

Frameworks

Bruna Diirr

brunadiirr@ic.uff.br

Introdução

Programar não é trivial...

Estratégia: Lançar mão de formas de reuso que vão além de código → reuso de análise, design, código

Framework orientado a objeto

O que é um framework?

“Aplicação semicompleta reutilizável que, quando especializada, produz aplicações personalizadas” (Johnson & Foote, 1988)

É um conjunto de classes cooperantes que constroem um projeto reutilizável para uma determinada categoria de software

- um framework orientado a construção de editores gráficos para diferentes domínios

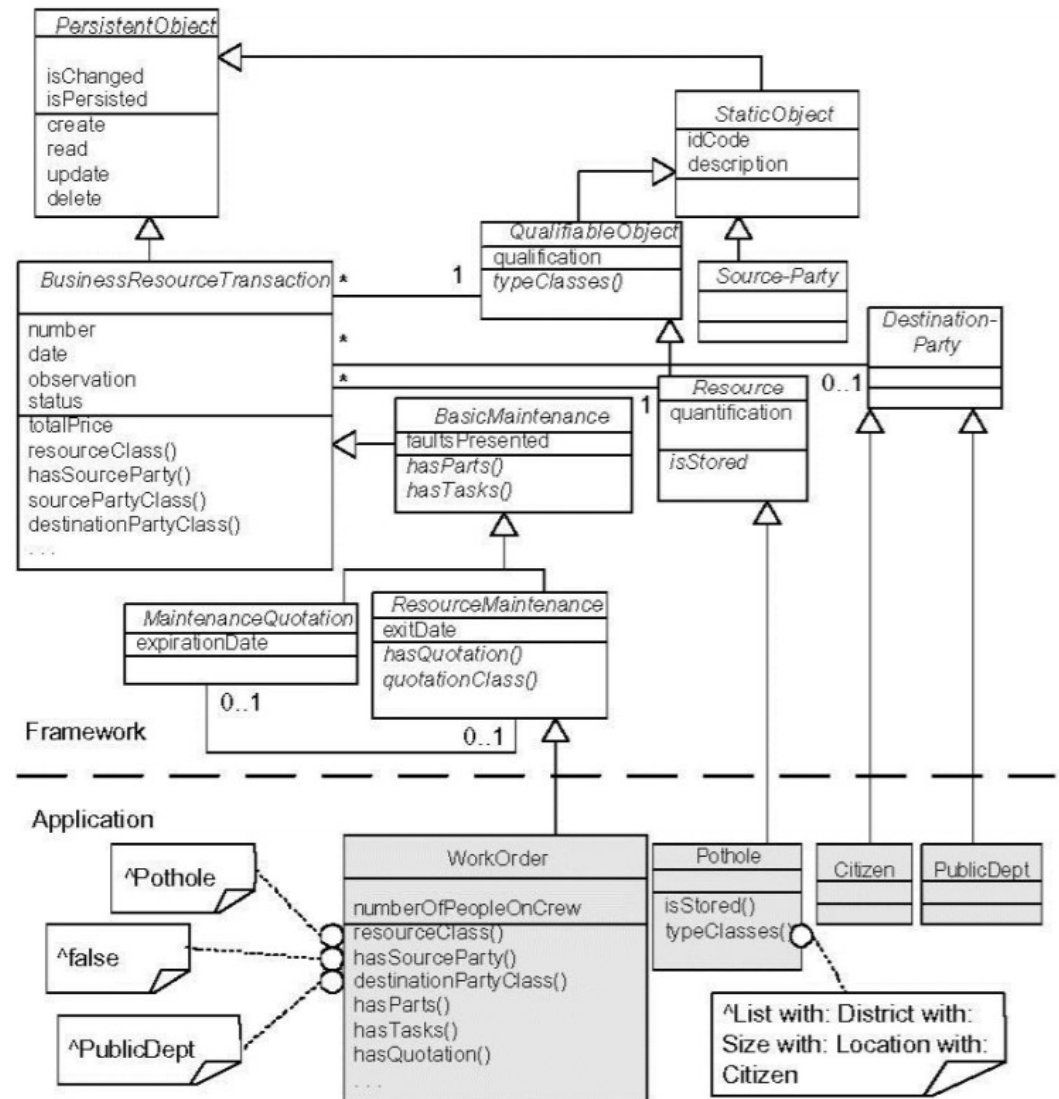
- outro ajuda a produzir compiladores para diferentes linguagens de programação

O que é um framework?

Framework é uma aplicação QUASE completa

Seu trabalho consiste em prover os pedaços faltantes, que são específicos para sua aplicação

Criação de subclasses específicas para a aplicação, derivadas das classes abstratas do framework



O que é um framework?

Um framework

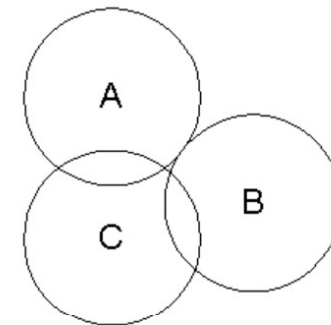
Provê solução para uma família de problemas semelhantes

Usando classes e interfaces que mostram como decompor essa família de problemas

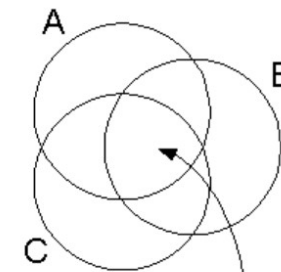
E como objetos dessas classes colaboram para cumprir suas responsabilidades

Devendo ser flexível e extensível para permitir a construção de várias aplicações com pouco esforço

Especificando apenas as particularidades de cada aplicação



Impossível criar Framework



Interseção grande
Possível criar Framework

Frameworks x Biblioteca de classes

Biblioteca de classes

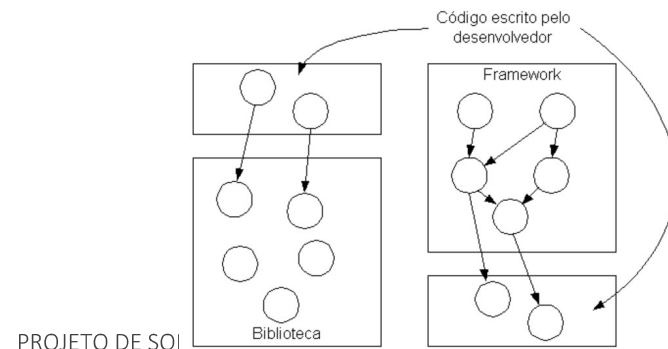
Cada classe é única e independente das outras

Comunicação entre objetos não está definida → Cliente responsável por chamar cada método

Não se pode embutir conhecimento do domínio

Não tem fluxo de controle pré-definido

Não possui comportamento default



Framework

Dependências/colaborações estão embutidas

Comunicação entre objetos já está definida → Framework responsável por chamar cada método

Hollywood principle: "Don't call us, we'll call you"

Pode embutir conhecimento do domínio

Controla fluxo de controle

Possui comportamento default

Frameworks x Padrões

Aparentemente, ambos consistem de classes, interfaces e colaborações prontas, mas...

Padrões

Menores

Formados por duas ou três classes

Mais abstratos

Representam conhecimento e experiência sobre software

Possuem natureza lógica

Devem ser estudados e programados a cada uso

Menos especializados

Podem ser usados em diversos tipos de aplicação, não ditando uma arquitetura particular

Framework

Maiores

Formados por muitas classes

Frameworks contêm padrões, mas não o inverso

Menos abstratos

É um software executável

Possuem natureza física

Programados para ser executados e reusados diretamente

Mais especializados

Em geral, são desenvolvidos para um domínio de aplicação específico

Frameworks x Aplicação

O desenvolvimento de uma aplicação pode ser feito com base em mais de um framework.

Por exemplo, no desenvolvimento de uma ferramenta CASE:

- o framework “Model-View-Controller” pode ser usado para desenvolver a interface; e

- o framework “Hotdraw” para desenvolver a parte de elaboração de diagramas

Características

Deve ser reusável

Propósito final!

Para isso, deve primeiro
ser *usável*

Bem documentado

Fácil de usar

Deve ser extensível

Contém funcionalidade abstrata
(sem implementação) que deve
ser completada

Deve ser de uso seguro

O desenvolvedor de aplicações
não pode destruir o framework

Deve ser eficiente

Devido a uso em muitas
situações, algumas delas podem
necessitar de eficiência

Deve ser completo

Para endereçar o domínio do
problema pretendido

Frameworks

Vantagens

Aproveitamento da experiência de especialistas do domínio

Maximização de reuso

Maior consistência e integração entre aplicações

Redução da manutenção

- Aplicações com menos linhas

- Reparos no framework se propagam pelas aplicações

Melhora da produtividade

- Programadores não precisam reinventar a roda

Economia a longo prazo

Desvantagens

Maior esforço para construção e aprendizado

Programas mais difíceis de depurar

Necessária documentação de manutenção e apoio

Vencer “Not Made Here Syndrome”

“The most profoundly elegant framework will never be reused unless the cost of understanding it and using its abstractions is lower than the programmer's perceived cost of writing them from scratch” (Booch, Dr Dobbs Journal, 1994)

Benefícios percebidos no longo prazo

Tipos de frameworks

Classificação de frameworks de acordo com duas dimensões:

Como o framework é usado?

Onde o framework é usado?

Tipos de frameworks

“Como é usado?”

Inheritance-focused

Também chamado de caixa-branca ou orientado à arquitetura

Estende ou modifica funcionalidade pela definição de subclasses com sobrescrita de métodos

Mais fácil de projetar

Composition-focused

Também chamado de caixa-preta ou orientado a dados

Usa a funcionalidade já presente no framework

As coisas internas do framework não podem ser vistas ou alteradas

Deve-se usar as interfaces fornecidas

As instanciações e composições feitas determinam as particularidades da aplicação

Mais fácil de usar

Híbridos

A maioria dos frameworks é inheritance-focused com alguma funcionalidade pronta (composition-focused)

Tipos de frameworks

“Onde é usado?”

Framework de suporte

Provê serviços de nível de sistema operacional (e não de aplicação)

São raros

Framework de aplicação

Também chamado de framework horizontal

Encapsula conhecimento aplicável a vasta gama de aplicações

Resolve apenas uma fatia do problema da aplicação

Exemplo: framework para construção de interface GUI

Framework de domínio

Também chamado de framework vertical

Encapsula conhecimento aplicável a aplicações de um domínio particular de problema

Resolve boa parte da aplicação

Exemplo: Framework para construir aplicações de controle de manufatura

Exemplos

Frameworks horizontais

Interface do Usuário (GUI)

Apple MacApp

Corel/Inprise/Borland OWL

Java Foundation Classes (JFC/Swing)

Microsoft Foundation Classes (MFC)

Sistemas operacionais

Aplicações são extensões do sistema

Frameworks para persistência

TopLink ou CocoBase

Middleware

Enterprise Java Beans

Microsoft MTS/COM+

Frameworks MVC

Struts

IBM servletmanager

Oracle9iAS MVC Framework

WebFlow em WebLogic Portal 4.0 da BEA

WebWork

Tapestry

Turbine

Java Infobus

Frameworks verticais

Editores de desenho estruturado

Visio

JHotdraw

Aplicações financeiras complexas

IBM San Francisco

Jogos de dados

Landin

Niklasson

Monitoramento de risco financeiro

Rogers

Como construir frameworks?

Frameworks são abstrações

As pessoas fazem generalização a partir de exemplos concretos

A concepção de código reutilizável requer iteração

Frameworks embutem conhecimento do domínio

Clientes do framework são os programadores de aplicações

Generalizando:

Encontre coisas com nomes diferentes mas que sejam as mesmas

Parametrize para eliminar diferenças

Quebre coisas grandes em coisas pequenas, de forma que componentes similares possam ser encontradas

Categorize coisas similares

Frameworks

Bruna Diirr

brunadiirr@ic.uff.br