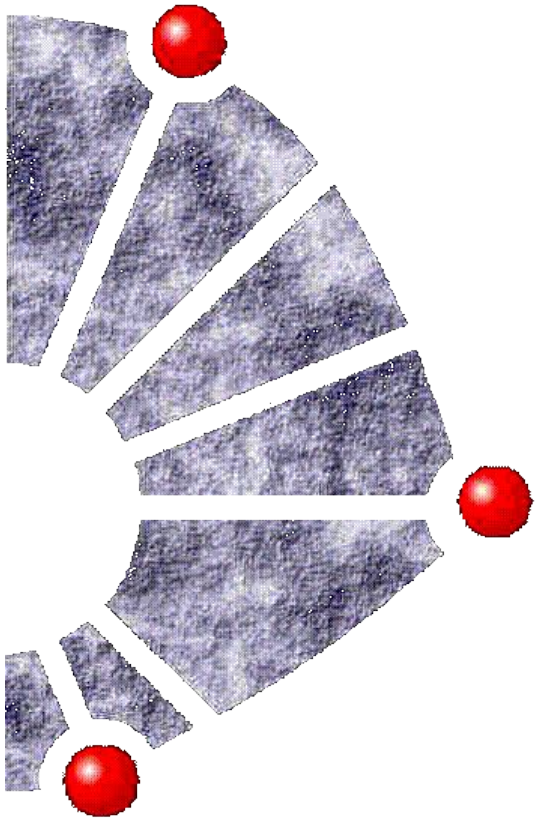
Java Avançado



AULA 5

Java Beans

- Java Beans
  - tecnologia de componentes para a plataforma Java (equivalente a um OCX ou ActiveX na plataforma Microsoft)
  - permite a criação de componentes reutilizáveis e independentes de plataforma
  - podem ser utilizados em applets e aplicações ou utilizados de maneira composicional para a criação de novos Beans
  - tecnologia incorporada a partir do JDK1.1 - qualquer browser que implemente uma máquina virtual compatível com o JDK1.1 implicitamente suporta o modelo de Beans
- Componentes
  - são unidades de software auto-contidas e reutilizáveis que podem ser visualmente compostas em unidades mais complexas, podendo ser gerenciadas e manipuladas por ferramentas de construção visual

Java Beans

- Características dos Beans
  - expoem seus recursos (e.g. métodos públicos e eventos), de modo que ferramentas visuais de construção possam manipulá-los de maneira padronizada
  - essa exposição é possível, pois os Beans seguem um conjunto de *design patterns* para essa finalidade
  - uma ferramenta que seja JavaBeans-enabled pode então se servir e utilizar de novos Beans de maneira transparente
  - novos Beans são automaticamente incluídos em palhetas ou toolboxes e manipulados como se fossem nativos da ferramenta
  - Beans podem ser manipulados utilizando-se drag-and-drop, ter sua aparência e comportamento modificados, ter definida sua interação com outros Beans e com algum applet ou aplicação, sem que haja a necessidade de se escrever uma linha de código que seja

Java Beans

- Introspeção
  - processo por meio do qual ferramentas de construção podem descobrir as características de um Bean (i.e. suas propriedades, métodos e eventos)
  - pode ser implementada de duas formas
    - utilizando-se um conjunto de design patterns na nomeação das características do Bean
    - provendo explicitamente as propriedades, métodos e eventos em uma classe de informações de Beans associada ao Bean
- No primeiro caso
  - classe java.beans.Introspector examina o Bean procurando por nomes-padrão, de forma a descobrir as características do Bean
- No segundo caso
  - classe de informações de Beans implementa a interface BeanInfo, fornecendo as informações necessárias

Java Beans

- Propriedades (Properties)
  - características da aparência e de comportamento dos Beans, que podem ser mudadas em tempo de design
  - ferramentas de desenvolvimento efetuam a introspecção sobre o Bean para descobrir suas propriedades e expô-las consequentemente para a manipulação do usuário
  - customização dos Beans pode ser feita de 2 modos:
    - utilizando-se editores de propriedades
    - utilizando-se *Beans customizers* mais sofisticados
- Eventos
  - são utilizados pelos Beans para se comunicar com outros Beans
  - um Bean que deseja receber eventos de um determinado tipo se registra como interessado com o Bean que dispara o evento em questão
  - ferramentas de desenvolvimento podem, por introspecção, descobrir quais eventos um determinado Bean pode gerar e quais está apto a manipular

Java Beans

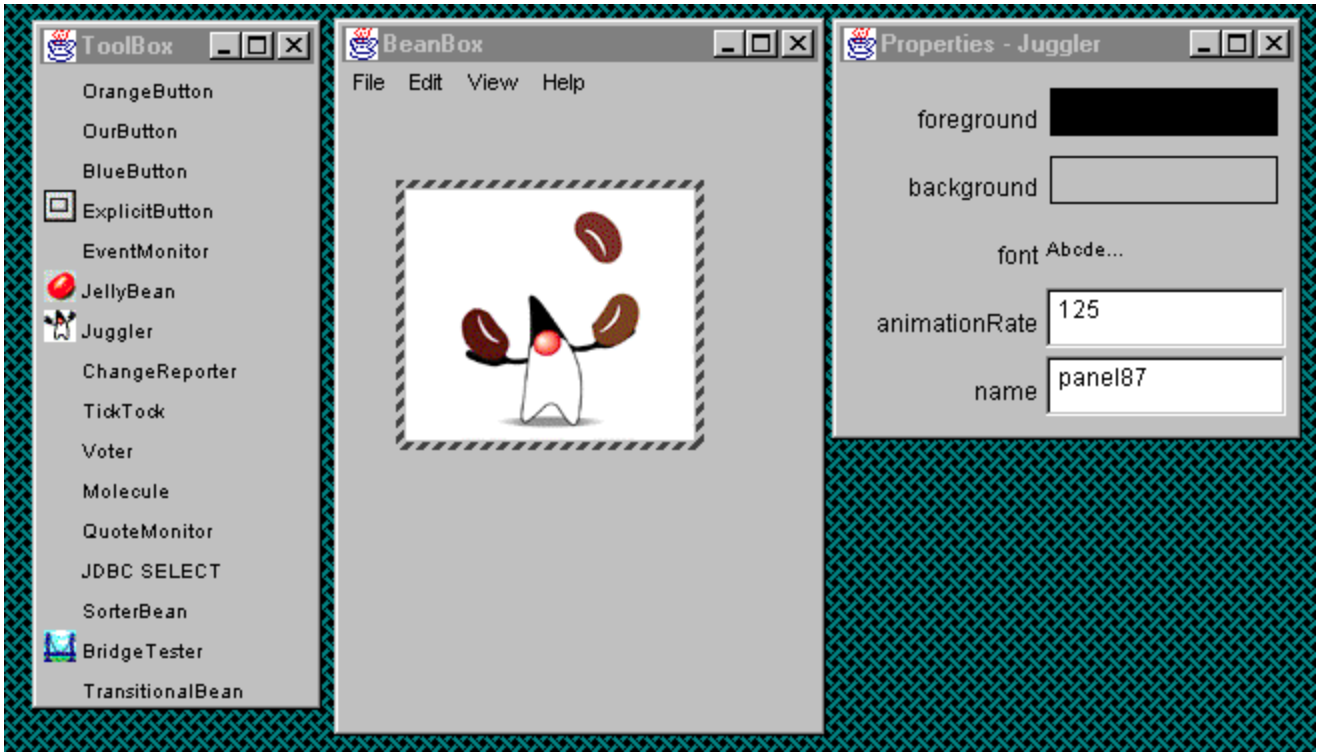
- Persistência
  - habilita Beans a salvar e restaurar seu estado
  - JavaBeans utiliza o mecanismo de serialização de objetos do Java para suportar persistência
- Métodos
  - métodos dos Beans são os próprios métodos do Java, podendo ser invocados de outros Beans ou de qualquer objeto Java que tenha referência para o Bean
  - de maneira default, todos os métodos públicos são exportados
- Princípio Básico
  - JavaBeans foram projetados de modo a serem manipulados transparentemente por ferramentas de desenvolvimento
  - apesar disso, toda a API, incluindo o suporte a eventos, propriedades e persistência pode ser facilmente compreensível também por programadores humanos

Java Beans

- BDK - Beans Development Kit
  - framework de programação, distinto do JDK, que disponibiliza ferramentas para o desenvolvimento de JavaBeans
  - download gratuito a partir do site da Sun:
    - http://java.sun.com/products/javabeans/software/index.html
- BeanBox
  - é uma ferramenta simples que vem com o BDK e que permite testarmos novos Beans, ao mesmo tempo que permite aprendermos como manipular visualmente as propriedades e eventos de um Bean
  - não é uma ferramenta de desenvolvimento de Beans, entretanto pode ser utilizado para um aprendizado maior sobre os Beans
  - pode ser utilizado para a geração automática de um applet que inclui o Bean sendo manipulado

Java Beans

- Apresentação Visual do BeanBox
  - quanto o BeanBox inicia, ele automaticamente carrega no ToolBox todos os Beans que encontra no diretório beans/jars



Gerando Um Bean Simples

- Edição de Código

```
import java.awt.*;
import java.io.Serializable;

public class SimpleBean extends Canvas implements Serializable{

//Constructor sets inherited properties
public SimpleBean(){
        setSize(60,40);
        setBackground(Color.red);
       }
```

- }Empacotando em um arquivo JAR
  - usando o seguinte arquivo *manifest*

```
Name: SimpleBean.class
Java-Bean: True
```

Implementando Propriedades

- Implementando Propriedades
  - valores do tipo private que podem ser acessadas por meio de métodos do tipo getXXX e setXXX, onde XXX é um atributo que corresponde à propriedade em questão
- Tipos de Propriedades
  - **Simple Properties:** propriedades simples
  - **Bound Properties:** propriedades que, quando têm seus valores modificados geram eventos de notificação para outros objetos
  - **Constrained Properties:** propriedades para as quais uma mudança deve sofrer uma consulta, podendo ser vetada por outros objetos
  - **Indexed Properties:** propriedades com múltiplos valores

Implementando Propriedades

- Editores de Propriedades
  - ao descobrir uma propriedade, uma ferramenta de desenvolvimento automaticamente associa a esta propriedade um editor de propriedades adequado para a manipulação da propriedade envolvida
  - esse editor dependerá do tipo de objeto relacionado à propriedade
- Como Editores de Propriedades são associados a tipos de Propriedades não-padrão
  - Associação explícita por meio de um objeto BeanInfo, com o método setPropertyEditorClass
  - Registro explícito por meio do método estático java.Beans.PropertyEditorManager.registerEditor
  - Busca de nomes: se uma classe não está associada explicitamente a um editor de propriedades, o objeto PropertyEditorManager busca por uma classe XXXEditor

Implementando Propriedades

- Customizers
  - implementam a interface java.beans.Customizer de modo a criar um editor de propriedades customizado para editar todas as propriedades de um determinado Bean
- Editores de Propriedades
  - implementando a interface PropertyEditor, pode-se criar um editor de propriedades para uma propriedade específica de um Bean
- Cada classe Bean pode ter uma classe BeanInfo associada
  - customiza como o Bean deve aparecer na ferramenta de desenvolvimento
  - pode definir propriedades, métodos, eventos, textos a serem incluídos nos Beans e um pequeno help

Modelo de Eventos para Java Beans

- Eventos
  - devem estender a classe EventObject
    - e.g. class XXXEvent extends EventObject
- Listener para o Evento
  - deve estender a interface EventListener
    - e.g. interface XXXListener extends EventListener
- Classes suportanto um determinado tipo de evento
  - devem implementar um método do tipo addXXXListener, que cadastre o objeto no objeto-fonte de tais eventos
  - da mesma maneira deve implementar um método removeXXXListener
  - é por meio dos nomes destes métodos que uma ferramenta de desenvolvimento descobre quais os eventos suportados por um Bean usando introspecção

Persistência em Beans

- Bean
  - pode tornar-se persistente, tendo suas propriedades, atributos e informações de estado salvas e restauradas em um meio persistente (arquivo)
  - mecanismo que torna a persistência possível é a serialização de objetos. Quando um bean é serializado, ele é convertido em um stream de dados e transferido para um arquivo
  - qualquer applet ou aplicação que precise do bean pode então reconstituí-lo por meio de uma deserialização
  - JavaBeans usa a API de serialização de objetos do JDK
- Para tornar um Bean persistente
  - basta definir a classe como implementando java.io.Serializable, que seus atributos serão salvos
  - atributos que não necessitam ser salvos podem ser marcados com os keywords `transient` ou `static`

Arquivos JAR

- JavaBeans são distribuídos em arquivos JAR
- Arquivo JAR
  - é um arquivo do tipo ZIP que opcionalmente pode conter um arquivo de manifesto
  - pode conter arquivos do tipo .class, beans serializados (.ser), arquivos de help em formato HTML e recursos (imagens, áudio, texto, etc)
- Arquivo de Manifesto
  - descreve o conteúdo do arquivo JAR
  - se o arquivo JAR não tem um arquivo de manifesto, então todas as classes e objetos serializados são tratados como beans
  - o arquivo de manifesto permite especificarmos quais classes são beans por meio da entrada “Java-Bean: True”

Arquivos JAR

- Vantagens dos Arquivos JAR
  - **segurança:** arquivos JAR podem ser assinados digitalmente, de tal forma que um applet pode ter sua origem confirmada antes de ser executado
  - **menor tempo de download:** uma única transação HTTP é utilizada para trazer todo as classes relacionadas a um applet
  - **compressão:** o arquivo JAR é um arquivo ZIP
  - **empacotamento para extensões:** frameworks de extensão à API Java podem ser distribuídos em arquivos JAR
  - **selamento de pacotes:** pacotes armazenados em arquivos JAR podem ser opcionalmente selados (todas as classes definidas em um pacote devem estar no mesmo arquivo JAR), de modo a garantir versões consistentes de classes e objetos
  - **controle de versão de pacotes:** arquivo JAR pode conter dados sobre os arquivos que contém
  - **portabilidade:** mecanismo para o manuseio de arquivos JAR é padrão na plataforma Java

Arquivos JAR

- O Programa JAR
  - A implementação padrão do JDK provê um programa aplicativo chamado **jar** (Java ARchiver) para manipular arquivos JAR
  - sintaxe do programa **jar** é equivalente à do programa **tar** em Unix
- Exemplos de Uso
  - Para criar um arquivo JAR
    - jar cf jar-file input-file(s)
  - Para ver o conteúdo de arquivos JAR
    - jar tf jar-file
  - Para extrair o conteúdo de um arquivo JAR
    - jar xf jar-file
  - Para extrair arquivos específicos do arquivo JAR
    - jar xf jar-file archived-file(s)

Outras Aplicações de Arquivos JAR

- Para rodar aplicações empacotadas em um arquivo JAR
  - (JDK 1.1) jre -cp app.jar MainClass
  - (JDK 1.2 - requer Main-Class manifest header) java -jar app.jar
- Para invocar um applet empacotado em um arquivo JAR
  - ```
    <applet code=AppletClassName.class
    archive="JarFileName.jar"
    width=width height=height>
    </applet>
    ```
- Para assinar um arquivo JAR
  - jarsigner jar-file alias
    - alias identifica a chave privada que será utilizada para a assinatura do arquivo JAR, junto com o certificado associado à chave em questão
    ```

Arquivos JAR Assinados

- Assinando e Verificando Arquivos JAR
  - a opção de assinar um arquivo JAR com uma assinatura eletrônica permite maior segurança no uso de Applets em redes Intranet e Internet
- Assinatura
  - números especiais chamados de chaves públicas e privadas, que vêm sempre aos pares, exercendo papéis complementares
- Chave Privada
  - é utilizada para a assinatura eletrônica de um arquivo JAR
  - é secreta, permanecendo somente sob a posse de quem assina o arquivo
- Chave Pública
  - permite a verificação da autenticidade de uma assinatura

Arquivos JAR Assinados

- Somente o uso de Chaves Públicas e Privadas
  - não garante totalmente a segurança da assinatura
  - é necessário confirmar que a chave pública realmente pertence à pretensa pessoa que assinou o arquivo
- Elemento Adicional
  - certificado incluído em um arquivo JAR assinado, que é assinado digitalmente por alguma autoridade reconhecida de certificação, que indica quem a quem pertence a chave pública em questão
- Autoridades de Certificação
  - são entidades (normalmente firmas especializadas) que têm confiabilidade no mercado para assinar e emitir certificados de chaves e seus proprietários

Arquivos JAR Assinados

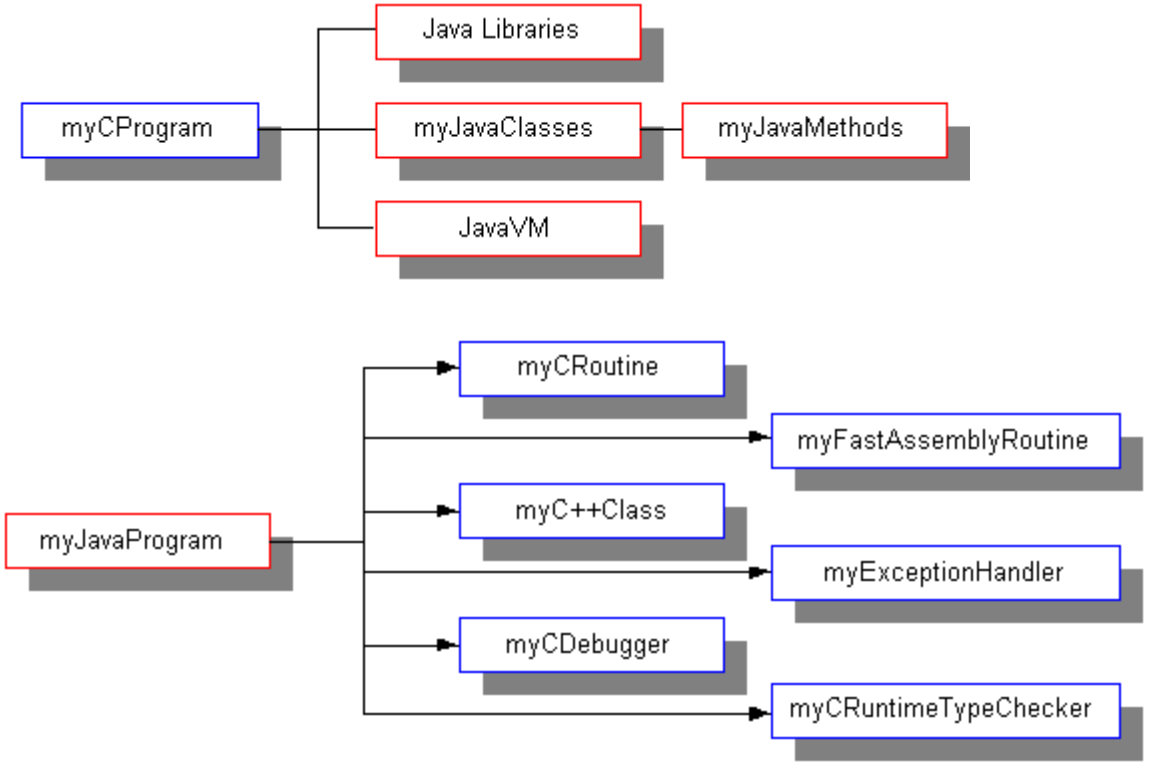
- Certificados
  - podem ser armazenados em servidores externos pagos, tais como VeriSign, Thawte, Entrust
  - podem ser armazenados em servidores internos específicos, utilizando produtos tais como Netscape/Microsoft Certificate Servers ou Entrust CA
  - podem ser auto-assinados
- Ferramenta para a Assinatura e Verificação de Arquivos
  - jarsigner
- Ferramenta para o Gerenciamento de Chaves e Certificados
  - keytool

JNI - Java Native Interface

- JNI - Java Native Interface
  - interface nativa do Java para o acesso a porções do código que não tenham sido escritas em Java
  - JNI permite que código Java inclua código escrito em outras linguagens tais como o C, C++ e Assembly
  - JNI permite que se inclua código Java e uma máquina virtual Java em aplicações desenvolvidas em outras linguagens
- Situações de Uso do JNI
  - Biblioteca de classes Java não suporta algum feature dependente de plataforma em alguma aplicação
  - Existe uma biblioteca ou aplicação já desenvolvida em outra linguagem, que se deseja aproveitar
  - Algumas partes do programa que sejam críticas em relação ao tempo e que portanto têm que ser desenvolvidas em outras linguagens

JNI - Java Native Interface

- Casos de Uso do JNI



Java IDEs

- Programação em Java
  - pode ser facilitada com o uso de ambientes de desenvolvimento integrados (IDE - Integrated Development Environment)
- Principais Vantagens
  - organização e facilidade na edição de uma árvore de classes
  - desenvolvimento visual da interface gráfica, com o uso de JavaBeans
    - criação das interfaces
    - manipulação de propriedades da interface
    - teste da interface
  - debugger integrado
  - manipulação automática de arquivos JAR
  - geração automática de documentação com o uso da ferramenta javadoc

Java IDEs

- Principais Ferramentas
  - JBuilder
    - Ferramenta de desenvolvimento da Borland/Imprise para Java
  - Forté for Java
    - Sucessor dos antigos Java Workshop da Sun Microsystems e NetBeans Developer
  - VisualAge for Java
    - Ferramenta de desenvolvimento Java da IBM
  - Visual Café
    - Ferramenta da Symantec para desenvolvimento em Java
  - Visual J++
    - Ferramenta da Microsoft, parte da plataforma Visual Studio .net