

Introdução à Programação Orientada a Objetos com Java

Aplicação das Diretrizes Básicas para Programação

Paulo Borba
Centro de Informática
Universidade Federal de Pernambuco

Depois de aprender **como** programar...

Precisamos exercitar!



Quem vai usar o programa?

Considerando que o nosso cliente
é um banqueiro...

■ Pessoas

- Um cliente do banco via Internet
- Um cliente do banco via ATM
- Os funcionários do banco (gerente, caixas, etc.)

Quem vai usar o programa?

Considerando que o nosso cliente
é um banqueiro...

■ Sistemas

- Operadora de DOC (transferências de um banco para outro)
- A aplicação da operadora do cartão de débito direto
- Prestadoras de serviços que façam cobranças via débito automático

Que serviços precisam ser implementados?

■ Cliente Internet

- Transferência
- Envio de DOC
- Emissão de extrato

■ Cliente ATM

- Os mesmos do cliente Internet
- Saque
- Emissão de cheques avulsos

■ Funcionários

- Os mesmos que os clientes
- Cadastro de cliente
- Criação de conta
- Habilitar o acesso via Internet
- Fechamento de conta

Que serviços precisam ser implementados?

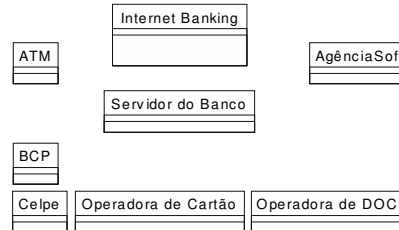
Temos que identificar os serviços
de interesse de cada um dos usuários,
pessoas ou sistemas

Que serviços precisam ser implementados?

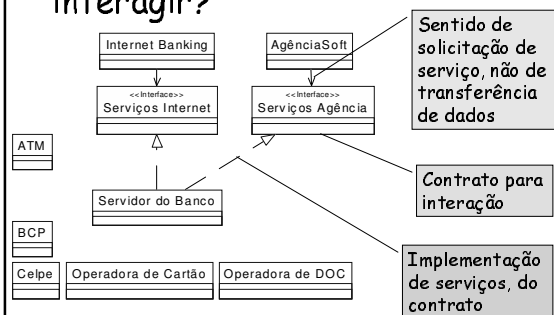
Todos os usuários têm
"login no sistema"
e "alteração de senha"

São considerados
também os serviços
auxiliares!

Que entidades vão
implementar os serviços?



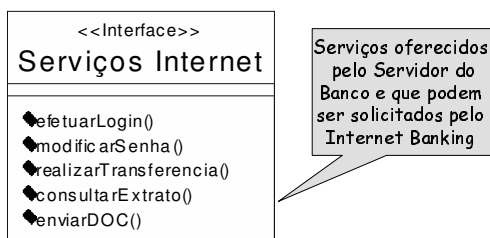
Como as entidades vão
interagir?



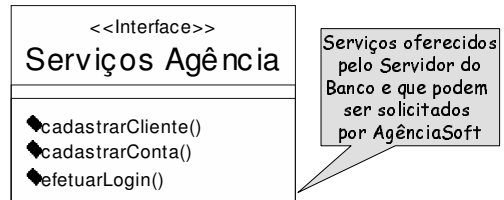
Como as entidades vão
interagir?

Faltou mostrar as **solicitações de serviços** entre os outros componentes, e em **sentidos diferentes...**

Como as entidades vão
interagir?



Como as entidades vão
interagir?



Como cada entidade realiza os seus serviços?

cadastrarCliente (do Servidor do Banco)

1. Verificar se as informações do cliente estão consistentes
2. Verificar se um cliente com o mesmo CPF ou CGC já foi cadastrado
3. Armazenar as informações do cliente no cadastro de clientes do banco

Seqüência de instruções

Versão 0.1

Como cada entidade realiza os seus serviços?

Depois de pensar na seqüência normal de execução das instruções, pensar nas seqüências excepcionais

Usar comandos condicionais: **Se**

cadastrarCliente, versão 0.2

1. Verificar se as informações do cliente estão consistentes
2. **Se** não estiverem, avise que o serviço não pode ser realizado e pare a execução
3. Verificar se um cliente com o mesmo CPF ou CGC já foi cadastrado
4. **Se** já existe um cliente cadastrado com o mesmo CPF ou CGC, avise que o serviço não pode ser realizado e pare a execução
5. Armazenar as informações do cliente no cadastro de clientes do banco

Como cada entidade realiza os seus serviços?

Normalmente é necessário detalhar melhor uma determinada instrução...

E repetir as mesmas instruções até que uma condição seja satisfeita

cadastrarCliente, versão 0.3

1. Verificar se as informações do cliente estão consistentes
2. **Se** não estiverem, avise que o serviço não pode ser realizado e pare a execução
3. Verificar se um cliente com o mesmo CPF ou CGC já foi cadastrado
4. **Se** já existe um cliente cadastrado com o mesmo CPF ou CGC, avise que o serviço não pode ser realizado e pare a execução
5. Armazenar as informações do cliente no cadastro de clientes do banco

O mesmo que na versão 0.2

3. Verificar se um cliente com o mesmo CPF ou CGC já foi cadastrado

Enquanto houver clientes a serem analisados

- a. Analise mais um cliente
- b. Compare o CPF ou CGC dele com o do cliente a ser cadastrado
- c. Se for diferente, continue; senão, avise que já existe um cliente com o mesmo CPF ou CGC e pare a execução desta seqüência

Detalhamento e agrupamento de instruções

Para um dado serviço devemos ter...

- uma seqüência principal de instruções de alto nível
- seqüências auxiliares de instruções que detalham cada uma das instruções da principal
- e recursivamente...

O processo de detalhamento pode se repetir

Novas entidades e serviços auxiliares...

Podem surgir quando detalhando como uma entidade realiza seus serviços

Na verdade, as necessidades dos clientes mudam, sempre...

- Novos tipos de usuários podem surgir
- Novos serviços podem ser necessários
- Modificações em serviços existentes podem ser necessárias

E as soluções também mudam, sempre...

Pode ser necessário ter

- Novas entidades
- Novas formas de interação entre as mesmas
- Novas formas de implementar serviços existentes

Um processo de desenvolvimento simplificado

1. Identificação dos requisitos
(Que usuários e serviços?)
2. Projeto e implementação dos requisitos
(Que entidades e iterações?
Como realizar cada serviço?)
3. Testar a implementação
(Os serviços estão funcionando?)
4. Voltar para o Item 2 caso o resultado dos testes não seja satisfatório

Na prática, o processo é iterativo...

1. Requisitos	1. Requisitos	...
2. Projeto	2. Projeto	
3. Implementação	3. Implementação	
4. Testes	4. Testes	
Iteração 1	Iteração 2	

Repetem-se as atividades, construindo o sistema incrementalmente... E permitindo mudanças nos requisitos e nas soluções.

O processo não é necessariamente top-down...

- Pode ser bottom-up
- Mas, na prática, é normalmente uma mistura dos dois estilos



As perguntas não têm que ser respondidas na ordem apresentada!

Resumindo...

- Para programar temos que fazer, iterativamente, as seguintes perguntas:
 - Quais são os clientes e usuários interessados?
 - Que serviços eles necessitam?
 - Que entidades vão, em conjunto, realizar estes serviços?
 - Como estas entidades vão interagir?
 - Como cada entidade realiza os seus serviços?
- A gente ainda está aprendendo a ser pedreiro, falta muito para engenheiro...

