

## Material de Apoio Aula 3

### BlueJ – Um ambiente Java interativo

O ambiente BlueJ foi desenvolvido como parte de um trabalho de pesquisa universitário cujo enfoque foi dado ao ensino da orientação a objetos a programadores iniciantes. O sistema continua sendo desenvolvido, e mantido, por um grupo reunindo pesquisadores das universidades de Daking (Melbourne, Austrália) e de Kent (Canterbury, UK). O projeto é financiado pela Sun Microsystems. O objetivo do BlueJ é prover um ambiente didático de fácil utilização para programação em Java para estudantes de primeiro ano de graduação. Este ambiente encoraja a experimentação dos processos de desenvolvimento orientado a objetos através do uso de técnicas de visualização e interação com o ambiente.

### Informações adicionais

Homepage: <http://www.bluej.org>

Download: <http://www.bluej.org/download/download.html> (para Windows há um executável para instalação. Para os demais ambientes, há um arquivo .jar que pode ser executado através de uma máquina virtual.)

### Primeiros passos com BlueJ

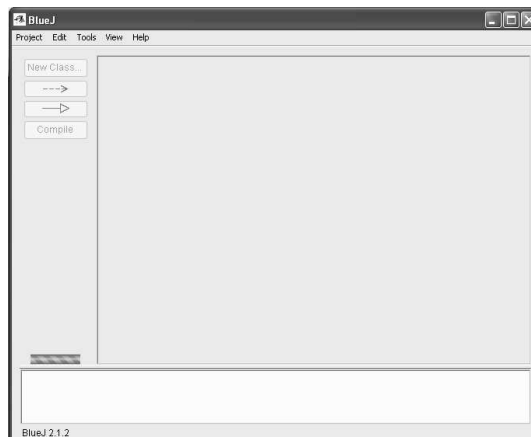
O BlueJ será utilizado na disciplina como plataforma de desenvolvimento de aplicações orientadas a objetos em Java. Seu fim didático nos permitirá acompanhar a evolução de um programa considerando os seus objetos e a interação entre eles. Repita a seqüência de passos apresentada neste material de aula para se familiarizar com o BlueJ.

### Exercício preliminar

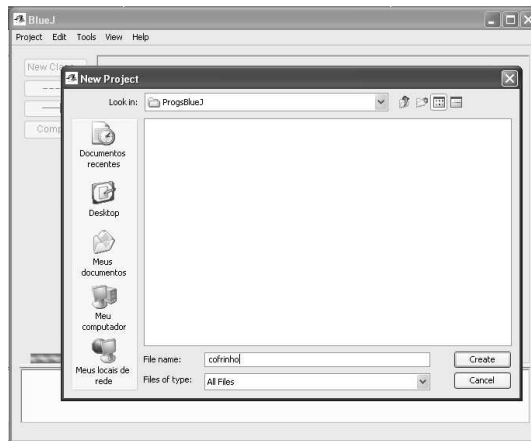
- Construa uma classe, em Java, para representar objetos Cofrinhos.
  - Identifique o(s) atributo(s).
  - Identifique a interface de serviços.
  - Identifique os métodos.

### Criando um novo projeto em BlueJ

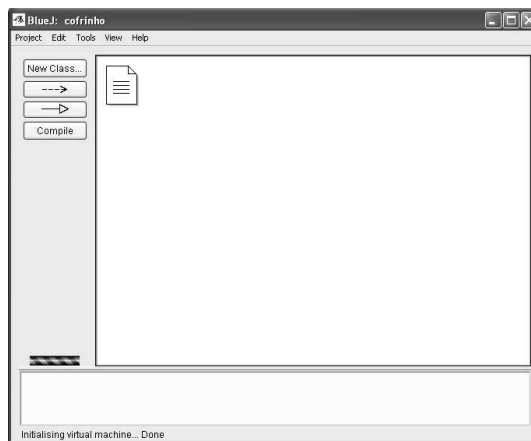
Lance a execução do BlueJ. A janela que deverá aparecer é a seguinte:



Em seguida, crie um novo *projeto*, entrando no menu Project, selecionando um diretório para trabalho (sugiro a criação de um diretório para programas feitos com o BlueJ, no exemplo, eu criei o diretório ProgsBlueJ). No meu exemplo, o nome dado ao projeto é cofrinho.



Como resultado da criação de um projeto, aparecerá a seguinte iconografia no BlueJ.

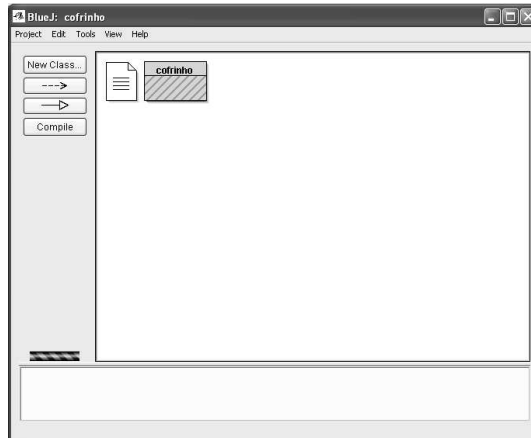


### Criando uma nova classe

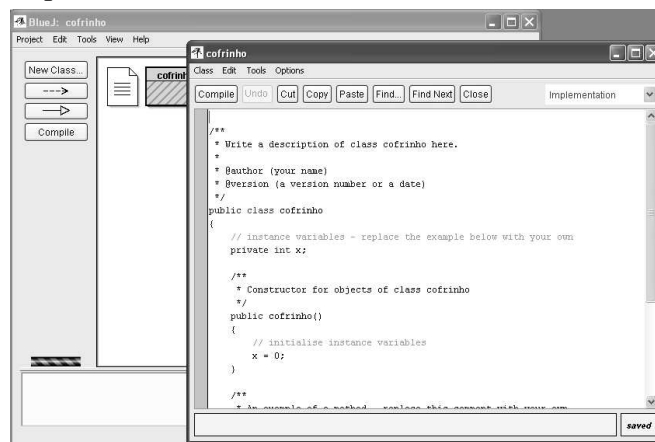
Em um projeto é possível implementar classes. Com o botão `New Class...` selecione a opção de criação de uma classe (Class). Algumas das demais opções serão vistas com o decorrer da disciplina. No exemplo, o nome dado à classe é `cofrinho`.



O resultado da criação de uma classe é visualizado graficamente conforme a figura a seguir. Observe que o ícone da nova classe encontra-se com “ranhuras”, isto significa que a classe não foi editada.



Edite a classe, inserindo o código desejado. Observe que um esqueleto inicial do código já é apresentado, incluindo alguns comentários. Note, no entanto, que este código não é, nem de perto, o código desejado para um cofrinho...

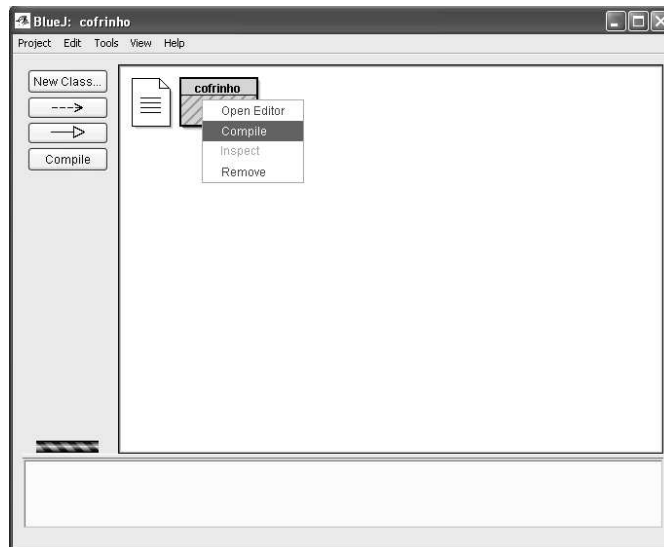


Digite, portanto, seu código (aquele, desenvolvido no exercício preliminar):

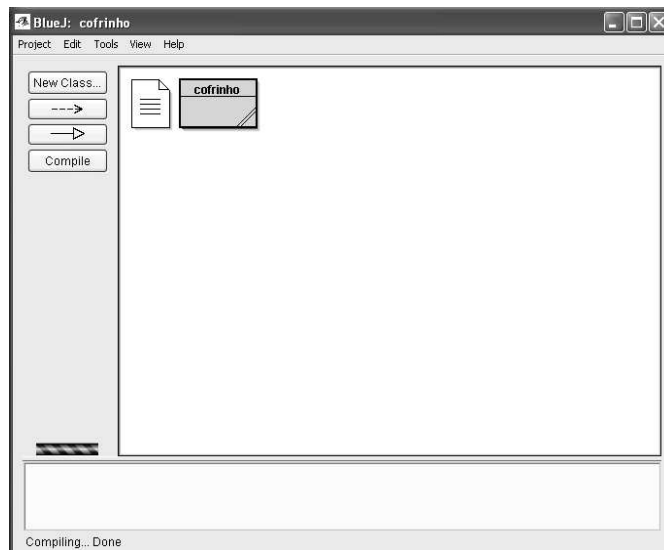


### Compilando uma classe

Uma vez tendo sido editada a classe, ela necessita ser compilada. Com o botão da direita do mouse, entre na opção **Compile**:

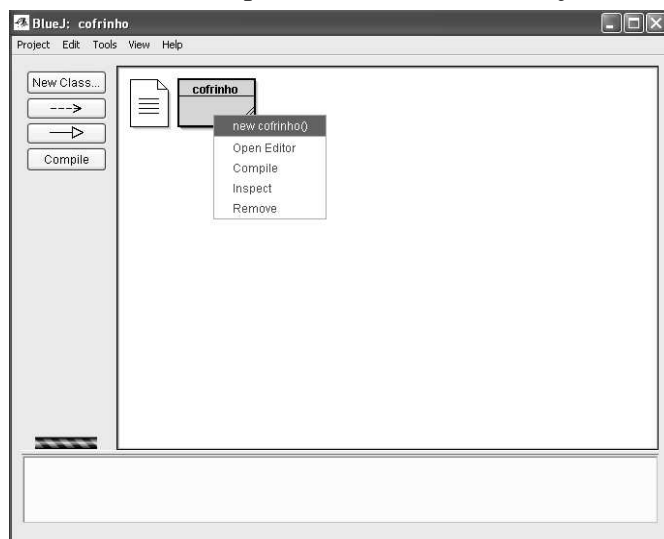


Como resultado da compilação, o bytecode encontra-se disponível:

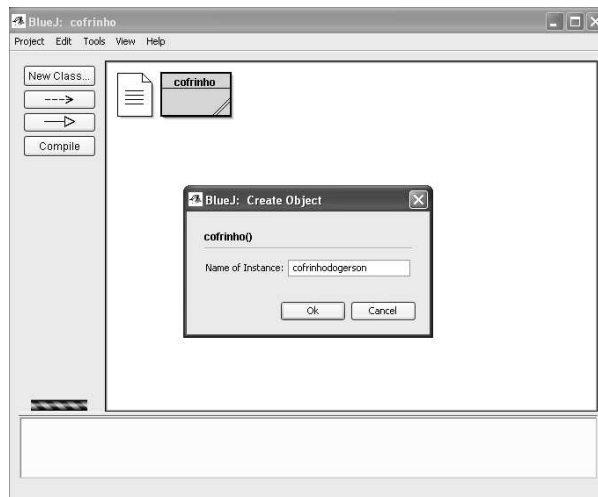


### **Criando um novo objeto**

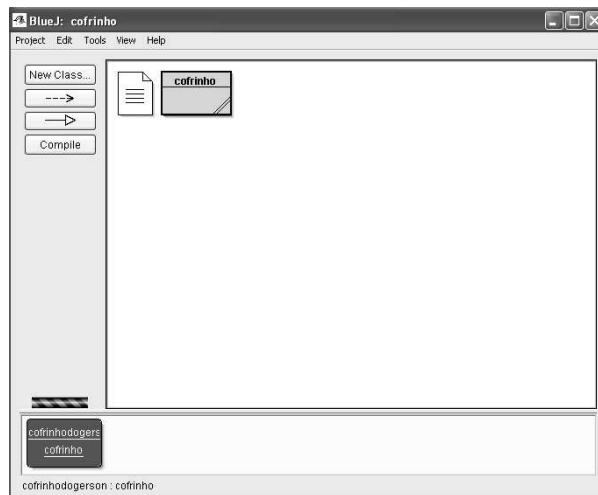
A partir deste momento, objetos da classe `cofrinho` podem ser criados. Com o botão direito do mouse, a opção `new cofrinho()` permite criar um novo objeto:



Antes de realizar a criação propriamente dita, será solicitado um identificador para a referência ao objeto a ser criado. No exemplo, o identificador informado é `cofrinhodogerson`:

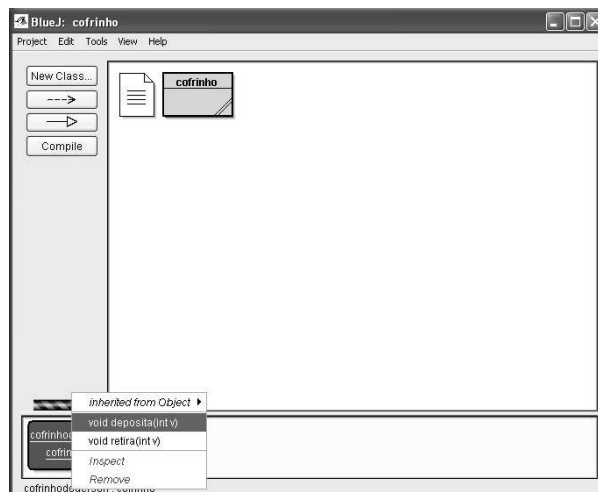


Então, o objeto referenciado por `cofrinhodogerson` pode ser visualizado na janela de visualização de objetos:

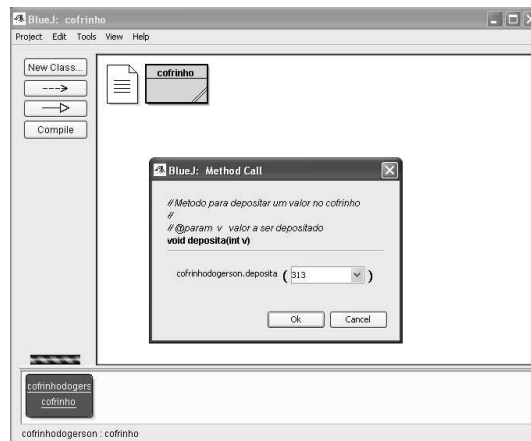


### Enviando mensagens a objetos

Com o botão da direita clicado sobre o objeto referenciado por `cofrinhodogerson` é possível verificar quais são os serviços disponíveis na interface de serviços do objeto. Neste momento, o usuário pode solicitar a ativação de um serviço. No exemplo, será depositado um valor.

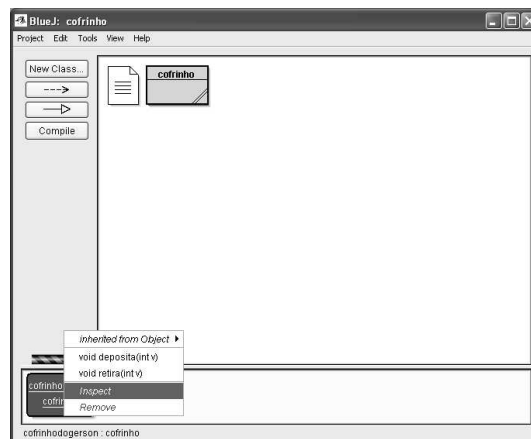


A janela aberta solicita os parâmetros de entrada para o método e informa os comentários que foram inseridos no código. No exemplo esta sendo depositado 313 *dinheiros*:

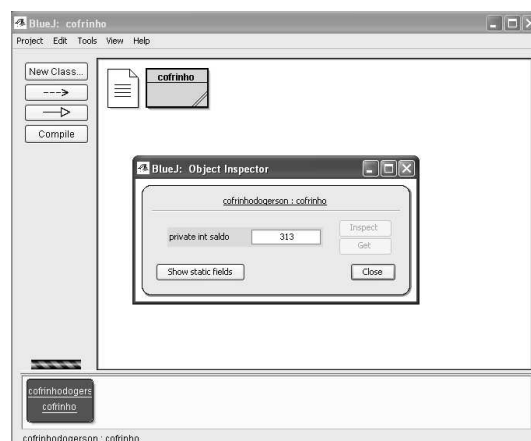


### Verificando o estado interno de um objeto

O estado interno de um objeto pode ser visualizado através da opção `Inspect`.



Este estado interno reflete o conjunto de alterações sofridas pelos atributos do objeto a cada invocação de método. No caso de nosso exemplo, o atributo interno `saldo` indica que o `cofrinhodogerson` tem 313 *dinheiros*.



### Exercícios

1. Deposite e retire valores do objeto `cofrinhodogerson`.
2. Crie novos objetos: `cofrinhodofulano`, `cofrinhobeltrano`, ...
3. Depositando valores em um destes cofrinhos, o outro altera seu estado interno?
4. O ambiente BlueJ pode ser considerado um objeto interagindo com os objetos cofrinhos? Justifique.
5. Force a ocorrência de um erro de sintaxe, o qual o compilador é capaz de detectar. Observe a mensagem de erro gerada. Force ao erro da seguinte forma: ao invés de colocar `int`, coloque `itn` em `private int saldo`.
6. Implemente uma classe `ContaEmBanco`.