

MBA em Engenharia de Software a Distância

**Disciplina: Análise e Projeto
Orientado a Objetos: UML**

MBA em Engenharia de Software a Distância

- **Modelagem de classes de projeto**
- O **modelo de classes de projeto** é resultante de refinamentos no modelo de classes de análise.
- Esse modelo é construído em paralelo com o **modelo de interações**.
- O modelo de classes de projeto contém detalhes úteis para a **implementação** das classes nele contidas.



MBA em Engenharia de Software a Distância

- Aspectos a serem considerados na fase de projeto para a modelagem de classes:
 - Estudo de novos elementos do diagrama de classes que são necessários à construção do modelo de projeto.
 - Descrever transformações pelas quais passam as classes e suas propriedades com o objetivo de transformar o modelo de classes análise no modelo de classes de projeto.
 - Adição de novas classes ao modelo



MBA em Engenharia de Software a Distância

- Especificação de atributos, operações e de associações
- Descrever refinamentos e conceitos relacionados à herança, que surgem durante a modelagem de classes de projeto
- Utilização de padrões de projeto (*design patterns*)



MBA em Engenharia de Software a Distância

- **Transformação de classes de análise em classes de projeto**
- **Especificação de classes de fronteira**
- Não devemos atribuir a essas classes responsabilidades relativas à lógica do negócio.
 - Classes de fronteira devem apenas servir como um ponto de captação de informações, ou de apresentação de informações que o sistema processou.



MBA em Engenharia de Software a Distância

- A única inteligência que essas classes devem ter é a que permite a elas realizarem a comunicação com o ambiente do sistema.
- Há diversas razões para isso:
 - Se o sistema tiver que ser implantado em outro ambiente, as modificações resultantes sobre seu funcionamento propriamente dito seriam mínimas.



MBA em Engenharia de Software a Distância

- O sistema pode dar suporte a diversas formas de interação com seu ambiente
- Essa separação resulta em uma melhor **coesão**.



MBA em Engenharia de Software a Distância

- Durante a análise, considera-se que há uma única classe de fronteira para cada ator. No projeto, algumas dessas classes podem resultar em várias outras.
- Interface com seres humanos: **projeto da interface gráfica** produz o detalhamento das classes.



MBA em Engenharia de Software a Distância

- Outros sistemas ou equipamentos: devemos definir uma ou mais classes para encapsular o protocolo de comunicação.
 - É usual a definição de um **subsistema** para representar a comunicação com outros sistemas de software ou com equipamentos.
 - É comum nesse caso o uso do padrão Façade (mais adiante)



MBA em Engenharia de Software a Distância

- O projeto de objetos de fronteira é altamente dependente da natureza do ambiente...
 - Clientes WEB clássicos
 - Clientes móveis
 - Clientes stand-alone
 - Serviços WEB (WEB services)



MBA em Engenharia de Software a Distância

- **Especificação de classes de entidade**
- A maioria das classes de entidade normalmente permanece na passagem da análise ao projeto.
- Durante o projeto, um aspecto importante a considerar sobre classes de entidade é identificar quais delas geram objetos que devem ser persistentes.
- Um aspecto importante é a forma de representar associações, agregações e composições entre objetos de entidade.



MBA em Engenharia de Software a Distância

- Outro aspecto relevante para classes de entidade é o modo como podemos identificar cada um de seus objetos unicamente.
 - Isso porque, principalmente em sistemas de informação, objetos de entidade devem ser armazenados de modo persistente.
- A manipulação dos diversos atributos identificadores possíveis em uma classes pode ser bastante trabalhosa.



MBA em Engenharia de Software a Distância

- Para evitar isso, um **identificador de implementação** é criado, que não tem correspondente com atributo algum do domínio.
 - Possibilidade de manipular identificadores de maneira uniforme e eficiente.
 - Maior facilidade quando objetos devem ser mapeados para um SGBDR



MBA em Engenharia de Software a Distância

- **Especificação de classes de controle**
- Com relação às classes de controle, no projeto devemos identificar a real utilidade das mesmas.
 - Em casos de uso simples, classes de controle não são realmente necessárias. Neste caso, classes de fronteira podem repassar os dados fornecidos pelos atores diretamente para as classes de entidade correspondentes.



MBA em Engenharia de Software a Distância

- Entretanto, é comum a situação em que uma classe de controle de análise ser transformada em duas ou mais classes no nível de especificação.
- No refinamento de qualquer classe proveniente da análise, é possível a aplicação de padrões de projeto.



MBA em Engenharia de Software a Distância

- Normalmente, cada classe de controle deve ser particionada em duas ou mais outras classes para controlar diversos aspectos da solução.
- Alguns exemplos dos aspectos de uma aplicação cuja coordenação é de responsabilidade das classes de controle.



MBA em Engenharia de Software a Distância

- Um tipo comum de controlador é o **controlador de caso de uso**, responsável pela coordenação da realização de um caso de uso.
- As seguintes responsabilidades são esperadas de um controlador de caso de uso:
 - Coordenar a realização de um caso de uso do sistema.
 - Servir como canal de comunicação entre objetos de fronteira e objetos de entidade.



MBA em Engenharia de Software a Distância

- Se comunicar com outros controladores, quando necessário.
- Mapear ações do usuário (ou atores de uma forma geral) para atualizações ou mensagens a serem enviadas a objetos de entidade.
- Estar apto a manipular exceções provenientes das classes de entidades.



MBA em Engenharia de Software a Distância

- Em aplicações WEB, é comum a prática de utilizar outro tipo de objeto controlador chamado ***front controller*** (FC).
- Um FC é um controlador responsável por receber todas as requisições de um cliente.
- O FC identifica qual o controlador (de caso de uso) adequado para processar a requisição, e a despacha para ele.



MBA em Engenharia de Software a Distância

- Sendo assim, um FC é um ponto central de entrada para as funcionalidades do sistema.
 - Vantagem: mais fácil controlar a autenticação dos usuários.
- O FC é um dos *padrões de projeto* do catálogo J2EE



MBA em Engenharia de Software a Distância

- **Especificação de outras classes**
- Além do refinamento de classes preexistentes, diversas outros aspectos demanda a identificação de novas classe durante o projeto.
 - Persistência de objetos
 - Distribuição e comunicação
 - Autenticação/Autorização
 - Logging
 - Configurações
 - Threads



MBA em Engenharia de Software a Distância

- Classes para testes
- Uso de **bibliotecas, componentes e frameworks**
- Conclusão: a tarefa de identificação de classes não termina na análise.



MBA em Engenharia de Software a Distância

- **Especificação de atributos**
- **Especificação de operações**
- **Refinamento de Atributos e Métodos**
- Os atributos e métodos de uma classe a habilita a cumprir com suas **responsabilidades**.
- **Atributos:** permitem que uma classe armazene informações necessárias à realização de suas tarefas.



MBA em Engenharia de Software a Distância

- **Métodos:** são funções que manipulam os valores do atributos, com o objetivo de atender às **mensagens** que o objeto recebe.



MBA em Engenharia de Software a Distância

- **Visibilidade e Encapsulamento**
- Os três qualificadores de visibilidade aplicáveis a atributos também podem ser aplicados a operações.
 - + representa visibilidade pública
 - # representa visibilidade protegida
 - representa visibilidade privativa



MBA em Engenharia de Software a Distância

- O real significado desses qualificadores depende da linguagem de programação em questão.
- Usualmente, o conjunto das operações públicas de uma classe são chamadas de **interface** dessa classe.
 - ***Note que há diversos significados para o termo interface.***



MBA em Engenharia de Software a Distância

- **Membros estáticos**
- Membros estáticos são representados no diagrama de classes por declarações sublinhadas.
 - **Atributos estáticos** (variáveis de classe) são aqueles cujos valores valem para a classe de objetos como um todo.
 - Diferentemente de atributos não-estáticos (ou variáveis de instância), cujos valores são particulares a cada objeto.



MBA em Engenharia de Software a Distância

- **Métodos estáticos** são os que não precisam da existência de uma instância da classe a qual pertencem para serem executados.
 - Forma de chamada:
NomeClasse.Método(argumentos)



MBA em Engenharia de Software a Distância

- **Projeto de métodos**
- Métodos de construção (criação) e destruição de objetos
- Métodos de acesso (getX/setX) ou propriedades
- Métodos para manutenção de associações (conexões) entre objetos.
- Outros métodos:
 - Valores derivados, formatação, conversão, cópia e clonagem de objetos, etc.



MBA em Engenharia de Software a Distância

- Alguns métodos devem ter uma operação inversa óbvia
 - e.g., habilitar e desabilitar; tornarVisível e tornarInvisível; adicionar e remover; depositar e sacar, etc.
- Operações para desfazer ações anteriores.
 - e.g., padrões de projeto GoF: Memento e Command



MBA em Engenharia de Software a Distância

```
public class Turma {  
    private Set<OfertaDisciplina> ofertasDisciplina = new HashSet();  
  
    public Turma() {  
    }  
  
    public void adicionarOferta(OfertaDisciplina oferta) {  
        this.ofertasDisciplina.add(oferta);  
    }  
  
    public boolean removerOferta(OfertaDisciplina oferta) {  
        return this.ofertasDisciplina.remove(oferta);  
    }  
  
    public Set getOfertasDisciplina() {  
        return  
        Collections.unmodifiableSet(this.ofertasDisciplina);  
    }  
}
```



MBA em Engenharia de Software a Distância

- **Detalhamento de métodos**
- **Diagramas de interação** fornecem um indicativo sobre como métodos devem ser implementados.
- Como complemento, **notas explicativas** também são úteis no esclarecimento de como um método deve ser implementado.
- O **diagrama de atividades** também pode ser usado para detalhar a lógica de funcionamento de métodos mais complexos.




MBA em Engenharia de Software a Distância

- **Especificação de associações**
- O conceito de dependência
- O ***relacionamento de dependência*** indica que uma classe depende dos serviços (operações) fornecidos por outra classe.
- Na análise, utilizamos apenas a ***dependência por atributo*** (ou estrutural), na qual a classe dependente possui um atributo que é uma referência para a outra classe.



MBA em Engenharia de Software a Distância

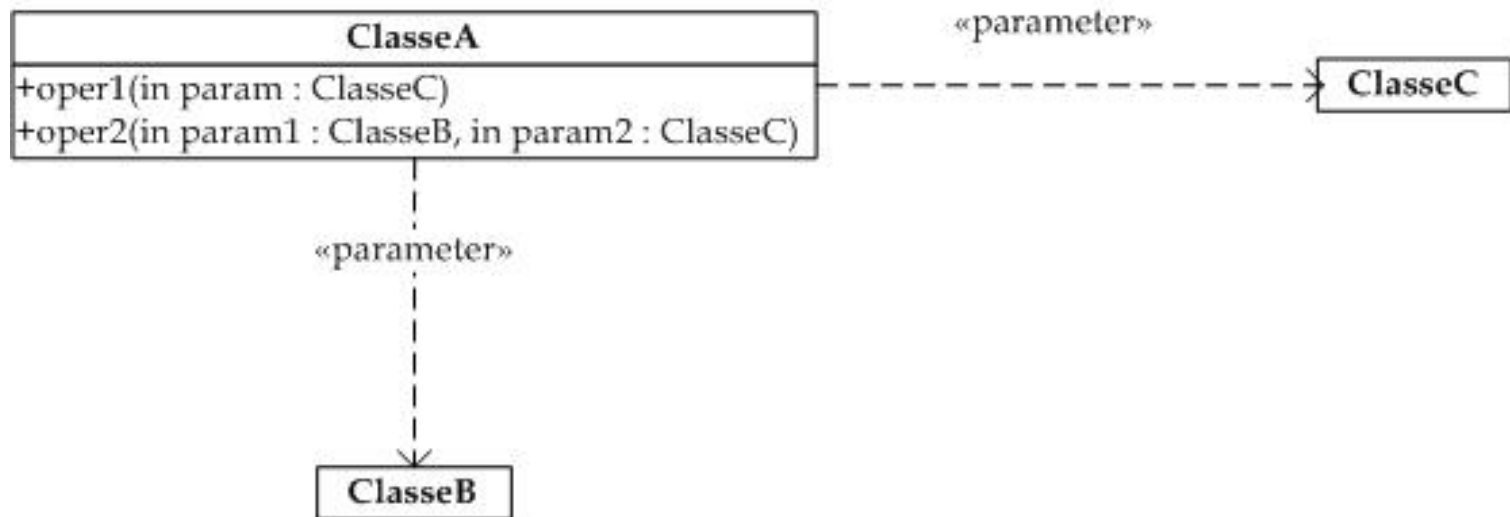
- Entretanto, há também as ***dependências não estruturais***:
 - Na **dependência por variável global**, um objeto de escopo global é referenciado em algum método da classe dependente.
 - Na **dependência por variável local**, um objeto recebe outro como retorno de um método, ou possui uma referência para o outro objeto como uma variável local em algum método.
 - Na **dependência por parâmetro**, um objeto recebe outro como parâmetro em um método. 

MBA em Engenharia de Software a Distância

- Dependências não estruturais são representadas na UML por uma linha tracejada direcionada e ligando as classes envolvidas.
 - A direção é da classe dependente (**cliente**) para a classe da qual ela depende (**fornecedora**).
 - Estereótipos predefinidos: <<global>>, << local>>, << parameter>>.



MBA em Engenharia de Software a Distância



MBA em Engenharia de Software a Distância

- **De associações para dependências**
- Durante o projeto de classes, é necessário avaliar, para cada associação existente, se é possível transformá-la em uma dependência não estrutural.
- Objetivo: aumentar o encapsulamento de cada classe e diminuir o acoplamento entre as classes.



MBA em Engenharia de Software a Distância

- A dependência por atributo é a forma mais forte de dependência.
- Quanto menos dependências por atributo houver no modelo de classes, maior é o encapsulamento e menor o acoplamento.



MBA em Engenharia de Software a Distância

- **Navegabilidade de associações**
- Associações podem ser **bidirecionais** ou **unidirecionais**.
 - Uma **associação bidirecional** indica que há um conhecimento mútuo entre os objetos associados.
 - Uma **associação unidirecional** indica que apenas um dos extremos da associação tem ciência da existência da mesma.



MBA em Engenharia de Software a Distância

- A escolha da navegabilidade de uma associação pode ser feita através do estudo dos **diagramas de interação**.
 - O sentido de envio das mensagens entre objetos influencia na necessidade ou não de navegabilidade em cada um dos sentidos.



MBA em Engenharia de Software a Distância

- **Implementação de associações**
- Há três casos, em função da conectividade: 1:1, 1:N e N:M
- Para uma associação 1:1 entre duas classes A e B:
 - Se a navegabilidade é unidirecional no sentido de A para B, é definido um atributo do tipo B na classe A.
 - Se a navegabilidade é bidirecional, podemos aplicar o procedimento acima para as duas classes.



MBA em Engenharia de Software a Distância

- Para uma associação 1:N ou N:M entre duas classes A e B:
 - São utilizados atributos cujos tipos representam coleções de elementos.
 - É também comum o uso de classes parametrizadas.
 - Idéia básica: definir uma classe parametrizada cujo parâmetro é a classe correspondente ao lado *muitos* da associação.



MBA em Engenharia de Software a Distância

- **Classe Parametrizada**
- Uma coleção pode ser representada em um diagrama de classes através uma **classe parametrizada**.
 - Def.: é uma classe utilizada para definir outras classes.
 - Possui operações ou atributos cuja definição é feita em função de um ou mais parâmetros.



MBA em Engenharia de Software a Distância

- Uma coleção pode ser definida a partir de uma classe parametrizada, onde o parâmetro é o tipo do elemento da coleção.



MBA em Engenharia de Software a Distância

- **Herança**
- Na modelagem de classes de projeto, há diversos aspectos relacionados ao de ***relacionamento de herança***.
 - Tipos de herança
 - Classes abstratas
 - Operações abstratas
 - Operações polimórficas
 - Interfaces



MBA em Engenharia de Software a Distância

- Acoplamentos concreto e abstrato
- Reuso através de delegação e através de generalização
- Classificação dinâmica



MBA em Engenharia de Software a Distância

- **Tipos de herança**
- Com relação à quantidade de superclasses que certa classe pode ter.
 - *herança múltipla*
 - *herança simples*
- Com relação à forma de reutilização envolvida.
 - Na **herança de implementação**, uma classe reusa alguma implementação de um “ancestral”.



MBA em Engenharia de Software a Distância

- Na **herança de interface**, uma classe reusa a interface (conjunto das assinaturas de operações) de um “ancestral” e se compromete a implementar essa interface.



MBA em Engenharia de Software a Distância

- **Classes abstratas**
- Usualmente, a existência de uma classe se justifica pelo fato de haver a possibilidade de gerar instâncias a partir da mesma.
 - Essas classes são chamadas de **classes concretas**.
- No entanto, podem existir classes que não geram instâncias “diretamente”.
 - Essas classes são chamadas de **classes abstratas**.



MBA em Engenharia de Software a Distância

- Classes abstratas são usadas para organizar hierarquias.
 - Propriedades comuns a diversas classes podem ser organizadas e definidas em uma classe abstrata a partir da qual as primeiras herdam.
- Também propiciam a implementação do **princípio do polimorfismo.**



MBA em Engenharia de Software a Distância

- **Operações abstratas**
- Uma classe abstrata possui ao menos uma operação abstrata, que corresponde à especificação de um serviço que a classe deve fornecer.
- Uma classe qualquer pode possuir tanto operações abstratas, quanto operações concretas (ou seja, operações que possuem implementação).



MBA em Engenharia de Software a Distância

- Entretanto, uma classe que possui pelo menos uma operação abstrata é, por definição abstrata.
- Uma operação abstrata definida com visibilidade pública em uma classe também é herdada por suas subclasses.
- Quando uma subclasse herda uma operação abstrata e não fornece uma implementação para a mesma, esta classe também é abstrata.



MBA em Engenharia de Software a Distância

- **Operações polimórficas**
- Uma subclasse herda todas as propriedades de sua superclasse que tenham visibilidade pública ou protegida.
- Entretanto, pode ser que o comportamento de alguma operação herdada seja diferente para a subclasse.



MBA em Engenharia de Software a Distância

- Nesse caso, a subclasse deve redefinir o comportamento da operação.
 - A assinatura da operação é reutilizada.
 - Mas, a implementação da operação (ou seja, seu **método**) é diferente.
- Operações polimórficas são aquelas que possuem mais de uma implementação.



MBA em Engenharia de Software a Distância

- Operações polimórficas possuem sua assinatura definida em diversos níveis de uma hierarquia gen/spec.
 - A assinatura é repetida na(s) subclasse(s) para enfatizar a redefinição de implementação.
 - O objetivo de manter a assinatura é garantir que as subclasses tenham uma interface em comum.

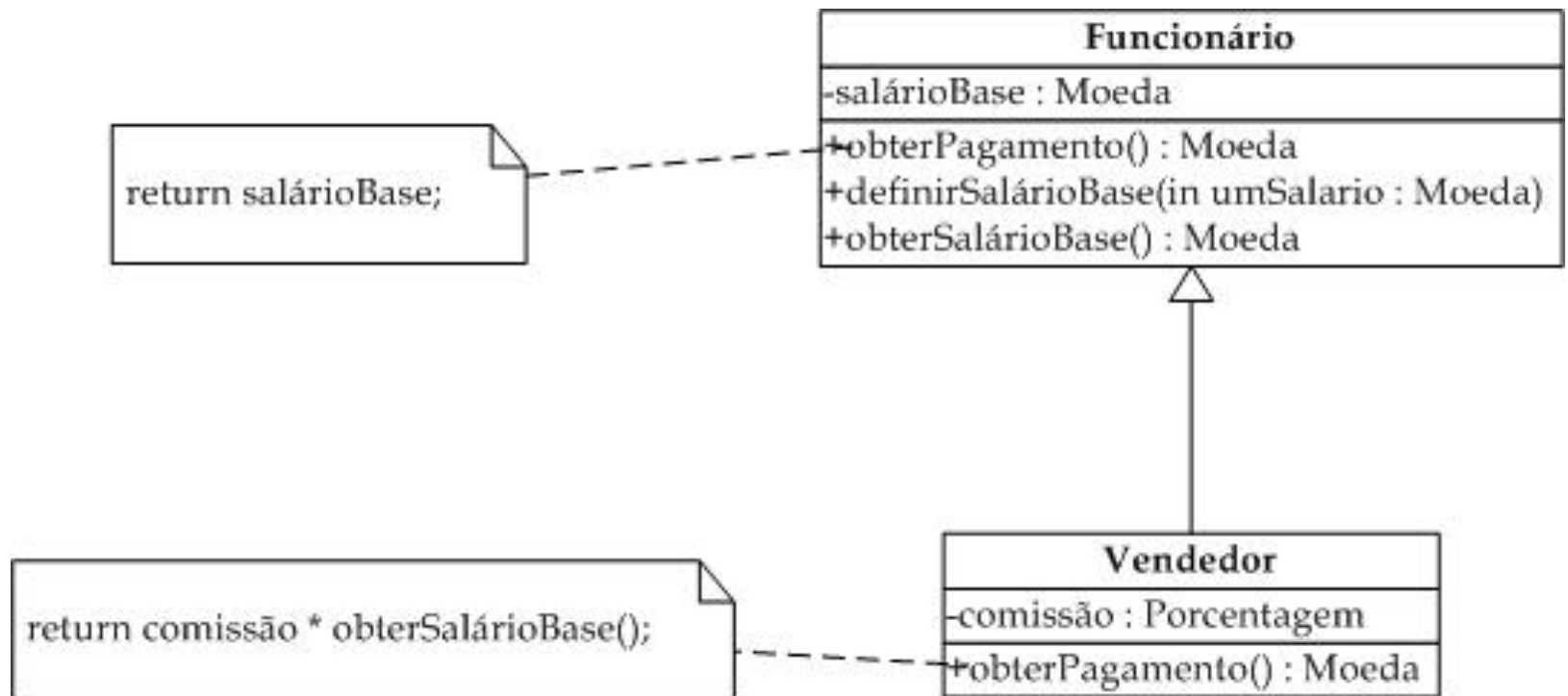


MBA em Engenharia de Software a Distância

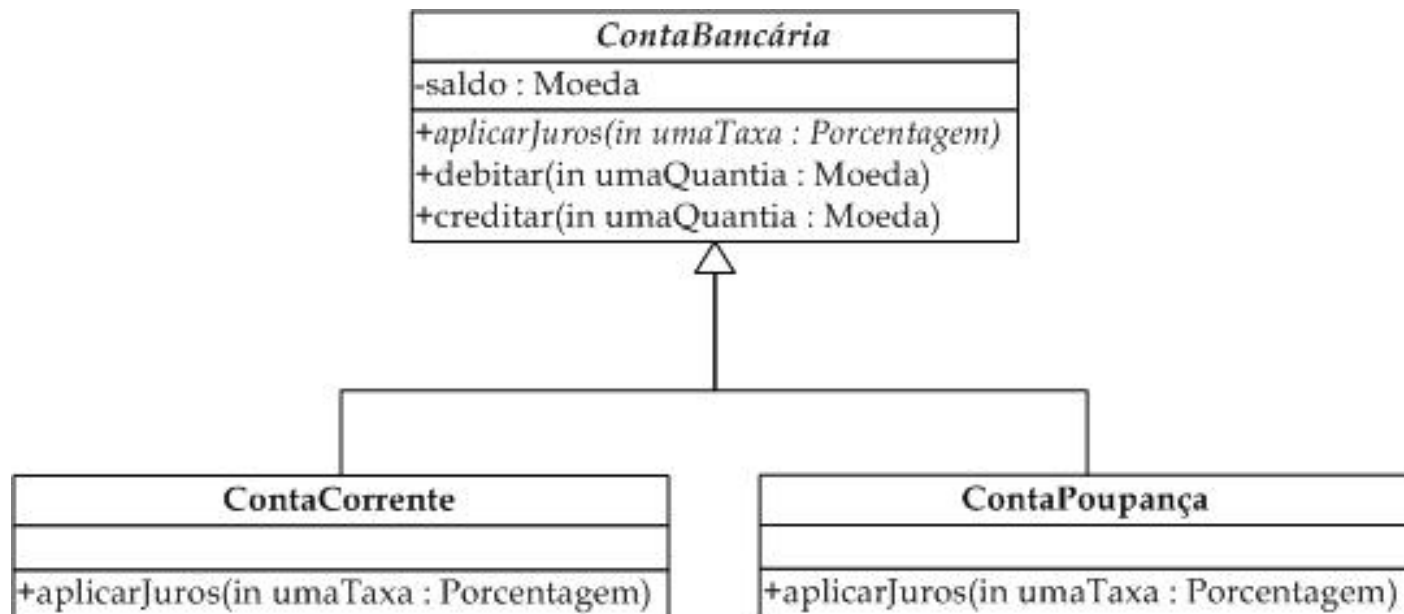
- Operações polimórficas facilitam a implementação.
 - Se duas ou mais subclasses implementam uma operação polimórfica, a mensagem para ativar essa operação é a mesma para todas essas classes.
 - No envio da mensagem, o remetente não precisa saber qual a verdadeira classe de cada objeto, pois eles aceitam a mesma mensagem.
 - A diferença é que os métodos da operação são diferentes em cada subclasse.



MBA em Engenharia de Software a Distância



MBA em Engenharia de Software a Distância



MBA em Engenharia de Software a Distância

- **Interfaces**
- Uma **interface** entre dois objetos compreende um conjunto de **assinaturas de operações** correspondentes aos serviços dos quais a classe do objeto cliente faz uso.
- Uma interface pode ser interpretada como um **contrato de comportamento** entre um objeto cliente e eventuais objetos fornecedores de um determinado serviço.



MBA em Engenharia de Software a Distância

- Interfaces são utilizadas com os seguintes objetivos:
 - 1. Capturar semelhanças entre classes não relacionadas sem forçar relacionamentos entre elas.
 - 2. Declarar operações que uma ou mais classes devem implementar.
 - 3. Revelar as operações de um objeto, sem revelar a sua classe.

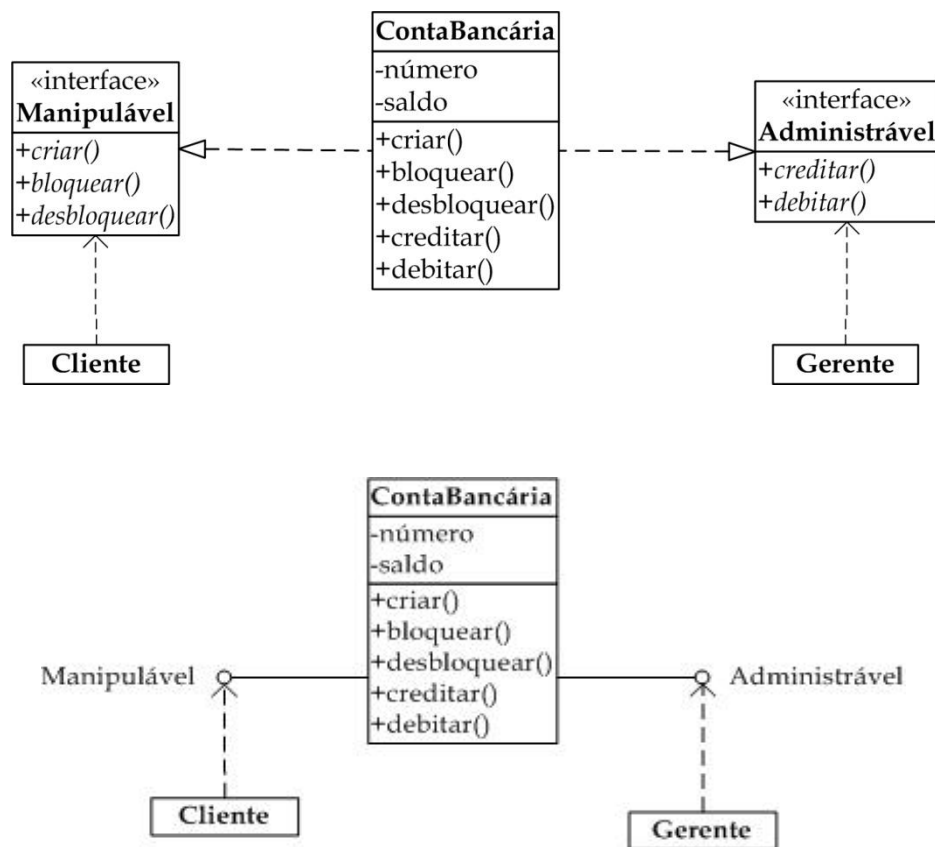


MBA em Engenharia de Software a Distância

- 4. Facilitar o desacoplamento entre elementos de um sistema.
- Nas LPOO modernas (Java, C#, etc.), interfaces são definidas de forma semelhante a classes.
 - Uma diferença é que todas as declarações em uma interface têm visibilidade pública.
 - Adicionalmente, uma interface não possui atributos, somente declarações de assinaturas de operações e (raramente) constantes.



MBA em Engenharia de Software a Distância



MBA em Engenharia de Software a Distância

- **Acoplamentos concreto e abstrato**
- Usualmente, um objeto A faz referência a outro B através do conhecimento da classe de B.
 - Esse tipo de dependência corresponde ao que chamamos de ***acoplamento concreto***.
- Entretanto, há outra forma de dependência que permite que um objeto remetente envie uma mensagem para um receptor sem ter conhecimento da verdadeira classe desse último.

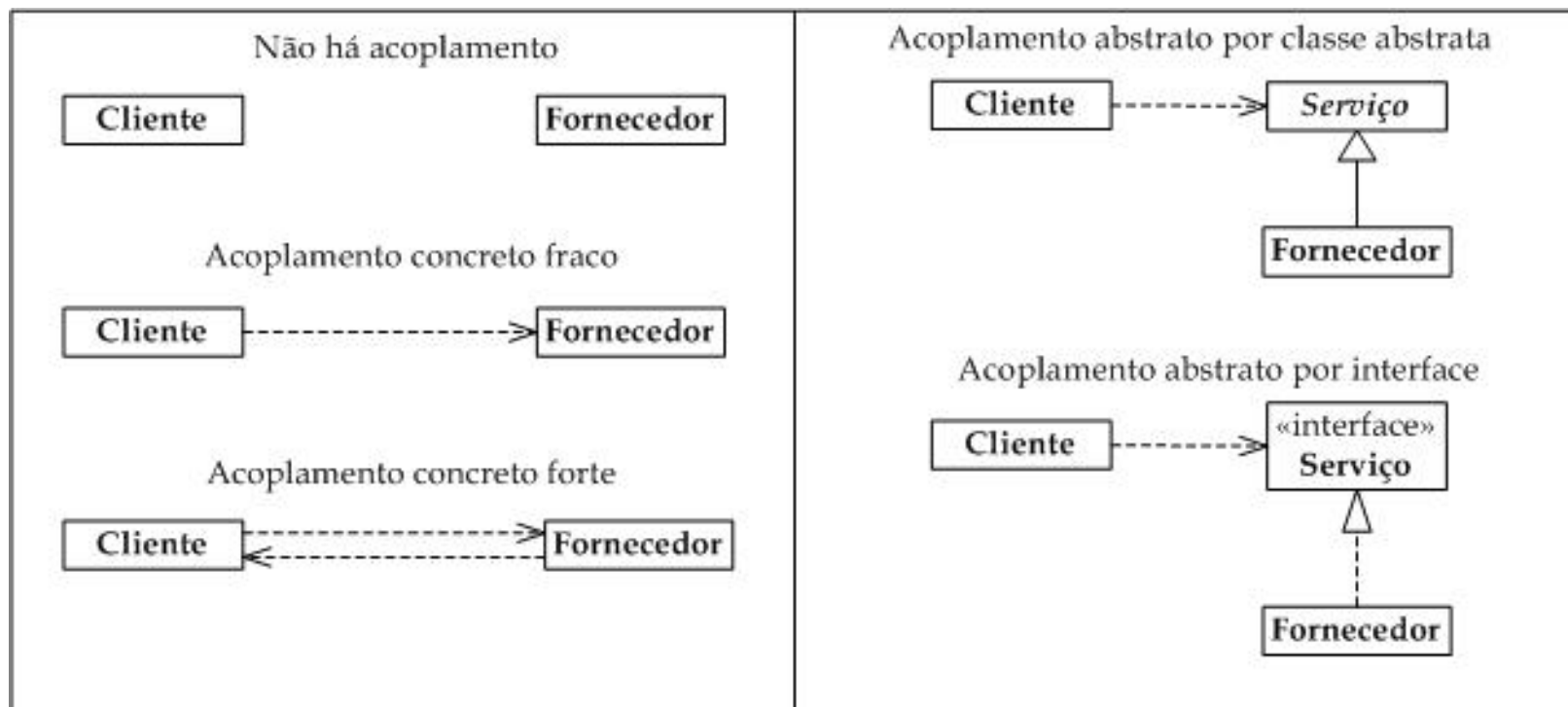


MBA em Engenharia de Software a Distância

- Essa forma de dependência corresponde ao que chamamos de **acoplamento abstrato**.
- O acoplamento abstrato é preferível ao acoplamento concreto.
- Classes abstratas e interface permitem implementar o acoplamento abstrato.



MBA em Engenharia de Software a Distância



MBA em Engenharia de Software a Distância

- **Reuso através de generalização**
- No reuso por generalização, subclasses que herdam comportamento da superclasse.
 - Exemplo: um objeto ContaCorrente não tem como atender à mensagem para executar a operação debitar só com os recursos de sua classe. Ele, então, utiliza a operação herdada da superclasse.
- Vantagem: fácil de implementar.



MBA em Engenharia de Software a Distância

- Desvantagem:
 - Exposição dos detalhes da superclasse às subclasses (Violação do *princípio do encapsulamento*).
 - Possível violação do *Princípio de Liskov* (regra da substituição).



MBA em Engenharia de Software a Distância

- **Reuso através de delegação**
- A delegação é outra forma de realizar o reuso.
- “Sempre que um objeto não pode realizar uma operação por si próprio, ele delega uma parte dela para outro(s) objeto(s)”.
- A delegação é mais genérica que a generalização.
 - um objeto pode reutilizar o comportamento de outro sem que o primeiro precise ser uma subclasse do segundo.



MBA em Engenharia de Software a Distância

- O compartilhamento de comportamento e o reuso podem ser realizados em tempo de execução.
- Desvantagens:
 - desempenho (implica em cruzar a fronteira de um objeto a outro para enviar uma mensagem).
 - não pode ser utilizada quando uma classe parcialmente abstrata está envolvida.



MBA em Engenharia de Software a Distância

- Há vantagens e desvantagens tanto na generalização quanto na delegação.
- De forma geral, não é recomendado utilizar generalização nas seguintes situações:
 - Para representar papéis de uma superclasse.
 - Quando a subclasse herda propriedades que não se aplicam a ela.
 - Quando um objeto de uma subclasse pode se transformar em um objeto de outra subclasse💡

MBA em Engenharia de Software a Distância

- **Classificação dinâmica**
- Problema na especificação e implementação de uma generalização.
 - Um mesmo objeto pode pertencer a múltiplas classes simultaneamente, ou passar de uma classe para outra.
- Considere uma empresa em que há empregados e clientes.
 - Pode ser que uma pessoa, em um determinado momento, seja apenas cliente;



MBA em Engenharia de Software a Distância

- depois pode ser que ela passe a ser também um empregado da empresa.
- A seguir essa pessoa é desligada da empresa, continuando a ser cliente.
- As principais LPOO (C++, Java, Smalltalk) não dão suporte direto à implementação da classificação dinâmica.
 - se um objeto é instanciado como sendo de uma classe, ele não pode pertencer posteriormente a uma outra classe.

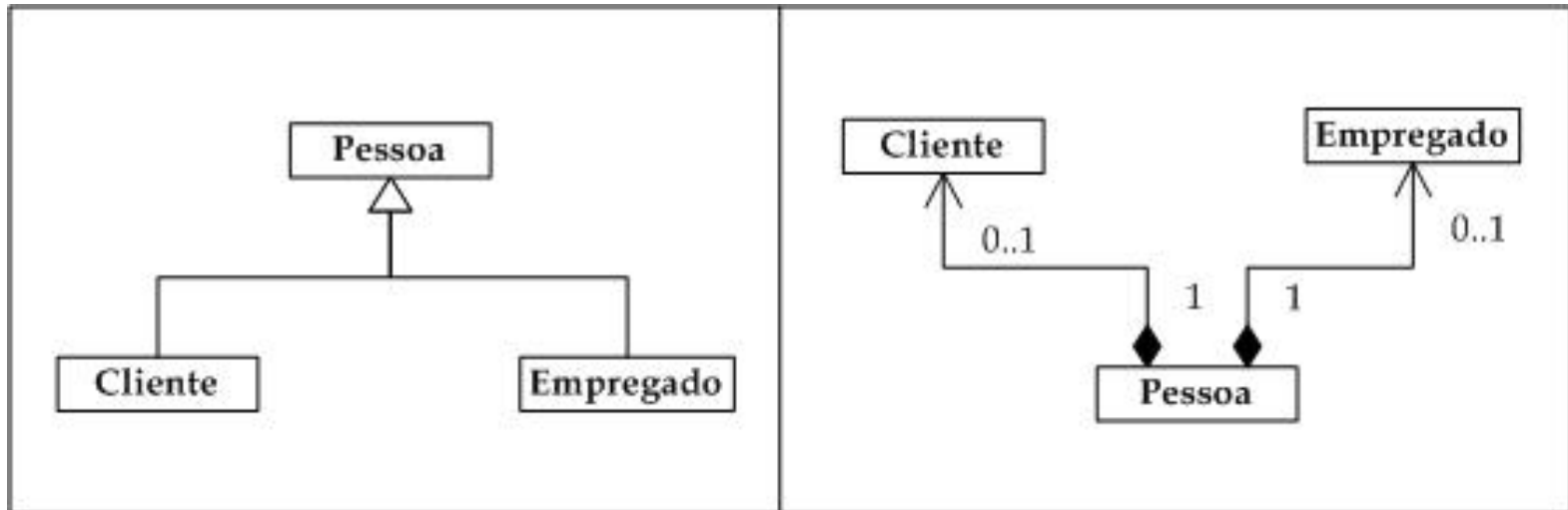


MBA em Engenharia de Software a Distância

- Solução parcial: definir todas as possíveis subclasses em uma determinada situação.
- Não resolve o problema todo:
 - Pode ser que um objeto mude de classe! (*metamorfose*)
 - A adição de novas classes à hierarquia torna o modelo ainda mais complexo.
- Uma melhor solução: utilizar a delegação.
 - Uma generalização entre cada subclasse e a superclasse é substituída por uma composição.



MBA em Engenharia de Software a Distância



MBA em Engenharia de Software a Distância

- **Padrões de projeto**
- É da natureza do desenvolvimento de software o fato de que os mesmos problemas tendem a acontecer diversas vezes.
- Um ***padrão de projeto*** corresponde a um esboço de uma solução reusável para um problema comumente encontrado em um contexto particular.

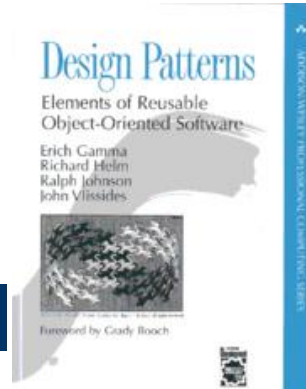


MBA em Engenharia de Software a Distância

- Estudar esses padrões é uma maneira efetiva de aprender com a experiência de outros.
- O texto clássico sobre o assunto é o de Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides.
 - Esses autores são conhecidos *Gang of Four*.
 - Nesse livro, os autores catalogaram 23 padrões.



MBA em Engenharia de Software a Distância



- **Padrões GoF**
- Os padrões GoF foram divididos em três categorias:
 1. ***Criacionais:***
 2. ***Estruturais:***
 3. ***Comportamentais:***



MBA em Engenharia de Software a Distância

Criacionais	Estruturais	Comportamentais
Abstract Factory Builder Factory Method Prototype Singleton	Adapter Bridge Composite Decorator Façade Flyweight Proxy	Chain of Responsibility Command Interpreter Iterator Mediator Memento Observer State Strategy Template Method Visitor



MBA em Engenharia de Software a Distância

- **Padrões de Criação**
- **Abstract Factory** - Provê uma interface para criar famílias de objetos relacionados ou interdependentes sem especificar suas classes concretas.
- **Builder** - Separa a construção de um objeto complexo da sua representação de forma que o mesmo processo de construção possa criar representações diferentes.



MBA em Engenharia de Software a Distância

- **Factory Method** - Define uma interface para criar um objeto, mas deixa as subclasses decidirem qual classe instanciar. O padrão *Factory Method* deixa uma classe repassar a responsabilidade de instânciação para subclasses.
- **Prototype** - Especifica os tipos de objetos a criar usando uma instância-protótipo e cria novos objetos copiando este protótipo.



MBA em Engenharia de Software a Distância

- **Singleton** - Garante que uma classe tenha uma única instância e provê um ponto global de acesso à instância.



MBA em Engenharia de Software a Distância

- **Padrões Estruturais**
- **Adapter** - Converte a interface de uma classe em outra interface com a qual os clientes estão prontos para lidar. O padrão *Adapter* permite que classes trabalhem conjuntamente apesar de interfaces incompatíveis.
- **Bridge** - Desacopla uma abstração de sua implementação de forma que as duas possam mudar independentemente uma da outra.



MBA em Engenharia de Software a Distância

- **Composite** - Compõe objetos em estruturas de árvore para representar hierarquias do tipo Parte-Todo. O padrão *Composite* permite que clientes tratem objetos individuais e composições de objetos de maneira uniforme.
- **Decorator** - Adiciona responsabilidades a um objeto dinamicamente. Decoradores provêem uma alternativa flexível à herança para estender funcionalidade.



MBA em Engenharia de Software a Distância

- **Façade** - Provê uma interface unificada para um conjunto de interfaces num subsistema. Define uma interface de mais alto nível, deixando o subsistema mais fácil de usar.
- **Flyweight** - Usa o compartilhamento para dar suporte eficiente ao uso de um grande número de objetos de granularidade pequena.
- **Proxy** - Provê um objeto procurador ou *placeholder* para outro objeto para controlar o acesso a ele.



MBA em Engenharia de Software a Distância

- **Padrões de Comportamento**
- **Chain of Responsibility** - Evita acoplar o remetente de um pedido ao receptor dando oportunidade a vários objetos para tratarem do pedido. Os receptores são encadeados e o pedido é passado na cadeia até que um objeto o trate.
- **Command** - Encapsula um pedido num objeto, permitindo assim parametrizar clientes com pedidos diferentes, enfileirar pedidos, fazer log de pedidos, e dar suporte a operações de undo.



MBA em Engenharia de Software a Distância

- **Interpreter** - Dada uma linguagem, define uma representação de sua gramática e um interpretador que usa a representação da gramática para interpretar sentenças da linguagem.
- **Iterator** - Provê uma forma de acessar os elementos de uma coleção de objetos sequencialmente sem expor sua representação subjacente.



MBA em Engenharia de Software a Distância

- **Mediator** - Define um objeto que encapsule a forma com a qual um conjunto de objetos interagem. O padrão *Mediator* promove o acoplamento fraco evitando que objetos referenciem uns aos outros explicitamente e permite que suas interações variem independentemente.
- **Memento** - Sem violar o princípio de encapsulamento, captura e externaliza o estado interno de um objeto de forma a poder restaurar o objeto a este estado mais tarde.



MBA em Engenharia de Software a Distância

- **Observer** - Define uma dependência um-para-muitos entre objetos de forma a avisar e atualizar vários objetos quando o estado de um objeto muda.
- **State** - Permite que um objeto altere seu comportamento quando seu estado interno muda. O objeto estará aparentemente mudando de classe com a mudança de estado.



MBA em Engenharia de Software a Distância

- **Strategy** - Define uma família de algoritmos, encapsula cada um, e os deixa intercambiáveis permitindo que o algoritmo varie independentemente dos clientes que o usem.
- **Template Method** - Define o esqueleto de um algoritmo numa operação, deixando que as subclasses completem algumas das etapas do algoritmo sem alterar a sua estrutura.



MBA em Engenharia de Software a Distância

- **Visitor** – Representa uma operação a ser realizada nos elementos de uma estrutura de objetos. O padrão *Visitor* permite que se defina uma nova operação sem alterar as classes dos elementos nos quais a operação age.

