

Resumo por Kaléu Caminha

Diretrizes Para o Ensino de Orientação a Objetos com Java

Michael Kölling | John Rosenberg

As ferramentas e técnicas desenvolvidas para o ensino de Programação Orientada a Objetos (OO) não são tão maduras quanto as ferramentas de ensino de programação estruturada, o resultado disso, é uma maior dificuldade em ensinar OO para alunos novatos em programação.

Uma ferramenta que oferece suporte ao ensino de OO é o BlueJ, que corresponde a um ambiente integrado de desenvolvimento java, criado especialmente para o ensino, com características inexistentes em outros ambientes de desenvolvimento populares.

Estas características foram desenvolvidas para oferecer suporte, dentre outras, às oito diretrizes apresentadas neste artigo.

O Ambiente Bluej

O BlueJ é focado em interatividade, visualização e simplicidade.

É possível visualizar todas as classes criadas na forma de um diagrama semelhante ao diagrama de classes da UML (Imagem 1), criar objetos a partir destas classes (Foto 2) e visualizá-los em um “painel de objetos”.

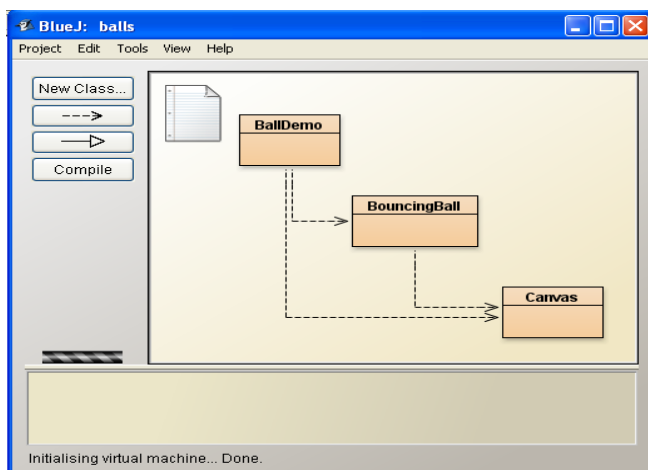


Imagem 1 – Classes de um programa representas em um diagrama semelhante ao diagrama de classes da UML.

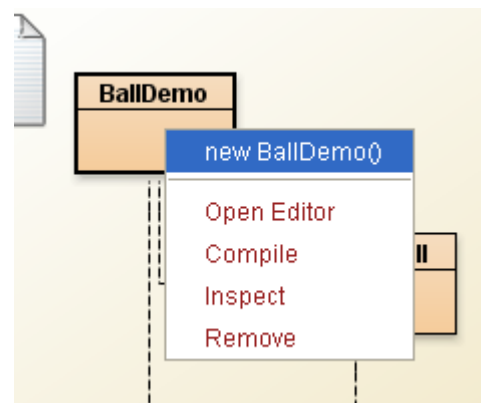


Imagem 2 – Criação de Objetos a partir de uma Classe.

Com os objetos dispostos em um painel, é possível interagir com eles visualizando seu estado e realizando chamadas de métodos diretamente (Imagem 3), apenas com cliques do mouse.

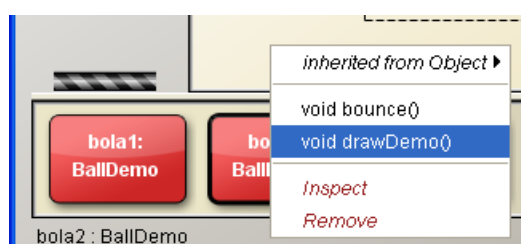


Imagem 3 – Chamada de métodos de um objeto da bancada de objetos.

O programa também não precisa de um método main, pois qualquer método de qualquer classe pode ser executado em qualquer instante.

O BlueJ também fornece interface gráfica para digitação dos parâmetros (Imagem 5) de métodos e a informação dos nomes das variáveis que armazenarão um objeto (Imagem 4).

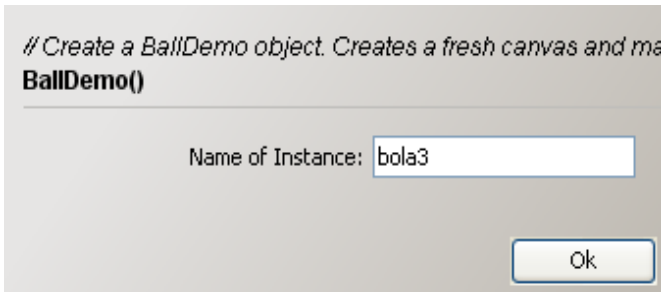


Imagem 4 – Informando o nome da variável que irá armazenar um objeto.

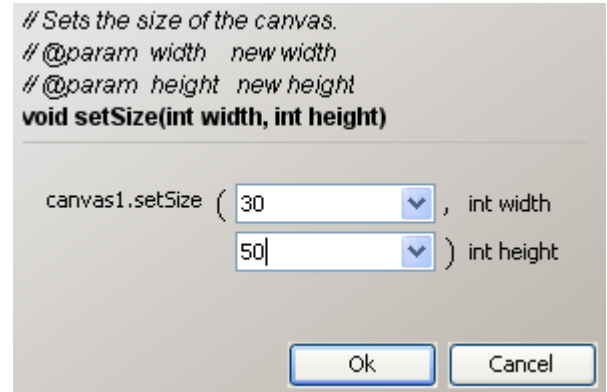


Imagem 5 – Informando os parâmetros largura e altura para o método setSize() de um canvas..

Por ser voltado para simplicidade, ele não possui muitas das ferramentas comuns em ambientes profissionais, seu uso é especialmente voltado para o ensino em cursos introdutórios de modo que o estudante possa colocar seu esforço no aprendizado dos principais conceitos de Programação Orientada a Objetos.

Questões sobre o Ensino com BlueJ

Qualquer exemplo de programação Java pode ser utilizado e trabalhado dentro do BlueJ, porém, é importante que cursos que o utilizem sejam especialmente desenvolvimentos para fazer uso dos recursos específicos que o ambiente fornece.

As diretrizes aqui apresentadas são independentes do BlueJ, porém, este é preparado para fornecer suporte nativo à todas elas.

1ª Diretriz: Objetos primeiro

É consenso entre muitos professores de Programação Orientada a Objetos que é melhor ensinar a criar e manipular objetos antes de qualquer outra coisa. Esta atividade, porém, é difícil nos ambientes tradicionais, pois para executar o método de um objeto é preciso:

1. Uma classe contendo um método main estático com um array de Strings como parâmetro;
2. Uso de uma variável tipada;
3. Uma atribuição de um objeto à uma variável;
4. A instanciação de um objeto;
5. A sintaxe de chamada de um método;
6. A visualização do resultado da chamada do método.

Com BlueJ a primeira atividade é abrir um projeto existente e criar objetos a partir das classes, em seguida, os estudantes podem chamar métodos destes objetos e visualizar seu estado. Tudo isso sem escrever nenhuma linha de código.

2ª Diretriz: Não inicie com uma tela em branco.

Um dos maiores erros em programação OO é iniciar com uma tela vazia.

Escrever projetos OO envolvem a definição de quais classes o programa terá e quais métodos cada classe deverá ter, e este, é um exercício extremamente difícil para estudantes novatos.

Ao invés disso, pode-se iniciar com pequenas modificações e adições em código existente.

3ª Diretriz: Leia código

Muitos estudantes são forçados a começar a programar sem terem lido praticamente nenhum código. Isso não acontece em nenhuma outra disciplina.

Estudantes aprendem mais quando tem acesso a códigos bem escritos que eles possam utilizar como base para os seus projetos.

4ª Diretriz: Use projetos grandes

Programas curtos dificilmente deixam claro as vantagens em se utilizar Orientação a Objetos.

A sugestão é que sejam utilizados programas bem escritos com um número maior de classes e métodos que permitam aos estudantes compreenderem as vantagens do uso de OO, assimilarem a importância de um código limpo e comentado, pratiquem a leitura de código e estudem a interação entre as classes.

5ª Diretriz: Não inicie com o método main

O método main exige muitos conceitos para ser compreendido. O ideal é que os estudantes só sejam apresentados a ele após terem se envolvido com os conceitos principais como a criação de objetos e chamada de métodos.

6ª Diretriz: Não use “Hello world”

Para se fazer um “Hello World” em Java é preciso iniciar do método main, que por sua vez significa: 1) construir uma classe da qual nenhum objeto é criado. 2) construir um método (o main) que não age sobre um objeto.

Isso não é Programação OO e fere outras diretrizes já apresentadas.

7ª Diretriz: Mostre a estrutura do programa

Classes e seus relacionamentos são muito importantes. É essencial que os estudantes possam visualizar a estrutura dos programas nos quais trabalham.

O BlueJ exhibe a estrutura do programa em um diagrama semelhante ao diagrama de classes da UML. Quando o BlueJ não estiver sendo utilizado é importante que o professor encontre meios de mostrar claramente esta estrutura.

8ª Diretriz: Tenha cuidado com a interface do usuário

É importante ensinar uma forma de realizar a interface com o usuário para fornecimento de parâmetros e para a impressão de resultados do software.

Em Java, podemos fazer isso com Applets, com a API de entrada e saída ou com frameworks GUI como o Swing.

Porém, nenhuma destas alternativas é simples o suficiente para ser utilizada em cursos introdutórios. São sugeridos então, dois caminhos para lidar com este problema:

1. Específico no BlueJ, são as janelas pop-up que solicitam ao usuário os valores para os parâmetros dos métodos e as janelas que exibem os retornos dos métodos;
2. Um projeto maior com classes que encapsulem a interface com o usuário e que os estudantes apenas precisem aprender a utilizar seus métodos.

Uma Sequência de Atividades para um curso de OO

Esta seção apresenta uma sequencia lógica de lições para o ensino de OO utilizando as diretrizes apresentas:

1. Estudantes iniciam criando objetos e executando métodos de classes existentes;
2. Estudantes podem fazer pequenas modificações em projetos existentes como a alteração de Strings;
3. Estudantes são desafiados a implementarem mudanças no corpo dos métodos em classes existentes;
4. Estudantes criam novos métodos para classes existentes;
5. Estudantes criam novas classes para um projeto existente;
6. Estudantes desenvolvem um projeto do zero.

Conclusão

O uso das diretrizes e da sequencia apresentada podem beneficiar qualquer curso de Programação Orientada a Objetos, oferecendo ainda mais benefícios se utilizado em conjunto com o BlueJ.

Michael Kölling and John Rosenberg. 2001. **Guidelines for teaching object orientation with Java**. In *Proceedings of the 6th annual conference on Innovation and technology in computer science education (ITiCSE '01)*. ACM, New York, NY, USA, 33-36.
DOI=10.1145/377435.377461 <http://doi.acm.org/10.1145/377435.377461>