

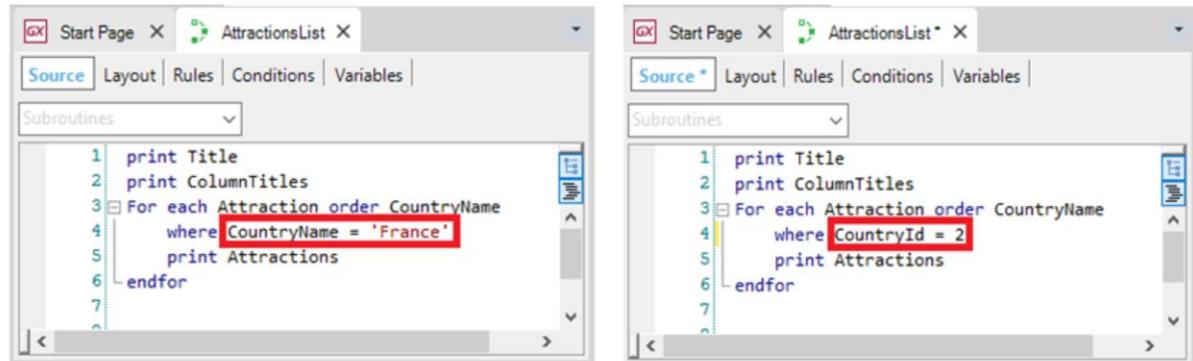
Comunicação entre objetos

Necessidade de invocar um objeto a partir de outro

GeneXus™ 16

Até o momento...

Utilizamos valores fixos para filtrar... neste caso por país



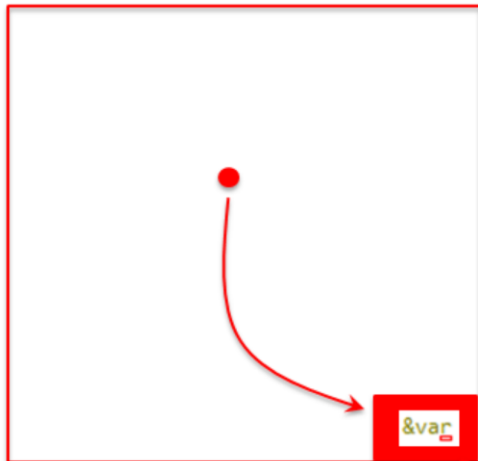
E se quisermos generalizar a lista e poder "receber" de alguma forma o país pelo qual filtrar?...

Em situações anteriores, tivemos que chamar um objeto a partir de outro objeto.

Por exemplo, quando implementamos o objeto procedimento AttractionsList, precisávamos filtrar as atrações cujo país se chamava "França", ou, da mesma forma, cujo identificador do país era 2 (que correspondia à "França")... Para isso, usamos valores fixos no código.

No entanto, isto implica que se quiséssemos filtrar as atrações de um outro país que não fosse a França, teríamos de mudar o código do procedimento toda vez!

Object A



Object B: AttractionsList



Idealmente, deveríamos ser capazes de "receber" neste objeto o valor pelo qual queremos filtrar. Em outras palavras, aquele outro objeto GeneXus permitir ao usuário escolher aquele valor... e depois enviá-lo para este objeto procedimento para ter as atrações listadas de acordo com o país recebido.

Em seguida, usaremos este exemplo para ver como implementar essa comunicação entre objetos GeneXus.

Definindo a comunicação entre objetos...

1) Criamos um web panel que peça o país a considerar.

Variável com tipo de controle **Dynamic Combo** para que ofereça ao usuário os países da base de dados

2) Adicionamos botão para chamar o procedimento **AttractionsList**.

Botão que tem associado o evento **ListAttractionsByCountry**

Variável que contém o país indicado no form do web panel.

Event 'List attractions by country'
AttractionsList(&CountryId)
Endevent

Name	Type	Is Co
Variables		
Standard Variables		
CountryId	Attribute:CountryId	

Primeiro, precisamos criar um objeto que forneça uma tela para solicitar valores ao usuário e faça alguma coisa com esses valores. O objeto que permite isto é o **web panel**, que será estudado em detalhes mais tarde. Por enquanto, digamos que é um painel visual, flexível que pode solicitar dados ao usuário, bem como mostrar as informações do banco de dados ou de outras fontes, entre outras coisas. Por exemplo, o elemento Work With attractions foi automaticamente implementado por GeneXus como um Web Panel.

Então, nós criaremos um objeto deste tipo e o chamaremos EnterAttractionsFilter. Note que um Web Form será criado para ser o objeto de tela. Ele contém uma única tabela

Nós adicionamos uma variável CountryId... Como seu nome é o mesmo do atributo, é baseado nele e, portanto, leva o mesmo tipo de dados. Desta forma, se mudarmos o tipo de dados do atributo, por exemplo de numérico(10) para numérico (4), a variável irá automaticamente assumir este valor.

Agora vamos editar as propriedades da variável e ver que sua propriedade **Control Type** tem o valor **Edit**. Isto significa que quando o web panel é executado, este campo espera que o usuário insira um valor numérico, mas não irá fornecer qualquer ajuda para escolher os valores existentes no banco de dados ou indicar o país correspondente. Mudamos o tipo de controle para **Dynamic ComboBox**. Desta forma, ao usuário será oferecida uma série de valores extraídos do banco de dados para ele escolher um. Que valores? Aqueles do atributo CountryId. Isso quer dizer, a tabela Country será percorrida e os CountryIds existentes serão carregados no combo box. Mas, como os identificadores geralmente não fornecem qualquer detalhe, mesmo que a variável irá armazenar um identificador de país, ao usuário será mostrado o conteúdo do atributo indicado na propriedade **Item Descriptions** da variável... Nós escolhemos mostrar o nome do país. Observe que a seta que indica o combo é exibida. Em suma, em tempo de execução irá oferecer um combo box com uma lista dos países armazenados no banco de dados para escolher um que nos interessa.

Além disso, adicionamos um botão. Nos solicita que informemos o nome do evento

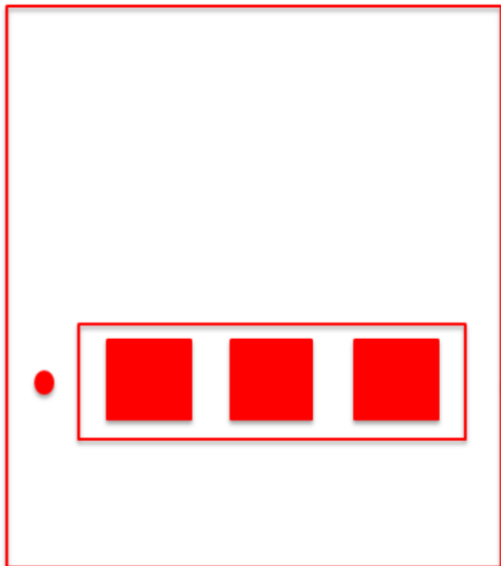
que será associado esse botão. Chamamos de: : "List Attractions By Country". Vemos que o texto do botão leva o mesmo nome padrão. Se clicarmos nele, botão direito do mouse, e selecionamos Go to Event... vemos que um evento com esse nome foi criado e, alterada automaticamente da guia Web Form para a guia Events. Além disso, o cursor está aguardando que digitemos o código que será executado quando esse evento for acionado. Isso quer dizer, quando o usuário pressionar o botão associado.

O que precisamos fazer agora é chamar o objeto procedimento AttractionsList que lista as atrações e enviar o país que queremos usar para filtrá-las.

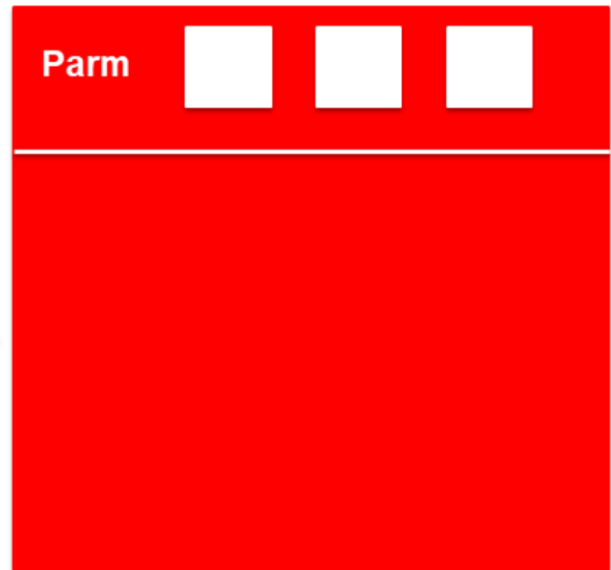
Note que quando pressionarmos o botão e executarmos esse código, a variável &CountryId irá conter o identificador do país selecionado pelo usuário no combo box exibido na tela. Anteriormente, vimos que uma variável é uma parte da memória que é dado um nome e usada para salvar temporariamente um item de dados. Também, que cada objeto tem sua seção de variáveis. Isso quer dizer, as variáveis definidas em um objeto são conhecidas apenas neste objeto. Então, se dois objetos têm uma variável CountryId, mesmo se elas tiverem o mesmo nome, serão duas variáveis diferentes.

Definindo a comunicação entre objetos...

Object A



Object B

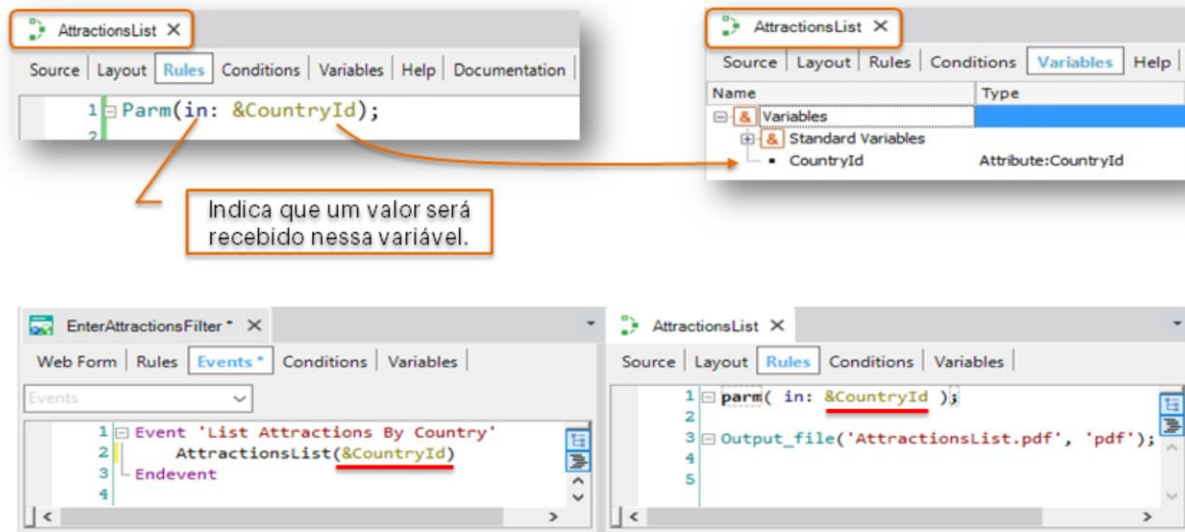


Então, como fazer um objeto A chamar outro objeto B em um determinado momento, enviando valores? Além disso, este objeto B deve ser capaz de carregar em suas variáveis internas os valores enviados a ele, a fim de fazer alguma coisa com essa informação.

Para um objeto para ser capaz de receber valores (os quais chamamos de **parâmetros**), temos que abrir sua seção de regras e escrever uma regra **Parm**. Esta regra de **Parm** declara os parâmetros que o objeto pode receber e/ou retornar ao chamador.

Regra Parm

Para que um objeto possa receber valores (parâmetros), devemos utilizar a regra Parm.



A partir de nosso exemplo os valores serão recebidos pelo objeto procedimento AttractionsList, abrimos o objeto e vamos para a seção de regras. Vamos escrever a regra parm acima.

Com "in" indicamos que a variável &CountryId será um parâmetro de **entrada**. Isso significa que ele será usado apenas para receber um valor do chamador. Não retornará nenhum valor. Podemos pular essas informações e ter GeneXus a inferi-lo.

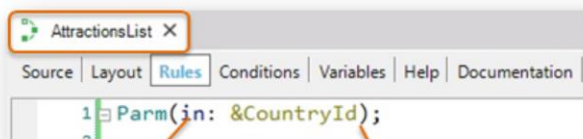
Neste objeto (AttractionsList), nós criamos a variável com o mesmo nome e tipo de dados como a que usamos no web panel para o usuário para entrar o país.

No entanto, como já dissemos, são duas variáveis diferentes. Uma só é válida no web panel e a outra no procedimento. Poderíamos ter usado nomes diferentes em ambos os objetos, mas para a comunicação e compartilhamento de dados ser correto, o **data type** do chamador e objeto chamado devem ser o mesmo.

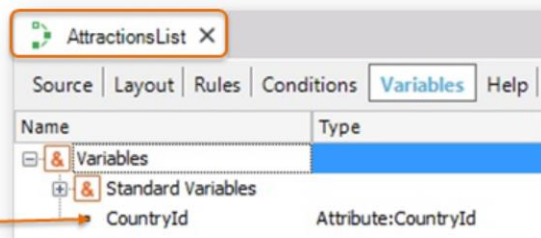
Agora, nosso objeto procedimento está pronto para receber um identificador de país, neste caso, do web panel EnterAttractionsFilter.

Regra Parm

Para que um objeto possa receber valores (parâmetros), devemos utilizar a regra Parm.



Indica que um valor está sendo recebido nessa variável.



Modificamos o Source:

```
print Title
print ColumnTitles
|For each Attraction order CountryName
  where CountryId = 2
  print Attractions
endfor
```

Mudamos para...

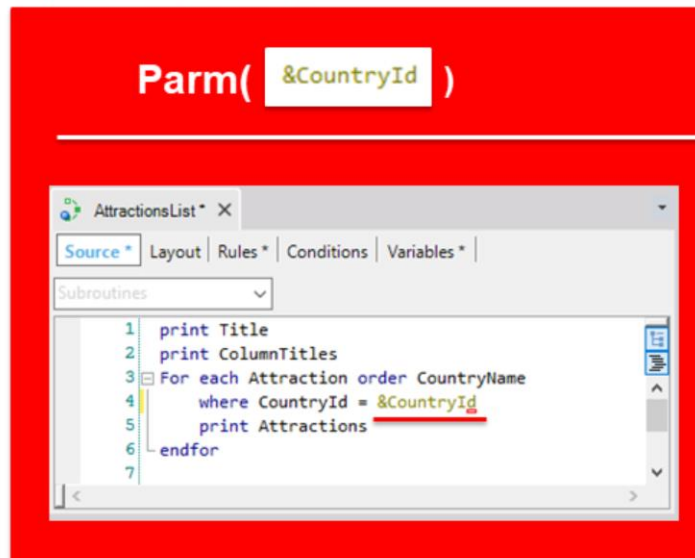
```
print Title
print ColumnTitles
For each Attraction order CountryName
  where CountryId = &CountryId
  print Attractions
endfor
```

Agora só temos que remover o filtro fixo que tínhamos (país valor 2) no comando For Each e alterá-lo para a variável cujo valor é recebido como um parâmetro.

Object A



Object B: AttractionsList



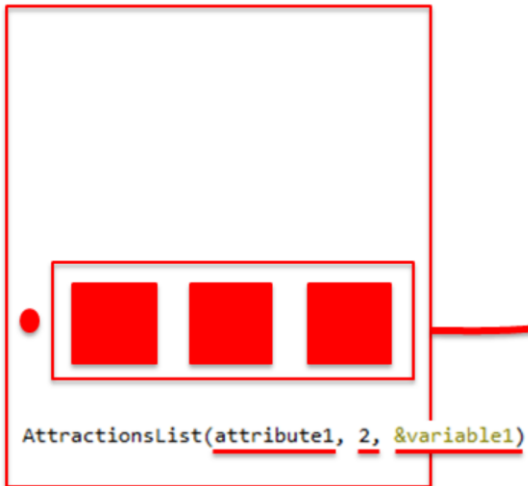
Note que, a partir da regra Parm, indicou-se desta forma que, de agora em diante, qualquer objeto que chama o procedimento será capaz de (e terá que) enviar o valor do identificador de país. Já não será possível chamar esse procedimento sem enviar a ele um valor deste tipo. É por isso que o procedimento AttractionsList não será mais exibido no Developer Menu.

No caso do web panel tivemos esse valor em uma variável (que o usuário entrou na tela). Mas se tivéssemos o dado em um atributo, incluiria o atributo correspondente entre parênteses.

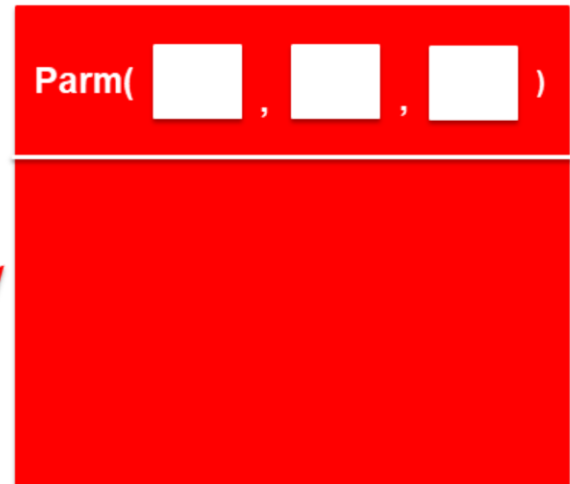
Podemos também enviar um valor.

Definindo a comunicação entre objetos...

Object A



Object B



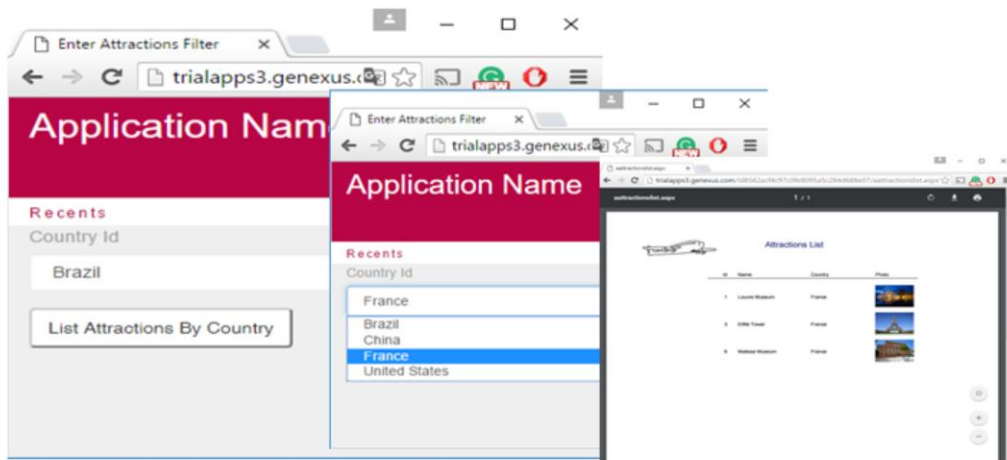
... Ou, se tivermos que enviar dois ou mais valores, enviáremos vários atributos, e/ou valores explícitos, e/ou variáveis separados por vírgulas.

Esses parâmetros também são declarados na regra parm de forma ordenada, separada por vírgulas.

Obviamente, um objeto que não recebe parâmetros não deve declarar a regra Parm.

DEMO

- Testemos em GeneXus tudo que fizemos: F5
Observar que o procedimento AttractionsList já não aparece no Developer Menu



Testamos o que fizemos, pressionando F5. Vemos que o procedimento AttractionsList não é mais exibido. Agora podemos apenas chamá-lo através do web panel...

No combo país, escolhemos a França... e pressionamos o botão.

Escolhendo o valor França do combo dinâmico, o valor do identificador de França internamente foi selecionado (no caso, o valor 2); Esse valor é enviado para o procedimento AttractionsList.

Vemos que o relatório foi executado, mostrando somente as atrações do país França.

Listar as atrações em um intervalo de nomes determinado

No web panel:

The screenshot shows a web panel with a form. At the top is a 'Country Id' dropdown menu. Below it are two text input fields: 'Attraction Name From' and 'Attraction Name To'. At the bottom are two buttons: 'List Attractions By Country' and 'List Attractions By Name'. An orange arrow points from the 'List Attractions By Name' button to the event definition on the right.

Event 'List attractions by name'
 AttractionsByName(&AttractionNameFrom, &AttractionNameTo)
 Endevent

```
Parm(in:&NameFrom, in:&NameTo);  

Output_file('AttractionByName.pdf', 'PDF');
```

No procedimento AttractionsByName:

Name	Type	Is
Variables		
Standard Variables		
NameFrom	Attribute:AttractionName	
NameTo	Attribute:AttractionName	

```
print Title  

print ColumnTitles  

For each Attraction order CountryName  

  where AttractionName >= &NameFrom  

  where AttractionName <= &NameTo  

  print Attractions  

endfor
```

Agora, vamos supor que queremos listar todas as atrações cujos nomes correspondem a um intervalo de valores selecionados pelo usuário. Por exemplo, entre "A" e "D".

Para fazer isso, para o web panel que criamos anteriormente, nós iremos adicionar a possibilidade ao usuário para inserir um nome inicial e um nome final. Desta forma, pressionando um botão irá chamar uma lista para mostrar todas as atrações turísticas cujos nomes estão dentro do intervalo.

Abrimos web panel EnterAttractionsFilter e adicionamos uma tabela com duas variáveis:

- &AttractionNameFrom... é baseada no atributo AttractionName,
- E &AttractionNameTo, que baseia-se também na definição de AttractionName. Como já dissemos antes, isso significa que a definição de variável está ligada à definição do atributo, e se mudarmos o tipo de dados do atributo, a variável será automaticamente alterada em conformidade.

Em seguida, adicionamos um botão de evento chamado "List Attractions By Name". Clicamos no botão, que acabamos de adicionar, botão direito do mouse e selecionamos Go to event. A partir daqui, nós precisamos chamar o procedimento que irá imprimir as atrações turísticas dentro do intervalo selecionado.

Já tínhamos o relatório de AttractionsList... mas ele recebeu em um parâmetro o identificador do país, não o intervalo de nome. Vamos salvá-lo com outro nome, AttractionsByName e alterar sua regra Parm, de modo que agora recebe dois parâmetros de entrada: A variável &NameFrom e a variável &NameTo

Definimos as duas variáveis com base no atributo AttractionName.

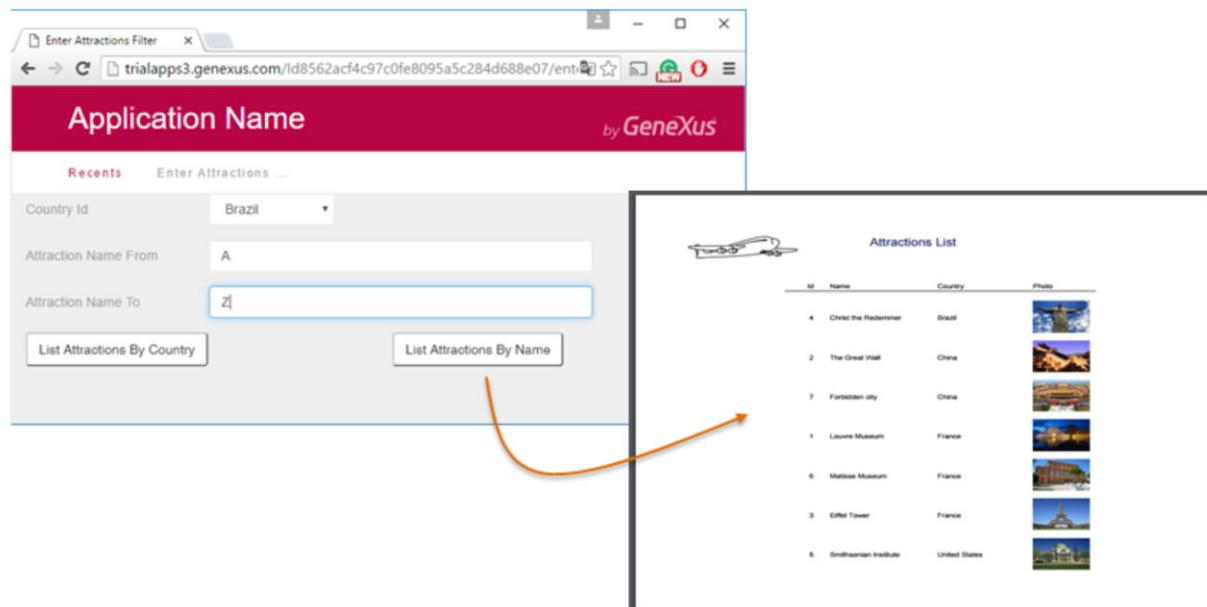
Observe que para as variáveis usamos nomes diferentes daqueles das variáveis que criamos no web panel. O que é mais importante é que os **tipos de dados enviados e recebidos** devem corresponder.

O primeiro parâmetro que escrevemos na chamada será carregado no primeiro parâmetro definido na regra Parm do objeto chamado, e o segundo parâmetro da chamada será carregado no segundo parâmetro

do objeto invocado. Devemos prestar atenção para a ordem usada na invocação e definição da regra Parm. É recomendável usar nomes relacionados, como já fizemos aqui, a fim de entender melhor o código.

Com isso, o procedimento está pronto. Vamos pressionar F5 para executá-lo...

Em execução...



Application Name by GeneXus

Recents Enter Attractions ...








Country Id Brazil

Attraction Name From A

Attraction Name To Z

List Attractions By Country List Attractions By Name

Attractions List

Id	Name	Country	Photo
4	Christ the Redeemer	Brazil	
2	The Great Wall	China	
7	Forbidden city	China	
1	Louvre Museum	France	
6	Musée d'Orsay	France	
3	Eiffel Tower	France	
8	Smithsonian Institution	United States	



Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications