Semana Nacional de Ciência e Tecnologia (SNCT)

IFBA Campus Vitória da Conquista

19 a 21 de outubro de 2015

Linha de Produtos de Software (SPL) em Java: Teoria e **Prática**

Prof. Me. Pablo Freire Matos - pablofmatos@ifba.edu.br
Prof. Me. Djan Almeida Santos - djan.santos@ifba.edu.br
Informática - Instituto Federal da Bahia (IFBA)

Agenda

- Reuso de Software
  - Motivação
  - Contextualização
  - Objetivo
- Criação/Instanciação de uma SPL
- Prática
- Estudo de Caso

Motivação

- Reuso de Software:
  - 40-60% código é reusável
  - 75% funções são comuns e podem ser reutilizáveis em mais de um programa
  - 15% código é único
- Avanço em ES:
  - POO
  - DBC
  - SPL

O que é o Reuso de Software?

- Reutilização de software é o processo de incorporar em um novo produto:
  - código
  - especificações de requisitos e projeto
  - planos de teste
  - qualquer produto gerado durante desenvolvimentos anteriores
  - conhecimento em geral

Definição Reuso

- O processo de criar sistemas de software a partir de software existente é preferível do que construir do zero {Krueger}

Benefícios da Reutilização

- Produtos:
  - Melhor qualidade
  - Mais confiáveis
  - Consistentes
  - Padronizados

  } **Aumento da Produtividade**
- Redução do esforço:
  - Custo de manutenção
  - Tempo de mercado
  - Documentação
  - Trabalho redundante

Dificuldades

- Aumento nos custos de manutenção
- Falta de ferramentas de apoio
- Barreiras psicológicas:
  - síndrome do não foi inventado aqui
- Barreiras legais e econômicas
- Necessidade da criação de incentivos à reutilização

Técnicas para Reuso

- Bibliotecas de Classes
- Padrões de software
- Componentes
- *Frameworks*
- Famílias de Produtos

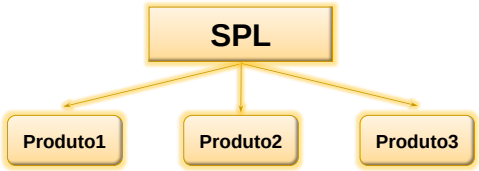
O que é o Família de Produtos?

- Conjunto de aplicações relacionadas, que têm uma arquitetura de domínio específico em comum
  - Contudo cada aplicação específica é especializada de alguma maneira
- O núcleo em comum da família é reutilizado cada vez que uma nova aplicação é desejada
- O novo produto pode exigir que algumas partes de código sejam escritos ou modificados

Exemplo de Família de Produtos

Objetivo do Minicurso: Parte Prática

- Criar uma Linha de Produto de Software
  - Instanciar diferentes produtos



Etapas para a Criação da SPL

1. **Definição do Domínio da SPL**
   - Requisitos
2. Mapa de Produto e Modelo de Feature
3. Desenvolvimento da SPL
   - Diagramas de Caso de Uso e de Classe
   - Interface Java em linha de comando
   - Definição das diretivas de compilação
4. Instanciação

① Definição do Domínio da SPL

① Requisitos

- Objetivo Geral:
  - Criar n produtos com diferentes funcionalidades
- Objetivos Específicos:
  - Proporcionar aos usuários o gerenciamento e o armazenamento persistente dos contatos
  - Permitir importar e exportar a agenda
  - Acrescentar manual ou automaticamente um dígito no início do número de contato

① Requisitos

- Gerenciamento dos Dados
  - adicionar, alterar, excluir, pesquisar e visualizar
- Dados
  - nome, sobrenome, ddd e número
- Armazenamento Persistente
  - importar e exportar a agenda
- Acréscimo de Dígito
  - acrescentar manualmente ou automaticamente um dígito no início de um número

Etapas para a Criação da SPL

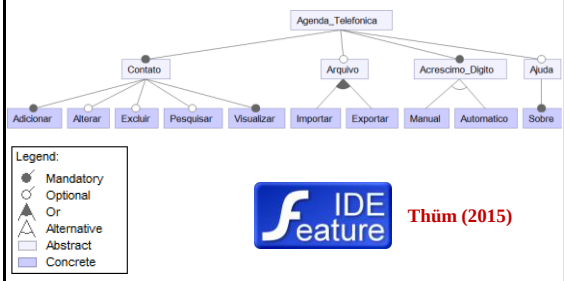
1. Definição do Domínio da SPL
   - Requisitos
2. **Mapa de Produto e Modelo de Feature**
3. Desenvolvimento da SPL
   - Diagramas de Caso de Uso e de Classe
   - Interface Java em linha de comando
   - Definição das diretivas de compilação
4. Instanciação

② Mapa de Produto

SPL Agenda Telefônica

Feature	*Subfeature*	*Agenda Lite*	*Agenda Standard*	*Agenda Pro*
Contato	Adicionar	**Sim**	**Sim**	**Sim**
	Alterar	Não	**Sim**	**Sim**
	Excluir	Não	**Sim**	**Sim**
	Pesquisar	Não	**Sim**	**Sim**
	Visualizar	**Sim**	**Sim**	**Sim**
Arquivo	Importar	Não	Não	**Sim**
	Exportar	Não	Não	**Sim**
Acrescimo Digito	Manual	**Sim**	**Sim**	Não
	Automático	Não	Não	**Sim**
Ajuda	Sobre	Não	Não	**Sim**
		3 F	**6 F**	**9 F**

② Modelo de Feature: Diagrama



FeatureIDE

Thüm (2015)

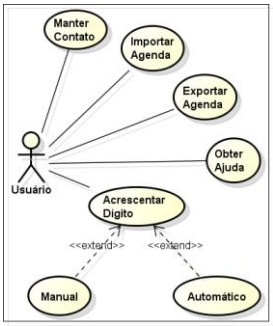
② Modelo de Feature: XML

```
<struct>
    <and abstract="true" mandatory="true" name="Agenda_Telefonica">
        <and abstract="true" mandatory="true" name="Contato">
            <feature mandatory="true" name="Adicionar"/>
            <feature name="Alterar"/>
            <feature name="Excluir"/>
            <feature name="Pesquisar"/>
            <feature mandatory="true" name="Visualizar"/>
        </and>
        <or abstract="true" name="Arquivo">
            <feature mandatory="true" name="Importar"/>
            <feature mandatory="true" name="Exportar"/>
        </or>
        <alt abstract="true" mandatory="true" name="Acrescimo_Digito">
            <feature mandatory="true" name="Manual"/>
            <feature mandatory="true" name="Automatico"/>
        </alt>
        . . .
    </and>
</struct>
```

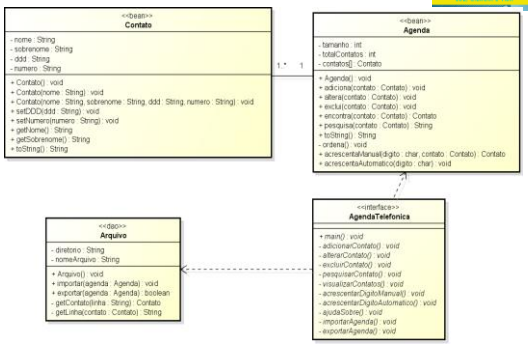
Etapas para a Criação da SPL

1. Definição do Domínio da SPL
   - Requisitos
2. Mapa de Produto e Modelo de Feature
3. **Desenvolvimento da SPL**
   - Diagramas de Caso de Uso e de Classe
   - Interface Java em linha de comando
   - Definição das diretivas de compilação
4. Instanciação

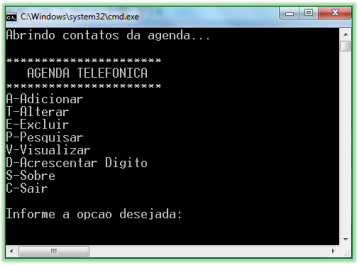
③ Diagrama de Caso de Uso



③ Diagrama de Classe



③ Interface Java em Linha de Comando



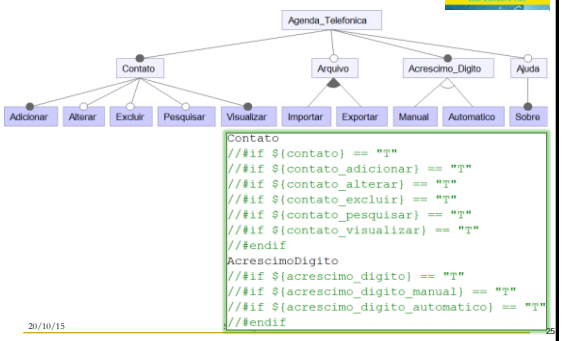
③ Diretivas de Compilação: Definição

- Java (Compilação Condicional)
  - Preprocessador JavaPP → Diretivas #IF DEF
  - Kropf (2010)

```
Conditions
-----------------
    #if expression [and|or expression]
    [#elif expression [and|or expression]]
    [#else]
    #endif
Conditions can be nested.
Expressions are combined using the 'and', 'or' operators.
```

```
Expressions
-----------------
    operand <operator> operand | [!]defined(prop)
Expressions contain an operator and two operands.
The supported operators are '==', '!=', '<=', '<', '>=', '>'.
```

3 Diretivas de Compilação: Definição




```
Contato
//#if ${contato} == "T"
//#if ${contato_adicionar} == "T"
//#if ${contato_alterar} == "T"
//#if ${contato_excluir} == "T"
//#if ${contato_pesquisar} == "T"
//#if ${contato_visualizar} == "T"
//#endif
AcrescimoDigito
//#if ${acrescimo_digito} == "T"
//#if ${acrescimo_digito_manual} == "T"
//#if ${acrescimo_digito_automatico} == "T"
//#endif
```


3 Diretivas de Compilação: Exemplo

```
public class HelloWorld {
public static void main(String[] args) {
    //#if ${hello} == "T"
    System.out.println("Hello");
    //#endif
    //#if ${feature_wonderful} == "T"
    System.out.println("Wonderful");
    //#endif
    //#if ${feature_beautiful} == "T"
    System.out.println("Beautiful");
    //#endif
    //#if ${world} == "T"
    System.out.println("World");
    //#endif
}
}
```

Diretivas inseridas no código fonte

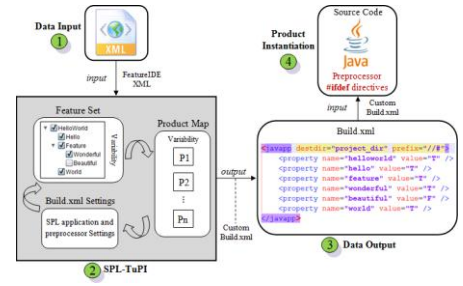
Arquivo XML interpretado pelo JavaPP

```
<?xml version="1.0" encoding="UTF-8"?>
<project default="run">
<taskdef resource="javapp-defs.xml" classpath="javapp.jar" />
<target name="preprocess" depends="clean">
<javapp destdir="HelloWorld_Wonderful/src/spl" prefix="//#">
    <fileset dir="src/spl" includes="**/*.java" />
    <property name="helloworld" value="T" />
    <property name="hello" value="T" />
    <property name="feature" value="T" />
    <property name="feature_wonderful" value="T" />
    <property name="feature_beautiful" value="F" />
    <property name="world" value="T" />
</javapp>
<target/>
  .
  .
  .
</project>
```

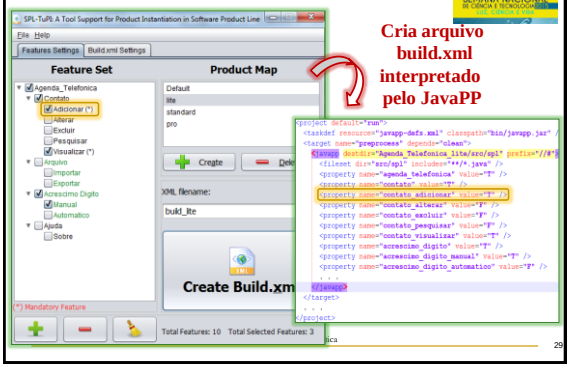

Etapas para a Criação da SPL

1. Definição do Domínio da SPL
   - Requisitos
2. Mapa de Produto e Modelo de Feature
3. Desenvolvimento da SPL
   - Diagramas de Caso de Uso e de Classe
   - Interface Java em linha de comando
   - Definição das diretivas de compilação
4. **Instanciação**

4 Instanciação

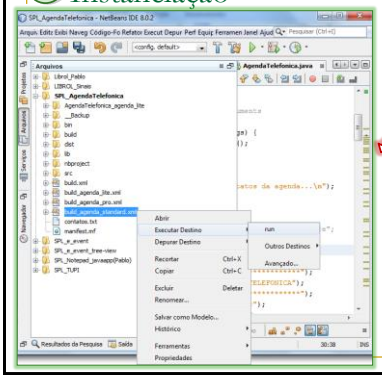


4 Instanciação



Cria arquivo build.xml interpretado pelo JavaPP

4 Instanciação



Instancia os produtos na IDE NetBeans

Prática

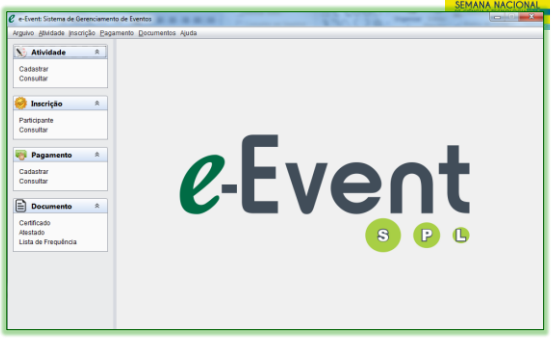
- **Atividade 1:**
  - Criar o arquivo XML na ferramenta FeatureIDE
  - Baixar a ferramenta SLP-TuPI (site minicurso)
  - Criar o build.xml dos 3 produtos utilizando a ferramenta SPL-TuPI
- **Atividade 2:**
  - Baixar o código fonte da SPL AgendaTelefônica (site minicurso)
  - Definir as diretivas de compilação #IF DEF no código fonte de acordo com o mapa de produto
    - **Definir somente para a classe Agenda**
- **Atividade 3:**
  - Instanciar os 3 produtos no NetBeans a partir dos arquivos XMLs criados

Prática

- Como ficou a diretiva #IF DEF para o método encontra da classe Agenda?

```
//#if ${contato_alterar} == "T" or ${acrescimo_digito_manual} == "T"
public Contato encontra(Contato contato) {
    for (int i = 0; i < totalContatos; i++) {
        if (contato.compareTo(this.contatos[i]) == 0) {
            return this.contatos[i];
        }
    }
    return null; // Não encontrou
}
//#endif
```


Estudo de Caso SPL e-Event



Mapa de Produto

Feature	Subfeature	Basic e-Event	Intermediate e-Event	Advanced e-Event
Atividade	Palestra	Sim	Não	Sim
	Minicurso	Sim	Sim	Sim
	Mesa Redonda	Não	Sim	Sim
Inscrição	Participante	Sim	Sim	Sim
Pagamento	À vista	Não	Sim	Sim
	Depósito	Não	Sim	Sim
	Boleto	Não	Não	Sim
	Com desconto Institucional	Não	Sim	Sim
	Com desconto SBC	Não	Não	Sim
Certificado	Ministrante	Não	Não	Sim
	Organização	Sim	Não	Sim
	Participação	Não	Sim	Sim
Documento	Atestado	Sim	Não	Sim
	Lista de Frequência	Não	Sim	Sim
		5 F	8 F	14 F

Modelo de Feature

FeatureIDE Thüm (2015)

Minicurso ∨ "Mesa Redonda" ⇒ "À vista"
Minicurso ∨ "Mesa Redonda" ⇒ "Membro Institucional"
Minicurso ∨ "Mesa Redonda" ⇒ Participação

Legend:
- Mandatory
- Optional
- Or
- Abstract
- Concrete

Diagrama de Caso de Uso

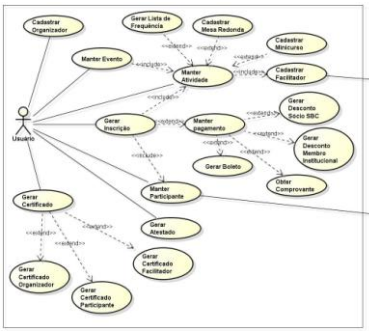
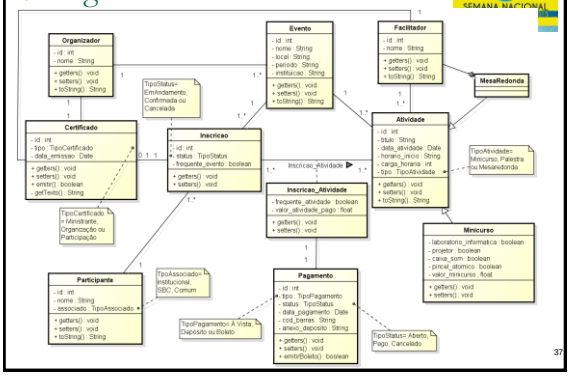
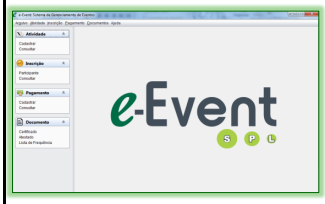


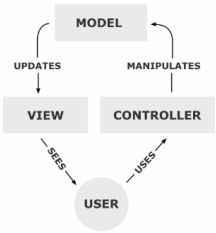
Diagrama de Classe



Desenvolvimento



Arquitetura



e-Event-SPL: Produtos Criados

lib 4,320 KB

Linhas de Produto	COM Lib	SEM Lib	Redução de Código
Without Features	4.780 KB	460 KB	---------
Basic	5.010 KB	690 KB	230 KB (34.47%)
Intermediate	5.088 KB	768 KB	308 KB (12.25%)
Advanced	5.131 KB	811 KB	351 KB

Linhas de Produto	Arquivos Fonte	LOC	BLOC	CLOC	SLOC-P	SLOC-L
Without Features	8	1.811	401	113	1.297	1.009
Basic	51	7.723	1.504	348	5.871	4.253
Intermediate	63	9.503	1.801	432	7.270	5.249
Advanced	72	10.863	2.043	495	8.325	6.056

http://www.locmetrics.com/ ***Diretórios: 31***

Referências

- KANG, K.; COHEN, S.; HESS, J.; NOVAK, W.; PETERSON, A. **Feature-Oriented Domain Analysis (FODA) Feasibility Study**. Software Engineering Institute, Carnegie Mellon University, 1990.
- KROPF, J. **Java Preprocessor for Apache Ant**. 2010. Disponível em: <http://git.slashdev.ca/javapp>. Acesso em: 19 out. 2015.
- SPENCER, H.; COLLYER, G. **#ifdef Considered Harmful, or Portability Experience With C News**. San Antonio, TX: University of Toronto, 1992. p.
- THÜM, T. **FeatureIDE:** Get Started. 2015. Disponível em: <http://wwwiti.cs.uni-magdeburg.de/iti_db/research/featureide/slides/featureide-2-getstarted.pdf>. Acesso em: 19 out. 2015.
- THÜM, T.; KÄSTNER, C.; BENDUHN, F.; MEINICKE, J.; SAAKE, G.; LEICH, T. FeatureIDE: An extensible framework for feature-oriented software development. **Science of Computer Programming**, v. 79, p. 70-85, 2014.

Material do Minicurso

http://goo.gl/9QatNl

Semana Nacional de Ciência e Tecnologia (SNCT)

IFBA Campus Vitória da Conquista

19 a 21 de outubro de 2015

Linha de Produtos de Software (SPL) em Java: Teoria e **Prática**

Prof. Me. Pablo Freire Matos - pablofmatos@ifba.edu.br
Prof. Me. Djan Almeida Santos - djan.santos@ifba.edu.br
Informática - Instituto Federal da Bahia (IFBA)