

# Dia 4: Desenvolvendo com InterSystems Objects e SQL

Estou participando do desenvolvimento com a InterSystems Objects e SQL com o Joel Solon. O curso é muito legal e vou compartilhar com vocês algumas dicas que recebi durante o treinamento. Dicas apresentadas no dia 4:

1. Todos os dados são armazenados em Globais e nomes Globais começam com ^. Exemplo de global: ^animal. A Global pode ter múltiplos locais de dados("sub dado"). Exemplo: ^animal(1).
2. ^%\* globais são acessíveis de qualquer sistema (de qualquer namespace).
3. Globais permite que o IRIS ofereça suporte a dados multimodelos (objeto, relacional, documento, multidimensional, etc.).
4. Para ver Globais vá até Portal de Gerenciamento > Explorer > Globais > Selecione Globais > Exibir ou em Terminal digite ^%G ou zwrite ^global.
5. Há uma correspondência automática entre classes persistentes e Tabelas SQL:
  1. Pacote é SQL Schema;
  2. Classe é uma Tabela;
  3. Propriedade é uma Coluna;
  4. Método é um procedure de armazenamento; (quando sqlProc);
  5. Relacionamento entre classes é uma Chave Estrangeira SQL constraint (deve ser bidirecional);
  6. Objeto é uma linha.
6. Uma tabela pode corresponder a várias classes, mas a classe serial faz parte da tabela da classe persistente (não tem uma tabela específica).
7. Uma classe pode corresponder a várias tabelas.
8. Temos alguns tipos de classes:
  1. Não registrada: não é um objeto de classes (container apenas para métodos);
  2. Registrada: objetos transitórios;
  3. Persistente: persistências SQL em tabelas;
  4. Serial: persistência SQL na tabela principal (serial é embutido);
  5. Tipos de dados: não objeto de classe é usado para fazer novas validações e conversões para tipos de dados de base.
9. Classes podem ser compostas por:
  1. Propriedades;
  2. Métodos;
  3. Consulta de classes: SQL Selecione as declarações;
  4. Parâmetros: constante do usuário ou constante do sistema para configurar o comportamento da classe;
  5. Chaves estrangeiras: para integridade referencial;
  6. Índices: para melhorar o desempenho e faz valores únicos;
  7. Gatilhos: chamada de métodos associados a eventos de persistência;
  8. XData: XML ou JSON definições associadas à classe;
  9. Armazenamento: descrição do armazenamento de dados.

10. Classes por convenção tem a primeira letra da palavra em maiúscula. Exemplo: CountryOrigin. Parâmetros são todos maiúsculo. Exemplo: COLORNUMBER.
11. Atributos de classe qualificam/configuram uma classe. Exemplo: [SqlTableName = Animal] definir o nome da tabela para uma classe. [Final] não permitem herança. [Private] não permite chamar métodos ou usar propriedades para não subclasses.
12. Internamente, o IRIS gera as propriedades Get e Set, e não são visíveis mas podem ser declaradas para alterar o comportamento.
13. É possível sobrescrever um método da superclasses, para fazer isso, repita o nome da classe, os argumentos. É possível aumentar o número de argumentos, não diminuir.
14. Use ##super() para chamar o método da classe base.
15. Para criar uma classe abstrata use [Abstract] e evite a instanciação.
16. É possível estender várias classes. Exemplo Class Person extends (%Persistent, %Animal). (Persistente deve ser o primeiro nas extensões, veja a dica do joel nos comentários)
17. REST é transferência de estado representacional. É baseado no protocolo HTTP. Use verbos HTTP: GET (select), POST (insert), PUT (update) e DELETE (delete).
18. Para expor sua classe como recurso REST estendido %CSP.REST.
19. Use URLMap dentro do bloco XData para configurar as rotas do seu serviço REST. Usando o Portal, crie uma Aplicação Web, habilite REST e especifique a classe Dispatch.
20. %JSONAdaptor fornece conversão entre objetos e JSON. Use obj.%JSONImport(jsonObj) para atribuir DynamicObject a um objeto. Use obj.%JSONExportToString(.jsonString) para gravar uma String JSON em um objeto.
21. %JSON.Formatter formata uma String JSON para facilitar a leitura humana.