

Padrões de Projeto

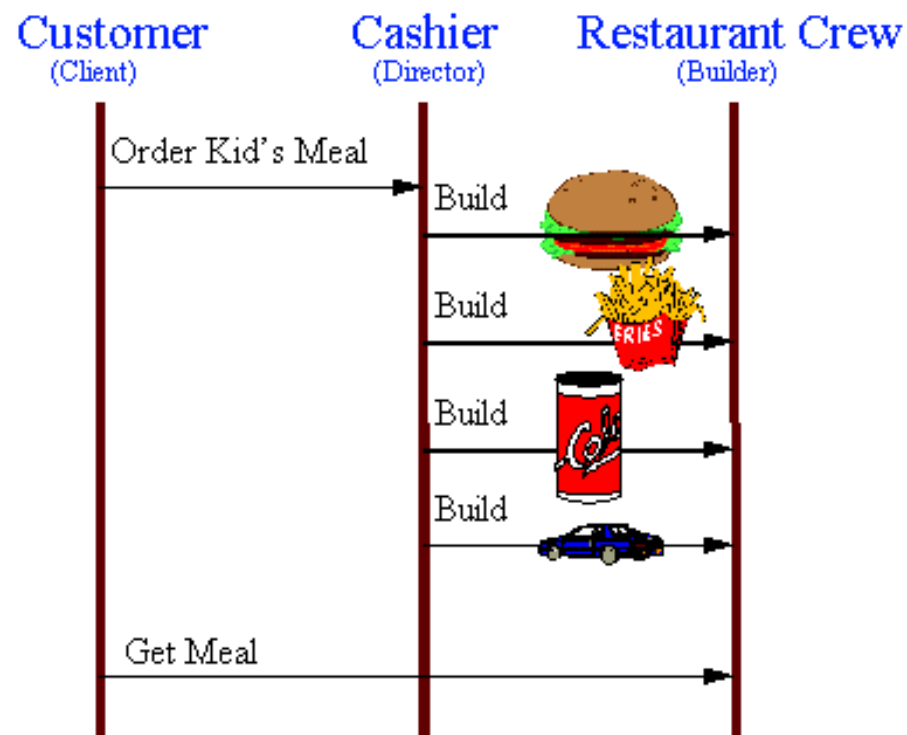
Prof. Adilson Vahldick

Departamento de Engenharia de Software

Udesc Ibirama

Objetivos da aula

- Conhecer e aplicar o padrão
 - Builder



Problema (1)

- Temos uma linha de montagem de veículos. Cada veículo tem um conjunto (tipo e quantidade) de peças distintas. Constrói de acordo com a encomenda:

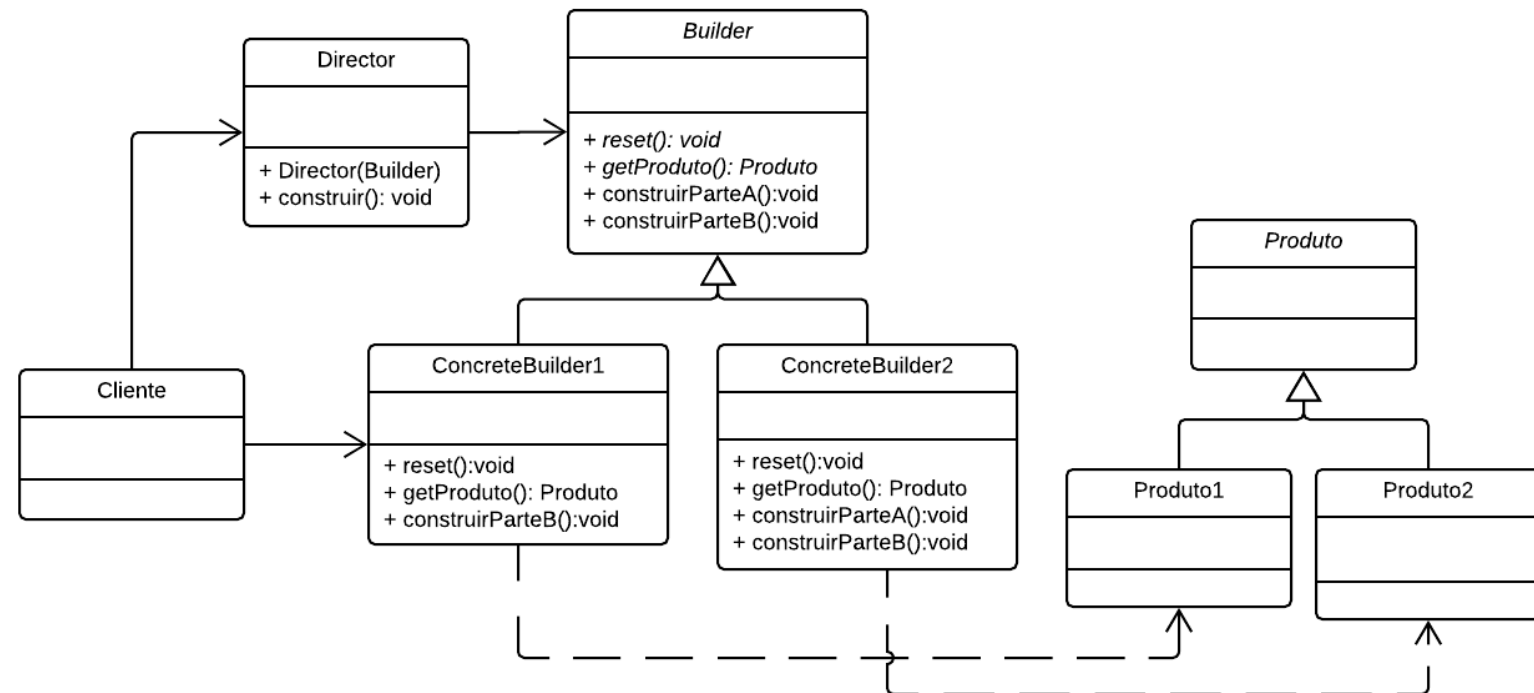
[illegible]

Problema (2)

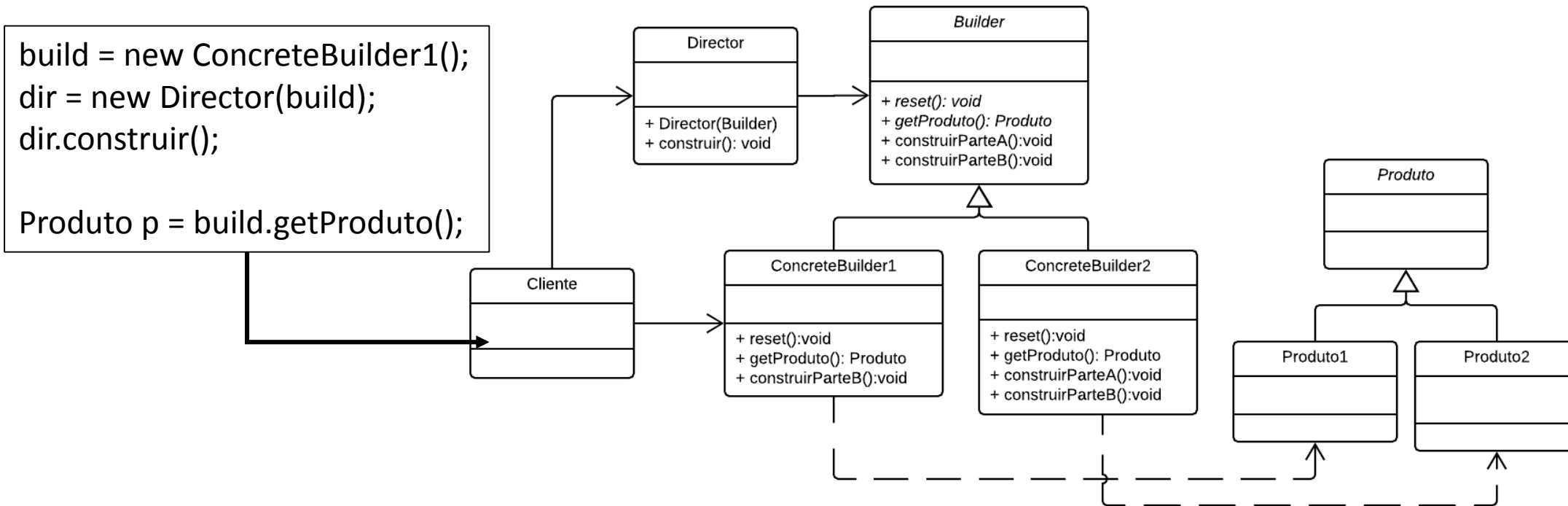
- Qual o problema com essa abordagem ?
 - Novos tipos de carros exigem mudanças no algoritmo de construção
 - Tipos de carros podem ter características (p.e. número de pneus) que modificam a forma de construí-lo

Solução (1)

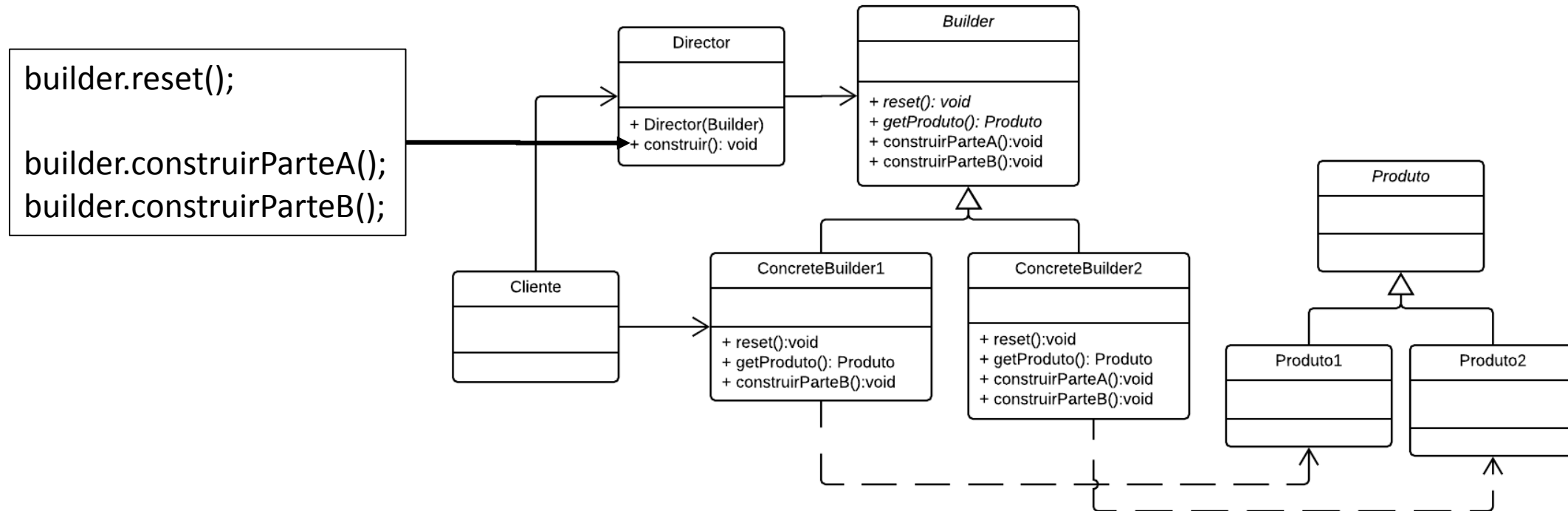
- **Builder:** separar a construção de um objeto complexo da sua representação de modo que o mesmo processo de construção possa criar diferentes representações



Solução (2)



Solução (3)

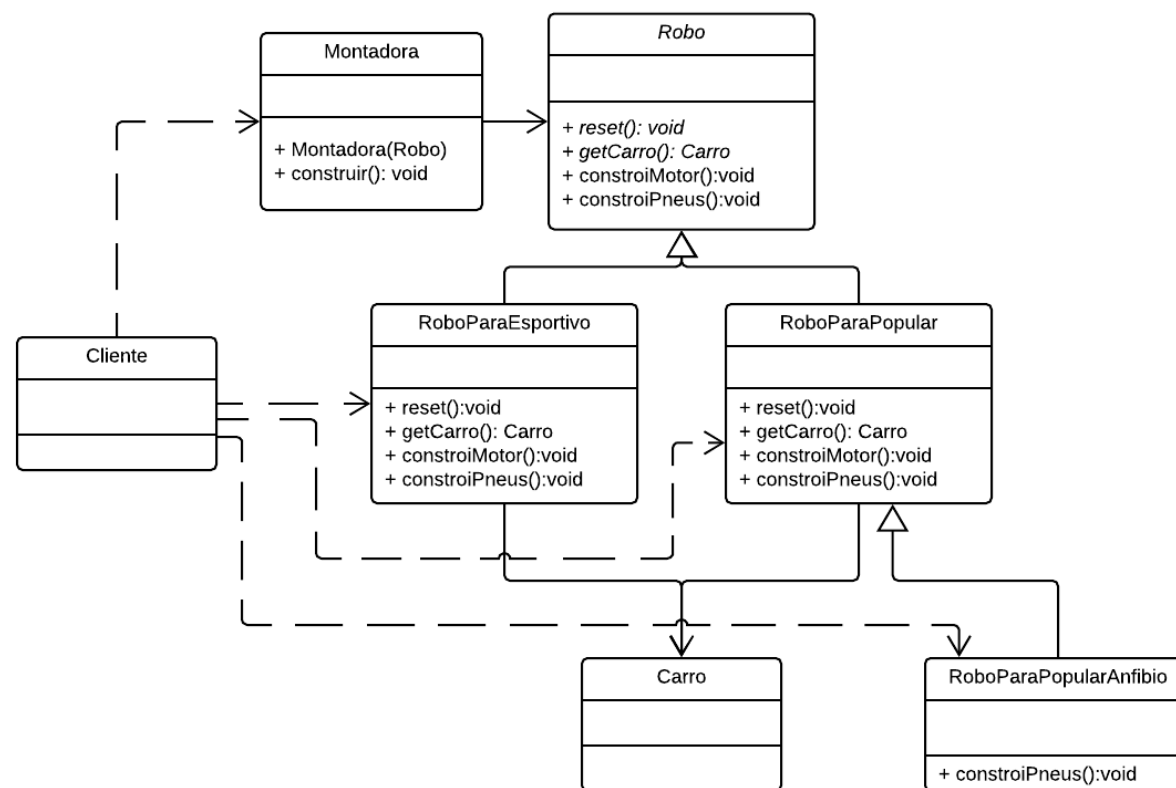


Problema (3) – builder1

```
Carro carro = new Carro();
if (tipo == "esportivo") {
    carro.setMotor(new MotorV8());
    carro.setPneus(new Pneu[]{new Pneu19(), new Pneu19(),
                                new Pneu19(), new Pneu19()});
} else {
    carro.setMotor(new Motor1_0());
    if (tipo == "anfíbio")
        carro.setPneus(new Pneu[]{new Pneu14(), new Pneu14(), new Pneu14(),
                                    new Pneu14(), new Pneu14(), new Pneu14()});
    else
        carro.setPneus(new Pneu[]{new Pneu14(),
                                    new Pneu14(), ew Pneu14(), new Pneu14()});
}
```

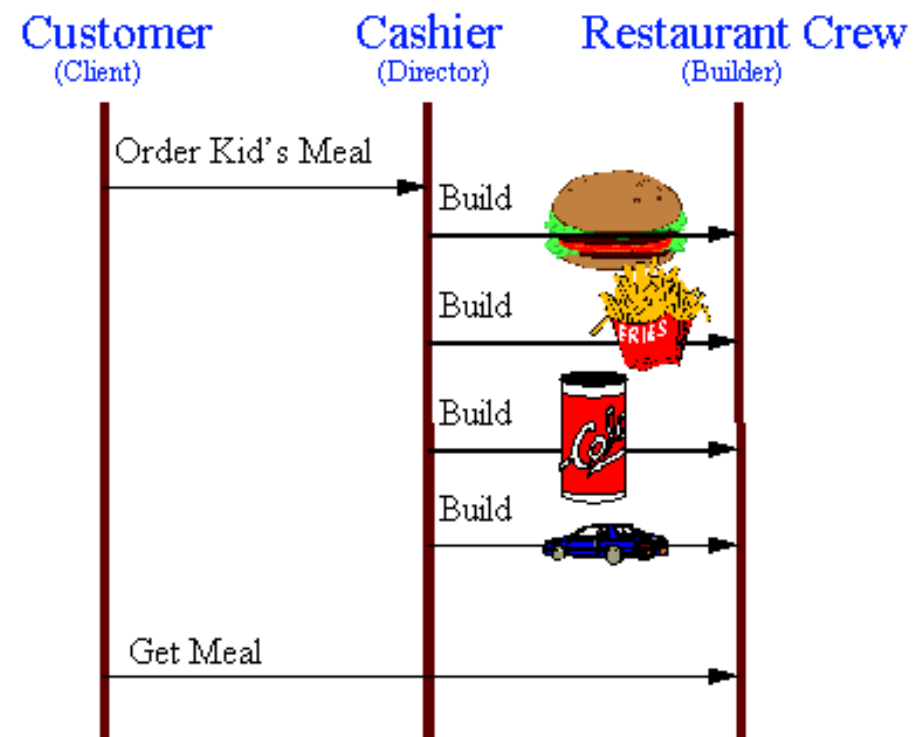
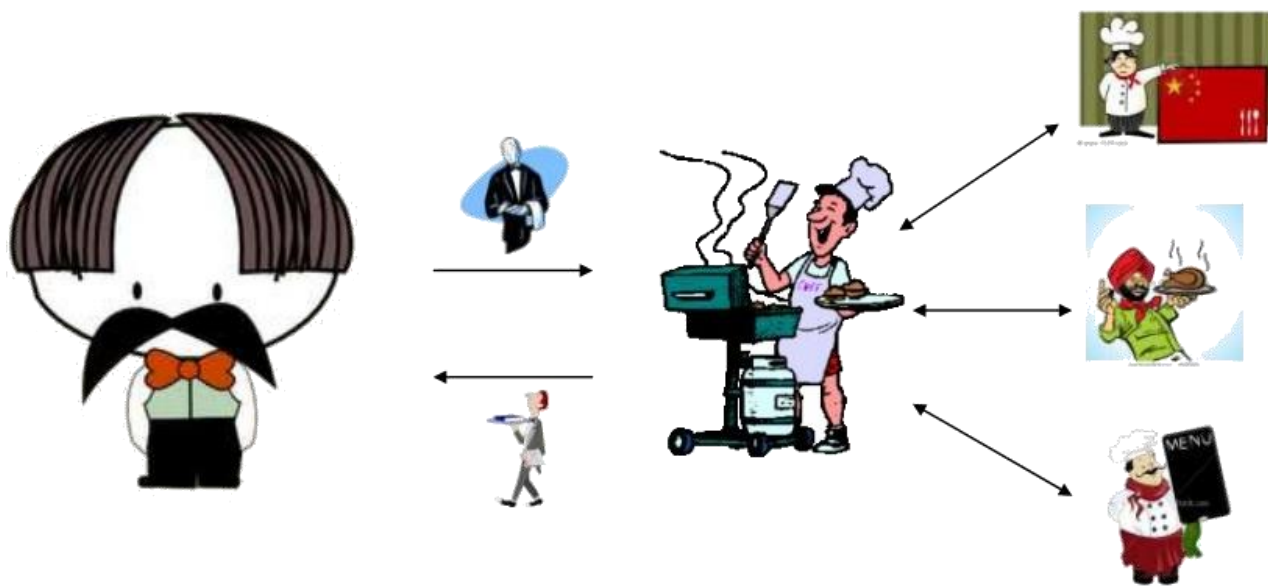

Problema (4) – builder1

Acompanhar com o código do projeto



Abstract Factory X Builder

Qual a diferença entre esses padrões ?





Para refletir no uso do Builder para o trabalho

```
String[] opcoes = {"A", "B", "C"};
int opcao = JOptionPane.showOptionDialog(this,
    "Escolha uma opção:", "JOGO",
    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE,
    null, opcoes, opcoes[0]);
switch (opcao) {
    case 1: // 10 mil linhas para construir o tabuleiro para opção 1
    case 2: // 25 mil linhas construir o tabuleiro para opção 2
    case 3: // 5 mil linhas construir o tabuleiro para opção 3
}
```

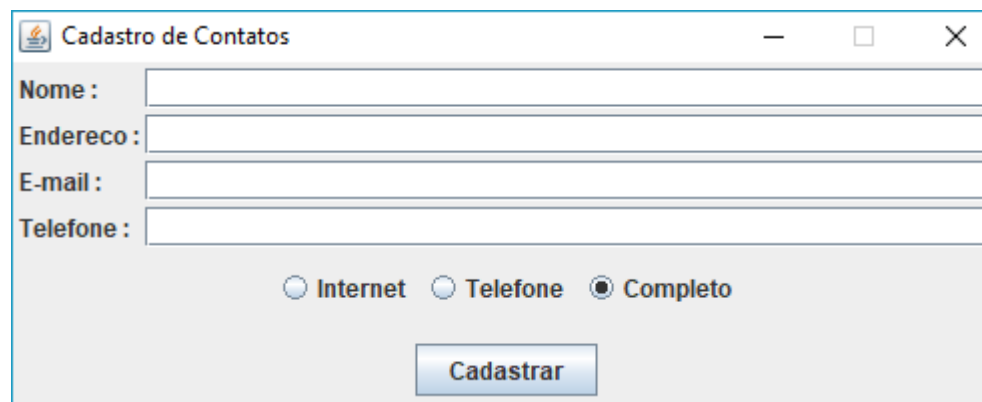


Exercício 1 (2) – builder2

- Cada banco tem uma forma diferente de imprimir o boleto e calcular o número digitável. Todos os boletos tem os mesmos dados.
- Considerando apenas os campos indicados com setas, aplique o padrão Builder para desenvolver boletos para os bancos Caixa e Bradesco.
 - Considere todas as informações como String (não estamos interessados na lógica do boleto !!!)
 - Caixa: o nome do cedente é armazenado em maiúsculas (toUpperCase())
 - Bradesco: a carteira sempre será 06
- Faltam os ConcreteBuilders
- Atualizar a classe Cliente

Exercício 2 – builder3

- Contatos são criados de acordo com um dos três tipos:
 - Internet: Nome e E-mail
 - Telefone: Nome e Telefone
 - Completo: Todos os campos



Cadastro de Contatos

Nome :

Endereco :

E-mail :

Telefone :

☐ Internet ☐ Telefone ☒ Completo

- Somente o cliente foi criado, porém existem instruções para guiá-lo no evento do botão Cadastrar

Exercício 3 – builder4

- Na cadeia de fast-food existe um padrão para montagem de lanches de crianças.
- Lanche 1 = Sanduíche: Hambúrguer, Batata: Pequena, Brinquedo: Carrinho
- Lanche 2 = Sanduíche: Hambúrguer, Batata: Pequena, Brinquedo: Bonequinha
- Lanche 3 = Sanduíche: Cheeseburguer, Batata: Média, Brinquedo: Carrinho
- Lanche 4 = Sanduíche: Cheeseburguer, Batata: Média, Brinquedo: Bonequinha

Imprimir a bandeja no final