES.GET(i).IMPRIMIR DADOS();
        CLIENTES.GET(i).SOMARPEDIDOS();
    }
}
BREAK;
```

CASE 5:

```
IF(!CLIENTES.ISEMPTY()){
    FOR (int i=0; i < CLIENTES.SIZE(); i++) {
        FOR (int j=0; j < CLIENTES.GET(i).GETPEDIDOS().SIZE(); j++) {
            CLIENTES.GET(i).GETPEDIDOS().GET(j).IMPRIMIRDADOS();
        }
        CLIENTE.GET(i).IMPRIMIR DADOS();
    }
} ELSE { System.out.println("Não há cliente cadastrado"); }
```

CASE 6:

```
System.out.println("Nome:");
STRING NOME2 = SC.NEXTLINE;

CLIENTE* P1 = NULL;
FOR (int i=0; i < CLIENTES.SIZE(); i++) {
    IF(CLIENTES.GET(i).GETNOME() == NOME2) {
        CLIENTES.GET(i).SOMARPEDIDOS();
        P1 = CLIENTES.GET(i);
    }
} IF(P1 == NULL) {
    System.out.println("O cliente não está cadastrado");
}
```

BREAK;

DEFAULT:

System.out.println("Opção inválida");

BREAK;

}