Minicurso SINFORM 2016 Linha de Produto de Software

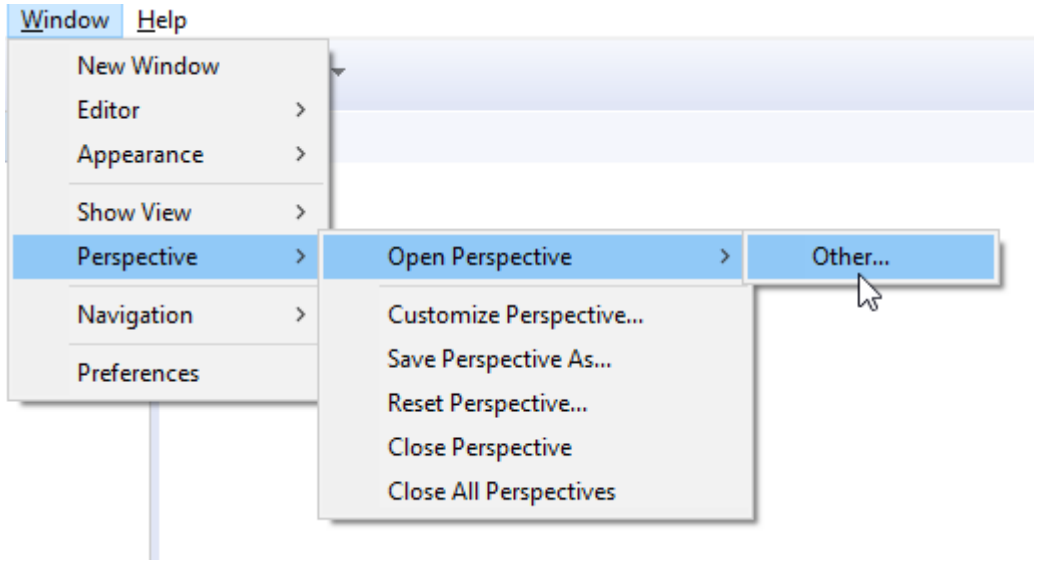
Mateus Passos Soares Cardoso

Roteiro

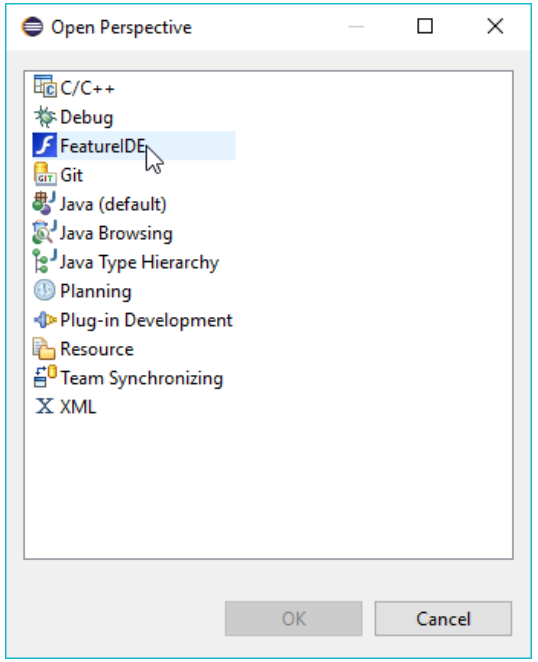
- Configurando o ambiente de trabalho
- Criando o projeto Hello World com o Composer AHEAD
- Criando o projeto Hello World com Compilação Condicional

Ativando a perspectiva do Feature IDE

- Window -> Perspective -> Open Perspective -> Other...

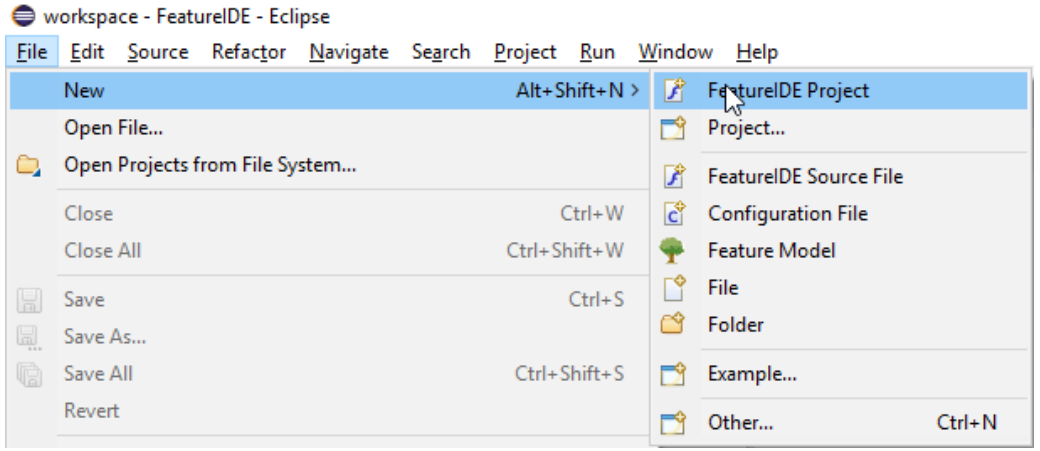


Ativando a Perspectiva do FeatureIDE



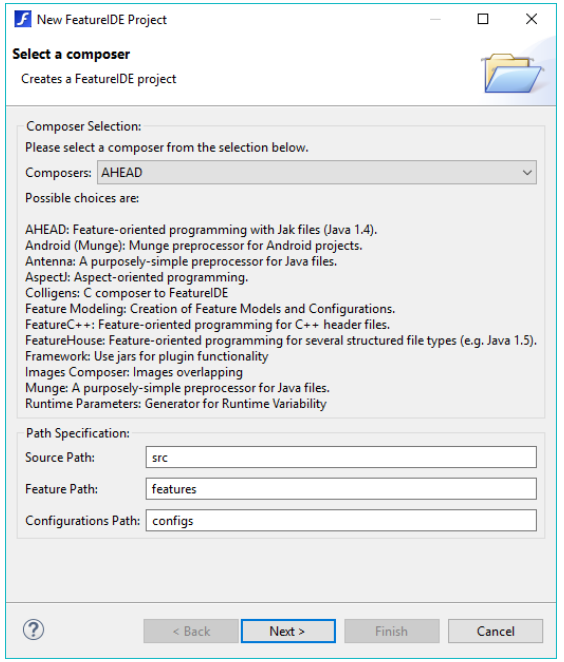
Criando um novo projeto do FeatureIDE

- File -> New -> Feature IDE Project

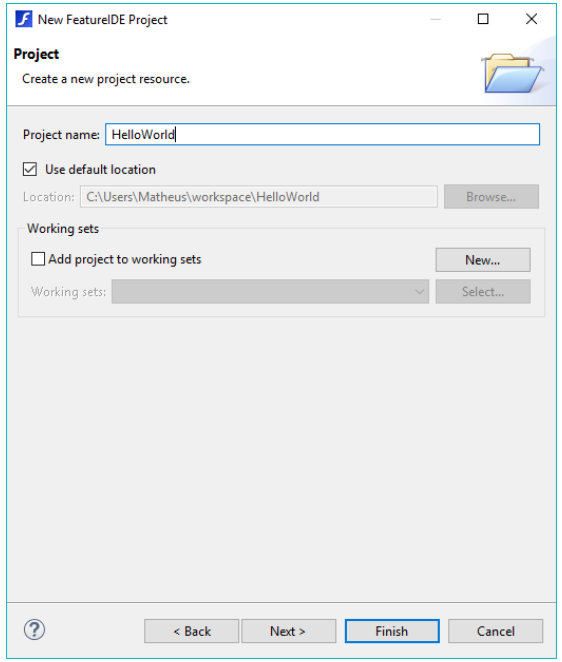


Escolhendo um Composer

- A função do composer é selecionar montar o produto a partir do conjunto de features selecionado.

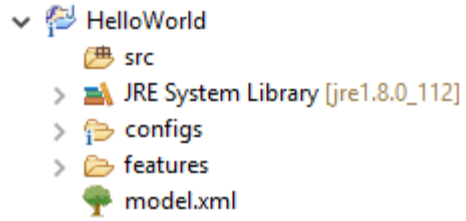


Definindo o nome do projeto



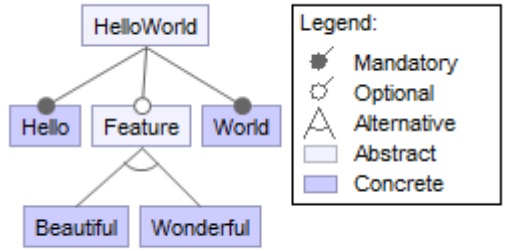
Estrutura do Projeto

- src – Contém o código do produto.
- Configs – Arquivo contendo as configurações dos produtos a serem gerados.
- Features – Contém o código fonte das features.
- Model.xml – Contém o diagrama de features.



Criando o modelo de features

- Abra o arquivo model
- Para adicionar uma features basta clica com o botão direito do mouse sobre umas da features disponíveis.



Criando o código das features Composer AHEAD

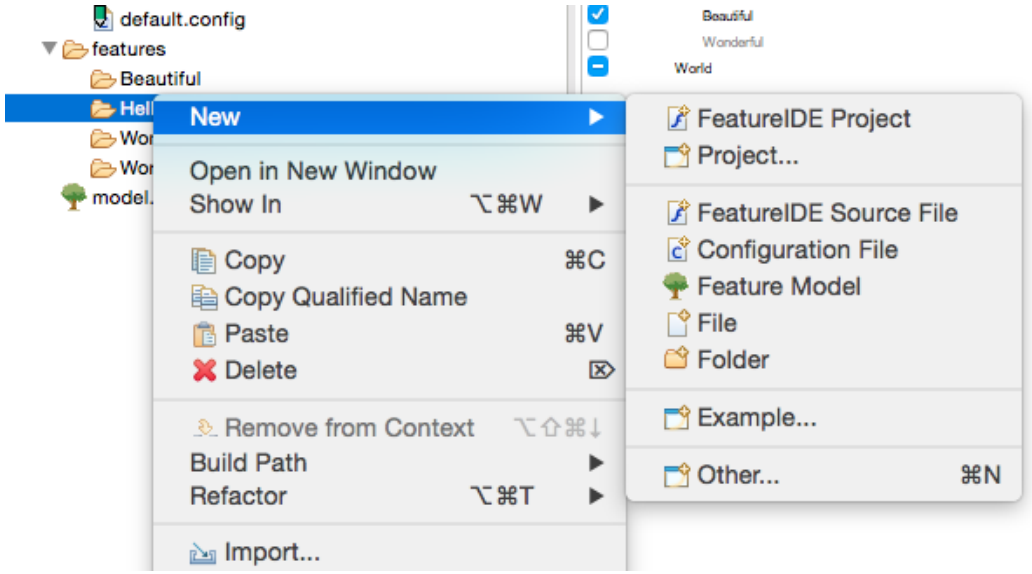
- No composer AHEAD, trabalhamos com arquivos .jak
- Arquivos .jak estendem as palavras chave com novas palavras voltadas para o desenvolvimento de projetos LPS.

Criando o código das features Composer AHEAD

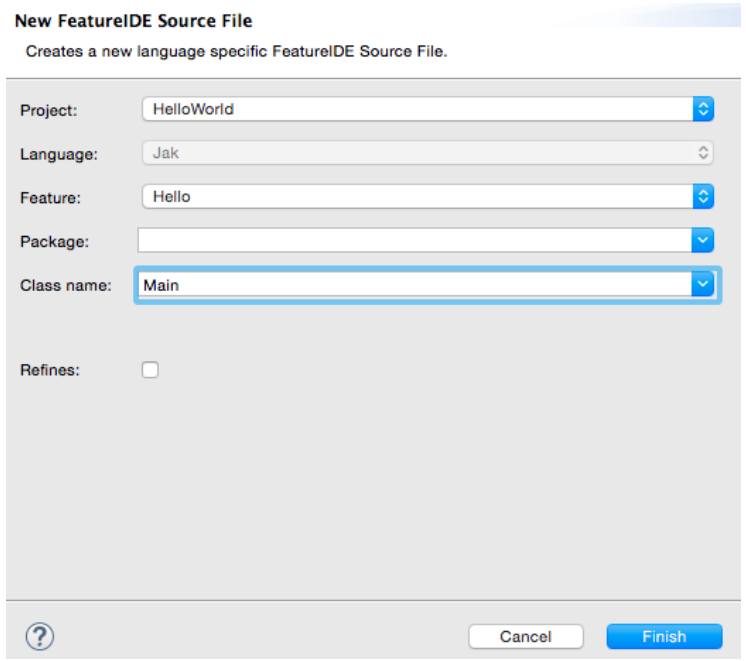
- refines
  - Usada para denotar o refinamento de uma classe já existente
- Super()
  - Usada para chamar um método refinado

Criando o código das features
Composer AHEAD

- Começaremos criando o código da feature Hello.



Criando o código das features Composer AHEAD



Criando o código das features Composer AHEAD

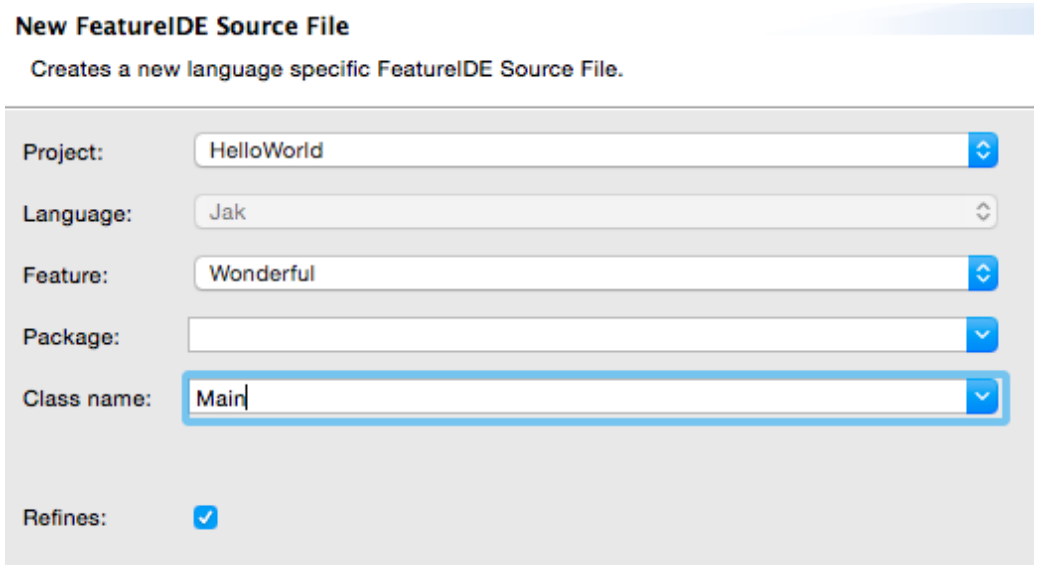
```
public class Main {
    public void print(){
        System.out.print("Hello");
    }

    public static void main(String args[]){
        new Main().print();
    }

}
```

Criando o Código das Features Composer AHEAD

- Para as features Beautiful, Wonderful e World iremos refinar a classe Main que acabamos de criar.

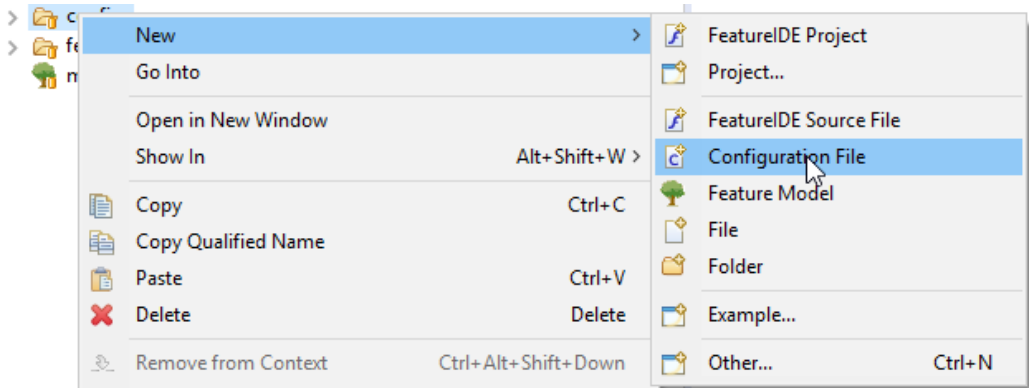


Criando o Código das Features Composer AHEAD

```
public refines class Main {
   public void print(){
      Super().print();
      System.out.print(" world!");
   }
}
```

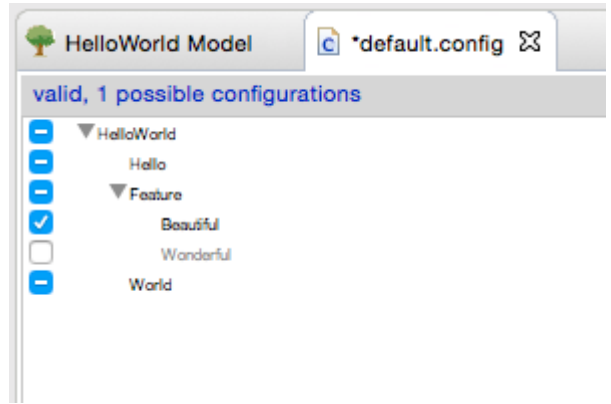
Criando uma configuração de produto

- File -> New -> Configuration File



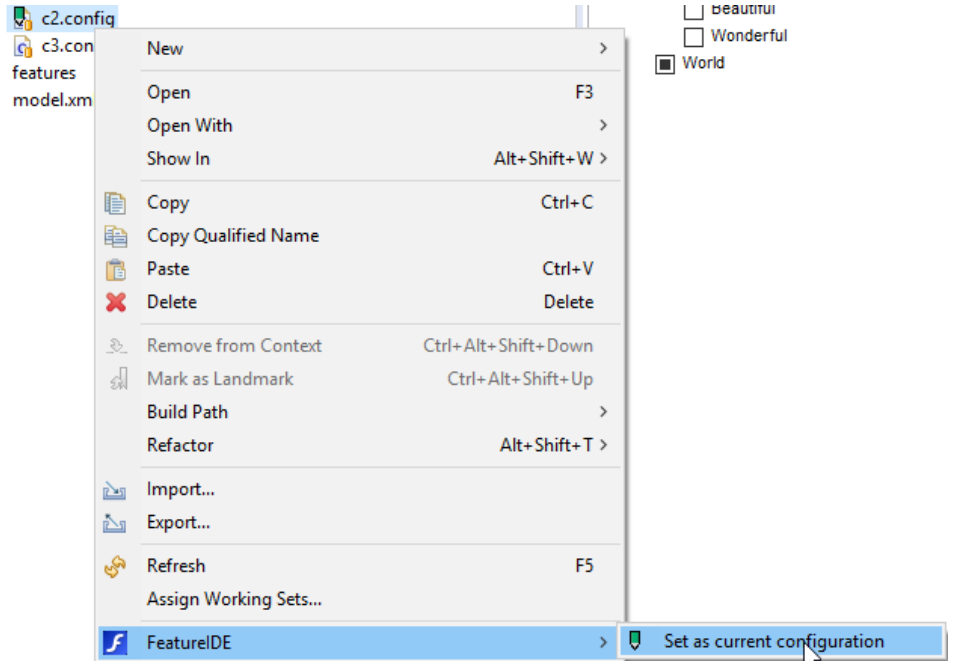
Criando uma configuração de produto

- Com o arquivo de configuração aberto, basta escolher as features que irão compor o produto e salvar.



Executando o produto gerado

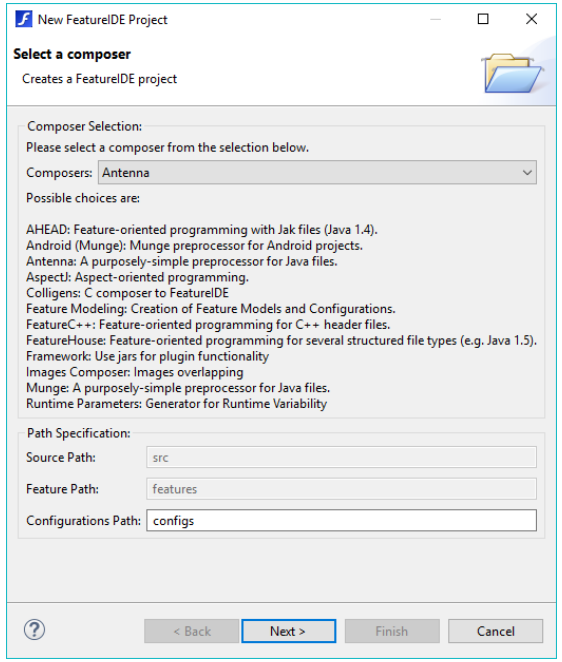
- Crie quantas configurações achar necessário
- Para definir uma configuração para gerar o produto basta clicar com o botão direito em cima do arquivo de configuração -> FeatureIDE -> Set as current configuration
- Após escolher uma configuração execute o projeto.



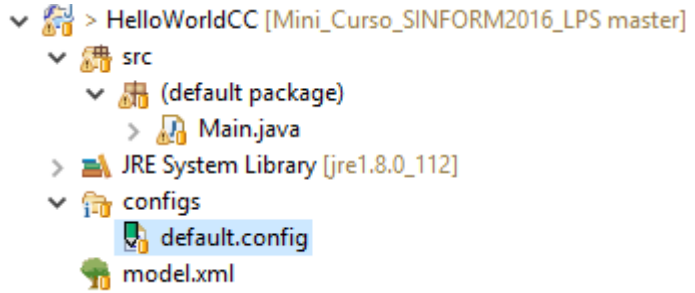
Utilizando compilação condicional

- É o método mais utilizado para criação de projetos LPS.
- Neste método utilizamos diretivas de código fonte para definir trechos do código que serão compilados ou não baseado na escolha de features.
- Crie um novo projeto no FeatureIDE e desta vez utilize o composer Antenna.

Utilizando compilação condicional



Utilizando Compilação Condicional
Estrutura do projeto



Utilizando Compilação Condicional Diretivas de Código

#if [expressão]	Utilizado para definir quais partes do código irão implementar uma features, deve ser terminado a diretiva #endif
#elif [expressão]	Funciona como o else if
#else	Funciona como o else das linguagens de programação tradicionais.
#endif	Usada para encerrar a diretiva #if

Utilizando Compilação Condicional Operadores Lógicos

!	Negação
&&	E lógico
^	Xor
||	Ou lógico

Utilizando Compilação Condicional
Código da Classe Main

```
public class Main {
    public static void main(String args[]){
        //#if Hello
        System.out.print("Hello");
        //#endif
        ...
        //#if World
        System.out.print(" world!");
        //#endif
    }
}
```

Utilizando Compilação Condicional Atividade

- Complete o código .
- Defina arquivos de configuração e execute seu projeto.