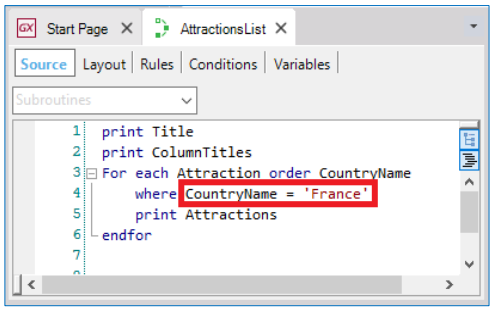
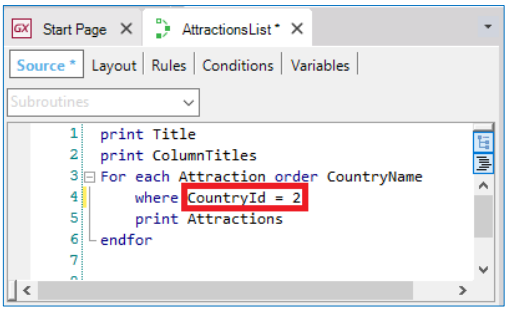
Invocações entre objetos

GeneXus™

GeneXus

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



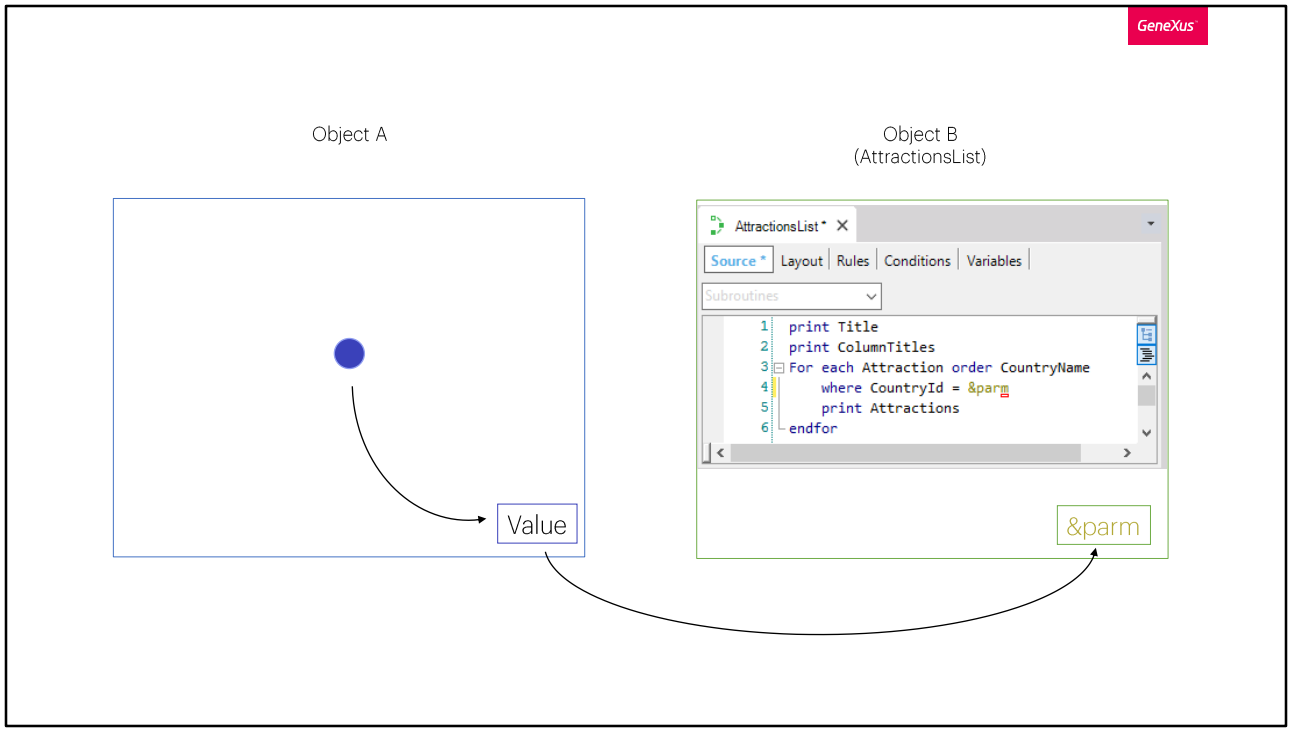


E se quisermos generalizar a lista e poder "receber" de alguma forma o país pelo qual filtrar?...

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!

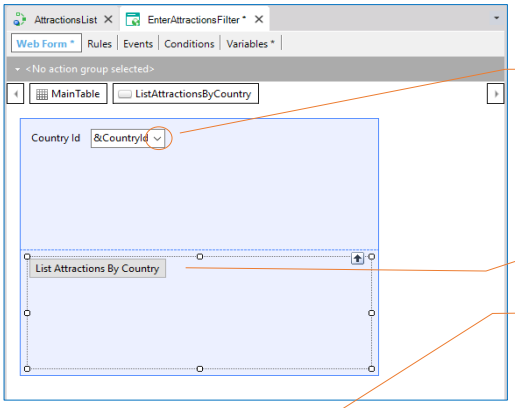


Idealmente, deveríamos ser capazes de "receber" neste objeto o valor pel o qual
queremos filtrar . Em outras palavras, aquele outro objeto GeneXus permit ir ao usuário escolher aquele valor... e depois enviá-lo para este objeto procedimento para ter as atrações listadas de acordo com o país recebido.

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

GeneXus

Definindo a comunicação entre objetos...



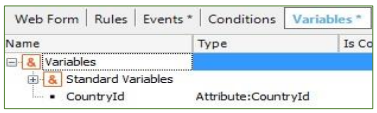
1) Criamos um web panel que peça o país a considerar.

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 ListAttractiosByCountry

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



Event:

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

Primeiro, precisamos criar um objeto que forneça uma tela para solicitar v alores 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 sol icitar 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 f orma, 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 n o 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.

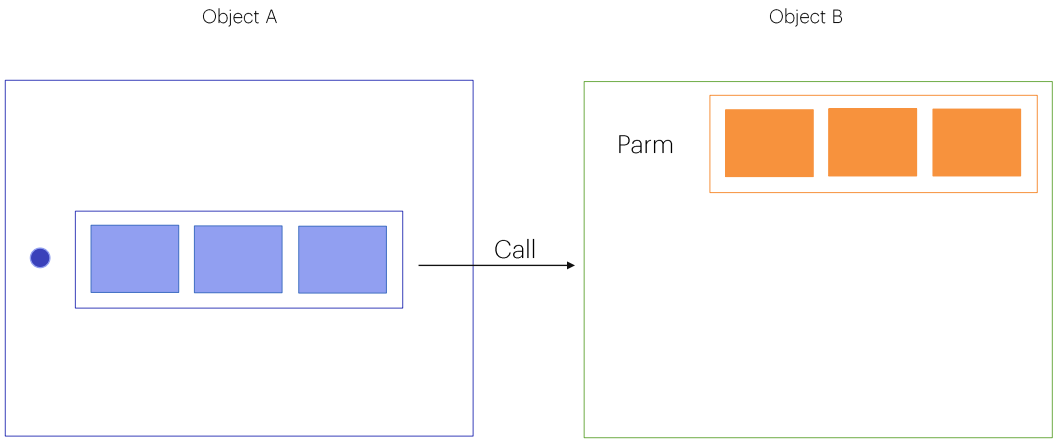
Além disso, adicionamos um botão. Nos solicita que informemos o nome do evento que será associado esse botão. Chamamos de: : “List Attractions By Country”. Vemos que o texto do
botão leva o mesmo nome padrão. Se clicarmos nele, botão direito do mouse, e selecionamos Go to Event...
vemos que um evento com esse nome foi criado e, alterada automaticam ente 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.

GeneXus

Definindo a comunicação entre objetos...



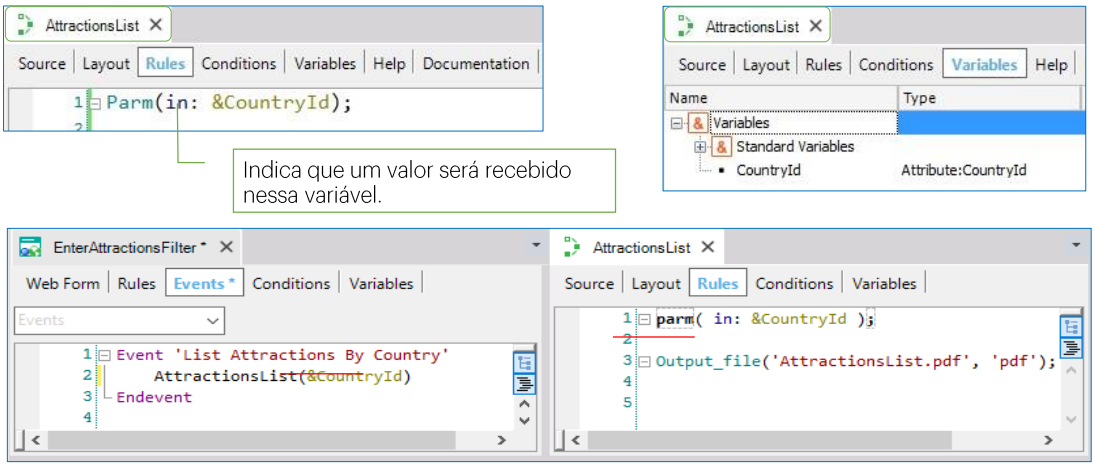
Então, como fazer um objeto A chamar outro objeto B em um determinado momento, enviando valores? Além disso, este objeto B deve ser capaz de carregar em suas variáveis internas os valores enviados a ele, a fim de fazer alguma coisa com essa informação.

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.

GeneXus

Regra Parm

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



Como em nosso exemplo quem vai receber os valores é o objeto procedimento AttractionsList, abrimos o objeto e vamos para sua seção de regras. Escrevemos:
Observemos que na Toolbox são oferecidas todas as regras que poderíamos escrever em um objeto desse tipo. Entre elas, a parm. Poderíamos tê-la arrastado de lá.

Vemos também que nos informa que precisamos substituir isto por um atributo ou variável. Depois veremos o caso dos atributos. Por enquanto, vemos apenas o caso das variáveis.
Com "in" estamos indicando que a variável &CountryId será um parâmetro de entrada. Isto significa que somente será utilizado para receber um valor de quem o chame. Não para devolver. Podemos omitir esta informação e deixar que GeneXus determine.

Escrevemos o nome da variável, mas não a inserimos como variável no objeto. Para fazer isso, uma das maneiras é nos posicionar sobre o nome, clicar com o botão direito do mouse e escolher "Add variable":
Se agora formos para a guia de variáveis, vemos que ela foi definida, com base, por padrão, no tipo do atributo CountryId. Isso ocorre porque a chamamos igual a um atributo.

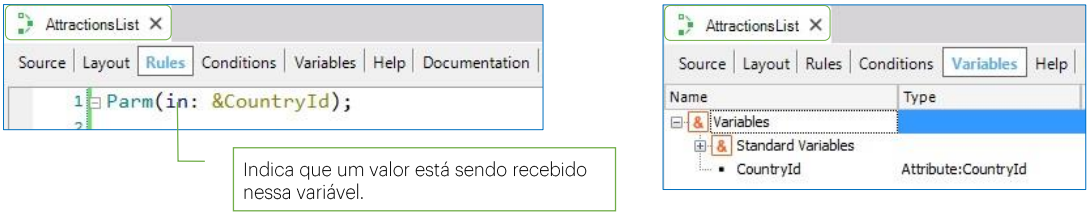
Neste objeto, definimos a variável com o mesmo nome e tipo de dados que usamos no web panel para o usuário inserir o país:
No entanto, como dissemos, são duas variáveis diferentes. Uma válida apenas no web panel e a outra no procedimento. Poderíamos tê-las chamado diferente em ambos os objetos, mas, para que a comunicação e a passagem de informações seja correta, o tipo de dados deve coincidir entre quem chama e quem é chamado.

Agora, nosso objeto procedimento está preparado para receber um identificador de país, neste caso, a partir do webpanel EnterAttractionsFilter.

GeneXus

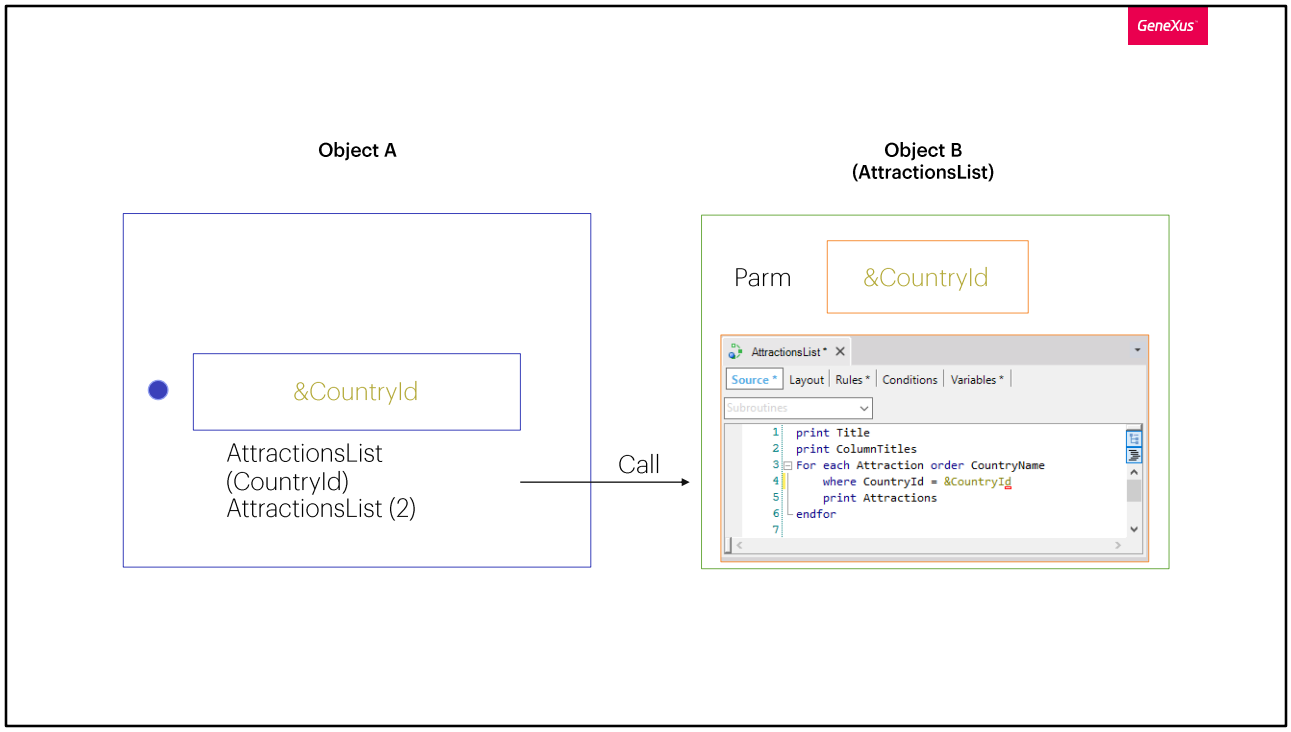
Regra Parm

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



Modificamos o Source:

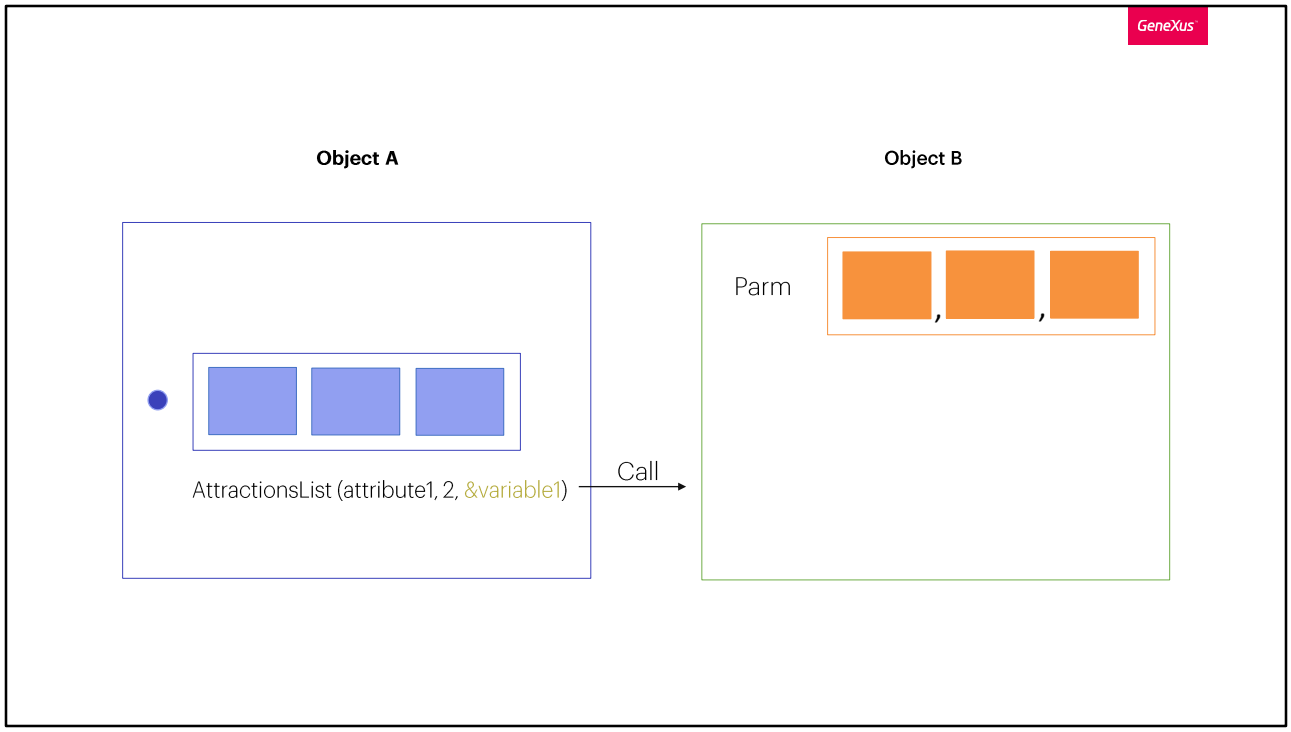
Agora só temos que remover o filtro fixo que tínhamos (país valor 2) no comando For Each e alterá-lo para a variável cujo valor é recebido como um parâmetro.



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.

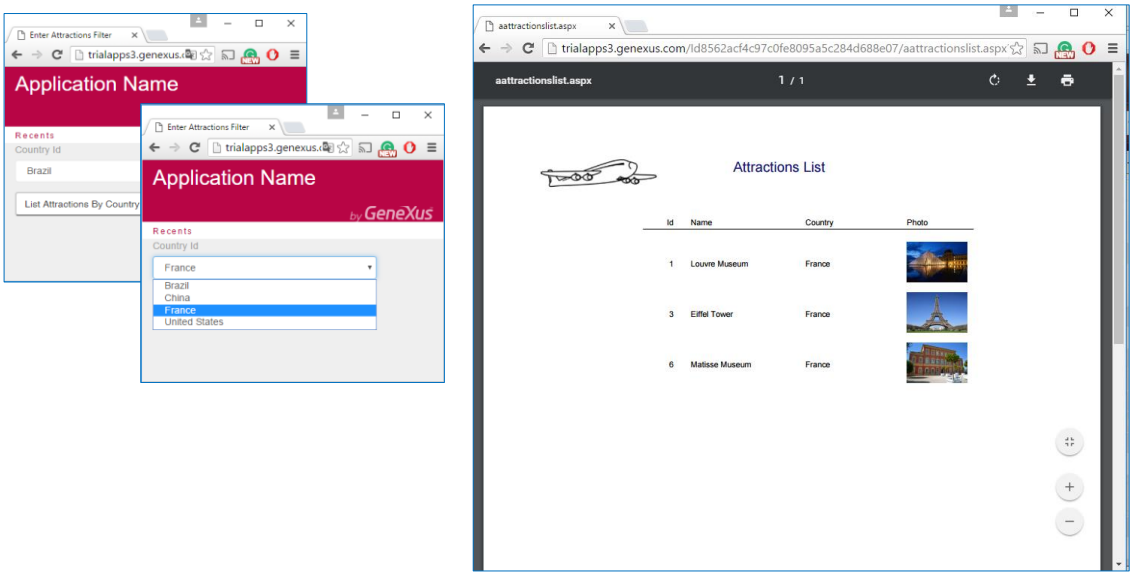
Podemos também enviar um valor.



Ou, se tivermos que enviar dois ou mais valores, enviaríamos 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.



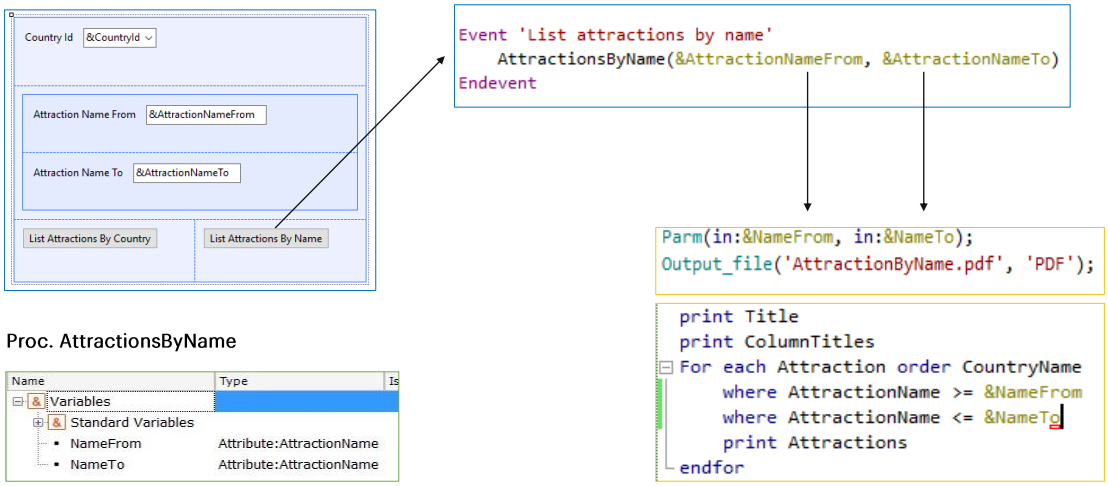
Testamos o que fizemos, pressionando F5. Vemos que o procedimento AttractionsList não é mais exibido. Agora podemos apenas chamá-lo através do web panel...

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 AttractionslList.

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

Listar as atrações em um intervalo de nomes determinado



Proc. AttractionsByName

Agora, vamos supor que queremos listar todas as atrações cujos nomes c orrespondem a um intervalo de valores selecionados pelo usuário. Por exe mplo, 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 lig ada à definição do atributo, e se mudarmos o tipo de dados do atributo, a variável será automaticamente alterada em conformidade.

Luego agregamos un botón de evento “List Attractions By Name”.

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 AttractionsList... mas recebia por parâmetro o identificador do país, não o intervalo de nomes. Vamos salvá-lo com outro nome, AttractionsByName, e modificar sua regra parm, para que agora receba dois parâmetros de entrada: a variável &NameFrom e a variável &NameTo

Temos que defini-las como variáveis e especificar o tipo de dados, então clicamos com o botão direito do mouse sobre a primeira e pressionamos “Add Variable” e editando suas propriedades, a baseamos no atributo AttractionName:

Fazemos o mesmo com &NameTo

Agora, vamos utilizar estas variáveis/parâmetros no For each..., para ficarmos com as atrações que atendam que seu AttractionName seja maior ou igual ao valor da variável &NameFrom, bem como menor ou igual ao valor da variável &NameTo:

Com isto o procedimento está pronto, e somente nos resta a chamá-lo a partir do web panel. Vamos ao webpanel e adicionamos a invocação. Arrastamos o relatório AttractionsByName desta janela para não ter que digitá-lo e, entre parênteses, escrevemos os parâmetros separados por vírgula; neste caso, as variáveis &AttractionNameFrom e &AttractionNameTo que são oferecidas ao usuário na tela.

Observemos que 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 chamado.

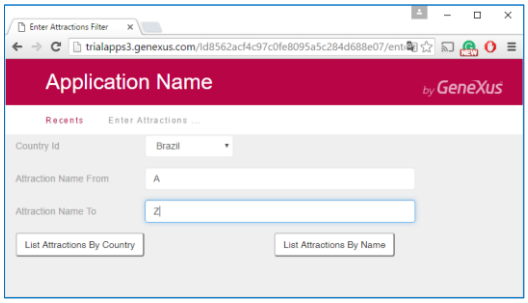
Devemos ser cuidadosos respeitando a ordem na chamada e a definição da regra Parm. É boa prática usar nomes relacionados como fizemos aqui, com o objetivo de entender melhor o código.

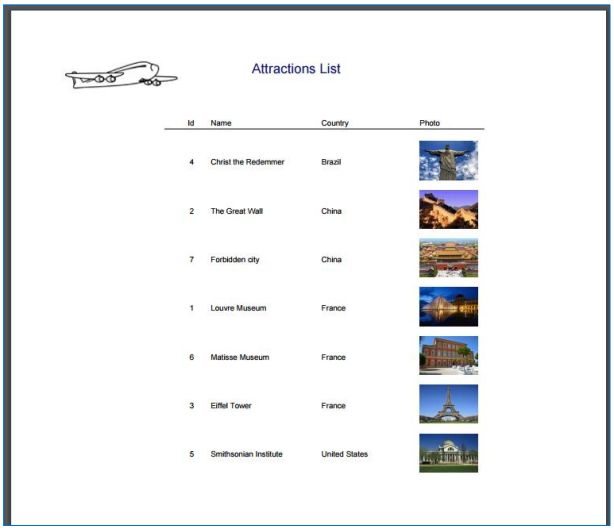
Notemos que os nomes que escolhemos para as variáveis do web panel e

as do procedimento são diferentes.
Como já dissemos, o importante é que os tipos de dados enviados e recebidos coincidam.

Pressionemos F5 para executar.

Em execução...



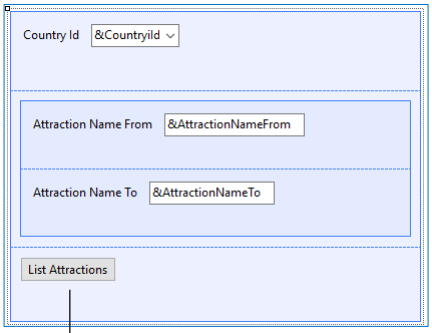


Abrimos o webpanel e, para começar, queremos ver todas as atrações entre “A” e “Z”.
Pressionamos o botão “List Attractions By Name”... e vemos que são listadas todas as atrações.

Agora vamos diminuir um pouco mais o intervalo, colocamos entre ‘A’ e ‘F’. E vemos como funcionou o filtro.

Liste as atrações com um único botão e procedimento.

Web panel




```
Event 'List Attractions'
    AttractionsByNameAndCountry(&CountryId, &AttractionNameFrom, &AttractionNameTo)
Endevent
```

Proc. AttractionByNameAndCountry

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

```
Output_file("AttractionsList.pdf", "pdf");

parm(in: &CountryId, in: &NameFrom, in: &NameTo);
```

```
print Title
print ColumnTitles
For each Attraction order CountryName
    where AttractionName >= &NameFrom when not &NameFrom.IsEmpty()
    where AttractionName <= &NameTo when not &NameTo.IsEmpty()
    where CountryId = &CountryId when not &CountryId.IsEmpty()
    print Attractions
endfor
```

Poderíamos também ter implementado a mesma coisa com um único procedimento em vez de dois, vejamos como.

Primeiro, vamos voltar ao webpanel EnterAttractionsFilter, vamos configurar a propriedade Empty Item da variável CountryId como true e, na propriedade Empty Item Text, inserimos “Select”. Desta forma, sempre que entrarmos neste painel, não haverá um país selecionado por padrão como até agora, mas será exibido o texto “Select” e deverá ser selecionado um.

Agora, criamos um procedimento que será uma mistura entre AttractionsByName e AttractionsList. Para facilitar sua criação, fazemos um “Save As...” de AttractionsByName e o chamamos de AttractionsByNameAndCountry.

Neste novo procedimento, devemos receber três parâmetros, que corresponderão aos três filtros que criamos no webpanel. Portanto, adicionamos àqueles que já inserimos, a variável CountryId e a adicionamos à lista de variáveis deste procedimento.
Na seção Source, adicionamos outra cláusula where para contemplar o filtro por país, where CountryId é igual à variável CountryId.

Agora, precisamos apenas criar o novo botão no web panel para executar este procedimento. Removemos os dois que havíamos criado e geramos um chamado List Attractions.

Clicamos com o botão direito do mouse sobre ele e, em seguida, Go To Event, para programar seu funcionamento.
Comentamos os eventos anteriores, pois não vamos utilizá-los e, a partir do novo evento, invocamos o novo objeto procedimento criado, AttractionsByNameAndCountry, passando por parâmetro as três variáveis que criamos e que são oferecidas ao usuário na tela, CountryId, AttractionNameFrom e AttractionNameTo.
Salvamos as alterações e executamos a aplicação para testar o funcionamento.

Selecionamos, por exemplo, o país França e queremos que nos mostre apenas as atrações com nomes que começam com as letras de “F” a “Z”.
Pressionamos o botão List Attractions e vemos que filtra corretamente o que queremos

Agora, se por exemplo, selecionamos novamente o país França e não inserimos filtros por nome, pois nos interessam todas as atrações deste país, se pressionamos o botão, observamos que nenhuma atração aparece na lista.
O mesmo acontece se não escolhermos nenhum país e queremos filtrar as atrações apenas por nome, independentemente do país. Suponhamos que queremos ver as atrações compreendidas entre “A” e “T”.
Pressionamos o botão List Attractions e vemos que a lista aparece vazia.
Por que isso acontece?
Isso ocorre porque nas cláusulas where do procedimento, não consideramos a possibilidade de que algum dos valores das variáveis pudesse estar vazio, ou seja, que o usuário não insira nenhum valor nesse filtro.
Para resolver isto, utilizamos as cláusulas when.

Ao utilizar a cláusula When, na sequência da definição da cláusula Where, estamos indicando que queremos aplicar uma restrição a esta última.
Veremos em vídeos futuros mais detalhadamente seu funcionamento. Por exemplo, para o caso do nosso primeiro where, adicionamos ao final when not &NameFrom. isEmpty(), ou seja quando a variável NameFrom não tenha valor vazio.
O mesmo fazemos para as outras duas condições.

O que estamos fazendo aqui é indicar que queremos que seja aplicada a condição do where somente quando a variável tenha algum valor atribuído, ou seja, quando não estiver vazia. Caso contrário, não será aplicada esta cláusula e continuará com a linha seguinte.

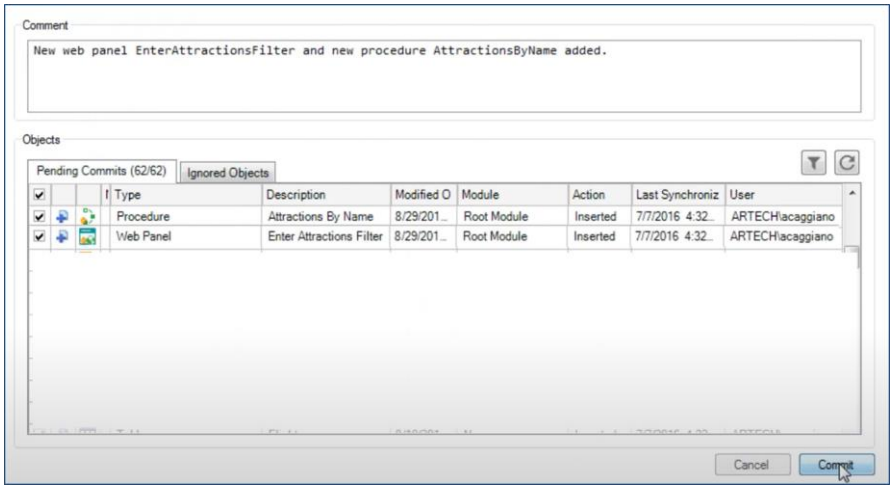
Agora sim, salvamos e executamos novamente com F5

Selecionamos o país França e não aplicaremos filtros por nome.
Pressionamos o botão List Attractions.
A lista com todas as atrações da França aparece como esperávamos.

Agora testaremos deixando o país sem selecionar, e queremos que sejam listadas as atrações compreendidas entre “A” e “G”. Vemos que aparecem na lista todas as atrações que atendem a esta condição, independentemente do país, que é exatamente o que procurávamos.

Como acabamos de ver, fazendo desta maneira, podemos filtrar apenas por país, somente por nome, ou por país e por nome. Além de ter apenas um procedimento e um único botão em nosso web panel.

Enviamos tudo o que fizemos para o GeneXus Server.



Para finalizar, enviamos tudo o que fizemos para o GeneXus Server.

No próximo vídeo veremos outras formas de enviar e receber parâmetros, incluindo os efeitos de colocar um atributo na regra Parm, no lugar de uma variável.

GeneXus™

training.genexus.com
wiki.genexus.com