



# **Preparatório para Certificação Delphi Developer**



**Essa certificação tem ênfase na sintaxe da linguagem, assimilação e utilização da IDE, domínio de técnicas básicas de programação e programação com banco de dados**



# Requisitos

**Não existem pré-requisito, mas é altamente recomendado uma experiência mínima de 2 anos desenvolvendo diariamente com Delphi**



# Muita atenção

# A Prova é toda em Inglês



**Custo da prova é de  
\$59,00**

**<https://www.embarcaderoacademy.com/>**



**A prova é 100% Online através do site**  
**<https://www.embarcaderoacademy.com/>**

***Sem nenhuma restrição ou exigência***



**Você recebe seu resultado imediatamente após a execução do exame**



**Caso não seja aprovado  
você poderá refazer a prova por \$29,00**





**A prova contém 60 perguntas divididas em  
11 grupo de assuntos distintos**



- **Princípios fundamentais do Delphi**
- **Tipos de Dados, variáveis e constantes**
- **Procedimentos e Funções**
- **Classes e Objetos**
- **Rotinas Padrão de I/O**
- **Bibliotecas e Pacotes**
- **Interfaces de Objetos**
- **Gerenciamento de Memória**
- **Generics, Atributos e Métodos Anônimos**
- **Exceptions e Assertions**
- **Desenvolvimento e Conceitos de Banco de Dados**



**A quantidade de perguntas para cada grupo é aleatória, porém existe a obrigatoriedade de pelo menos 1 pergunta de cada grupo seja feita**



**O tempo total para realização da Prova é de 60 minutos o que resulta em 1 minuto para resolução de cada questão**



**Todas as questões são de múltipla escolha**



**Para aprovação o candidato necessita atingir um acerto mínimo de 80% da prova, o que resulta em 48 questões corretas.**



## Guia de Estudo



# Fundamentos do Delphi

**Engloba o conhecimento sobre a linguagem e a ferramenta.**





# Linguagem Delphi

- **Linguagem Compilada**
- **Baseada em Object Pascal**
- **Considerada de Alto nível**
- **Fortemente Tipada**
- **Permite tanto o desenvolvimento estruturado quanto Orientado a Objetos**



# Linguagem Delphi

**Necessário o conhecimento da sintaxe da Linguagem, suas palavras reservadas e suas nuances relacionadas à declaração de variáveis, procedimentos e funções.**



## IDE

- **RAD (Rapid Application Development)**
- **Característica de Arrastar e Soltar componentes**



# IDE

**Necessário o conhecimento sobre as capacidades essenciais da IDE do Delphi e os tipos de arquivos envolvidos no contexto de um projeto Delphi.**



# Tipos de Dados

- **Tipos Simples (String, Integer...)**
- **Tipos Estruturados (Record, Array...)**



# Variáveis

**Local reservado na memória e cujo valor pode ser alterado em tempo de execução do programa.**

**Podendo ser classificada como Global ou Local, quando definida como local, somente fica disponível ao escopo de sua declaração, enquanto que quando definida como global, fica acessível para o contexto da aplicação.**



# Constantes

**Identificadores cujo valor não pode ser alterado em tempo de execução. Por este motivo, o próprio compilador é capaz de identificar seu tipo, a partir de seu valor definido.**



**Nesse tema de **fundamentos do Delphi** é importante um claro entendimento sobre a sintaxe de declaração, suas diferenças e seu uso.**





**Conhecimento sobre Ponteiros, Records, enumerations, matrizes estáticas e dinâmicas (Arrays), String, Char, AnsiChar, AnsiString, WideChar e UnicodeString.**



**Também é necessário um conhecimento das funções da RTL (Run Time Library)**



## Procedimentos e Funções



**São blocos independentes de instruções de código que podem ser chamados a partir de diferentes pontos da aplicação**



**A grande diferença se dá pelo fato de que uma Procedure, quando chamada, não retorna valor, enquanto que uma Function retorna**



**O conhecimento exigido abrange a sintaxe de declaração, suas formas de convenções de chamada, aninhamento (Procedures e Funções Aninhadas) até a sobrecarga de métodos (Overloading). Além disso, outros pontos relacionados também são considerados, tais como parâmetros e valores de retorno (no caso de funções), diretivas e formas de declaração (global e local)**



- Ponteiros : <http://docwiki.embarcadero.com/Libraries/Tokyo/en/System.Pointer>
- Records: [http://docwiki.embarcadero.com/RADStudio/Rio/en/Structured\\_Types\\_\(Delphi\)](http://docwiki.embarcadero.com/RADStudio/Rio/en/Structured_Types_(Delphi))
- Enumerations: [http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Simple\\_Types\\_\(Delphi\)](http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Simple_Types_(Delphi))
- Matrizes [http://docwiki.embarcadero.com/RADStudio/Rio/en/Structured\\_Types\\_\(Delphi\)](http://docwiki.embarcadero.com/RADStudio/Rio/en/Structured_Types_(Delphi))
- Strings [http://docwiki.embarcadero.com/RADStudio/Tokyo/en/String\\_Types\\_\(Delphi\)](http://docwiki.embarcadero.com/RADStudio/Tokyo/en/String_Types_(Delphi))
- RTL [http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Using\\_the\\_RTL\\_\(Run-Time\\_Library\)](http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Using_the_RTL_(Run-Time_Library))
- Funções Aninhadas <https://objectpascalprogramming.com/posts/funcoes-aninhadas/>



## Classes e Objetos





**Uma classe nada mais é que uma abstração para um objeto com características, traduzidos em objetos (métodos), bem como os atributos pertinentes.**



**Um objeto nada mais é que uma instância da classe.**



**Para esse tema é importante o pleno conhecimento dos conceitos de Herança, Abstração, Construtores, Destrutores, Polimorfismo e Visibilidade (Public, Private, Protected, Published...)**



## Rotinas Padrão e I/O



**Esse tema exige conhecimento sobre o uso da RTL (Run Time Library), essa que é a biblioteca de tempo de execução do Delphi.**

**Inclui também a manipulação de Strings e Alocação de Memória.**



**Rotinas de I/O (Input e Output), tais como  
Read/ReadLn e Write/WriteLn.**

**Tratamento de Erros e Manipulação de arquivos e  
diretórios também são abordados.**



## Bibliotecas e Pacotes



**Esse tema aborda a utilização e criação de DLL's  
(Dynamic Link Libraries) e Packages.**





**Uma DLL pode ser definida como sendo uma biblioteca de vínculo dinâmico, cuja representação é um arquivo compilado.**

**Uma vez construída, uma DLL pode ser chamada a partir de qualquer tipo de aplicação.**



**De forma semelhante, às Packages se apresentam como um tipo especial de DLL, exclusivas para Delphi. São representadas com a extensão .bpl (Borland Package Library).**

**As Packages podem ser de dois tipos, Design Time e Runtime, e sua utilização inclui desde a instalação de componentes no IDE ao envolvimento com a aplicação.**



**Para a prova é exigido o conhecimento na criação e utilização de rotinas em DLLs para Windows e para as Packages, deve-se ter o entendimento sobre os fundamentos básicos de um pacote, que envolve convenção de nomenclatura, tipos (Design Time e RunTime) e compreensão das cláusulas contains e required.**



## Interfaces



**Esse tema aborda interfaces de objetos e não as interfaces do usuário (GUI)**



**O conhecimento básico de implementação de interfaces por classes ajudará a se sair bem nesse tópico.**



**O conceito principal de que toda interface representa um contrato e que a classe que a implementa deve respeitar esse contrato, implementando seus procedimentos e métodos.**



## Gerenciamento de Memória





**O Delphi possui seu próprio gerenciador de memória e, conseqüentemente, uma gama de recursos para a realização do processo.**



**É importante conhecer as funções de gerenciamento (automático e manual), bem como a configuração do próprio gerenciador.**



## Outros temas importantes são:

- Tratamento de vazamentos de memória (memory leaks)
- Áreas de memória (Stack e Heap)
- Alocação de Memória para tipos de dados, classes, objetos, matrizes e strings



## **Generics, Atributos e Métodos Anônimos**



**Generics podem ser definidos como sendo estruturas parametrizadas por um tipo à definir.**

**Um exemplo muito utilizado são as listas genéricas `TList<T>` onde “T” será o nosso Generic**



**Atributtes, são um recurso da linguagem Delphi que permitem se fazer “anotações” com objetos especiais em um tipo, como classes e seus membros.**

**Posteriormente essas anotações podem ser lidas via RTTI para realizar as mais variadas funções**



**Métodos Anônimos é um método, seja ele procedure ou function que não tem um nome associado, cujo bloco de código pode ser utilizado como parâmetro de um método ou atribuído a uma variável.**



**A exigência desses 3 temas na prova é de conhecimento bem básico, sua preocupação maior deve ser sobre a sintaxe de declaração de cada uma e suas utilizações mais comuns.**





**O Guia de estudo pontua ainda sobre RTTI, Generics  
Constrains e os tipos de classes de coleção.**



## Exceptions e Assertions



**Uma Exception é uma ação levantada no momento em que ocorre um erro na execução do programa e que acaba por interromper seu fluxo normal.**



**Para se fazer a devida captura e tratamento de erros deste tipo, convertendo-os em simples exceções, basta que a aplicação ou unit em questão faça uso da unit SysUtils.**



**Exceção nada mais é do que um objeto, ou seja, a instância de uma classe.**



**O Delphi possui várias classes de Exceptions, cada qual com sua finalidade e direcionamento como, por exemplo, EMathError, EZeroDivide e EDatabaseError**



**Assertions podem ser definidos basicamente como sendo procedimentos baseados em uma condição e uma mensagem de erro opcional.**

**São indicados para testes de condição.**



**Nesses temas é desejável que você saiba como utilizar e declarar Exceptions e Assertions no Delphi, criar manipuladores de exceção; ter um conhecimento sobre as Classes de Exceção disponíveis, saber como trabalhar com Exceptions aninhadas, como fazer o raise, usar o método Abort, executar exceções silenciosas e ativação e desativação de assertions.**





## Desenvolvimento com Banco de Dados



**Nesse tema o desejado é que você tenha um conhecimento geral sobre os conceitos envolvidos no desenvolvimento Delphi com Banco de Dados, tais como:**



- **Uso dos Componentes de Acesso a Dados**
- **A forma e a sequência dos métodos e eventos disponibilizados por estes componentes.**
- **Uso dos componentes data-aware, os famosos componentes “DB” como TDBEdit, TDBGrid e etc...**
- **Detalhes que envolvam a conexão com o banco de dados a partir do Delphi**
- **Manipulação de Registros no Banco de Dados, Insert, Delete, Update, Select**
- **Conhecimento das classes TDataset e TField**



## PERGUNTAS E RESPOSTAS DA PROVA DE CERTIFICAÇÃO



**Ao criarmos mensagens customizadas para serem transmitidas entre aplicações Windows, é recomendável utilizarmos valores acima da constante base WM\_APP. Qual o valor (decimal) desta constante?**

- 1024
- 2048
- 8000
- 32768
- 65535



**Ao criarmos mensagens customizadas para serem transmitidas entre aplicações Windows, é recomendável utilizarmos valores acima da constante base WM\_APP. Qual o valor (decimal) desta constante?**

- 1024
- 2048
- 8000
- 32768
- 65535

[http://docwiki.embarcadero.com/RADStudio/XE2/en/Declaring\\_a\\_Message\\_Identifier](http://docwiki.embarcadero.com/RADStudio/XE2/en/Declaring_a_Message_Identifier)



## **Qual a tecla de atalho e nome do diálogo que permite o gerenciamento de threads em tempo de debug?**

- Ctrl+Shift+T; Thread Status
- Ctrl+Alt+T; Thread Status
- Ctrl+T; Thread Manager
- Ctrl+Shift+T; Thread Manager
- Ctrl+Alt+T; Thread Manager



## Qual a tecla de atalho e nome do diálogo que permite o gerenciamento de threads em tempo de debug?

- Ctrl+Shift+T; Thread Status
- **Ctrl+Alt+T; Thread Status**
- Ctrl+T; Thread Manager
- Ctrl+Shift+T; Thread Manager
- Ctrl+Alt+T; Thread Manager

<http://docwiki.embarcadero.com/RADStudio/en/Threads>





**O Record TStopClock permite um controle mais preciso ao monitorarmos a execução de rotinas em nossos sistemas. Qual a unit deve ser declarada para utilizarmos este recurso?**

- Diagnostics
- Precision
- Classes
- SysUtils
- Este Record não existe



**O Record TStopClock permite um controle mais preciso ao monitorarmos a execução de rotinas em nossos sistemas. Qual a unit deve ser declarada para utilizarmos este recurso?**

- Diagnostics
- Precision
- Classes
- SysUtils
- Este Record não existe

O record TStopclock não existe. Na realidade, o record em questão se chama TStopwatch, contido na unit Diagnostics.

<http://docwiki.embarcadero.com/VCL/en/Diagnostics.TStopwatch>



## Considerando o código a seguir, qual a função da diretiva default utilizada na declaração da propriedade XYZ?

```
TMinhaClasse = class(TComponent)  
property XYZ : Integer read FXYZ write SetXYZ default 1;
```

- Informar que propriedade não deve ser persistida no DFM, quando o seu valor for 1
- Informar que a propriedade XYZ é a propriedade default para objetos da classe
- Inicializar a propriedade XYZ com o valor 1 quando um objeto da classe for iniciado
- Nenhuma das anteriores



## Considerando o código a seguir, qual a função da diretiva default utilizada na declaração da propriedade XYZ?

```
TMinhaClasse = class(TComponent)  
property XYZ : Integer read FXYZ write SetXYZ default 1;
```

- Informar que propriedade não deve ser persistida no DFM, quando o seu valor for 1
- Informar que a propriedade XYZ é a propriedade default para objetos da classe
- Inicializar a propriedade XYZ com o valor 1 quando um objeto da classe for iniciado
- Nenhuma das anteriores

A diretiva default, neste caso, indica que a propriedade não deve ser persistida no DFM se o seu valor for igual a 1.

[http://docwiki.embarcadero.com/RADStudio/en/Default\\_Property\\_Values](http://docwiki.embarcadero.com/RADStudio/en/Default_Property_Values)



**Considerando que o código abaixo foi retirado de um método pertencente a uma classe herdada de TComponent, quais os tipos de variáveis que devemos utilizar para as variáveis locais XML1 e XML2, respectivamente?**

**Var**

**XML1 : ?**

**XML2 : ?**

**begin**

**XML1 := TXMLDocument.Create(Self);**

**XML2 := TXMLDocument.Create(nil);**

**end;**

- TXMLDocument e TXMLDocument
- IXMLDocument e TXMLDocument
- TXMLDocument e IXMLDocument
- IXMLDocument e IXMLDocument
- Todas as Anteriores



**Considerando que o código abaixo foi retirado de um método pertencente a uma classe herdada de TComponent, quais os tipos de variáveis que devemos utilizar para as variáveis locais XML1 e XML2, respectivamente?**

**Var**

XML1 : ?

XML2 : ?

**begin**

XML1 := TXMLDocument.Create(Self);

XML2 := TXMLDocument.Create(nil);

**end;**

- TXMLDocument e TXMLDocument
- IXMLDocument e TXMLDocument
- **TXMLDocument e IXMLDocument**
- IXMLDocument e IXMLDocument
- Todas as Anteriores

Objetos do tipo TXMLDocument se comportam como interfacedobjects quando são instanciados sem owner, e devem ser atribuídos a variáveis do tipo interface.

<http://docwiki.embarcadero.com/VCL/en/XMLDoc.TXMLDocument>



**Assim como DLL's, pacotes compartilham a área de dados global com a aplicação:**

- Verdadeiro
- Falso



**Assim como DLL's, pacotes compartilham a área de dados global com a aplicação:**

- Verdadeiro
- Falso

A afirmativa pode até ser verdadeira quanto ao gerenciamento de memória do pacote, mas a premissa do funcionamento da DLL está incorreta  
[http://docwiki.embarcadero.com/RADStudio/en/Packages\\_and\\_Standart\\_DLLs](http://docwiki.embarcadero.com/RADStudio/en/Packages_and_Standart_DLLs)





**O Componente TSQLConnection (da paleta dbExpress) permite a conexão com diversas fontes de dados. Falando especificamente de conexões com banco de dados, como podemos configurar um TSQLConnection para utilizar o recurso de pooling de conexões?**

- Devemos configurar duas conexões. Uma conexão com o driver DBXPool (A), e outra com os dados do SGBD a ser conectado (B). Por fim, o parâmetro DelegateConnection de B recebe o nome da conexão A.
- Devemos configurar duas conexões. Uma conexão com o driver DBXPool (A), e outra com os dados do SGBD a ser conectado (B). Por fim, o parâmetro DelegateConnection de A recebe o nome da Conexão B
- Basta configurar uma conexão com os dados do SGBD a ser conectado e adicionar os parâmetros MinConnections e MaxConnection desejados.
- O Pooling de conexões só pode ser configurado no SGBD
- Nenhuma das anteriores.



**O Componente TSQLConnection (da paleta dbExpress) permite a conexão com diversas fontes de dados. Falando especificamente de conexões com banco de dados, como podemos configurar um TSQLConnection para utilizar o recurso de pooling de conexões?**

- Devemos configurar duas conexões. Uma conexão com o driver DBXPool (A), e outra com os dados do SGBD a ser conectado (B). Por fim, o parâmetro `DelegateConnection` de B recebe o nome da conexão A.
- Devemos configurar duas conexões. Uma conexão com o driver DBXPool (A), e outra com os dados do SGBD a ser conectado (B). Por fim, o parâmetro `DelegateConnection` de A recebe o nome da Conexão B
- Basta configurar uma conexão com os dados do SGBD a ser conectado e adicionar os parâmetros `MinConnections` e `MaxConnection` desejados.
- O Pooling de conexões só pode ser configurado no SGBD
- Nenhuma das anteriores.

Duas conexões devem ser configuradas, uma conexão com o driver específico para Pool, e outra para o SGBD que queremos conectar, por fim, a conexão com o SGBD deve receber a primeira conexão em sua propriedade `DelegateConnection`

[http://docwiki.embarcadero.com/RADStudio/en/Connection\\_Pooling\\_\(Delegate\\_Drivers\\_tutorial\)](http://docwiki.embarcadero.com/RADStudio/en/Connection_Pooling_(Delegate_Drivers_tutorial))



**O DBX framework permite o acesso a dados em uma camada mais próxima ao DGBD. A classe TDBXConnectionFactory pode ser utilizada para criar objetos de conexão através de configurações previamente cadastradas. No entanto, conexões customizadas podem ser facilmente configuradas e criadas, com o auxílio da seguinte classe:**

- TDBXParams
- TDBXParameters
- TDBXConfig
- TDBXProperties
- NDA



**O DBX framework permite o acesso a dados em uma camada mais próxima ao DGBD. A classe TDBXConnectionFactory pode ser utilizada para criar objetos de conexão através de configurações previamente cadastradas. No entanto, conexões customizadas podem ser facilmente configuradas e criadas, com o auxílio da seguinte classe:**

- TDBXParams
- TDBXParameters
- TDBXConfig
- **TDBXProperties**
- NDA

TDBXProperties permite a criação de uma conexão customizada (em runtime) através do DBX Framework.

<http://docwiki.embarcadero.com/Libraries/en/Data.DBXCommon.TDBXProperties>



**O procedimento RegisterPropertyEditor pode ser utilizado para registrar editores de propriedades para um tipo de propriedade, sem necessariamente informar o nome da propriedade:**

- Verdadeiro
- Falso



**O procedimento RegisterPropertyEditor pode ser utilizado para registrar editores de propriedades para um tipo de propriedade, sem necessariamente informar o nome da propriedade:**

- Verdadeiro
- Falso

Sim, podemos registrar um editor de propriedades sem necessariamente atrelá-lo a uma propriedade específica, ficando o editor de propriedade ativo para todas as propriedades de um determinado tipo.

<http://docwiki.embarcadero.com/VCL/en/DesignIntf.RegisterPropertyEditor>



**O Datasnap permite a inclusão de filtros em seus canais de comunicação.  
Quais filtros são nativos no Delphi XE?**

- PC1
- RC6
- RSA
- 7ZipCompression
- ZLibCOmpression



**O Datasnap permite a inclusão de filtros em seus canais de comunicação.  
Quais filtros são nativos no Delphi XE?**

- PC1
- RC6
- RSA
- 7ZipCompression
- ZLibCOmpression

Os filtros nativos no Delphi são PC1 (Criptografia), RSA (Criptografia) e ZLibCompression (Compressão).

[http://docwiki.embarcadero.com/RADStudio/en/Filtering\\_Datasnap\\_Byte\\_Stream](http://docwiki.embarcadero.com/RADStudio/en/Filtering_Datasnap_Byte_Stream)





## Qual das alternativas apresenta a definição do método `GetEnumName` da `unit TypInfo`?

- `function GetEnumName(EnumInfo : Pointer; Value : TEnum) : String;`
- `function GetEnumName(TypeInfo : Pointer; Value : TEnum) : String;`
- `function GetEnumName(TypeInfo : PTypeInfo; Value TEnum) : String;`
- `function GetEnumName(TypeInfo : Pointer; Value : Integer) : String;`
- `function GetEnumName(TypeInfo : PTypeInfo : Value : Integer) : String;`



## Qual das alternativas apresenta a definição do método `GetEnumName` da unit `TypeInfo`?

- `function GetEnumName(EnumInfo : Pointer; Value : TEnum) : String;`
- `function GetEnumName(TypeInfo : Pointer; Value : TEnum) : String;`
- `function GetEnumName(TypeInfo : PTypeInfo; Value TEnum) : String;`
- `function GetEnumName(TypeInfo : Pointer; Value : Integer) : String;`
- `function GetEnumName(TypeInfo : PTypeInfo; Value : Integer) : String;`

O método `GetEnumName` retorna o nome de uma determinado valor enumerado. Para isto, deve ser passado um ponteiro para o `TypeInfo` do tipo e valor no formato ordinal

<http://docwiki.embarcadero.com/VCL/en/TypeInfo.GetEnumName>



**Ao criarmos uma aplicação do tipo serviço, o Delphi estende uma classe base, onde implementamos as rotinas específicas do novo serviço. Qual o nome da classe base para serviços no Delphi.**

- TBackgroundApplication.
- TService
- TServiceApplication
- Nenhuma das Anteriores



**Ao criarmos uma aplicação do tipo serviço, o Delphi estende uma classe base, onde implementamos as rotinas específicas do novo serviço. Qual o nome da classe base para serviços no Delphi.**

- TBackgroundApplication.
- **TService**
- TServiceApplication
- Nenhuma das Anteriores

A classe TService encapsula as chamadas da API do Windows para a criação de serviços

<http://docwiki.embarcadero.com/VCL/en/SvcMgr.TService>



**Qual o nome da unit e respectiva função que nos dão acesso à área de transferência do Windows?**

- Clipboard.GetClipboard;
- Clipbrd.GetClipboard;
- Clipbrd.Clipboard;
- System.Clipboard;



## Qual o nome da unit e respectiva função que nos dão acesso à área de transferência do Windows?

- Clipboard.GetClipboard;
- Clipbrd.GetClipboard;
- **Clipbrd.Clipboard;**
- System.Clipboard;

O método Clipboard da unit Clipbrd retorna um objeto que interage com a área de transferência do Windows

<http://docwiki.embarcadero.com/VCL/en/Clipbrd.Clipboard>



**Podemos dizer que o TXMLDocument, responsável pela manipulação de arquivos XML, encontra-se na unit?**

- XMLDocument.pas
- XisEMLDoc.pas
- XMLDoc.pas
- Nenhuma das Anteriores, ele faz parte do Kernel do Delphi.



## Podemos dizer que o TXMLDocument, responsável pela manipulação de arquivos XML, encontra-se na unit?

- XMLDocument.pas
- XisEMLDoc.pas
- **XMLDoc.pas**
- Nenhuma das Anteriores, ele faz parte do Kernel do Delphi.

O Componente TXMLDocument pode representar e manipular um XML em sua totalidade. O Componente pode ser encontrado na unit XMLDoc.  
<http://docwiki.embarcadero.com/VCL/en/XMLDoc.TXMLDocument>





## Qual o tipo de retorno do método `TXMLDocument.AddChild`?

- `IXMLNode`
- `TXMLNode`
- `TXMLChild`
- `TXMLChildNode`



## Qual o tipo de retorno do método `TXMLDocument.AddChild`?

- `IXMLNode`
- `TXMLNode`
- `TXMLChild`
- `TXMLChildNode`

O método `AddChild` é declarado inicialmente na interface `IXMLDocument`, sendo que seu retorno é do tipo `IXMLNode`.

<http://docwiki.embarcadero.com/VCL/en/XMLDoc.TXMLDocument.AddChild>



**Em qual arquivo de um projeto VCL Forms Application é armazenado o ícone que será atribuído a aplicação?**

- MeuProjeto.pas
- MeuProjeto.res
- MeuProjeto.ico
- MeuProjeto.dpr
- MeuProjeto.dproj



## Em qual arquivo de um projeto VCL Forms Application é armazenado o ícone que será atribuído a aplicação?

- MeuProjeto.pas
- **MeuProjeto.res**
- MeuProjeto.ico
- MeuProjeto.dpr
- MeuProjeto.dproj

O ícone da aplicação é armazenado dentro do arquivo de recursos da aplicação

[http://docwiki.embarcadero.com/RADStudio/en/Language\\_Overview](http://docwiki.embarcadero.com/RADStudio/en/Language_Overview)



## Qual a classe inicial para manipulação da RTTI?

- TRTTIMETHOD
- TRTTIFIELD
- TRTTICONTEXT
- TRTTITYPE



## Qual a classe inicial para manipulação da RTTI?

- TRTTIMETHOD
- TRTTIFIELD
- **TRTTICONTEXT**
- TRTTITYPE

Por mais que TRttiContext se pareça e se comporte como uma classe, TRttiContext é um Record. Uma “instância” de TRttiContext deve ser utilizada como ponto inicial para a leitura de dados sobre tipos públicos da aplicação.

<http://docwiki.embarcadero.com/VCL/en/RTTI.TRttiContext>



## Qual a principal diferença entre JSON e XML?

- XML possibilita maior quantidade de registros num único arquivo
- JSON tem a sintaxe mais simples e arquivo final mais leve
- Por ser mais velho o JSON, está melhor estruturado
- Pela estrutura robusta o XML é interpretado mais rapidamente



## Qual a principal diferença entre JSON e XML?

- XML possibilita maior quantidade de registros num único arquivo
- **JSON tem a sintaxe mais simples e arquivo final mais leve**
- Por ser mais velho o JSON, está melhor estruturado
- Pela estrutura robusta o XML é interpretado mais rapidamente

A serialização de dados em JSON gera um resultado sensivelmente mais leve que um arquivo XML

<http://docwiki.embarcadero.com/RADStudio/en/JSON>





**Quais das seguintes alternativas o parâmetro tem seu valor inicial descartado e pode apenas ser utilizado para retornar valores?**

- a) `procedure ex(out AString);`
- b) `procedure ex(var AString);`
- c) `procedure ex(in AString);`
- d) `procedure ex(const AString);`



**Quais das seguintes alternativas o parâmetro tem seu valor inicial descartado e pode apenas ser utilizado para retornar valores?**

- a) `procedure ex(out AString);`
- b) `procedure ex(var AString);`
- c) `procedure ex(in AString);`
- d) `procedure ex(const AString);`

Parâmetros do tipo out devem ser utilizados somente para saída de valores de um método, nunca como entrada.

[http://docwiki.embarcadero.com/RADStudio/en/Parameters\\_\(Delphi\)#Out\\_Parameters](http://docwiki.embarcadero.com/RADStudio/en/Parameters_(Delphi)#Out_Parameters)



**Ao declararmos uma classe, método ou propriedade qual conjunto de teclas quando pressionado executa o comando “Class Completion” no code editor, encarregado de completar a declaração?**

- a) Ctrl+Shift+Space
- b) Ctrl+Alt+C
- c) Ctrl+Shift+C
- d) Ctrl+K+I



**Ao declararmos uma classe, método ou propriedade qual conjunto de teclas quando pressionado executa o comando “Class Completion” no code editor, encarregado de completar a declaração?**

- a) Ctrl+Shift+Space
- b) Ctrl+Alt+C
- c) **Ctrl+Shift+C**
- d) Ctrl+K+I

Ctrl+Shift+C é o atalho para o comando “Class Completion”

[http://docwiki.embarcadero.com/RADStudio/en/Using\\_Class\\_Competition](http://docwiki.embarcadero.com/RADStudio/en/Using_Class_Competition)



**Quais das seguintes opções é uma declaração válida de um Array dinâmico?**

- a) `var vetor[0.. 100] of string;`
- b) `var vetor of string;`
- c) `var vetor: array of string;`
- d) `var vetor: array[0.. 100] of string;`



## Quais das seguintes opções é uma declaração válida de um Array dinâmico?

- a) `var vetor[0.. 100] of string;`
- b) `var vetor of string;`
- c) `var vetor: array of string;`
- d) `var vetor: array[0.. 100] of string;`

A opção c declara corretamente um vetor dinâmico.

[http://docwiki.embarcadero.com/RADStudio/en/Structure\\_Type#Dynamic\\_Arrays](http://docwiki.embarcadero.com/RADStudio/en/Structure_Type#Dynamic_Arrays)



## O método “SetLength” pode ser utilizado para?

- a) Alterar o tamanho de uma Classe
- b) Alterar o tamanho de um Array
- c) Aumentar o tamanho de um Inteiro
- d) Alterar o tamanho de um arquivo



## O método “SetLength” pode ser utilizado para?

- a) Alterar o tamanho de uma Classe
- b) **Alterar o tamanho de um Array**
- c) Aumentar o tamanho de um Inteiro
- d) Alterar o tamanho de um arquivo

O comando SetLength permite a alteração do tamanho de um array dinâmico. Outra possibilidade de uso deste comando está em setar o tamanho de uma string

<http://docwiki.embarcadero.com/VCL/en/System.SetLength>





## Qual a utilidade da propriedade MainFormOnTaskBar?

- a) Disponibiliza opções do formulário (como Maximize, Minimize, Close...) principal na task bar
- b) Determina a janela que será associada ao botão da aplicação na task bar
- c) Quando setada para True, minimiza a aplicação deixando-a disponível somente na task bar
- d) Quando setada para True, restaura e exibe o formulário principal da aplicação



## Qual a utilidade da propriedade MainFormOnTaskBar?

- a) Disponibiliza opções do formulário (como Maximize, Minimize, Close...) principal na task bar
- b) **Determina a janela que será associada ao botão da aplicação na task bar**
- c) Quando setada para True, minimiza a aplicação deixando-a disponível somente na task bar
- d) Quando setada para True, restaura e exibe o formulário principal da aplicação

A propriedade MainFormOnTaskBar determina qual janela será associada ao botão da aplicação na task bar. Esta opção foi criada (na versão 2007 do Delphi) para possibilitar, entre outras coisas, a utilização de novos recursos introduzidos pelo Windows Aero. Quando a propriedade é setada para False, a aplicação apresenta o mesmo comportamento de aplicações compiladas antes do Delphi 2007, ou seja, a Task Bar apresenta as informações da janela escondida da aplicação (objeto Application). Quando True, a Task Bar apresenta informações do formulário principal da aplicação (como a caption). Outro efeito da propriedade (quando ligada) pode ser visto quando executamos o Flip 3D (tecla Windows + Tab), as aplicações feitas em Delphi somente são exibidas no Flip 3D quando a propriedade MainFormOnTaskBar está setada para True.

<http://docwiki.embarcadero.com/VCL/en/Forms.TApplication.MainFormOnTaskBar>



**Qual a classe da VCL introduz a propriedade de Padding?**

- a) TPersistent
- b) TComponent
- c) TControl
- d) TGraphicControl
- e) TWinControl



## Qual a classe da VCL introduz a propriedade de Padding?

- a) TPersistent
- b) TComponent
- c) TControl
- d) TGraphicControl
- e) TWinControl

A propriedade Padding permite a criação de uma margem interna para componentes do tipo Container, evitando que os componentes filho se alinhem além da margem especificada. Como somente componentes do tipo TWinControl podem ser containers, a resposta correta é “e”.

<http://docwiki.embarcadero.com/VCL/en/Controls.TwinControl.Padding>



## Selecione todas as alternativas corretas:

- a) A Uses que devemos ter para utilizar String RawByteString é System.
- b) O novo tipo de dados String RawByteString foi incorporado a partir do Delphi 2010, pelo motivo de possibilitar o suporte de unicode.
- c) A runtime library é utilizada somente para manipular de forma centralizada os conjuntos de caracteres internacionais para Unicode
- d) Pode-se utilizar o tipo String RawByteString para armazenamento de Blobs
- e) A Uses que devemos ter para utilizar StringRawByteString é Strings.



## Selecione todas as alternativas corretas:

- a) A Uses que devemos ter para utilizar String RawByteString é System.
- b) O novo tipo de dados String RawByteString foi incorporado a partir do Delphi 2010, pelo motivo de possibilitar o suporte de unicode.
- c) A runtime library é utilizada somente para manipular de forma centralizada os conjuntos de caracteres internacionais para Unicode
- d) Pode-se utilizar o tipo String RawByteString para armazenamento de Blobs
- e) A Uses que devemos ter para utilizar StringRawByteString é Strings.

O tipo RawByteString está contido na unit System. O objetivo do RawByteString é reduzir a necessidade de várias sobrecargas de procedimentos que lêem dados do tipo string;

<http://docwiki.embarcadero.com/VCL/en/System.RawByteString>



## Sobre o método Constructor podemos afirmar:

- I. Constructors são métodos especiais que criam e instanciam objetos.
- II. Uma classe pode ter um único método construtor.
- III. Caso ocorra uma exceção durante a execução do método Constructor o método destrutor é chamado automaticamente.

Selecione a alternativa correta:

- a) Apenas III está correta.
- b) Apenas I está correta.
- c) Apenas II está correta.
- d) I e III estão corretas.
- e) I e II estão corretas.



## Sobre o método Constructor podemos afirmar:

- I. Constructors são métodos especiais que criam e instanciam objetos.
- II. Uma classe pode ter um único método construtor.
- III. Caso ocorra uma exceção durante a execução do método Constructor o método destrutor é chamado automaticamente.

Selecione a alternativa correta:

- a) Apenas III está correta.
- b) Apenas I está correta.
- c) Apenas II está correta.
- d) I e III estão corretas.
- e) I e II estão corretas.

Um construtor é um método definido pela palavra constructor, embora a declaração não especifique nenhum valor de retorno, um construtor retorna uma referência ao objeto criado por ele, sempre do mesmo tipo da classe onde ele é declarado.

Uma classe pode conter mais de um construtor, desde que ambos tenham parâmetros de entrada diferentes e contenha a diretiva overload, ou os nomes utilizados para os métodos sejam diferentes, uma vez que o nome "Create" não é obrigatório, podendo ser utilizado o nome que preferir para seu construtor.

Vale ressaltar também que caso ocorra um erro no momento da execução do construtor, o método Destroy é chamado automaticamente.

<http://docwiki.embarcadero.com/RADStudio/en/Methods#Constructors>





**“...são um conjunto de ferramentas de abstração que permitem a separação de um algoritmo (como um procedimento ou função) ou uma estrutura de dados (como uma classe, interface ou record) de um ou mais tipos específicos que o algoritmo ou estrutura de dados utiliza”**

O Texto acima se refere a:

- a) Métodos anônimos
- b) Generic
- c) Polimorfismo
- d) Class Helpers



**“...são um conjunto de ferramentas de abstração que permitem a separação de um algoritmo (como um procedimento ou função) ou uma estrutura de dados (como uma classe, interface ou record) de um ou mais tipos específicos que o algoritmo ou estrutura de dados utiliza”**

O Texto acima se refere a:

- a) Métodos anônimos
- b) Generic**
- c) Polimorfismo
- d) Class Helpers

“O termo “Generics” ou Tipos Genéricos são um conjunto de ferramentas de abstração que permitem a separação de um algoritmo (como um procedimento ou função) ou uma estrutura de dados (como um classe, interface ou record) de um ou mais tipos específicos que o algoritmo ou estrutura de dados utiliza”

[http://docwiki.embarcadero.com/RADStudio/en/Overview\\_of\\_Generics](http://docwiki.embarcadero.com/RADStudio/en/Overview_of_Generics)



**Quando é necessário que nada mais seja executado após uma determinada condição em meio a um procedimento, poderá ser feita a seguinte chamada:**

- a) Exit;
- b) Pause;
- c) Abort;
- d) End;



**Quando é necessário que nada mais seja executado após uma determinada condição em meio a um procedimento, poderá ser feita a seguinte chamada:**

- a) Exit;
- b) Pause;
- c) **Abort;**
- d) End;

A principal diferença entre Exit e Abort é que o comando Exit finalizará o procedimento em questão, porém a execução continuará a partir do ponto onde o procedimento foi chamado, enquanto o Abort irá interromper toda a execução a partir de sua chamada.

<http://docwiki.embarcadero.com/RADStudio/en/Abort>

<http://docwiki.embarcadero.com/VCL/en/System.Exit>



**Que método quando executado abre uma caixa de diálogo contendo uma caixa de edição permitindo que seja inserido uma string, um integer ou um double?**

- a) InputText
- b) InputValue
- c) InputQuery
- d) InputTextValue



**Que método quando executado abre uma caixa de diálogo contendo uma caixa de edição permitindo que seja inserido uma string, um integer ou um double?**

- a) InputText
- b) InputValue
- c) **InputQuery**
- d) InputTextValue

Quando executamos o método InputQuery é aberto uma caixa de diálogo contendo uma caixa de edição permitindo que seja inserido uma string, um integer ou um double. InputQuery retorna True quando clicado no botão OK e False quando clicando no botão Cancel.

<http://docwiki.embarcadero.com/VCL/en/Dialogs.InputQuery>



**Para criarmos um diretório incluindo todas as pastas intermediárias quando estas não existirem utilizamos que método?**

- a) CreateDirectories
- b) ForceDirectories
- c) CreateDir
- d) ExtractDir



**Para criarmos um diretório incluindo todas as pastas intermediárias quando estas não existirem utilizamos que método?**

- a) CreateDirectories
- b) ForceDirectories**
- c) CreateDir
- d) ExtractDir

ForceDirectories cria um novo diretório conforme seja especificado no seu parâmetro "Dir, caso seja necessário criar mais de uma pasta para completar o caminho do diretório o método se encarrega de criá-las.

<http://docwiki.embarcadero.com/VCL/en/SysUtils.ForceDirectories>





## O que irá indicar a diretiva `{ $DENYPACKGEUNIT ON }` ao ser declarada em uma unit?

- a) Que está unit poderá apenas ser utilizada dentro de um pacote, seja ele de Design-time ou Run-time.
- b) Que está unit poderá apenas ser utilizada dentro de um pacote de Design-time.
- c) Que está unit poderá apenas ser utilizada dentro de um pacote de Run-time.
- d) Que está unit não poderá ser utilizada dentro de um pacote.



## O que irá indicar a diretiva `{ $DENYPACKGEUNIT ON }` ao ser declarada em uma unit?

- a) Que está unit poderá apenas ser utilizada dentro de um pacote, seja ele de Design-time ou Run-time.
- b) Que está unit poderá apenas ser utilizada dentro de um pacote de Design-time.
- c) Que está unit poderá apenas ser utilizada dentro de um pacote de Run-time.
- d) **Que está unit não poderá ser utilizada dentro de um pacote.**

A diretiva `{ $DENYPACKAGEUNIT ON }` impede que uma unit possa ser utilizada dentro de um pacote

<http://docwiki.embarcadero.com/VCL/en/Packages>



**As DLLs são significativamente mais limitadas do que os pacotes quando consideramos o que elas podem exportar, uma vez que DLL não pode exportar constantes, tipos e variáveis normais.**

- a) Verdadeiros
- b) Falso



**As DLLs são significativamente mais limitadas do que os pacotes quando consideramos o que elas podem exportar, uma vez que DLL não pode exportar constantes, tipos e variáveis normais.**

- a) Verdadeiros
- b) Falso

Verdadeira. As DLLs são significativamente mais limitadas do que os pacotes quando consideramos o que elas podem exportar, uma vez que DLLs não pode exportar constantes, tipos e variáveis normais. Pacotes são a melhor alternativa para exportar itens que não sejam simples procedimentos e funções. Devemos considerar o uso de Dlls apenas quando a interoperabilidade com outra programação é um requisito.

[http://docwiki.embarcadero.com/VCL/en/Libraries\\_and\\_Packages\\_Index](http://docwiki.embarcadero.com/VCL/en/Libraries_and_Packages_Index)



**Para retornar uma string formatada usando uma máscara de edição a unit MaskUtils possui o seguinte método?**

- a) FormatString
- b) FormatMaskText
- c) FormatStr
- d) FormatText



**Para retornar uma string formatada usando uma máscara de edição a unit MaskUtils possui o seguinte método?**

- a) FormatString
- b) **FormatMaskText**
- c) FormatStr
- d) FormatText

O método FormatMaskText é usado para retornar uma string formatada usando uma máscara de edição. FormatMaskText possui dois parâmetros, o EditMask onde informamos a máscara que será aplicada e o Value onde informamos o texto a ser formatado.

<http://docwiki.embarcadero.com/VCL/en/MaskUtils.FormatMaskEdit>



**Qual será o retorno da seguinte instrução  
FormatDateTime('dddddd',StrToDate('06/10/2018'))?**

- a) '06'.
- b) 'sáb'.
- c) 'sábado'.
- d) 'sábado, 6 de outubro de 2018'.



**Qual será o retorno da seguinte instrução  
`FormatDateTime('dddddd',StrToDate('06/10/2018'))`?**

- a) '06'.
- b) 'sáb'.
- c) 'sábado'.
- d) 'sábado, 6 de outubro de 2018'.

Utilizando como formato "dddddd" o `FormatDateTime` irá retornar a data com formato especificado pela variável global `LongDateFormat`, que em nosso exemplo fica "sábado, 6 de outubro de 2018".

<http://docwiki.embarcadero.com/VCL/en/SysUtils.FormatDateTime>





**Quais das seguintes classes não são herdadas de TComponent?**

- a) TFont
- b) TScreen
- c) TBitmap
- d) TImage



## Quais das seguintes classes não são herdadas de TComponent?

- a) TFont
- b) TScreen
- c) TBitmap
- d) TImage

TImage e TScreen são descendentes de TComponent

<http://docwiki.embarcadero.com/VCL/en/ExtCtrls.TImage>

<http://docwiki.embarcadero.com/VCL/en/Forms.TScreen>



**O Delphi permite que sejam configurados comandos para serem executado tanto antes como depois da geração do build.**

- a) Verdadeiro
- b) Falso



**O Delphi permite que sejam configurados comandos para serem executado tanto antes como depois da geração do build.**

- a) Verdadeiro
- b) Falso

Use o “Build Event” para configurar comando para serem executados antes e/ou após de executarmos um build  
[http://docwiki.embarcadero.com/RADStudio/en/Build\\_Events](http://docwiki.embarcadero.com/RADStudio/en/Build_Events)



**Na Unit Generics.Collections são declaradas diversas classes genéricas para facilitar o trabalho com coleções, listas e pilhas. Qual destas classes não pertence a Unit Generics.Collections?**

- a) TArray
- b) TObjectList
- c) TCollection
- d) TDictionary



**Na Unit Generics.Collections são declaradas diversas classes genéricas para facilitar o trabalho com coleções, listas e pilhas. Qual destas classes não pertence a Unit Generics.Collections?**

- a) TArray
- b) TObjectList
- c) TCollection
- d) TDictionary

Na Unit Generics.Collections são declaradas diversas classes genéricas para facilitar o trabalho com coleções, listas e pilhas. Mas apesar do nome sugestivo TCollection não pertence a unit Generics.Collections. Há várias outras classes que já existem, como o TObjectList, e possuem uma versão genérica nesta unit. A versão original continua valendo, mas temos a opção genérica para facilitar nossa vida.

<http://docwiki.embarcadero.com/VCL/en/Generics.Collections>



**Qual os três valores que uma variável do tipo TCollectionNotification pode assumir?**

- a) cnAdded
- b) cnRemoved
- c) cnExtracted
- d) cnInserted



## Qual os três valores que uma variável do tipo TCollectionNotification pode assumir?

- a) `cnAdded`
- b) `cnRemoved`
- c) `cnExtracted`
- d) `cnInserted`

TCollectionNotification pode assumir os seguintes valores, `cnAdded`, `cnRemoved` e `cnExtracted`. Ele é utilizado em alguns eventos das coleções genéricas das classes contidas na unit `Generics.Collections` e indica se um item está sendo incluído, removido ou extraído. A diferença entre removido e extraído é que no segundo caso o item além de removido seu valor é retornado.

<http://docwiki.embarcadero.com/VCL/en/Generics.Collections.TCollectionNotification>





**Qual das seguintes classes não é um descendente de TButtonControl.**

- a) TCheckBox
- b) TSpinButton
- c) TButton
- d) TBitBtn



**Qual das seguintes classes não é um descendente de TButtonControl.**

- a) TCheckBox
- b) TSpinButton
- c) TButton
- d) TBitBtn

Você já deve ter ouvido falar dos tais componentes que fazem “wrappers” dos controles do Windows. Eu gosto de descrever o “wrappers” da seguinte maneira, o Windows possui uma série de controles nativos, como botões e caixas de texto, o Delphi pega estes controles e da um “tunadas” neles. Por exemplo, o TButtonControl que faz “wrapper” do controle Button do Windows. Então o Delphi pega esse Button, põem umas rodas de liga leve, um aerofólio e temos o TButtonControl como qual o conhecemos. Uma coisa interessante é que para o Windows um Button, um CheckBox, um RadioButton é tudo a mesma coisa, todos eles não passam de uma button com um “estilo” diferente, logo, para o Delphi tanto o TButton, o TCheckBox e o TRadioButton herdam de TButtonControl.

<http://docwiki.embarcadero.com/VCL/en/StdCtrls.TButtonControl>



**Para atribuirmos uma imagem a um TButton qual propriedade devemos configurar?**

- a) Picture
- b) Glyph
- c) Image
- d) Icon



## Para atribuirmos uma imagem a um TButton qual propriedade devemos configurar?

- a) Picture
- b) Glyph
- c) Image
- d) Icon

Na propriedade Images de um TButton atribuímos um ImageList com as imagens que poderemos usar. Configure a propriedade ImageIndex o índice da imagem que será exibida no Botão.

<http://docwiki.embarcadero.com/VCL/en/StdCtrls.TButton.Images>



## Referente ao código de exemplo a seguir, como deve ser declarado o TFormClass a fim de fazer com que esta rotina crie uma instância de qualquer descendente TForm?

```
Procedure CriarFormulario(pFormClass : TFormClass);  
begin  
    pFormClass.Create(nil);  
end;
```

- a) type TFormClass = Class(TForm);
- b) type TFormClass = Class of TForm;
- c) type TFormClass = Set of TForm;
- d) type TFormClass = TCustomForm;



## Referente ao código de exemplo a seguir, como deve ser declarado o TFormClass a fim de fazer com que esta rotina crie uma instância de qualquer descendente TForm?

```
Procedure CriarFormulario(pFormClass : TFormClass);  
begin  
    pFormClass.Create(nil);  
end;
```

- a) type TFormClass = Class(TForm);
- b) type TFormClass = Class of TForm;**
- c) type TFormClass = Set of TForm;
- d) type TFormClass = TCustomForm;

Estamos acostumados a passar um objeto como parâmetros, mas passar uma referência de classe não é algo muito comum para maioria dos programadores, apesar de ser uma tarefa bastante simples. Para isso basta criar um tipo novo e atribuir um "Class of" da classe para eles. Em nosso exemplo, type TFormClass = Class of TForm.

[http://docwiki.embarcadero.com/RADStudio/en/Class\\_References](http://docwiki.embarcadero.com/RADStudio/en/Class_References)



**Assumindo que o código acima foi devidamente implementado, qual método Draw será executado:**

```
type
  TShape = class
    procedure Draw; virtual; abstract;
  end;

  TPolygon = class(TShape)
    procedure Draw; override;
  end;

procedure TForm1.FormCreate(Sender : TObject);
var
  IShape : TShape;
begin
  IShape := TRectangle.Create;
  IShape.Draw;
end;
```

- a) Shape
- b) Polygon
- c) Rectangle
- d) Uma exceção do tipo EAbstractError



**Assumindo que o código acima foi devidamente implementado, qual método Draw será executado:**

```
type
  TShape = class
    procedure Draw; virtual; abstract;
  end;

  TPolygon = class(TShape)
    procedure Draw; override;
  end;

procedure TForm1.FormCreate(Sender : TObject);
var
  IShape : TShape;
begin
  IShape := TRectangle.Create;
  IShape.Draw;
end;
```

- a) Shape
- b) Polygon**
- c) Rectangle
- d) Uma exceção do tipo EAbstractError





Esta é uma das questões mais difíceis de explicar, pois requer um bom conhecimento de OO. Aqui tratamos do conceito de polimorfismo e muita gente ainda se perde nesse quesito. Sempre que quisermos instanciar um objeto devemos fazer a declaração de tipo antes. O tipo declarado não necessariamente deve ser o mesmo tipo que será instanciado. Parece simples certo? De certa forma é, se não fosse alguns detalhezinhos. Em nosso exemplo declaramos uma variável do tipo TShape, mas instanciamos um objeto do tipo TRectangle. Em memória teremos um objeto do tipo TRectangle mas toda referência se dará para um tipo TShape. Se você tentar acessar um método que tenha sido declarado apenas em TRectangle deverá fazer um type cast para isso, pois como nossa variável é do tipo TShape, o Delphi só irá reconhecer os métodos desta classe. Mas quando temos métodos com o mesmo nome no descendente e no ancestral o Delphi irá executar os métodos do descendente e no ancestral o Delphi irá executar os métodos do descendente, desde que eles tenham sido implementados com a diretiva “Override”. No nosso caso apenas a classe TPolygon reimplementa o método Draw com esta diretiva, então esta será executada. Na classe TRectangle o método Draw é considerado um método novo, ele quebra a hierarquia, então não será executado em nosso exemplo

<http://docwiki.embarcadero.com/RADStudio/en/Methods>



## Quais os únicos contextos em que parâmetros default são permitidos?

- a) Typed const e value parameters
- b) Var e out parameters
- c) Var e value parameters
- d) Class reference parameters



## Quais os únicos contextos em que parâmetros Default são permitidos?

- a) Typed const e value parameters
- b) Var e out parameters
- c) Var e value parameters
- d) Class reference parameters

Muitas vezes criamos métodos que possuem parâmetros que em sua grande maioria terão sempre os mesmos valores e não teriam a necessidade de serem informados o tempo todo. Nestes casos podemos atribuir um valor “Default” para o parâmetro. Os únicos contextos onde podemos utilizar parâmetros default são em parâmetros do tipo Const e Value.

[http://docwiki.embarcadero.com/RADStudio/en/Parameters\\_\(Delphi\)](http://docwiki.embarcadero.com/RADStudio/en/Parameters_(Delphi))



**Qual das seguintes opções mostra a ordem correta que os eventos são executados para os descendentes TForm?**

- a) OnCreate, OnPaint, OnActivate, OnShow
- b) OnCreate, OnShow, OnActivate, OnPaint
- c) OnCreate, OnPaint, OnShow, OnActivate
- d) OnCreate, OnShow, OnPaint, OnActivate



**Qual das seguintes opções mostra a ordem correta que os eventos são executados para os descendentes TForm?**

- a) OnCreate, OnPaint, OnActivate, OnShow
- b) OnCreate, OnShow, OnActivate, OnPaint
- c) OnCreate, OnPaint, OnShow, OnActivate
- d) OnCreate, OnShow, OnPaint, OnActivate

Aqui o lance é decorar mesmo, não tem muito que fazer, a ordem de eventos ao chamarmos um form é OnCreate, OnShow, OnActivate, OnPaint.  
<http://docwiki.embarcadero.com/VCL/en/Forms.TForm.OnCreate>



**O evento OnKeyPress é acionado quando quais os tipos de teclas são pressionadas?**

- a) Arrow keys
- b) Numeric keys (0-9)
- c) Alphabetic keys (A-Z)
- d) Function keys (F1-F12)



## O evento OnKeyPress é acionado quando quais os tipos de teclas são pressionadas?

- a) Arrow keys
- b) Numeric keys (0-9)
- c) Alphabetic keys (A-Z)
- d) Function keys (F1-F12)

O evento OnKeyPress registra o carácter ASCII da tecla pressionada, sendo assim capturamos as Numeric Keys (0-9) e as Alphabetic keys (A-Z)  
<http://docwiki.embarcadero.com/VCL/en/Controls.TWinControl.OnKeyPress>



**Qual a propriedade de um descendente de TDataSet indica se o conjunto de dados pode ou não ser alterado?**

- a) CanEdit
- b) IsChangable
- c) AutoEdit
- d) CanModify





**Qual a propriedade de um descendente de TDataSet indica se o conjunto de dados pode ou não ser alterado?**

- a) CanEdit
- b) IsChangable
- c) AutoEdit
- d) CanModify

Todo DataSet possui uma propriedade chamada CanModify, essa propriedade indica se seus dados poderão ser ou não alterados. Datasets unidirecionais, como o TSqlDataset tem sempre essa propriedade atribuída como "False". Nos demais o CanModify reflete o valor da propriedade ReadOnly

<http://docwiki.embarcadero.com/VCL/en/DB.TDataSet.CanModify>



**No evento OnMouseWheel de um formulário temos o parâmetro WheelDelta cujo o propósito é:**

- a) Informar o estado de mudança durante a rotação da roda do mouse.
- b) Informar o valor pelo qual a roda do mouse é girada.
- c) Informar a atual coordenada do mouse.
- d) Informar se alguma tecla foi pressionada quando a roda do mouse é girada.



## No evento OnMouseWheel de um formulário temos o parâmetro WheelDelta cujo o propósito é:

- a) Informar o estado de mudança durante a rotação da roda do mouse.
- b) Informar o valor pelo qual a roda do mouse é girada.
- c) Informar a atual coordenada do mouse.
- d) Informar se alguma tecla foi pressionada quando a roda do mouse é girada.

O parâmetro WheelDelta informar o valor pelo qual roda do mouse é girada. Esse valor é um inteiro sempre múltiplo de 120. Sendo que ele será positivo se girarmos a “rodinha” para cima e negativo se girarmos para baixo.



**Qual dos seguintes itens não é uma string type:**

- a) CharString
- b) AnsiString
- c) UnicodeString
- d) WideString



**Qual dos seguintes itens não é uma string type:**

- a) CharString
- b) AnsiString
- c) UnicodeString
- d) WideString

CharString não existe, ou a variável é um char ou é uma string. Char é conhecido por ser um tipo simples, juntamente com os tipos Inteiros, Booleanos, Enumerados e Reais (numéricos). Já Strings se tratam de um tipo um pouco complexo, já que no mínimo estamos falando de um conjunto de Char.



**Que tipo é usado para armazenar os ponteiros de códigos e de dados de um método?**

- a) TObject
- b) TMethod
- c) TClass
- d) TStringList



**Que tipo é usado para armazenar os ponteiros de códigos e de dados de um método?**

- a) TObject
- b) TMethod**
- c) TClass
- d) TStringList

Sem pega ratão desta vez. A não ser o fato da dúvida de algo tão óbvio. TMethod é usado para armazenar os ponteiros de códigos e de dados de um método. Longe de ser algo trivial e de estar presente em nosso dia-dia, a dica mesmo é passar algumas horas “fuçando” o help do Delphi



## Qual a finalidade do método FilePos?

- a) Retornar a posição corrente de um arquivo aberto.
- b) Retornar a posição de um arquivo no diretório.
- c) Retornar a posição de um stream passado como parâmetro dentro de um arquivo.
- d) Retornar a posição de um trecho de arquivo na área de memória alocado para ele.





## Qual a finalidade do método FilePos?

- a) Retornar a posição corrente de um arquivo aberto.
- b) Retornar a posição de um arquivo no diretório.
- c) Retornar a posição de um stream passado como parâmetro dentro de um arquivo.
- d) Retornar a posição de um trecho de arquivo na área de memória alocado para ele.

FilePos retorna a posição corrente do ponteiro de arquivo aberto. É como se fosse o RecNo do DataSet só que para arquivos.



**Em um conjunto determinado por um tipo enumerado queremos retornar o primeiro elemento deste conjunto. Para isso utilizamos o método?**

- a) First();
- b) Pos();
- c) Star();
- d) Low();



**Em um conjunto determinado por um tipo enumerado queremos retornar o primeiro elemento deste conjunto. Para isso utilizamos o método?**

- a) First();
- b) Pos();
- c) Star();
- d) Low();

Sempre que quiser buscar o primeiro elemento de um conjunto de tipo enumerado utilizamos o método Low(), para retornar o último utilize o método High();



## Qual dos seguintes itens melhor define o método DupeString?

- a) Retorna uma string resultado da concatenação de dois Variants.
- b) Retorna a uma string sem espaços e outros caracteres de controle.
- c) Retorna a concatenação de duas strings.
- d) Retorna a concatenação de uma string com ela própria num determinado número de repetições



## Qual dos seguintes itens melhor define o método DupeString?

- a) Retorna uma string resultado da concatenação de dois Variants.
- b) Retorna a uma string sem espaços e outros caracteres de controle.
- c) Retorna a concatenação de duas strings.
- d) **Retorna a concatenação de uma string com ela própria num determinado número de repetições**

Eu utilizo muito o método StringOfChar, ele retorna uma string repetindo o mesmo caractere o número de vezes que especificamos no parâmetro. DupeString funciona da mesma forma, mas utilizando um string ao invés de uma char, Ele retornará uma string resultante da concatenação de uma string com ela própria num determinado número de repetições especificado em seu parâmetro.



**Qual dos seguintes métodos retorna o índice de byte do último caractere que corresponde a qualquer caractere em um determinado conjunto de delimitadores.**

- a) Pos;
- b) PosEx;
- c) LastDelimiter;
- d) RightPos;



**Qual dos seguintes métodos retorna o índice de byte do último caractere que corresponde a qualquer caractere em um determinado conjunto de delimitadores.**

- a) Pos;
- b) PosEx;
- c) **LastDelimiter;**
- d) RightPos;

LastDelimiter é bem interessante, ele retorna o índice do byte do último caractere que corresponde a qualquer caractere em um determinado conjunto de delimitadores. Ele possui dois parâmetros, o primeiro você passa uma string com uma lista de caracteres que você quer encontrar, o segundo a string onde esses caracteres serão procurados, o retorno será a posição do último caractere do segundo parâmetro que seja igual a algum caractere do primeiro.



## Utilizamos o método `ExpandFileName` para seguinte situação:

- a) Retornar o nome completo do caminho de uma determinado nome de arquivo.
- b) Extrair parte do nome e extensão de um determinado nome de arquivo.
- c) Retornar a unidade e o diretório de um determinado nome de arquivo.
- d) Retornar a extensão de um determinado nome de arquivo.





## Utilizamos o método `ExpandFileName` para seguinte situação:

- a) Retornar o nome completo do caminho de uma determinado nome de arquivo.
- b) Extrair parte do nome e extensão de um determinado nome de arquivo.
- c) Retornar a unidade e o diretório de um determinado nome de arquivo.
- d) Retornar a extensão de um determinado nome de arquivo.

O método `ExpandFileName` retorna o nome completo do caminho de um determinado nome de arquivo. Sempre é retornado o caminho qualificado/absoluto (aquele que inicia com o nome de um Driver), nunca relativo.



**Qual método arredonda uma variável em direção ao infinito negativo.**

- a) Floor
- b) Ceil
- c) Trunc
- d) Round



**Qual método arredonda uma variável em direção ao infinito negativo.**

- a) Floor
- b) Ceil
- c) Trunc
- d) Round

O método Floor retorna um inteiro arredondando uma variável em direção ao infinito negativo. Por exemplo, o número -2,9 será arredondado para -3 e 2,9 será arredondado para 3.



**Ao criarmos um novo projeto DataSnap Server podemos optar entre três tipos de aplicações. Qual dos seguintes item não é uma opção válida para um projeto DataSnap Server.**

- a) VCL Forms Application
- b) Console Application
- c) Service Application
- d) DataSnap Application



**Ao criarmos um novo projeto DataSnap Server podemos optar entre três tipos de aplicações. Qual dos seguintes item não é uma opção válida para um projeto DataSnap Server.**

- a) VCL Forms Application
- b) Console Application
- c) Service Application
- d) DataSnap Application

O que queremos saber aqui é justamente de que formas os aplicativos DataSnap podem ser concebidos. Podemos ter servidores DataSnap construídos em forma de Serviço do Windows, Console ou VCL Forms.



**Qual destes componentes não possui a propriedade Caption?**

- a) TStaticText
- b) TRadioButton
- c) TEdit
- d) TCheckBox



## Qual destes componentes não possui a propriedade Caption?

- a) TStaticText
- b) TRadioButton
- c) TEdit
- d) TCheckBox

TEdit não possui uma propriedade chamada caption. A propriedade responsável por exibir o texto de um TEdit é a propriedade Text



**Qual dos seguintes componentes não é um Vista Dialog Controls?**

- a) TFileOpenDialog
- b) TFileFindDialog
- c) TTaskDialog
- d) TFileSaveDialog





## Qual dos seguintes componentes não é um Vista Dialog Controls?

- a) TFileOpenDialog
- b) TFileFindDialog
- c) TTaskDialog
- d) TFileSaveDialog

Temos apenas três componentes especialmente desenhados para Windows Vista e consequentemente as versões sucessoras, então grave bem, são eles o TFileOpenDialog, o TFileSaveDialog e o TTaskDialog



**O componente TTimer possui uma propriedade chamada interval, que especificamente qual o intervalo de tempo entre as execuções do evento OnTimer do componente. Esse tempo específico na propriedade é dado em que unidade.**

- a) Milésimos de segundo
- b) Centésimos de segundo
- c) Décimos de segundo
- d) Segundos



**O componente TTimer possui uma propriedade chamada interval, que especificamente qual o intervalo de tempo entre as execuções do evento OnTimer do componente. Esse tempo específico na propriedade é dado em que unidade.**

- a) Milésimos de segundo
- b) Centésimos de segundo
- c) Décimos de segundo
- d) Segundos

A propriedade interval do componente configura em milésimos de segundo o intervalo de execução do método OnTimer.



**Ao utilizarmos o Refactor “Rename”, podemos alterar o nome de praticamente qualquer coisa no nosso projeto, sendo que uma destas exceções é a alteração do nome dos componentes.**

- a) Verdadeiro
- b) Falso



**Ao utilizarmos o Refactor “Rename”, podemos alterar o nome de praticamente qualquer coisa no nosso projeto, sendo que uma destas exceções é a alteração do nome dos componentes.**

- a) Verdadeiro
- b) Falso

O Refactor “Rename” pode ser usado inclusive para alterar nome de um componente.



**Qual destes controles não permite ser container de outro controle?**

- a) TPanel
- b) TScrollBar
- c) TBevel
- d) TGroupBox



## Qual destes controles não permite ser container de outro controle?

- a) TPanel
- b) TScrollBar
- c) TBevel
- d) TGroupBox

Essa é uma das perguntas mais clássicas da prova de certificação. E possui uma quantidade enorme de variações. O que devemos observar aqui é que estamos falando de componentes contidos em outros, para isso usamos componentes herdados de TWinControl, que tem por característica poder conter outros componentes. Lembre-se que estes componentes possuem uma propriedade chamada Controls, que é a lista de controles inseridos em um componente descendente de TWinControl. Os demais controles herdam de TGraphicControl, como o TImage e o TBevel, estes componentes não possuem a capacidade de serem contêiner de outros controles.



**Qual opção de BoderStyle de um TFrom permite visualizar o botão de Help no formulário?**

- a) bsSingle
- b) bsDialog
- c) bsSizeable
- d) bsNone





**Qual opção de BoderStyle de um TFrom permite visualizar o botão de Help no formulário?**

- a) bsSingle
- b) bsDialog**
- c) bsSizeable
- d) bsNone

Ta achando que é só chegar e habilitar o botão de Help no formulário (BorderIcons biHelp) que ele vai aparecer numa boa lá em cima do form? Pois é velhinho, não é por ai não! O formulário só exibe este botão se o BorderStyle dele estiver com bsDialog atribuido e o BoderIcons biSystemMenu habilitado.



**Qual das seguintes seções do Project Options possui a opção para alterar o Ícone do executável gerado?**

- a) Forms
- b) Version Info
- c) Delphi Compiler
- d) Application



**Qual das seguintes seções do Project Options possui a opção para alterar o Ícone do executável gerado?**

- a) Forms
- b) Version Info
- c) Delphi Compiler
- d) **Application**

Na seção Application podemos configurar o Nome exibido pelo aplicativo na barra de tarefas, o arquivo de help, o ícone da aplicação, se utilizará os temas do Windows (para vista ou sucessores) e a extensão do aplicativo, caso achamos que o .exe não seja a mais indicada.



## Qual alternativa preenche as lacunas corretamente.

Ao utilizarmos um ClientDataset podemos ordená-lo por um determinado campo especificando seu nome na propriedade \_\_\_\_\_ ou incluindo um índice aos \_\_\_\_\_ e especificando seu nome na propriedade \_\_\_\_\_.

- a) IndexName, IndexList, IndexItem
- b) IndexFieldName, IndexDefs, IndexName
- c) SortFieldName, SortFieldList, SortName
- d) IndexField, IndexFieldDef, IndexFieldName



## Qual alternativa preenche as lacunas corretamente.

Ao utilizarmos um ClientDataset podemos ordená-lo por um determinado campo especificando seu nome na propriedade \_\_\_\_\_ ou incluindo um índice aos \_\_\_\_\_ e especificando seu nome na propriedade \_\_\_\_\_.

- a) IndexName, IndexList, IndexItem
- b) IndexFieldName, IndexDefs, IndexName
- c) SortFieldName, SortFieldList, SortName
- d) IndexField, IndexFieldDef, IndexFieldName

Ao utilizarmos um ClientDataset podemos ordená-lo por um determinado campo especificando seu nome na propriedade IndexFieldName ou incluindo um índice aos IndexDefs e especificando seu nome na propriedade IndexName. Uma das vantagens do Indexname é que podemos fazer a ordenação de forma decrescente.



## Quais destes estilos estão disponíveis para o componente Ribbon?

- a) Obsidian
- b) Office
- c) Silver
- d) XP



## Quais destes estilos estão disponíveis para o componente Ribbon?

- a) Obsidian
- b) Office
- c) Silver
- d) XP

O componente Ribbon por default possui três estilos disponíveis, Lua, Obsidian e Silver



**O Record TSearchRec possui uma nova propriedade responsável por retornar à data de modificação de arquivos e diretórios. Qual o nome desta propriedade?**

- a) FileDate.
- b) SysDate.
- c) TimeStamp.
- d) DateTime.





**O Record TSearchRec possui uma nova propriedade responsável por retornar à data de modificação de arquivos e diretórios. Qual o nome desta propriedade?**

- a) FileDate.
- b) SysDate.
- c) **TimeStamp.**
- d) DateTime.

A propriedade TimeStamp do Record TSearchRec é responsável por retornar à data de modificação de arquivos e diretórios.



## Quais das seguintes alternativas são valores válidos para um TFieldAttribute?

- a) faHiddenCol, faReadOnly, faRequired, faLink, faUnNamed, faFixed.
- b) faFile, faFileSystem, faDirectory, faAttribute.
- c) faName, faDate, faReadOnly, faType, faVersion.
- d) faTemporary, faSymLink, faCompressed, faencrypted, faVirtual, faAnyFile.



## Quais das seguintes alternativas são valores válidos para um TFieldAttribute?

- a) `faHiddenCol`, `faReadOnly`, `faRequired`, `faLink`, `faUnNamed`, `faFixed`.
- b) `faFile`, `faFileSystem`, `faDirectory`, `faAttribute`.
- c) `faName`, `faDate`, `faReadOnly`, `faType`, `faVersion`.
- d) `faTemporary`, `faSymLink`, `faCompressed`, `faencrypted`, `faVirtual`, `faAnyFile`.

TFieldAttributes é um conjunto de TFieldAttribute utilizado para descrever atributos de um campo de uma tabela do banco de dados. Os valores disponíveis para TFieldAttribute são, `faHiddenCol`, `faReadOnly`, `faRequired`, `faLink`, `faUnNamed` e `faFixed`.



**Conforme o código abaixo, declaramos uma variável para o tipo TFormatSettings. Para carregar este record com os formatos atuais do Delphi, qual dos seguintes códigos deve ser executado?**

var

  IFormatSettings : TFormatSetting;

- a) IFormatSettings.GetDefault;
- b) GetFormatSttings(IFormatSttings);
- c) IFormatSettings := TFormatSettings.Create;
- d) IFormatSettings := SysUtils.FormatSettings;



**Conforme o código abaixo, declaramos uma variável para o tipo TFormatSettings. Para carregar este record com os formatos atuais do Delphi, qual dos seguintes códigos deve ser executado?**

var

IFormatSettings : TFormatSetting;

- a) IFormatSettings.Default;
- b) GetFormatSttings(IFormatSttings);
- c) **IFormatSettings := TFormatSettings.Create;**
- d) IFormatSettings := SysUtils.FormatSettings;

TFormatSettings define uma estrutura de dados contendo informações básicas locais para rotinas de formatação de String, como formato de data, hora, separadores decimais entre outros. Muitas vezes precisamos de uma variável com estes formatos para atender alguma necessidade específica e temos popular nossa variável com os valores locais do sistema, uma vez que a variável por si só é inicializada com seus valores vazios. Para isso temos duas opções, uma é a chamada do método GetLocaleFormatSettings, mas este atualmente foi definido como “Deprecated”, deixando-nos a opção de utilizar o método Create do record TFormatSttings.



## As definições abaixo se refere, nesta mesma ordem, qual dos métodos da unit DateUtils:

1. Indica-se dois valores de data/hora que estão dentro de um determinado número de meses um do outro.
  2. Retorna ao número aproximado de meses entre dois valores TDateTime.
  3. Retorna ao número aproximado de meses, incluindo frações, entre dois valores TDateTime.
  4. Retorna ao mês do ano representado por um valor TDateTime.
- 
- a) WithinPastMonths, MonthsBetween, MonthSpan, MonthOf.
  - b) WithinPastMonths, MonthSpan, MonthsBetween, MonthOf.
  - c) MonthOf, MonthsBetween, MonthSpan, WithinPastMonths.
  - d) MonthsBetween, WithinPastMonths, MonthSpan, MonthOf.



## As definições abaixo se refere, nesta mesma ordem, qual dos métodos da unit DateUtils:

1. Indica-se dois valores de data/hora que estão dentro de um determinado número de meses um do outro.
  2. Retorna ao número aproximado de meses entre dois valores TDateTime.
  3. Retorna ao número aproximado de meses, incluindo frações, entre dois valores TDateTime.
  4. Retorna ao mês do ano representado por um valor TDateTime.
- 
- a) **WithinPastMonths, MonthsBetween, MonthSpan, MonthOf.**
  - b) WithinPastMonths, MonthSpan, MonthsBetween, MonthOf.
  - c) MonthOf, MonthsBetween, MonthSpan, WithinPastMonths.
  - d) MonthsBetween, WithinPastMonths, MonthSpan, MonthOf.

A Unit DateUtils possui vários métodos para facilitar nosso trabalho com informações de Data e Hora, entre eles podemos citar alguns utilizados especificamente para trabalhar como meses:

WithinPastmonths - Indica se dois valores de data/hora que estão dentro de um determinado número de meses um do outro.

MonthsBetween - Retorna ao número aproximado de meses entre dois valores TDateTime.

MonthSpan - Retorna ao número aproximado de meses, incluindo frações, entre dois valores TDateTime.

MonthOf - Retorna ao mês do ano representado por um valor TDateTime.



**Delphi nos permite incluir um breakpoint em um determinado ponto do código e configurar uma condição para que a execução seja parada neste ponto.**

- a) Verdadeiro
- b) Falso





**Delphi nos permite incluir um breakpoint em um determinado ponto do código e configurar uma condição para que a execução seja parada neste ponto.**

- a) Verdadeiro
- b) Falso

Verdadeiro. Ao definirmos um breakpoint podemos clicar nele com o botão direito do mouse e acessar suas propriedades. Lá podemos configurar condições para a parada ou números de vezes que o cursor passará pela linha antes de parar



**Ao conectar um DataSetProvider em SqlDataSet e um ClientDataset ao DataSetProvider, como podemos trazer todos os parâmetros do SqlDataset para o ClientDataset de uma só vez?**

- a) Alterar a propriedade Ative para True.
- b) Executar o ComponentEditor Fetch Params
- c) Executar o PropertyEditor da propriedade Params e executar o botão LoadParams
- d) Os parâmetros devem ser adicionados um a um



**Ao conectar um DataSetProvider em SqlDataSet e um ClientDataset ao DataSetProvider, como podemos trazer todos os parâmetros do SqlDataset para o ClientDataset de uma só vez?**

- a) Alterar a propriedade Ative para True.
- b) Executar o ComponentEditor "Fetch Params"**
- c) Executar o PropertyEditor da propriedade Params e executar o botão LoadParams
- d) Os parâmetros devem ser adicionados um a um

Ao conectar um DataSetProvider em um SQLDataset em um ClientDataset ao DataSetProvider, podemos trazer todos os parâmetros do SqlDataset para o ClientDataset de uma só vez, para isso basta clicar com o botão direito no ClientDataset para executar o ComponentEditor "Fetch Params"



**Qual desses eventos não pertencem a um TDataSourceProvider?**

- a) OnDataChange
- b) OnStateChange
- c) OnUpdateData
- d) OnAfterScroll



## Qual desses eventos não pertencem a um TDataSourceProvider?

- a) OnDataChange
- b) OnStateChange
- c) OnUpdateData
- d) OnAfterScroll

O TDataSource está longe de ser um mega componente cheio de eventos, apesar de ser um componente presente em larga escala na maioria dos programas. Então não existe desculpa para não conhecer os 3 eventos dele: o OnDataChange, o OnStateChange e o OnUpdateData



**Ao abrirmos um arquivo HTML no Delphi, a unica maneira de visualizar a página html é através de um browser, pois o Delphi não possui uma ferramenta de designer de paginas HTMLs.**

- a) Verdadeiro
- b) Falso



**Ao abrirmos um arquivo HTML no Delphi, a única maneira de visualizar a página html é através de um browser, pois o Delphi não possui uma ferramenta de designer de páginas HTMLs.**

- a) Verdadeiro
- b) Falso**

Ao abrirmos ou criarmos um arquivo HTML no Delphi, a própria ferramenta já disponibiliza uma área de designer para que possamos visualizar a página HTML sem a necessidade de um browser. Além disso, a tool palette é modificada, fornecendo componentes que possam ser utilizados para editar visualmente o arquivo



## Qual das seguintes situações ocorrerá ao executarmos o código abaixo?

```
var  
  D : double;  
  T : TDateTime;  
begin  
  T := StrToDateTime('01/05/2018');  
  D := T;
```

- a) Será levantada uma exceção do tipo EConvertError com a mensagem “01/05/2018’ is not valid floating point value’.
- b) Ocorrerá o erro de compilação E2010 Incompatible types: ‘Double’ and ‘TDateTime’.
- c) Ocorrerá o erro de compilação E2010 Incompatible types: ‘TDateTime’ and ‘Double’.
- d) O código será executado sem nenhum problema.





## Qual das seguintes situações ocorrerá ao executarmos o código abaixo?

```
var  
  D : double;  
  T : TDateTime;  
begin  
  T := StrToDateTime('01/05/2018');  
  D := T;
```

- a) Será levantada uma exceção do tipo EConvertError com a mensagem “01/05/2018’ is not valid floating point value’.
- b) Ocorrerá o erro de compilação E2010 Incompatible types: ‘Double’ and ‘TDateTime’.
- c) Ocorrerá o erro de compilação E2010 Incompatible types: ‘TDateTime’ and ‘Double’.
- d) **O código será executado sem nenhum problema.**

Nenhuma exceção será levantada, uma vez que ao transformarmos uma String em um TDateTime e ao atribuirmos seu valor para uma variável do tipo Double, nenhuma operação ocorrerá já que um TDateTime na verdade se trata de um tipo Double



**Representa um janela padrão do aplicativo.**

- a) TForm
- b) TApplication
- c) TPanel
- d) TScrollingWinControl



**Representa um janela padrão do aplicativo.**

- a) TForm
- b) TApplication
- c) TPanel
- d) TScrollingWinControl

A classe TForm é utilizada para representar uma janela no aplicativo. Quando você criar formulários no designer de formulário em tempo de design, eles são implementados como descendentes de TForm. Os formulários podem representar a janela principal do aplicativo, caixas de diálogo ou MDI children. Um formulário pode conter outros objetos, tais como TButton, TCheckBox e objetos TComboBox.



**É um wrapper para um console de edição do Windows com várias linhas.**

- a) TListBox
- b) TMemo
- c) TDBGrid
- d) TComboBox



**É um wrapper para um console de edição do Windows com várias linhas.**

- a) TListBox
- b) TMemo**
- c) TDBGrid
- d) TComboBox

TMemo é um wrapper para um console de edição do Windows com várias linhas. O TMemo implementa o comportamento genérico introduzindo em TCustomMemo. Ele publica muitas das propriedades herdadas TCustomMemo, mas não introduz qualquer novo comportamento.



**É um wrapper para um console de edição do Windows com única linha.**

- a) TLabel
- b) TStaticText
- c) TEdit
- d) TMaskEdit



**É um wrapper para um console de edição do Windows com única linha.**

- a) TLabel
- b) TStaticText
- c) TEdit
- d) TMaskEdit

TEdit é um wrapper para um console de edição do Windows com uma única linha. Além de um componente de edição, um TEdit é uma boa saída quando precisamos exibir um texto para o usuário e permiti-lo selecionar o texto e copiá-lo para a área de transferência.



**É um controle “não ajanelado” que exibe o texto em um formulário.**

- a) TStaticText
- b) TEdit
- c) TPanel
- d) TLabel





**É um controle “não ajanelado” que exibe o texto em um formulário.**

- a) TStaticText
- b) TEdit
- c) TPanel
- d) TLabel

TLabel é um controle “não ajanelado” que exibe o texto em um formulário. Este texto pode ser usado para rotular outro controle e pode definir o foco para esse controle quando o usuário digita uma tecla de atalho, utilizando Alt + letra com underline.



**É um controle “ajanelado” que exibe o texto em um formulário.**

- a) TStaticText
- b) TEdit
- c) TPanel
- d) TLabel



**É um controle “ajanelado” que exibe o texto em um formulário.**

- a) TStaticText
- b) TEdit
- c) TPanel
- d) TLabel

TStaticText é um controle “ajanelado” que exibe o texto em um formulário. O TStaticText é muito semelhante ao TLabel, mas, por fazer wrapper com o windows, permite alguns outros recursos, como interagir com mensagens do Windows.



**Combina uma caixa de edição com uma lista “rolável”.**

- a) TComboBox
- b) TListBox
- c) TMemo
- d) TScrollBar



**Combina uma caixa de edição com uma lista “rolável”.**

- a) TComboBox
- b) TListBox
- c) TMemo
- d) TScrollBar

Um componente TComboBox é uma caixa de edição com uma lista drop-down “rolável” anexada a ele. Os usuários podem seleccionar um item da lista ou digitar diretamente na caixa de edição, de acordo com sua propriedade Style.



## Exibe uma coleção de itens em uma lista “rolável”

- a) TListBox
- b) TComboBox
- c) TMemo
- d) TDbGrid



## Exibe uma coleção de itens em uma lista “rolável”

- a) **TListBox**
- b) TComboBox
- c) TMemo
- d) TDbGrid

Use TListBox para exibir uma lista rolável de itens que os usuários podem selecionar, adicionar ou excluir. TListBox é um wrapper para o controle ListBox do Windows



**É um conjunto de páginas usadas para fazer uma caixa de diálogo de várias páginas.**

- a) TNotebook
- b) TPageControl
- c) TTabSheet
- d) TPageDialog





**É um conjunto de páginas usadas para fazer uma caixa de diálogo de várias páginas.**

- a) TNotebook
- b) TPageControl**
- c) TTabSheet
- d) TPageDialog

TPageControl é um conjunto de páginas usadas para fazer uma caixa de diálogo de várias páginas. Essas páginas sobrepostas são objetos TTabSheet. O usuário seleciona uma página clicando na guia da página que aparece na parte superior do controle. Para adicionar uma nova página a um objeto TPageControl em tempo de design, clique com o direito do mouse no objeto TPageControl e escolha New Page.



**Gerência botões de ferramentas e outros controles, organizando-as em linha e automaticamente ajustando seus tamanhos e posições.**

- a) TButtonManager
- b) TPanel
- c) TToolBar
- d) THeader



**Gerência botões de ferramentas e outros controles, organizando-as em linha e automaticamente ajustando seus tamanhos e posições.**

- a) TButtonManager
- b) TPanel
- c) **TToolBar**
- d) THeader

TToolBar é um recipiente para os botões de ferramenta (TToolButton). Ele oferece uma maneira fácil de organizar e gerenciar controles visuais - é, mais ou menos. Todos os botões de ferramentas em um toolbar mantém a largura e altura uniforme



**Representa o estado da tela em que um aplicativo é executado.**

- a) TMonitor
- b) TScreen
- c) TDisplay
- d) TApplication



**Representa o estado da tela em que um aplicativo é executado.**

- a) TMonitor
- b) TScreen**
- c) TDisplay
- d) TApplication

TScreen representa o estado da tela em que um aplicativo é executado. Essa classe apresenta propriedades que especificam o tamanho e a resolução da tela e cursores e fontes que estão disponíveis para o aplicativo usar. Para ter acesso a essas informações existe a variável global Screen, do tipo TScreen, que é instanciada automaticamente para uso da aplicação



**Quando temos necessidade de integração de sistemas e plataformas, distribuição geográfica, segurança no envio e recebimento para a aplicação, a tecnologia melhor recomendada para esta situação é:**

- a) BDE
- b) DBX Framework
- c) DataSnap
- d) VCL Web



**Quando temos necessidade de integração de sistemas e plataformas, distribuição geográfica, segurança no envio e recebimento para a aplicação, a tecnologia melhor recomendada para esta situação é:**

- a) BDE
- b) DBX Framework
- c) **DataSnap**
- d) VCL Web

DataSnap é uma tecnologia altamente recomendada quando temos a necessidade de integração de sistemas e plataformas, distribuição geográficas, segurança no envio e recebimento de informações e escalabilidade para a aplicação.



**O componente responsável por gerenciar e visualizar todos os Actions contidos em uma aplicação é:**

- a) TActionList
- b) TActionManager
- c) TActionToolBar
- d) TActionLink





**O componente responsável por gerenciar e visualizar todos os Actions contidos em uma aplicação é:**

- a) TActionList
- b) TActionManager**
- c) TActionToolBar
- d) TActionLink

TActionManager prove mecanismos que possibilitam gerenciar e visualizar todos os Action contidos em uma aplicação



**Qual exceção é disparada quando um componente existe em um TForm, mas sua declaração de tipo foi removida?**

- a) Acces Violation
- b) EAbort
- c) EClassNotFound
- d) EComponentError



**Qual exceção é disparada quando um componente existe em um TForm, mas sua declaração de tipo foi removida?**

- a) Acces Violation
- b) EAbort
- c) **EClassNotFound**
- d) EComponentError

EClassNotFoound é gerado se um nome de classe que não estiver associada ao aplicativo atual é encontrado durante a leitura de um componente em um Stream (um arquivo dfm, por exemplo). Isso pode acontecer quando existe um componente em um formulário, mas a sua declaração de tipo foi removida.



**O Delphi gera classes proxies para diversas outras plataformas, para que possamos consumir os recursos disponibilizados por nosso servidor DataSnap. Entre estas plataformas estão:**

- a) JavaScript e PHP
- b) Python e Java
- c) PHP e Flash
- d) OSX e Android



**O Delphi gera classes proxies para diversas outras plataformas, para que possamos consumir os recursos disponibilizados por nosso servidor DataSnap. Entre estas plataformas estão:**

- a) **JavaScript e PHP**
- b) Python e Java
- c) PHP e Flash
- d) OSX e Android

Para que possamos utilizar nosso servidor DataSnap com outras tecnologias o Delphi possibilita a criação de classes proxies. Entre as plataformas que estão disponíveis na geração destas classes estão JavaScript e o PHP



**O ClienteDataSet suporta múltiplas atualizações em uma unica transação, desde que seus datasets estejam ligados por uma chave primária. Este recurso é conhecido como:**

- a) Client-Server
- b) Aggregates
- c) Inline transactions
- d) NestedDataSet



**O ClientDataSet suporta múltiplas atualizações em uma única transação, desde que seus datasets estejam ligados por uma chave primária. Este recurso é conhecido como:**

- a) Client-Server
- b) Aggregates
- c) Inline transactions
- d) **NestedDataSet**

O ClientDataSet suporta múltiplas atualizações em uma única transação, permitindo que os dados de vários ClientDataSet sejam persistidos no banco de dados utilizando apenas com um comando ApplyUpdates. Isso é possível através do recurso conhecido como NestedDataset, onde vários ClientDataSet estão ligados em um ClientDataSet principal por uma chave primária.



**Ao declararmos um método que poderá ser herdado por seus descendentes, devemos assinalá-lo com a diretiva:**

- a) Override;
- b) Overload;
- c) Virtual;
- d) Inherited;





**Ao declararmos um método que poderá ser herdado por seus descendentes, devemos assinalá-lo com a diretiva:**

- a) Override;
- b) Overload;
- c) **Virtual;**
- d) Inherited;

Para que um método possa ser herdado por seus descendentes ele deve estar assinalado com a diretiva Virtual ou Dynamic



**O que é necessário fazer para que ao deixar o mouse sobre controle na tela, apareça um quadro explicativo do controle.**

- a) Apenas informar a mensagem que será exibida na propriedade Message do controle.
- b) Informar qual a mensagem que será exibida na propriedade Message do controle e atribuir True para a propriedade ShowMessage do controle.
- c) Apenas informar a mensagem que será exibida na propriedade Hint do controle
- d) Informar qual a mensagem que será exibida na propriedade Hint do controle e atribuir True para a propriedade ShowHint do controle.



**O que é necessário fazer para que ao deixar o mouse sobre controle na tela, apareça um quadro explicativo do controle.**

- a) Apenas informar a mensagem que será exibida na propriedade Message do controle.
- b) Informar qual a mensagem que será exibida na propriedade Message do controle e atribuir True para a propriedade ShowMessage do controle.
- c) Apenas informar a mensagem que será exibida na propriedade Hint do controle
- d) **Informar qual a mensagem que será exibida na propriedade Hint do controle e atribuir True para a propriedade ShowHint do controle.**

Para habilitarmos o Hint de um controle (o quadro explicativo que aparece ao deixarmos o mouse sobre o controle), devemos informar o texto explicativo na propriedade Hint e atribuir True para a propriedade ShowHint do controle.



**Em um componente PageControl foram inseridas algumas abas, TabSheets, e queremos que algumas delas não fiquem visíveis, para isso devemos:**

- a) Atribuir False para a propriedade TabVisible do TabSheet.
- b) Atribuir False para a propriedade Visible do TabSheet.
- c) Atribuir False para a propriedade Active do TabSheet.
- d) Não é possível deixar um aba invisível em um componente PageControl



**Em um componente PageControl foram inseridas algumas abas, TabSheets, e queremos que algumas delas não fiquem visíveis, para isso devemos:**

- a) **Atribuir False para a propriedade TabVisible do TabSheet.**
- b) Atribuir False para a propriedade Visible do TabSheet.
- c) Atribuir False para a propriedade Active do TabSheet.
- d) Não é possível deixar um aba invisível em um componente PageControl

Utilize a propriedade TabVible do TabSheet para controlar se você quer que uma TabSheet apareça ou não em um PageControl.



**Qual das seguintes classes pode ser usada sempre que precisarmos de uma referência de classe genérica?**

- a) TObject;
- b) TStream;
- c) TClass;
- d) TComponent;



**Qual das seguintes classes pode ser usada sempre que precisarmos de uma referência de classe genérica?**

- a) TObject;
- b) TStream;
- c) TClass;
- d) TComponent;

TClass uma metaclasses de TObject:

type TClass = class of TObject;

Isso permite usa-la sempre que necessitamos de uma referencia para uma classe sem que tenhamos que nos preocupar com o tipo dela.



**Que método de um TClientDataset pode ser utilizado para validar a existência de um campo?**

- a) FieldExist
- b) FindField
- c) LocaleField
- d) FieldByName





**Que método de um TClientDataset pode ser utilizado para validar a existência de um campo?**

- a) FieldExist
- b) FindField**
- c) LocaleField
- d) FieldByName

Usamos FindField para verificar se um determinado campo existe em um Dataset. Se o FindField localizar o campo, será retornado um objeto do tipo TField, caso contrário o método retornará Nil.



**Que método atribui o foco para o próximo controle ativo.**

- a) NextFocus
- b) TabExec
- c) SelectNext
- d) SetFocus



## Que método atribui o foco para o próximo controle ativo.

- a) NextFocus
- b) TabExec
- c) **SelectNext**
- d) SetFocus

Podemos atribuir o foco para o próximo controle ativo utilizando o método SelectNext. Este método possui três parâmetros, o primeiro indica o controle que será usado como base para localizar o próximo controle a ser focado. Caso você queira buscar o controle seguinte ao que está atualmente ativo, passamos como parâmetros o ActiveControl de um Form. O segundo parâmetro indica a direção que queremos percorrer, se verdadeiro iremos navegar para frente, se false, para trás. O terceiro parâmetro quando ativo fará com que o método localize apenas controles onde a propriedade TabStop seja verdadeira. Caso o método SelectNext não localize nenhum controle que satisfaça a pesquisa o foco não será alterado



**Os seguintes comandos. “Run”, “Trace into”, “Step over” e “Run to cursor”, estão disponíveis no menu Run do Delphi. Qual das opções abaixo contém as teclas de atalho na mesma ordem dos comandos apresentados.**

- a) F9, F8, F7, F4
- b) F9, F7, F8, F4
- c) F5, F10, F11, F6
- d) F9, F8, F7, F6



**Os seguintes comandos. “Run”, “Trace into”, “Step over” e “Run to cursor”, estão disponíveis no menu Run do Delphi. Qual das opções abaixo contém as teclas de atalho na mesma ordem dos comandos apresentados.**

- a) F9, F8, F7, F4
- b) F9, F7, F8, F4**
- c) F5, F10, F11, F6
- d) F9, F8, F7, F6

Como o próprio nome já diz, teclas de atalho existem para nos fornecer um meio rápido para um determinado comando, e conhecê-las nunca é demais. Entre as mais utilizadas estão os comandos “Run” (F9), “Trace into” (F7), “Step over” (F8) e “Run to cursor” (F4).



**Qual das alternativas abaixo não pertence ao enumerador TFieldKind?**

- a) fkField
- b) fkLookup
- c) fkInternalCalc
- d) fkAggregate



## Qual das alternativas abaixo não pertence ao enumerador TFieldKind?

- a) **fkField**
- b) fkLookup
- c) fkInternalCalc
- d) fkAggregate

TFieldKind indica que tipo de informação será representada por um componente TField. Um TFieldKind pode assumir os seguintes valores: fkData, fkCalculated, fkLookup, fkInternalCalc ou fkAggregate.



## Qual o propósito da diretiva “External”?

- a) Indicar que a implementação real do método está em uma DLL ou um Arquivo-Objeto.
- b) Indicar as Units externas que serão acessadas por uma Unit
- c) Indicar os pacotes externos que serão utilizados por um pacote
- d) Indicar que o método estará disponível para uso externo à aplicação





## Qual o propósito da diretiva “External”?

- a) Indicar que a implementação real do método está em uma DLL ou um Arquivo-Objeto.
- b) Indicar as Units externas que serão acessadas por uma Unit
- c) Indicar os pacotes externos que serão utilizados por um pacote
- d) Indicar que o método estará disponível para uso externo à aplicação

A diretiva External é uma velha conhecida de todos que já tiveram a necessidade de utilizar um método contido em um arquivo DLL. A diretiva informa que a implementação real do método não está declarada em nosso código fonte e sim em uma DLL ou em algum outro Arquivo-Objeto.



## Qual o significado da sigla “Json”?

- a) JavaScript Orientation Notation
- b) JavaScript Object Notation
- c) Java Simplify Object Notation
- d) JavaScript Object Notation



## Qual o significado da sigla “Json”?

- a) JavaScript Orientation Notation
- b) **JavaScript Object Notation**
- c) Java Simplify Object Notation
- d) JavaScript Object Notation

A sigla JSON significa “JavaScript Object Notation”. Representa um subconjunto da notação JavaScript exclusivamente, basicamente um formato encapsulado de dados. Atualmente está sendo muito utilizado para transferência como alternativa ao XML, tendo como principal vantagem a menor quantidade de dados trafegando entre serviço e o consumidor



**O Delphi permite salvar os BreakPoints do projeto para que quando fecharmos e reabrirmos o Delphi eles continuem no projeto.**

- a) Verdadeiro
- b) Falso



**O Delphi permite salvar os BreakPoints do projeto para que quando fecharmos e reabrirmos o Delphi eles continuem no projeto.**

- a) Verdadeiro
- b) Falso

Realmente não me recordo em que versão o Delphi passou a salvar os BreakPoints, mas já faz um bom tempo. Para que este recurso esteja valendo, você deve marcar a opção Project desktop do Environment Options.



## Qual o significado da sigla “Rest”?

- a) Representational Script Trabsfer
- b) Resource Script Test
- c) REsource State Transfer
- d) Representational State Transfer



## Qual o significado da sigla “Rest”?

- a) Representational Script Trabsfer
- b) Resource Script Test
- c) REsource State Transfer
- d) **Representational State Transfer**

A sigla REST significa “Representational State Transfer” e é um protocolo de comunicação alternativo ao protocolo SOAP.



**O arquivo DPR de uma DLL sempre inicia com a palavra reservada:**

- a) Unit
- b) Program
- c) DynamicLL
- d) Library





**O arquivo DPR de uma DLL sempre inicia com a palavra reservada:**

- a) Unit
- b) Program
- c) DynamicLL
- d) **Library**

O arquivo DPR de um programa executável sempre inicial com a palavra reservada “Program”. Nas DLLs inicial com a palavra reservada “Library” e nos pacotes com “Packages”.



**Que evento devemos implementar para atribuir um valor a um campo calculado de um dataset?**

- a) AfterRefresh
- b) AfterFieldCalc
- c) OnCalcFields
- d) BeforeScrollCalcFields



## Que evento devemos implementar para atribuir um valor a um campo calculado de um dataset?

- a) AfterRefresh
- b) AfterFieldCalc
- c) OnCalcFields
- d) BeforeScrollCalcFields

Em algumas situações exibir em um grid colunas com valores que não tem como origem o banco de dados e nem serão armazenados nele. Os campos calculados têm como propósito exibir alguma informação apenas para um dado momento. Estes campos podem apresentar um cálculo de idade ou uma descrição para algum código ou situação. A atribuição dos valores para esses campos deve ser feita no evento OnCalcFields do dataset



**Qual classe da VCL introduz a propriedade Owner?**

- a) TObject
- b) TControl
- c) TCustomComponent
- d) TComponent



## Qual classe da VCL introduz a propriedade Owner?

- a) TObject
- b) TControl
- c) TCustomComponent
- d) TComponent

Todo componente possui um outro componente que é responsável por gerenciá-lo. Esse componente gerenciador fica armazenado na propriedade Owner, que é introduzida na classe TComponent.



**Qual cláusula identifica os pacotes externos dos quais um pacote depende?**

- a) Uses
- b) Includes
- c) Requires
- d) Contains



## Qual cláusula identifica os pacotes externos dos quais um pacote depende?

- a) Uses
- b) Includes
- c) Requires
- d) Contains

Quando criamos um componente ou uma aplicação que estará dividida em pacotes, muitas vezes estes pacotes terão que acessar outros pacotes. Dentro do arquivo DPR de um pacote encontramos a palavra reservada "Require", e é nesta seção que ficam declarados os pacotes externos.



**Que propriedade indica que uma barra de rolagem deverá aparecer automaticamente em um “scrolling windowed control”, quando ele não possuir espaço suficiente para exibir todos os controles inseridos nele?**

- a) PrintScale
- b) AutoSize
- c) AutoScroll
- d) Scaled





**Que propriedade indica que uma barra de rolagem deverá aparecer automaticamente em um “scrolling windowed control”, quando ele não possuir espaço suficiente para exibir todos os controles inseridos nele?**

- a) PrintScale
- b) AutoSize
- c) **AutoScroll**
- d) Scaled

Um “scrolling windowed control” é um controle que possui suporte a barras de rolagens, como é o caso de controles herdados de TScrollingWinControl, como o TScrollBar, o TFrame e o TForm. Todos estes controles possuem uma propriedade em comum, a “AutoScroll”, que indica que uma barra de rolagem deverá aparecer automaticamente quando o controle não possuir espaço suficiente para exibir todos os controles inseridos nele.



**Qual dos seguintes componentes pode ser usado como contêiner de outros componentes?**

- a) TStringGrid
- b) TPanel
- c) TImage
- d) TContainer



**Qual dos seguintes componentes pode ser usado como contêiner de outros componentes?**

- a) TStringGrid
- b) TPanel**
- c) TImage
- d) TContainer

O controle TPanel tem como função ajudar a organizar melhor nossa tela. Ele é utilizado como um contêiner de outros componentes.



**Em que *Form* você insere um *Panel*. No *Panel* você insere um *Button*. Qual objeto é o *Owner* do *Button*?**

- a) Form
- b) Application
- c) Button
- d) Panel



Em que *Form* você insere um *Panel*. No *Panel* você insere um *Button*. Qual objeto é o *Owner* do *Button*?

- a) Form
- b) Application
- c) Button
- d) Panel

Sempre que estivermos trabalhando em designer, não importa se inserirmos um controle diretamente no Form ou em outro controle, como um Panel ou um GroupBox, o Owner do controle sempre será o Form



**Os componentes em um *Component Template* são adicionados em um *Form* junto com suas propriedades e eventos pré-configurados.**

- a) Verdadeiro
- b) Falso



**Os componentes em um *Component Template* são adicionados em um *Form* junto com suas propriedades e eventos pré-configurados.**

- a) Verdadeiro
- b) Falso

Verdadeiro. Os componentes em um *Component Template* são adicionados em um *Form* junto com suas propriedades e eventos pré-configurados. Se você colocar algum código no evento onClick de um botão e transformá-lo em um *Component Template*, você verá que toda vez que esse Template for adicionado no formulário, o evento onClick já será declarado com o mesmo código fonte do componente original. Observe que os templates não irão apontar para o mesmo evento, cada um terá seu próprio evento declarado com uma cópia do código fonte original.



**Qual dos seguintes operadores ao ser inserido no exemplo abaixo atribuirá o valor 4 ao *Result*?**

Result := 16 {insert operator} 2;

- a) Div
- b) Xor
- c) Shl
- d) Shr
- e) Mod





**Qual dos seguintes operadores ao ser inserido no exemplo abaixo atribuirá o valor 4 ao *Result*?**

`Result := 16 {insert operator} 2;`

- a) Div
- b) Xor
- c) Shl
- d) Shr
- e) Mod

Se você está acostumado a desenvolver aplicações comerciais como controle de estoque e PDV's, provavelmente nunca precisou utilizar o operador Shr. Há duas formas de explicar este operador, então vamos lá. temos o número 16 que em binário é igual a 10000. Quando aplicamos o operador Shr estamos removendo número de bits especificado da direita, no caso 2 bits, desta forma ficamos com 100, que em decimal é igual a 4. Outra forma de compreender este operador é que sempre dividimos o primeiro número por 2 elevado ou outro fator, assim temos 16 dividido por 2 elevado a 2, ficará 16 dividido por 4 que resultará em 4. Shl segue o mesmo princípio, mas ao invés de retirar os bits da direita ele acrescenta sempre bits zero a direita, ou se preferir, ao invés de dividir ele irá multiplicar seu valor por 2 elevado ao fator especificado.



**Qual método é usado para realizar a verificação de tipo em *Variants*?**

- a) `TypeOf`
- b) `As`
- c) `GetVariantType`
- d) `VarType`
- e) `Is`



## Qual método é usado para realizar a verificação de tipo em *Variants*?

- a) `TypeOf`
- b) `As`
- c) `GetVariantType`
- d) `VarType`
- e) `Is`

Sabemos que em uma variável do tipo variant podemos armazenar praticamente qualquer tipo de valor, integers, strings, datetimes ou ponteiros. O método `VarType` retorna um valor do tipo `TVarData` que indica qual tipo do valor armazenado dentro de uma variável do Tipo variant.



**Uma classe pode possuir apenas um método construtor e um método destrutor.**

- a) Verdadeiro
- b) Falso



**Uma classe pode possuir apenas um método construtor e um método destrutor.**

- a) Verdadeiro
- b) Falso**

Falso, podemos ter quantos métodos construtores e destrutores queremos.



**Todo componente visual deve mapear a seguinte mensagem do Windows:**

- a) WM\_PAINT
- b) WM\_CTRL
- c) WM\_SYSCOMMAND
- d) WM\_MESSAGE



**Todo componente visual deve mapear a seguinte mensagem do Windows:**

- a) WM\_PAINT
- b) WM\_CTRL
- c) WM\_SYSCOMMAND
- d) WM\_MESSAGE

Para criarmos um componente visual é obvio que temos que desenhá-lo, para isso é necessário que o controle mapeie a mensagem WM\_PAINT. Mas não precisa ir tão longe não, classes como TGraphicControl e TCustomControl já possuem essa implementação e basta sobrescrevermos o método Paint.



**Qual das seguintes alternativas melhor define o método PostMessage.**

- a) O método envia uma mensagem diretamente para um controle do Windows e não aguarda retorno;
- b) O método envia uma mensagem diretamente para um controle do Windows e fica aguardando o retorno;
- c) O método envia uma mensagem para a fila do Windows e não aguarda retorno;
- d) O método envia uma mensagem para a fila do Windows e fica aguardando retorno;





## Qual das seguintes alternativas melhor define o método PostMessage.

- a) O método envia uma mensagem diretamente para um controle do Windows e não aguarda retorno;
- b) O método envia uma mensagem diretamente para um controle do Windows e fica aguardando o retorno;
- c) O método envia uma mensagem para a fila do Windows e não aguarda retorno;
- d) O método envia uma mensagem para a fila do Windows e fica aguardando retorno;

O método PPostMessage envia uma mensagem para a fila do Windows e não aguarda o retorno, praticamente é uma mensagem dentro de uma garrafa atirada no rio. Ao contrário do SendMessage em que o método envia uma mensagem diretamente para um controle do Windows e fica aguardando o retorno.



**Apenas propriedades definidas com escopo Published podem ser acessadas durante a execução pela RTTI**

- a) Verdadeiro
- b) Falso



## Apenas propriedades definidas com escopo Published podem ser acessadas durante a execução pela RTTI

- a) Verdadeiro
- b) Falso

Falso. Até algumas versões atrás isso seria verdade, mas a partir da nova RTTI o Delphi passou a dar suporte a acesso aos outros escopos durante o modo de execução



**Ao criarmos um componente que possui uma propriedade definida como uma classe, esta classe deve obrigatoriamente ser descendente de:**

- a) TComponent
- b) TObject
- c) TControl
- d) TPersistent



**Ao criarmos um componente que possui uma propriedade definida como uma classe, esta classe deve obrigatoriamente ser descendente de:**

- a) TComponent
- b) TObject
- c) TControl
- d) TPersistent

Para criar um componente devemos herdar de TComponent



**Para que um componente possa ser instalado ele deve informar a classe que será instalada e a paleta de componentes em que ele aparecerá. Onde estas informações são atribuídas?**

- a) No método Register de alguma unit do pacote;
- b) No constructor do componente
- c) Em um atributo do arquivo dpr;
- d) No Project Options do pacote;



**Para que um componente possa ser instalado ele deve informar a classe que será instalada e a paleta de componentes em que ele aparecerá. Onde estas informações são atribuídas?**

- a) No método Register de alguma unit do pacote;
- b) No constructor do componente
- c) Em um atributo do arquivo dpr;
- d) No Project Options do pacote;

Para instalarmos um componente no Delphi, devemos ter uma unit no nosso pacote que tenha um método Register implementado. Neste método, devemos fazer uma chamada para o RegisterComponents, passando o nome da paleta de componentes onde ele ficará e o array de classes dos componentes que será instalado.



## Qual a principal função da classe TControl

- a) Fornecer uma base para a construção de qualquer tipo de componentes.
- b) Fornecer métodos de desenho já implementados.
- c) Fornecer uma base para a construção de controles visuais.
- d) Fornecer uma interface para controle dos objetos na tela.





## Qual a principal função da classe TControl

- a) Fornecer uma base para a construção de qualquer tipo de componentes.
- b) Fornecer métodos de desenho já implementados.
- c) Fornecer uma base para a construção de controles visuais.
- d) Fornecer uma interface para controle dos objetos na tela.

Toda a vez que você quiser criar um controle visual é aconselhável herdar sua classe pelo menos de TControl, essa classe já fornece uma estrutura base para a construção desse tipo de componente.



**Quando abrimos um Form em tempo de design na IDE do Delphi, este Form é carregado trazendo as informações que estão armazenadas em seu arquivo dfm. Estas informações são carregadas durante que método do componente?**

- a) Fornecer uma base para a construção de qualquer tipo de componentes.
- b) Fornecer métodos de desenho já implementados.
- c) Fornecer uma base para a construção de controles visuais.
- d) Fornecer uma interface para controle dos objetos na tela.



**Quando abrimos um Form em tempo de design na IDE do Delphi, este Form é carregado trazendo as informações que estão armazenadas em seu arquivo dfm. Estas informações são carregadas durante que método do componente?**

- a) Fornecer uma base para a construção de qualquer tipo de componentes.
- b) Fornecer métodos de desenho já implementados.**
- c) Fornecer uma base para a construção de controles visuais.
- d) Fornecer uma interface para controle dos objetos na tela.

Quando trabalhamos com criação de componentes muitas vezes temos a necessidade de interceptar ou mesmo alterar alguma informação que é carregada do DFM. Para isso temos o método Loaded.



## Qual das seguintes alternativas está incorreta em relação do TGraphicControl?

- a) Herda de TCustomControl
- b) Não permite wrapper com windows.
- c) Não pode ser contêiner de outros componentes
- d) Depende de seu Parent para funcionalidades mais complexas.



## Qual das seguintes alternativas está incorreta em relação do TGraphicControl?

- a) Herda de TCustomControl
- b) Não permite wrapper com windows.
- c) Não pode ser contêiner de outros componentes
- d) Depende de seu Parent para funcionalidades mais complexas.

TGraphicControl é a classe base para criarmos componentes visuais e ela herda diretamente de TControl. Os controles herdados desta classe possuem algumas limitações como o fato de não fazer wrapper com o Windows, não poder ser contêiner de outros componente e depende de seu Parent para funcionalidade mais complexas.



**O Recurso do Delphi que permite abrir um popup menu com opções de design ao clicarmos com o botão direito do mouse sobre o componente é conhecido como:**

- a) Object Inspector
- b) Property Editor
- c) Object Repository
- d) Component Editor



**O Recurso do Delphi que permite abrir um popup menu com opções de design ao clicarmos com o botão direito do mouse sobre o componente é conhecido como:**

- a) Object Inspector
- b) Property Editor
- c) Object Repository
- d) **Component Editor**

Toda vez que eu quiser criar um Popup Menu para um componente em tempo de design devemos criar um Componente Editor



**O Recurso do Delphi que permite abrir um popup menu com opções de design ao clicarmos com o botão direito do mouse sobre o componente é conhecido como:**

- a) Object Inspector
- b) Property Editor
- c) Object Repository
- d) Component Editor





**O Recurso do Delphi que permite abrir um popup menu com opções de design ao clicarmos com o botão direito do mouse sobre o componente é conhecido como:**

- a) Object Inspector
- b) Property Editor
- c) Object Repository
- d) **Component Editor**

Toda vez que eu quiser criar um Popup Menu para um componente em tempo de design devemos criar um Componente Editor



## Qual das seguintes alternativas melhor define o método TypeInfo:

- a) É uma função que retorna um ponteiro para as informações de tipo em tempo de execução
- b) É uma função que retorna uma String contendo o nome do tipo em tempo de execução
- c) É uma Unit que contém métodos específicos para manipulação da RTTI
- d) É uma função que retorna o handler de uma classe



## Qual das seguintes alternativas melhor define o método TypeInfo:

- a) É uma função que retorna um ponteiro para as informações de tipo em tempo de execução
- b) É uma função que retorna uma String contendo o nome do tipo em tempo de execução
- c) É uma Unit que contém métodos específicos para manipulação da RTTI
- d) É uma função que retorna o handler de uma classe

Muito cuidado aqui, pois não estamos falando da unit TypeInfo nem do record TTypeInfo. A pergunta aqui se refere a “Função” TypeInfo, que é utilizada para retornar um ponteiro para as informações de tipo em tempo de execução. Esse assunto é tópico obrigatório para que quer ter um bom entendimento de RTTI



**Que validação nos assegura que uma variável do tipo TNotifyEvent está realmente apontando para um método.**

- a) FVariavel<>nil;
- b) Assigned(FVariavel);
- c) Not VarIsNull(FVariavel);
- d) FVariavel<>null;



**Que validação nos assegura que uma variável do tipo TNotifyEvent está realmente apontando para um método.**

- a) FVariavel<>nil;
- b) Assigned(FVariavel);**
- c) Not VarIsNull(FVariavel);
- d) FVariavel<>null;

Uma variável que aponta para um método não pode ser comparada utilizando a constante “nil”. Para verificarmos se ela está apontando ou não para alguém, devemos utilizar o método Assigned. O motivo para não podermos fazer a comparação com o “nil” é que o compilador entende que o método que é referenciado pela variável será executado e deverá retornar um valor, que será comparado ao “nil”, ao invés de apenas verificar se a variável aponta para algum lugar.



**O método SaveToFile de um TClientDataset é utilizado para exportar os seus dados para arquivos. por padrão esses arquivos possuem as seguintes extensões:**

- a) .date .xml
- b) .xml e .cds
- c) .date .csv
- d) .xml e .txt



**O método SaveToFile de um TClientDataset é utilizado para exportar os seus dados para arquivos. por padrão esses arquivos possuem as seguintes extensões:**

- a) .date .xml
- b) .xml e .cds**
- c) .date .csv
- d) .xml e .txt

Existem dois formatos que podem ser salvo por um TClientDataset, podendo ser binário ou xml. Por default estes formatos possuem as extensões cds e xml, respectivamente.



**Em um determinado trecho do código uma variável recebe um determinado valor. Durante a depuração existe a necessidade de alterar este valor. Essa alteração pode ser feita através da janela:**

- a) Watch
- b) Structure
- c) Evaluate/Modify
- d) ObjectInspector





**Em um determinado trecho do código uma variável recebe um determinado valor. Durante a depuração existe a necessidade de alterar este valor. Essa alteração pode ser feita através da janela:**

- a) Watch
- b) Structure
- c) Evaluate/Modify
- d) ObjectInspector

Mais popularmente conhecido pelo atalho, [Ctrl]+[F7]



## Como podemos declarar uma “Diretiva de Condicional” que seja válida para todo o projeto?

- a) Adicionando um {\$DECLARE DIRETIVA} no formulário principal da aplicação
- b) Adicionando a diretiva em MenuRun -> Parameters opção Parameters
- c) Adicionando a diretiva em Project Options -> Directories/Condicionais opções “Condicional defines”
- d) Adicionando um arquivo de recurso com a lista das diretivas



## Como podemos declarar uma “Diretiva de Condicional” que seja válida para todo o projeto?

- a) Adicionando um {\$DECLARE DIRETIVA} no formulário principal da aplicação
- b) Adicionando a diretiva em MenuRun -> Parameters opção Parameters
- c) Adicionando a diretiva em Project Options -> Directories/Condicionais opções “Condicional defines”
- d) Adicionando um arquivo de recurso com a lista das diretivas

Acessando o Project Options e indo em Directories/Condicionais, existe a opção “Condicional defines”. Nela, podemos adicionar diretivas condicionais que serão visíveis para todo o projeto



**Ao consumir um WebService verificamos que o certificado está expirado.  
Que opção devemos configurar no componente THTTPIO para ignorar esta  
validação?**

- a) UsesCertificate := False;
- b) HTTPWebNode.Enable := False;
- c) HTTPWebNoe.InvokeOptions := [solgnoreInvalidCerts];
- d) IgnoreInvalidCerts := True;



**Ao consumir um WebService verificamos que o certificado está expirado. Que opção devemos configurar no componente THTTPrIO para ignorar esta validação?**

- a) UsesCertificate := False;
- b) HTTPWebNode.Enable := False;
- c) **HTTPWebNoe.InvokeOptions := [solgnoreInvalidCerts];**
- d) IgnoreInvalidCerts := True;

O componente THTTPrIO possui uma propriedade chamada HTTPWebNode, essa propriedade é um objeto do tipo THHPReqResp e é utilizada pelo componente THTTPrIO para estabelecer a conexão com WebService. É na propriedade HTTPWebNode que podemos configurar como será o comportamento na hora de consumir nosso WebService. Uma dessas opções está em ignorar a validade de um certificado, bastando habilitar para isso a opção solgnoreInvalidCerts da propriedade InvokeOptions do HTTPWebNode.



## Qual a melhor definição para a propriedade KeyPreview de um formulário.

- a) Quando esta opção estiver habilitada é possível imprimir o formulário diretamente na impressora padrão do Windows
- b) Quando esta opção estiver habilitada os eventos de teclado irão ocorrer primeiro no formulário e depois no controle ativo
- c) Quando esta opção estiver desabilitada não é possível utilizar o teclado em nenhum controle do formulário
- d) Somente quando esta opção estiver ativa é possível utilizar o evento OnKeyPress do formulário.



## Qual a melhor definição para a propriedade KeyPreview de um formulário.

- a) Quando esta opção estiver habilitada é possível imprimir o formulário diretamente na impressora padrão do Windows
- b) Quando esta opção estiver habilitada os eventos de teclado irão ocorrer primeiro no formulário e depois no controle ativo
- c) Quando esta opção estiver desabilitada não é possível utilizar o teclado em nenhum controle do formulário
- d) Somente quando esta opção estiver ativa é possível utilizar o evento OnKeyPress do formulário.

Muitos de nós já tivemos a necessidade de fazer a famosa troca de Tab pelo Enter, para que o cursor mudasse de controle quando pressionássemos o Enter ao invés do Tab. É óbvio que ficaria um tanto quanto maçante alterar o evento OnKeyPress de cada controle da tela. Para isso temos a propriedade KeyPreview do formulário, que quando habilitada irá fazer com que cada evento de teclado ocorra primeiro no formulário e depois no componente, ou seja, quando você digitar alguma coisa, primeiro será executado o evento OnKeyPress no formulário e é nesta que fazemos a troca do Tab pelo Enter



**Referente ao bloco de código abaixo, considerando que o CDS iniciou o bloco com 10 registros, quantos registros restarão ao final do bloco?**

```
CDS.First;  
while not CDS.Eof do  
begin  
    CDS.Delete;  
    CDS.Next;  
end;
```

- a) 0;
- b) 5;
- c) 10;
- d) O código irá gerar uma exceção





**Referente ao bloco de código abaixo, considerando que o CDS iniciou o bloco com 10 registros, quantos registros restarão ao final do bloco?**

```
CDS.First;  
while not CDS.Eof do  
begin  
    CDS.Delete;  
    CDS.Next;  
end;
```

- a) 0;
- b) 5;**
- c) 10;
- d) O código irá gerar uma exceção

O bloco de exemplo é muito interessante por demonstrar um erro clássico de implementação. Ao efetuarmos um loop que deleta os registros de um dataset, devemos ter cuidado para não incluirmos a instrução next para mover o cursor para o próximo registro, uma vez que o registro atual está sendo excluído e o cursor já será posicionado no próximo. Se você mantiver o Next no código, sempre pulará um registro. Como nosso exemplo, onde será excluída apenas a metade dos registros.



## No evento BeforeUpdateRecord do TDataSetProvider qual a finalidade do parâmetro Applied?

- a) É utilizado para informarmos se as alterações efetuadas no TClientDataset serão enviadas para o banco de dados
- b) Informa se o método ApplyUpdates do TClientDataset foi chamado
- c) Informa se o registro foi “commitado” no banco de dados
- d) É utilizado para informarmos que registros serão atualizados no TClientDataset com as informações contidas no banco de dados



## No evento BeforeUpdateRecord do TDataSetProvider qual a finalidade do parâmetro Applied?

- a) É utilizado para informarmos se as alterações efetuadas no TClientDataset serão enviadas para o banco de dados
- b) Informa se o método ApplyUpdates do TClientDataset foi chamado
- c) Informa se o registro foi “commitado” no banco de dados
- d) É utilizado para informarmos que registros serão atualizados no TClientDataset com as informações contidas no banco de dados

Ao atribuirmos true para o parâmetro Applied do evento BeforeUpdateRecord de um TDataSetProvider, estamos informando ao componente que as alterações já foram aplicadas e não serão feitas pelo provider. Esta opção é utilizada quando queremos executar manualmente alguma alteração através de outros comandos SQL e não deixar que o ClientDataset e o DataSetProvider façam isso sozinhos.



## Qual das seguintes alternativas melhor define o TSoptwatch.

- a) É um novo componente criado para substituir o componente TTimer.
- b) É uma classe utilizada para cronometrar o tempo de vida de uma TThread.
- c) É uma classe utilizada para parar o relógio do sistema operacional
- d) é um Record utilizado para obter acesso a temporizadores de alta resolução que podem ser usados para monitorar o tempo gasto na realização de algumas operações.



## Qual das seguintes alternativas melhor define o TSoptwatch.

- a) É um novo componente criado para substituir o componente TTimer.
- b) É uma classe utilizada para cronometrar o tempo de vida de uma TThread.
- c) É uma classe utilizada para parar o relógio do sistema operacional
- d) é um Record utilizado para obter acesso a temporizadores de alta resolução que podem ser usados para monitorar o tempo gasto na realização de algumas operações.

É um Record utilizado para obter acesso a temporizadores de alta resolução que podem ser usado para monitorar o tempo gasto na realização de algumas operações.



**Qual das seguintes opções representa um tipo utilizado para trabalharmos com valores do tipo “Inteiro”?**

- a) String;
- b) Double;
- c) Integer;
- d) Boolean;



**Qual das seguintes opções representa um tipo utilizado para trabalharmos com valores do tipo “Inteiro”?**

- a) String;
- b) Double;
- c) **Integer;**
- d) Boolean;

O tipo integer é utilizado para representar valores numéricos inteiros, possui um faixa de -2147483648 a 2147483687. Double é usado para valores fracionados, também conhecido como ponto flutuante. Para caracteres alfanuméricos, letras e números, temos a String. E se você é daqueles que sim “sim” e não é “não”, existe o tipo Boolean, apenas com a possibilidade True ou False, nada de “talvez”.



**Qual destas opções é método utilizado para retornar um valor?**

- a) Function;
- b) Procedure;
- c) Class;
- d) Property;





## Qual destas opções é método utilizado para retornar um valor?

- a) **Function;**
- b) Procedure;
- c) Class;
- d) Property;

Quando falamos de métodos, estamos sempre nos referindo a uma Procedure, uma Function, um Constructor ou um Destructor. Em nossas opções temos então Function e Procedure. A diferença entre elas é que a Function retorna valor e a Procedure não. Mesmo assim, uma Function pode ser chamada sem a necessidade de atribuir seu resultado.



## Qual a funcionalidade do método StringReplace?

- a) Criar um arquivo de recurso (.res);
- b) Substituir uma sequência de caracteres dentro de uma string
- c) Verificar se a estrutura da String está correta
- d) Localizar uma sequência de caracteres dentro de uma string



## Qual a funcionalidade do método StringReplace?

- a) Criar um arquivo de recurso (.res);
- b) Substituir uma sequência de caracteres dentro de uma string
- c) Verificar se a estrutura da String está correta
- d) Localizar uma sequência de caracteres dentro de uma string

O StringReplace é uma função utilizada para substituir uma sequência de caracteres de uma string por outra.

Declarada da seguinte forma:

**function** StringReplace(**const** S : **String**; **const** OldPattern : **String**; **const** NewPattern : **String**; Flags : TReplaceFlags) : **String**; **overload**;

Onde **S** representa a String que terá o trecho da string definido pelo **OldPattern**, substituído pelo que estiver no parâmetro **NewPattern**. O parâmetro **Flag** é um Set(conjunto) que indica como o método irá se comportar. Este parâmetro pode ser atribuído com a combinação dos tipos **rfReplaceAll** e **rflgnoreCase**. **rfReplaceAll** indica que todas as incidências da subString **OldPattern** contida em **S** será substituída pela **NewPattern**. Caso não for atribuída apenas à primeira incidência é alterada. **rflgnoreCase** indica que para localizar a subString **OldPattern** o **CharCase** será ignorado.



**Para transformar um valor inteiro em uma string devemos utilizar o método:**

- a) String();
- b) StrToInt();
- c) ToString();
- d) IntToStr();



**Para transformar um valor inteiro em uma string devemos utilizar o método:**

- a) `String();`
- b) `StrToInt();`
- c) `ToString();`
- d) `IntToStr();`

`IntToStr` transforma um valor inteiro em uma string. Para a operação inversa temos o método `StrToInt`.



## Qual a principal diferença entre um ShowMessage e um MessageDlg?

- a) ShowMessage apenas exibe uma janela com um texto e um botão de OK, enquanto o MessageDlg exibe uma janela com um texto e nos permite escolher os botões como Yes e No, capturando o botão que foi clicado
- b) ShowMessage exibe uma janela com um texto e nos permite escolher os botões como Yes e No, enquanto o MessageDlg apenas exibe uma janela com um texto e um botão de OK
- c) ShowMessage abre uma janela padrão para envio de mensagens de e-mail, enquanto o MessageDlg permite escolher a ferramenta de e-mail
- d) Não existe diferença entre um ShowMessage e um MessageDlg, os dois métodos tem a mesma finalidade



## Qual a principal diferença entre um ShowMessage e um MessageDlg?

- a) ShowMessage apenas exibe uma janela com um texto e um botão de OK, enquanto o MessageDlg exibe uma janela com um texto e nos permite escolher os botões como Yes e No, capturando o botão que foi clicado
- b) ShowMessage exibe uma janela com um texto e nos permite escolher os botões como Yes e No, enquanto o MessageDlg apenas exibe uma janela com um texto e um botão de OK
- c) ShowMessage abre uma janela padrão para envio de mensagens de e-mail, enquanto o MessageDlg permite escolher a ferramenta de e-mail
- d) Não existe diferença entre um ShowMessage e um MessageDlg, os dois métodos tem a mesma finalidade

Para exibir uma mensagem de alerta para o usuário temos duas opções entre as mais utilizadas, ShowMessage e MessageDlg. ShowMessage é uma procedure e apenas exibe uma janela com um texto e um botão de OK. MessageDlg é uma Function que exibe uma janela com um texto e nos permite escolher entre um ou mais botões, como Yes, No, Cancel e Ok. Permite também que se possa escolher o ícone da janela, o que dá as características visuais de uma mensagem de Warning, Erro, Informação ou Confirmação. Além disso, por ser uma função, ela nos retorna o botão que foi clicado.



**Para que possamos utilizar métodos e classes que estão em outras units, devemos declarar estas na cláusula:**

- a) Export;
- b) Import;
- c) Uses
- d) Include;





**Para que possamos utilizar métodos e classes que estão em outras units, devemos declarar estas na cláusula:**

- a) Export;
- b) Import;
- c) **Uses**
- d) Include;

Toda vez que quisermos utilizar um método, classe, constante ou seja lá o que for, que não esteja em nossa unit, devemos declarar a unit que possui o método na cláusula Uses da unit onde o elemento será utilizado



**Qual destas opções representa a classe base de todas as classes do Delphi?**

- a) TComponent
- b) TObject
- c) TClass
- d) TForm



## Qual destas opções representa a classe base de todas as classes do Delphi?

- a) TComponent
- b) TObject
- c) TClass
- d) TForm

Todas as classes do Delphi tem como ponto de partida TObject.



## Quando o evento OnChange de um TEdit é executado?

- a) Quando o componente recebe foco
- b) Quando clicamos no componente
- c) Quando alteramos o tamanho do componente
- d) Quando alteramos o texto do componente



## Quando o evento OnChange de um TEdit é executado?

- a) Quando o componente recebe foco
- b) Quando clicamos no componente
- c) Quando alteramos o tamanho do componente
- d) Quando alteramos o texto do componente

O evento OnChange de um TEdit é chamado toda vez que alteramos a propriedade Text do componente, seja via formulário ou código.



## Qual a finalidade do método OnCloseQuery de um formulário?

- a) Validar se um formulário pode ser fechado ou não
- b) Capturar toda vez que uma Query é fechada dentro do formulário
- c) Executar algum procedimento antes do encerramento
- d) da aplicação
- e) O método existe apenas para compatibilidade com versões anteriores.



## Qual a finalidade do método OnCloseQuery de um formulário?

- a) Validar se um formulário pode ser fechado ou não
- b) Capturar toda vez que uma Query é fechada dentro do formulário
- c) Executar algum procedimento antes do encerramento
- d) da aplicação
- e) O método existe apenas para compatibilidade com versões anteriores.

Através do evento OnCloseQuery podemos validar se um formulário pode ser fechado ou não. Esse evento é chamado quando executamos o Close de um formulário. O evento possui um parâmetro chamado CanClose. Quando este parâmetro recebe False, não será permitido fechar o formulário.



**Para capturar uma exceção, o bloco de código deve estar dentro de uma instrução:**

- a) Try/Finally
- b) Try/Error
- c) Try/Catch
- d) Try/Except





## Para capturar uma exceção, o bloco de código deve estar dentro de uma instrução:

- a) Try/Finally
- b) Try/Error
- c) Try/Catch
- d) Try/Except

Para capturar uma exceção, ou seja, um erro durante a execução do programa, e tratá-la, devemos colocar nosso bloco de código entre uma instrução Try..Except. Para fazer tratamento desejado, seja o cancelamento da exceção ou sua propagação utilize o bloco seguinte, entre o Except..End;

Ex:

```
Try
{meu código}
Except
{tratamento da exceção}
end;
```



**Ao utilizarmos um componente Timer1 do tipo TTimer qual desses métodos nos permite pará-lo**

- a) Timer1.Stop;
- b) Timer1.Sleep;
- c) Timer1.Enable := False;
- d) Timer1.Free;



## Ao utilizarmos um componente Timer1 do tipo TTimer qual desses métodos nos permite pará-lo

- a) Timer1.Stop;
- b) Timer1.Sleep;
- c) **Timer1.Enable := False;**
- d) Timer1.Free;

A propriedade Enable do TTimer é responsável por iniciá-lo (True) ou pará-lo (False). Quando Ativo, o evento OnTimer é executado a cada intervalo (em milissegundos), definido na propriedade Interval.



## Em que classe é declarada a propriedade Touch?

- a) Persistent
- b) Component
- c) Control
- d) WinControl



## Em que classe é declarada a propriedade Touch?

- a) Persistent
- b) Component
- c) **Control**
- d) WinControl

A propriedade Touch é declarada na classe TControl. É isso que permite que todos componentes visuais tenham suporte a Gesture



**Qual dessas opções não é válida para envio de mensagens do Windows?**

- a) SendMessage
- b) Perform;
- c) PostMessage;
- d) ShowMessage;



## Qual dessas opções não é válida para envio de mensagens do Windows?

- a) SendMessage
- b) Perform;
- c) PostMessage;
- d) ShowMessage;

SendMessage, PostMessage e Perform podem ser utilizados para envio de mensagens do Windows. Cada um delas com uma peculiaridade. SendMessage é o que chamamos de envio de mensagem Síncrona, a mensagem é enviada para a fila do Windows e o sistema não aguarda sua execução. O Perform tem a mesma características do SendMessage, sua principal diferença é que ele é um método implementado na VCL, em quanto o SendMessage pertence a própria API do Windows.



**Para podermos depurar uma aplicação Web Broker no Delphi, ela deve ser desenvolvida como:**

- a) ISAPI
- b) CGI
- c) ASPX
- d) Web App Debugger





**Para podermos depurar uma aplicação Web Broker no Delphi, ela deve ser desenvolvida como:**

- a) ISAPI
- b) CGI
- c) ASPX
- d) Web App Debugger

Apesar de já existirem técnicas de depuração para aplicações CGI e ISAPI, o Web App Debugger foi construído especificamente para este fim.



## Como é possível fazer uma chamada da calculadora do Windows no menu do Delphi?

- a) Selecionando o menu File -> Open e selecionando o arquivo Calc.exe
- b) Selecionando o menu Tool -> Configure Tools e adicionar o arquivo Calc.exe
- c) Selecionando o menu Component -> Import Component e adicionar o arquivo Calc.exe
- d) Não é possível adicionar chamada para outros executáveis no Delphi.



## Como é possível fazer uma chamada da calculadora do Windows no menu do Delphi?

- a) Selecionando o menu File -> Open e selecionando o arquivo Calc.exe
- b) Selecionando o menu Tool -> Configure Tools e adicionar o arquivo Calc.exe**
- c) Selecionando o menu Component -> Import Component e adicionar o arquivo Calc.exe
- d) Não é possível adicionar chamada para outros executáveis no Delphi.

Selecionando o menu Tool -> Configure Tools podemos adicionar um chamada para qualquer arquivo executável (.exe, .com, .pif) ao menu do Delphi



**Qual dos escopos abaixo indica que um método será visível única e exclusivamente na classe origem e em suas descendentes?**

- a) Strict Private
- b) Private
- c) Strict Protected
- d) Protected



**Qual dos escopos abaixo indica que um método será visível única e exclusivamente na classe origem e em suas descendentes?**

- a) Strict Private
- b) Private
- c) **Strict Protected**
- d) Protected

O escopo Protected especifica que um método será visível na própria classe, suas descendentes e em suas Friend Class (classes que estão declaradas na mesma Unit). A partir do Delphi 2005 podemos utilizar o escopo String Protected, que remove a visibilidade das Friend Class, fazendo com que um método seja visível única e exclusivamente na classe de origem e em suas descendentes.



## Qual das sintaxes mostra uma propriedade somente leitura?

- a) property TotalValue : Integer Read GetTotalValue;
- b) property Speed : real read GetSpeed write SetSpeed;
- c) property Width : Integer read GetWidth write SetWidth;
- d) property Height : Integer read GetHeight write SetHeight;



## Qual das sintaxes mostra uma propriedade somente leitura?

- a) `property TotalValue : Integer Read GetTotalValue;`
- b) `property Speed : real read GetSpeed write SetSpeed;`
- c) `property Width : Integer read GetWidth write SetWidth;`
- d) `property Height : Integer read GetHeight write SetHeigh;`

Para construir propriedades Read-Only, basta não implementar a parte Write da propriedade.



## Qual a finalidade do método ParamCount da unit System?

- a) Retornar o número de parâmetros de um objeto TDataset
- b) Retornar o número de parâmetros passado para o programa por linha de comando.
- c) Retornar o número de itens de um objeto TParam.
- d) Retornar o número de parâmetros de um objeto TValueList





## Qual a finalidade do método ParamCount da unit System?

- a) Retornar o número de parâmetros de um objeto TDataset
- b) Retornar o número de parâmetros passado para o programa por linha de comando.
- c) Retornar o número de itens de um objeto TParam.
- d) Retornar o número de parâmetros de um objeto TValueList

Sabe quando você executa um programa no prompt de comando através de uma instrução como esta, "Programa.exe param1 param2". Cada item que é separado do nome do programa é um parâmetro da aplicação. O ParamCount retorna justamente a quantidade de parâmetros passado para o programa.



**IOTAIDEInsightService é uma interface que nos permite interagir com a “IDE Insight”. Um de seus métodos, o AddItem, permite que você adicione um elemento para a janela “IDE Insight”. Sendo que um dos parâmetros a ser passado é o item que queremos adicionar. Este “Item” deve ser um objeto de uma classe que implemente a seguinte interface:**

- a) IOTAIDEInsightServiceItem
- b) INTAIDEInsightItem
- c) IOTAIDEInsightItem
- d) IIDEInsightItem



**IOTAIDEInsightService é uma interface que nos permite interagir com a “IDE Insight”. Um de seus métodos, o AddItem, permite que você adicione um elemento para a janela “IDE Insight”. Sendo que um dos parâmetros a ser passado é o item que queremos adicionar. Este “Item” deve ser um objeto de uma classe que implemente a seguinte interface:**

- a) IOTAIDEInsightServiceItem
- b) INTAIDEInsightItem**
- c) IOTAIDEInsightItem
- d) IIDEInsightItem

A “IDE Insight”, um formulário de atalho que nos permite acessar qualquer ponto da IDE, seja menu, componente, configuração, e tudo mais que passar por sua cabeça e estiver presente na IDE. Mas o interessante, é que ela pode ser customizada, assim como praticamente qualquer coisa no Delphi. Sabendo a unit, interface e método certo, você pode “Tunar” completamente sua ferramenta. É como fazer nitroglicerina o difícil não é misturar ou descobrir seus componentes, mas sim não explodir durante o processo, a implementação errada pode dar muita dor de cabeça. Mas, voltando ao Delphi, cada item que temos na “IDE Insight” é um derivado da interface INTAIDEInsightItem.



**O atalho <Ctrl+I> permite a indentação do código selecione. Que outro atalho possui a mesma funcionalidade?**

- a) <Tab>
- b) <Shift+I>
- c) <Ctrl+U>
- d) <Del>



**O atalho <Ctrl+I> permite a indentação do código selecionado. Que outro atalho possui a mesma funcionalidade?**

- a) <Tab>
- b) <Shift+I>
- c) <Ctrl+U>
- d) <Del>

Para uma melhor adaptação dos programadores oriundos de outras linguagens, onde o <Tab> é o atalho para indentação do código selecionado, este foi incluído tendo a mesma funcionalidade do <Ctrl+I>



**O Delphi possui recurso de formatação de código, onde podemos, por exemplo, definir padrões de indentação e aplicá-los em um bloco de código ou em toda uma Unit.**

- a) Verdadeiro
- b) Falso



**O Delphi possui recurso de formatação de código, onde podemos, por exemplo, definir padrões de indentação e aplicá-los em um bloco de código ou em toda uma Unit.**

- a) Verdadeiro
- b) Falso

Verdadeiro. Hoje no menu "Edit" a opção "Format Source" ou <Ctrl+D>. Esta opção aplica padrões de formatação no código fonte. Esta formatação pode ser configurada em Tool -> Options -> Formatter -> Delphi



**Durante a depuração é possível mover o ponteiro indicativo da linha que está sendo depurada, tanto para uma linha anterior com para uma linha seguinte, possibilitando que você salte um trecho de código ou execute-o novamente**

- a) Verdadeiro
- b) Falso





**Durante a depuração é possível mover o ponteiro indicativo da linha que está sendo depurada, tanto para uma linha anterior com para uma linha seguinte, possibilitando que você salte um trecho de código ou execute-o novamente**

- a) Verdadeiro
- b) Falso

Verdadeiro. POder mover o depurador para outra linha é sensacional. Você passa por uma linha que tenha um determinado cálculo, usar o Evaluate\Modify para alterar o valor de uma variável e depois posicionar o ponteiro indicativo do depurados na linha do cálculo para testá-lo novamente



**Ao depurarmos nossa aplicação, muitas vezes necessitamos incluir informações no “Event Log”. O comando que nos permite incluir texto no Event Log é:**

- a) `EventLog.Add()`
- b) `Clipboard.asText`
- c) `InsertDebugString()`
- d) `OutputDebugString()`



**Ao depurarmos nossa aplicação, muitas vezes necessitamos incluir informações no “Event Log”. O comando que nos permite incluir texto no Event Log é:**

- a) `EventLog.Add()`
- b) `Clipboard.AsText`
- c) `InsertDebugString()`
- d) `OutputDebugString()`

Muitas vezes temos a necessidade de incluir alguma informação no Event Log, por exemplo, marcar o tempo que o programa levou para passar por uma determinadas linhas, sem ter que montar alguma coisa que terá que ser retirada ao gerar uma versão para distribuição. Para incluir alguma informação no Event Log utilizamos o comando `OutputDebugString()`, passando como parâmetro o texto que queremos incluir.



## TRTTIContext é:

- a) Uma Classe
- b) Um tipo enumerado
- c) Um Record
- d) Uma Interface



## TRTTIContext é:

- a) Uma Classe
- b) Um tipo enumerado
- c) Um Record
- d) Uma Interface

Está aí uma coisa que eu não consigo aceitar, pelo menos eu não faria assim, mas TRTTIContext é um Record.



**Quando utilizamos a TRTTIProperty para buscar o valor de uma propriedade, esse valor é retornado sempre com o seguinte tipo.**

- a) TValue
- b) Variant
- c) String
- d) TProperty



**Quando utilizamos a TRTTIProperty para buscar o valor de uma propriedade, esse valor é retornado sempre com o seguinte tipo.**

- a) TValue
- b) Variant
- c) String
- d) TProperty

O valor da propriedade retornado pelo TRTTIProperty é sempre do tipo TValue. E vão se acostumando, TValue veio para ficar, e pelo visto vai ficar cada vez mais comum encontrá-lo como retorno de métodos.



### **Qual das alternativas seguintes traz a melhor definição para o recurso conhecido como “class constructor”?**

- a) Class constructors são utilizados para que possamos chamar o método construtor sem a necessidade de ter um objeto instanciado.
- b) Basicamente, o class constructor é uma alternativa para o código de inicialização de uma Unit
- c) Construtores por si só já são métodos de classe, sendo o termo “class constructor” apenas uma convenção
- d) O recurso class constructor é utilizado apenas em aplicações .Net





## Qual das alternativas seguintes traz a melhor definição para o recurso conhecido como “class constructor”?

- a) Class constructors são utilizados para que possamos chamar o método construtor sem a necessidade de ter um objeto instanciado.
- b) Basicamente, o class constructor é uma alternativa para o código de inicialização de uma Unit
- c) Construtores por si só já são métodos de classe, sendo o termo “class constructor” apenas uma convenção
- d) O recurso class constructor é utilizado apenas em aplicações .Net

Para começar um construtor de classe não tem nada em comum com um construtor padrão. Ele é muito semelhante ao “Initialization” de uma unit, e pode ser utilizado como alternativa. Sendo que se os dois casos existirem, o construtor da classe é executado primeiro. Outra diferença é que o código do “Initialization” da unit é invariavelmente executado se a unit for compilada no programa, enquanto o construtor da classe é chamado apenas se a classe for realmente utilizada



## Qual a finalidade da diretiva “delayed”?

- a) Permitir que um trecho de código aguardasse até que algum processo seja finalizado.
- b) Fazer com que a carga de uma DLL implícita seja feita apenas quando ocorrer à primeira execução de um de seus métodos
- c) Fazer com que o método de uma DLL implícita seja carregada apenas quando ocorrer à primeira execução
- d) Permitir que DLL desenvolvidas em processadores de 64 bits possam ser executadas em programas desenvolvidos em plataformas de 32 bits



## Qual a finalidade da diretiva “delayed”?

- a) Permitir que um trecho de código aguardasse até que algum processo seja finalizado.
- b) Fazer com que a carga de uma DLL implícita seja feita apenas quando ocorrer à primeira execução de um de seus métodos
- c) Fazer com que o método de uma DLL implícita seja carregada apenas quando ocorrer à primeira execução
- d) Permitir que DLL desenvolvidas em processadores de 64 bits possam ser executadas em programas desenvolvidos em plataformas de 32 bits

A diretiva “delayed” é utilizada para “atrasar” o carregamento de funções até o momento em que são chamadas. Mas um detalhe, seu objectivo não é evitar o carregamento da DLL implícito, isso irá ocorrer de qualquer maneira.



## Referente ao código abaixo podemos afirmar que:

```
{$SCOPEDENUMS ON}
```

```
type
```

```
  TTipoPessoa = (tpFisica, tpJuridica)
```

- a) Ativar a diretiva “\$SCOPEDENUMS” significa que o tipo enumerado só poderá ser acessado pelo seu nome completo. Ex: TTipoPessoa.tpFisica
- b) Ativar a diretiva “\$SCOPEDENUMS” significa que os nomes dos itens do enumerado não poderá ser utilizado por nenhuma outra unit
- c) A diretiva “\$SCOPEDENUMS” permite que o enumerado possa ser acessado pela RTTI
- d) A diretiva “\$SCOPEDENUMS” existe apenas para fins de compatibilidade com versões anteriores e não causa qualquer mudança na funcionalidade do enumerado.



## Referente ao código abaixo podemos afirmar que:

```
{$SCOPEDENUMS ON}
```

```
type
```

```
  TTipoPessoa = (tpFisica, tpJuridica)
```

- a) Ativar a diretiva “\$SCOPEDENUMS” significa que o tipo enumerado só poderá ser acessado pelo seu nome completo. Ex: TTipoPessoa.tpFisica
- b) Ativar a diretiva “\$SCOPEDENUMS” significa que os nomes dos itens do enumerado não poderá ser utilizado por nenhuma outra unit
- c) A diretiva “\$SCOPEDENUMS” permite que o enumerado possa ser acessado pela RTTI
- d) A diretiva “\$SCOPEDENUMS” existe apenas para fins de compatibilidade com versões anteriores e não causa qualquer mudança na funcionalidade do enumerado.

Ativar a diretiva “\$SCOPEDENUMS” significa que o tipo enumerado só poderá ser acessado pelo seu nome completo (absoluto), ou seja, devemos proceder o nome do enumerado pelo nome do seu tipo. No nosso exemplo, TTipoPessoa.tpFisica.



**Ao criarmos um editor de propriedade, qual destas bibliotecas será requerida pelo pacote?**

- a) propedit.dcp
- b) designer.dcp
- c) designide.dcp
- d) interface.dcp



## Ao criarmos um editor de propriedade, qual destas bibliotecas será requerida pelo pacote?

- a) propedit.dcp
- b) designer.dcp
- c) designide.dcp
- d) interface.dcp

Quando trabalhamos com criação de componentes, muitas vezes teremos a necessidade de criarmos editores de propriedades e de componentes. Para que seja possível implementar estas funcionalidades em nossos componentes. Para que seja possível implementar estas funcionalidades em nossos componentes precisamos fazer uso de duas Units, a DesignIntf e a DesignEditors. Estas Units só estarão disponíveis ao adicionarmos a referência do pacote designide ao “requires” do pacotes do nosso componente



**Quando queremos vincular um novo gesture a um componente devemos atribuir a ele um GetureManager. A propriedade GetureManager de um componente está contida na seguinte propriedade**

- a) Gesture
- b) Touch
- c) MouseMove
- d) Action





**Quando queremos vincular um novo gesture a um componente devemos atribuir a ele um GetureManager. A propriedade GetureManager de um componente está contida na seguinte propriedade**

- a) Gesture
- b) Touch**
- c) MouseMove
- d) Action

Atualmente todo componente visual possui suporte a Touch e Gesture. Isso porque a propriedade responsável, a Touch, é declarada na classe TControl (classe base para todos os componentes visuais). Nessa propriedade atribuímos o GestureManager e mais uma série de configurações.



## Quais alternativas representam um estilo válido para um BalloonHint?

- a) bhsStandard
- b) BalloonStyle
- c) NoneStyle
- d) bhsSingle



## Quais alternativas representam um estilo válido para um BalloonHint?

- a) **bhsStandard**
- b) BalloonStyle
- c) NoneStyle
- d) bhsSingle

O BalloonHint é um componente utilizado para customização de Hints (aqueles balões que parecem com informações quando deixamos o cursor do mouse parado em cima de algum componente). E ele possui dois estilos `bhsStandard` e `bhsBalloon`.



**Atualmente o código fonte de um Type Library possui um formato texto.  
Esse arquivo é armazenado com a seguinte extensão.**

- a) .tbl
- b) .pas
- c) .ridl
- d) .dll



**Atualmente o código fonte de um Type Library possui um formato texto. Esse arquivo é armazenado com a seguinte extensão.**

- a) .tbl
- b) .pas
- c) .ridl
- d) .dll

A Type Library no formato texto permite que possamos alterá-la diretamente no código, possibilitando entre vários benefícios a utilização de comentários e seu versionamento através de comparação textual. Esses são salvos com a extensão ridl.



**Apenas WebService construídos em SOAP 1.1 podem ser importados pelos Delphi.**

- a) Verdadeiro
- b) Falso



**Apenas Webservice construídos em SOAP 1.1 podem ser importados pelos Delphi.**

- a) Verdadeiro
- b) Falso

A partir do Delphi 2010 é possível importar WebServices em SOAP 1.2 além do SOAP 1.1



## O que podemos afirmar sobre o código a seguir?

```
TCustomListControlHelper<T> = class  
  helper for TCustomListControl
```

- a) É uma classe helper que possibilita tornar uma classe normal em uma classe genérica.
- b) O código não é válido, pois não podemos transformar uma classe normal em uma classe genérica através de um class helper
- c) O código não é válido, pois devemos atribuir uma constraint ao tipo genérico limitando apenas classes que possuam construtores.
- d) É uma declaração de tipo para implementação de eventos que utilizam tipos genéricos.





## O que podemos afirmar sobre o código a seguir?

```
TCustomListControlHelper<T> = class  
  helper for TCustomListControl
```

- a) É uma classe helper que possibilita tornar uma classe normal em uma classe genérica.
- b) O código não é válido, pois não podemos transformar uma classe normal em uma classe genérica através de um class helper
- c) O código não é válido, pois devemos atribuir uma constraint ao tipo genérico limitando apenas classes que possuam construtores.
- d) É uma declaração de tipo para implementação de eventos que utilizam tipos genéricos.

Se fosse verdade seria muito bom, mas não é possível transformar uma classe normal em uma classe genérica utilizando class helper



**Quando uma classe é definida com a diretiva “abstract” nenhum de seus métodos podem ser implementados.**

- a) Verdadeiro
- b) Falso



**Quando uma classe é definida com a diretiva “abstract” nenhum de seus métodos podem ser implementados.**

- a) Verdadeiro
- b) Falso**

Declarar uma “class abstract” apenas indica que esta classe “teoricamente” se trata de uma classe abstract. Teoricamente por quê? A grande verdade é que nada se altera em sua classe, você pode herdá-la, implementar métodos declarar propriedades normalmente.



**O recurso que possibilita localizar qualquer coisa no IDE do Delphi, desde componentes a opções da ferramenta se chama?**

- a) Find Resource
- b) Project Manager
- c) Code Complete
- d) IDE Insight



**O recurso que possibilita localizar qualquer coisa no IDE do Delphi, desde componentes a opções da ferramenta se chama?**

- a) Find Resource
- b) Project Manager
- c) Code Complete
- d) IDE Insight

Através do <F6> ou <Ctrl+.> podemos acessar a IDE Insight localizar qualquer coisa na IDE do Delphi, seja componentes para o formulário, opções de projeto e ferramenta, itens de menu... Lembrou de mais alguma coisa, digita nele.



**Qual técnica de loop permite varrer os elementos de uma lista sem a necessidade de sabermos o número total de itens contidos nela?**

- a) for to
- b) for each
- c) for in
- d) while do



## Qual técnica de loop permite varrer os elementos de uma lista sem a necessidade de sabermos o número total de itens contidos nela?

- a) for to
- b) for each
- c) **for in**
- d) while do

Eu particularmente já sou viciado no “for in”, mas depois que surgiu o `TObjectList<T>` e as demais listas genéricas, ele se tornou indispensável. Observe o exemplo a seguir:

```
var
  ILista : TObjectList<TItem>;
  IItem : TItem;
  I : Integer;
begin
  //for to
  for I := 0 to ILista.Count - 1 do
    Memo1.Lines.Add(ILista.Item[I].Nome);

  //for in
  for IItem in ILista do
    Memo1.Lines.Add(IItem.Nome);
end;
```



## Como pode ser criado um novo atributo para ser utilizado em uma classe?

- a) Criando uma classe herdada de TCustomAttribute.
- b) Criando uma classe que implemente a Interface IAttribute.
- c) Instanciando um objeto AttributeManager e adicionando a classe que terá o novo atributo.
- d) Declarando uma diretiva de compilação {\$ATTRIBUTE NOME\_DO\_ATRIBUTO}





## Como pode ser criado um novo atributo para ser utilizado em uma classe?

- a) Criando uma classe herdada de TCustomAttribute.
- b) Criando uma classe que implemente a Interface IAttribute.
- c) Instanciando um objeto AttributeManager e adicionando a classe que terá o novo atributo.
- d) Declarando uma diretiva de compilação {\$ATTRIBUTE NOME\_DO\_ATRIBUTO}

Para criarmos um Atributo devemos primeiramente criar uma classe herdada de TCustomAttribute, que é a classe base para todos atributos que serão criados



**Para utilizarmos as novas funcionalidades da RTTI devemos fazer uso da seguinte Unit:**

- a) SysUtils
- b) TypInfo
- c) RTTI
- d) ToolsAPI



**Para utilizarmos as novas funcionalidades da RTTI devemos fazer uso da seguinte Unit:**

- a) SysUtils
- b) TypInfo
- c) **RTTI**
- d) ToolsAPI

Pode parecer pegadinha mas não é! É a unit RTTI mesmo.



## O que podemos afirmar sobre a declaração abaixo?

```
TList<T> = class(TEnumerable<T>)
```

```
  TMinhaClasse = class
```

```
  private
```

```
    FCodigo : integer;
```

```
  published
```

```
    property Codigo : integer read FCodigo write FCodigo;
```

```
end;
```

- a) TList é uma classe Herdada de TEnumerable que possui um parâmetro para um tipo genérico.
- b) A instrução “<T>” indica o nome do diagrama em que TList e TEnumerable foram criadas, utilizando do suporte a modelagem.
- c) TList é uma classe que possui um atributo de classe que foi herdado da classe TEnumerable.
- d) O código acima não é compatível.



## O que podemos afirmar sobre a declaração abaixo?

```
TList<T> = class(TEnumerable<T>)
```

```
TMinhaClasse = class
```

```
private
```

```
FCodigo : integer;
```

```
published
```

```
property Codigo : integer read FCodigo write FCodigo;
```

```
end;
```

- a) TList é uma classe herdada de TEnumerable que possui um parâmetro para um tipo genérico.
- b) A instrução “<T>” indica o nome do diagrama em que TList e TEnumerable foram criadas, utilizando do suporte a modelagem.
- c) TList é uma classe que possui um atributo de classe que foi herdado da classe TEnumerable.
- d) O código acima não é compatível.

A classe TList aqui se refere ao TList da unit Generics.Collections, o qual é classe genérica herdada de TEnumerable.



**Para definirmos um atributo para a propriedade código a sintaxe deve ser escrita da seguinte forma:**

- a) “{\$ATTRIBUTE NOME\_DO\_ATRIBUTO}” acima da linha onde foi declarado a property código
- b) No final da linha do property adicionar a diretiva “Attribute”
- c) “[NOME\_DO\_ATRIBUTO(<Parametros>)]” acima da linha onde foi declarado a property código
- d) Alterar a linha para “property código [ATRIBUTO]”: integer read FCodigo write FCodigo



**Para definirmos um atributo para a propriedade código a sintaxe deve ser escrita da seguinte forma:**

- a) “{\$ATTRIBUTE NOME\_DO\_ATRIBUTO}” acima da linha onde foi declarado a property código
- b) No final da linha do property adicionar a diretiva “Attribute”
- c) “[NOME\_DO\_ATRIBUTO(<Parametros>)]” **acima da linha onde foi declarado a property código**
- d) Alterar a linha para “property código [ATRIBUTO]”: integer read FCodigo write FCodigo

A declaração do atributo deve ser feita acima da declaração da linha property (como se fosse um atributo de um método, acima da declaração do método), com a seguinte sintaxe “[NOME\_DO\_ATRIBUTO(<Parametro>)]”. Caso você queira mais de um atributo, basta ir declarando um abaixo do outro.



**Qual destas classes é utilizada para retornar os métodos de uma classe através dos novos da RTTI?**

- a) TRTTIContext
- b) TRTTIType
- c) TRTTIMethod
- d) TRTTIClass





**Qual destas classes é utilizada para retornar os métodos de uma classe através dos novos da RTTI?**

- a) TRTTIContext
- b) TRTTIType**
- c) TRTTIMethod
- d) TRTTIClass

TRTTIType possui o método GetMethod que retorna um Array de TRTTIMethod, que é a classe que representa os métodos com os métodos da classe



**A classe TList da unit Classes existe apenas para fins de compatibilidade. Devemos sempre utilizar a TList da unit Generics.Collections.**

- a) Verdadeiro
- b) Falso



**A classe TList da unit Classes existe apenas para fins de compatibilidade. Devemos sempre utilizar a TList da unit Generics.Collections.**

- a) Verdadeiro
- b) Falso**

Falso. O TList continua existindo sem qualquer indicação de deprecated. Apesar do novo TList ser uma alternativa muito melhor que seu antecessor



**Referente ao código abaixo podemos afirmar que o tipo se trata de um:**

TIntProc = reference to procedure (n : Integer);

- a) Assinatura para um evento
- b) Assinatura para um método anônimo
- c) Tipo genérico
- d) Class Procedure



**Referente ao código abaixo podemos afirmar que o tipo se trata de um:**

TIntProc = reference to procedure (n : Integer);

- a) Assinatura para um evento
- b) Assinatura para um método anônimo**
- c) Tipo genérico
- d) Class Procedure

A declaração representa uma assinatura para um método anônimo.



## Referente ao código abaixo podemos afirmar:

```
function TMinhaClasse.RetornaValor(sl : TStringList; n : Integer) : string;  
var  
    l : integer;  
begin  
    for l := 0 to sl.Count do  
        if Pos(IntToStr(n), sl[l]) > 0 then  
            Exit(sl[l]);  
end;
```

- a) Que o método Exit recebe um parâmetro substituindo assim o uso do Result.
- b) O Desenvolvedor incluiu um método Exit para dentro da classe TMinhaClasse. Onde esse novo método necessita de um parâmetro.
- c) O método irá gerar um warning por não estar atribuindo valor ao result da função.
- d) O código acima não é compilável.



## Referente ao código abaixo podemos afirmar:

```
function TMinhaClasse.RetornaValor(sl : TStringList; n : Integer) : string;  
var  
    l : integer;  
begin  
    for l := 0 to sl.Count do  
        if Pos(IntToStr(n), sl[l]) > 0 then  
            Exit(sl[l]);  
    end;  
end;
```

- a) Que o método Exit recebe um parâmetro substituindo assim o uso do Result.
- b) O Desenvolvedor incluiu um método Exit para dentro da classe TMinhaClasse. Onde esse novo método necessita de um parâmetro.
- c) O método irá gerar um warning por não estar atribuindo valor ao result da função.
- d) O código acima não é compilável.

O Exit recebe agora uma nova funcionalidade, além de forçar o fim de um método, dentro de uma função ele pode receber um parâmetro, que será sempre do mesmo tipo do retorno da função, neste caso não haverá necessidade de utilizar o result. Resumindo ele é utilizado para aqueles casos que temos um “Result” seguido de “Exit” para que possamos utilizar apenas o Exit(<valor\_Result>).



## Que método é usado para liberar um pacote que foi carregado com a função LoadPackage?

- a) FreePackage
- b) UnLoadPackage
- c) DestroyPackage
- d) UnloadBlp





## Que método é usado para liberar um pacote que foi carregado com a função LoadPackage?

- a) FreePackage
- b) UnLoadPackage
- c) DestroyPackage
- d) UnloadBlp

Pacotes são liberados pelo método UnloadPackge, não confunda com as DLLs que são liberadas pelo FreeLibrary



**Qual propriedade do TForm contém a lista de todos os componentes em que o TForm é o Parent?**

- a) items
- b) objects
- c) controls
- d) components



**Qual propriedade do TForm contém a lista de todos os componentes em que o TForm é o Parent?**

- a) items
- b) objects
- c) controls**
- d) components

Controls armazena a lista de todos os componentes que estão “colados” diretamente no formulário. Um botão que foi colocado em um Panel que por sua vez está diretamente no formulário, não será listado na propriedade controls do Form e sim na do Panel



**Qual janela da IDE mostra as DLLs carregadas pelo aplicativo que está sendo depurado?**

- a) Process Window
- b) Call Stack
- c) Event List
- d) Watch List
- e) Modules



**Qual janela da IDE mostra as DLLs carregadas pelo aplicativo que está sendo depurado?**

- a) Process Window
- b) Call Stack
- c) Event List
- d) Watch List
- e) **Modules**

A janela modules nos permite visualizar os processos utilizados pela aplicação, isso inclui DLLs que foram carregadas por ela.



**Os três procedimentos abaixo estão incluídos na seção implementation da mesma unidade. Qual dos procedimentos a seguir não é um membro de uma classe?**

```
procedure PrintText (Text : String);  
begin  
    ShowMessage(Text);  
end;  
procedure TAnimal.Walk;  
begin  
    {code goes here}  
end;  
procedure TDog.Bark;  
begin  
    {code goes here}  
end;
```

- a) PrinText
- b) Somente Walk
- c) Somente Bark
- d) Walk e Bark
- e) Nenhum deles faz parte de uma classe



**Os três procedimentos abaixo estão incluídos na seção implementation da mesma unidade. Qual dos procedimentos a seguir não é um membro de uma classe?**

```
procedure PrintText (Text : String);  
begin  
    ShowMessage(Text);  
end;  
procedure TAnimal.Walk;  
begin  
    {code goes here}  
end;  
procedure TDog.Bark;  
begin  
    {code goes here}  
end;
```

- a) **PrintText**
- b) Somente Walk
- c) Somente Bark
- d) Walk e Bark
- e) Nenhum deles faz parte de uma classe

Observe que no método PrintText não aparece o nome da classe antes do nome do método, isso indica que este método não faz parte de uma classe.



## Qual é o comportamento do código de exemplo abaixo, quando atribuímos Zero para o valor de Denominador?

```
function Divide(Numerator, Denominator : integer) : integer;  
begin  
  try  
    Result := trunc(Numerator/Denominator);  
    if Result < 1 then  
      raise Exception.Create('Result is less than 1.');
```

```
  except on EZeroDivide do  
    ShowMessage('Divide by zero error.');
```

```
  else  
    raise;
```

```
end;
```

```
end;
```

- a) Nenhuma exceção é gerada e o controle é passado para o método que chamou a função Divide.
- b) Uma exceção é gerada com o texto 'Result is less than 1' e a exceção é propagada para a rotina de chamada.
- c) 'Divide by zero error' é exibida e a exceção é propagada para a sub-rotina chamada.
- d) 'Divide by zero error' é exibida e não é propagada exceção.
- e) Uma exceção é gerada com texto 'Result is less than 1' e esta não é propagada para a rotina de chamada





## Qual é o comportamento do código de exemplo abaixo, quando atribuímos Zero para o valor de Denominador?

```
function Divide(Numerator, Denominator : integer) : integer;  
begin  
  try  
    Result := trunc(Numerator/Denominator);  
    if Result < 1 then  
      raise Exception.Create('Result is less than 1.');
```

```
  except on EZeroDivide do  
    ShowMessage('Divide by zero error.');
```

```
  else  
    raise;
```

```
end;
```

```
end;
```

- a) Nenhuma exceção é gerada e o controle é passado para o método que chamou a função Divide.
- b) Uma exceção é gerada com o texto 'Result is less than 1' e a exceção é propagada para a rotina de chamada.
- c) 'Divide by zero error' é exibida e a exceção é propagada para a sub-rotina chamada.
- d) 'Divide by zero error' é exibida e não é propagada exceção.**
- e) Uma exceção é gerada com texto 'Result is less than 1' e esta não é propagada para a rotina de chamada

A linha "Result := trunc(Numerator/Denominator);" irá gerar uma exceção instanciando a classe EZeroDivide. Esta exceção será capturada e tratada no bloco Except, exibindo a mensagem "Divide by zero error." e não será mais propagada a exceção.



## O que a diretiva forward indica no código de exemplo abaixo?

**function** Add(X, Y : integer) : integer; forward;

- a) É utilizado para estender o escopo da função para um ponto anterior no código-fonte.
- b) É usado para especificar que a definição desta função não será na unit atual, mas na unit seguinte.
- c) Indica que o valor de retorno da função utiliza uma convenção diferente do Pascal.
- d) Indica que a função encaminha o resultado para uma variável global predefinida.
- e) Indica que essa função usa a convenção de chamada forward.



## O que a diretiva forward indica no código de exemplo abaixo?

```
function Add(X, Y : integer) : integer; forward;
```

- a) É utilizado para estender o escopo da função para um ponto anterior no código-fonte.
- b) É usado para especificar que a definição desta função não será na unit atual, mas na unit seguinte.
- c) Indica que o valor de retorno da função utiliza uma convenção diferente do Pascal.
- d) Indica que a função encaminha o resultado para uma variável global predefinida.
- e) Indica que essa função usa a convenção de chamada forward.

São raríssimos os lugares onde já vi um forward, nem lembro se já tive a necessidade de utilizá-lo. Esta diretiva é utilizada para estender o escopo da função para um ponto anterior no código-fonte. Se você programar sempre orientado a objetos nunca precisará utilizá-la, pois o método já estará declarado no escopo da classe



**Qual é a função padrão para verificar o tipo de uma variável variant?**

- a) TypeOf
- b) As
- c) GetVariantType
- d) VarType
- e) Is



## Qual é a função padrão para verificar o tipo de uma variável variant?

- a) TypeOf
- b) As
- c) GetVariantType
- d) **VarType**
- e) Is

VarType, função da unit Variants.pas. Ela nos permite identificar o tipo de valor armazenado em uma variável do tipo variant.



**Qual das seguintes declarações exibe uma mensagem ao ser inserida no final do método SetOperations, implementado no código do exemplo abaixo?**

```
type
  TDay = (wkSunday, wkMonday, wkTuesday, wkWednesday, wkThursday, wkFriday, wkSaturday);
  TDays = set of TDay;
procedure SetOperations;
var
  SomeDays, SomeOtherDays : TDays;
begin
  SomeOtherDays := [wkMonday, wkTuesday, wkWednesday, wkFriday];
  SomeDays := [wkMonday..wkFriday];
  {Statements inserted here}
end;
```

a) if (SomeOtherDays and SomeDays) then ShowMessage('Statement is True');

b) if (SomeOtherDays = SomeDays) then ShowMessage('Statement is True');

c) if (wkThursday in SomeOtherDays) then ShowMessage('Statement is True');

d) if (SomeOtherDays <= SomeDays) then ShowMessage('Statement is True');

e) if (SomeDays in SomeOtherDays) then ShowMessage('Statement is True');



## Qual das seguintes declarações exibe uma mensagem ao ser inserida no final do método SetOperations, implementado no código do exemplo abaixo?

```
type
  TDay = (wkSunday, wkMonday, wkTuesday, wkWednesday, wkThursday, wkFriday, wkSaturday);
  TDays = set of TDay;
procedure SetOperations;
var
  SomeDays, SomeOtherDays : TDays;
begin
  SomeOtherDays := [wkMonday, wkTuesday, wkWednesday, wkFriday];
  SomeDays := [wkMonday..wkFriday];
  {Statements inserted here}
end;
a) if (SomeOtherDays and SomeDays) then ShowMessage('Statement is True');
b) if (SomeOtherDays = SomeDays) then ShowMessage('Statement is True');
c) if (wkThursday in SomeOtherDays) then ShowMessage('Statement is True');
d) if (SomeOtherDays <= SomeDays) then ShowMessage('Statement is True');
e) if (SomeDays in SomeOtherDays) then ShowMessage('Statement is True');
```

Muitos devem ter acertado está, mas com a percepção errada de que SomeDays é maior que SomeOtherDays por possuir mais elementos. Na realidade, variáveis do tipo set no Delphi são, por debaixo dos panos, variáveis numéricas, onde cada valor do tipo enumerado é representado por um bit da variável numérica, sendo que o bit menos



**Que palavras chaves especifica seções que contém códigos que serão executados no início do programa e no seu término?**

- a) Initialization and Finalization
- b) Begin and End
- c) Onload and OnFree
- d) Create and Destroy
- e) Start and Finish





**Que palavras chaves especifica seções que contém códigos que serão executados no início do programa e no seu término?**

- a) Initialization and Finalization
- b) Begin and End
- c) Onload and OnFree
- d) Create and Destroy
- e) Start and Finish

Initialization e Finalization são executados na inicialização da aplicação e na sua finalização respectivamente.



**Referindo-se ao código de exemplo abaixo, que caso faz com que ocorra uma exceção no loop do EOF?**

```
aTable.First;  
while not aTable.eof do  
  begin  
    if aTable.Fields[1].IsNull then aTable.Next;  
    if ProcessRecord(aTable) then Break;  
    aTable.Next;  
  end;
```

- a) Qualquer tabela que tem apenas um campo
- b) Um Tabela com um registro que tenha um valor nulo no segundo campo
- c) Uma tabela sem referência
- d) Uma tabela com um registro que tenha um valor nulo no primeiro campo
- e) Uma tabela que não tem registros



## Referindo-se ao código de exemplo abaixo, que caso faz com que ocorra uma exceção no loop do EOF?

```
aTable.First;  
while not aTable.eof do  
begin  
  if aTable.Fields[1].IsNull then aTable.Next;  
  if ProcessRecord(aTable) then Break;  
  aTable.Next;  
end;
```

- a) Qualquer tabela que tem apenas um campo
- b) Um Tabela com um registro que tenha um valor nulo no segundo campo
- c) Uma tabela sem referência
- d) Uma tabela com um registro que tenha um valor nulo no primeiro campo
- e) Uma tabela que não tem registros

Ao tentar acessar a `aTable.Fields[1]` a tabela deve ter ao menos dois campos, `Fields[0]` e `Fields[1]`. Caso contrário uma exceção do tipo `EListError`, com a famosa frase, “List index out of bounds”, será levantada.



**Referindo-se ao código de exemplo abaixo, que caso faz com que ocorra uma exceção no loop do EOF?**

```
aTable.First;  
while not aTable.eof do  
  begin  
    if aTable.Fields[1].IsNull then aTable.Next;  
    if ProcessRecord(aTable) then Break;  
    aTable.Next;  
  end;
```

- a) Qualquer tabela que tem apenas um campo
- b) Um Tabela com um registro que tenha um valor nulo no segundo campo
- c) Uma tabela sem referência
- d) Uma tabela com um registro que tenha um valor nulo no primeiro campo
- e) Uma tabela que não tem registros



## Referindo-se ao código de exemplo abaixo, que caso faz com que ocorra uma exceção no loop do EOF?

```
aTable.First;  
while not aTable.eof do  
begin  
  if aTable.Fields[1].IsNull then aTable.Next;  
  if ProcessRecord(aTable) then Break;  
  aTable.Next;  
end;
```

- a) Qualquer tabela que tem apenas um campo
- b) Um Tabela com um registro que tenha um valor nulo no segundo campo
- c) Uma tabela sem referência
- d) Uma tabela com um registro que tenha um valor nulo no primeiro campo
- e) Uma tabela que não tem registros

Ao tentar acessar a `aTable.Fields[1]` a tabela deve ter ao menos dois campos, `Fields[0]` e `Fields[1]`. Caso contrário uma exceção do tipo `EListError`, com a famosa frase, “List index out of bounds”, será levantada.



**Parabéns Radizeiro, se você chegou até aqui estudando cada uma das questões e temas apresentados, tenho certeza que está mais do que preparado para gabaritar a prova de Certificação Delphi Developer**



**Agora se você sente que precisa de mais uma ajudinha, vou te dar um presente e tanto na próxima página**



## Pack Mestre Especialista

Um conjunto completo de treinamentos para lhe transformar em um mestre de qualidade de código no Delphi e lhe deixar mais do que preparado para a prova de **Certificação Delphi Developer**.

Esse Pack contém os meus treinamentos de:

- Certificação Especialista em Arquitetura MVC**
- Certificação Especialista em Design Patterns**
- Teste de Software**
- Técnicas Avançadas de Delphi**
- Migração do RAD ao Restful**







## Pack Mestre Especialista

Somados esses treinamentos juntos saem por um valor de **R\$ 1.185,00**, mas eu vou presentear você que deseja aprender cada vez mais e evoluir se tornando um profissional cada dia melhor.

Somente através do link desse e-book você terá um desconto de ~~R\$ 988,00~~ podendo adquirir esse treinamento por até

**12x R\$ 29,70** ou R\$ 297,00 á vista

[CLIQUE AQUI PARA APROVEITAR ESSE SUPER DESCONTO](#)





## Contatos

[contato@thuliobittencourt.com](mailto:contato@thuliobittencourt.com)

@thuliobittencourt

[www.thuliobittencourt.com](http://www.thuliobittencourt.com)

facebook.com/bittencourtthulio