

1. No blueJ, crie um novo projeto chamado **Modulo7**;
2. Crie a classe **Dado** com o seguinte código (copie exatamente como está abaixo) e a compile:

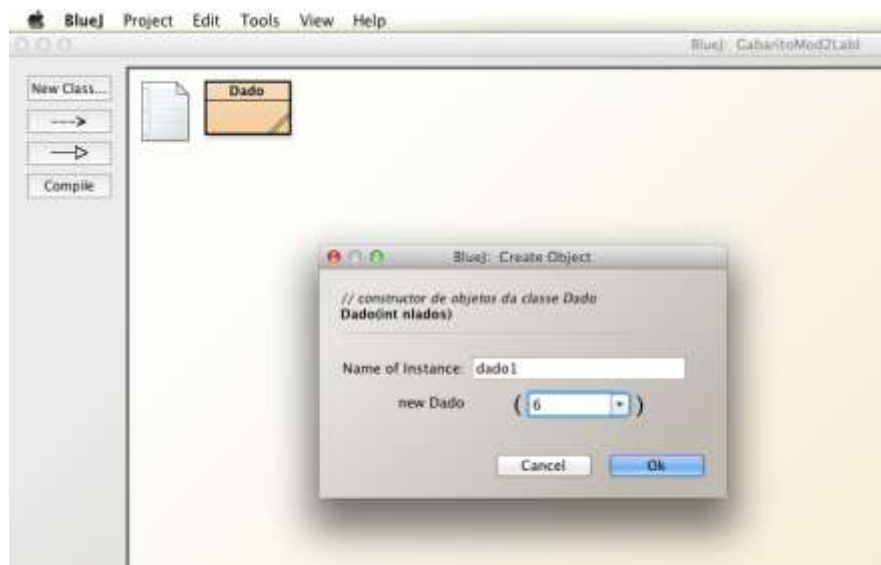
```
public class Dado
{
    // atributos
    private int lados;

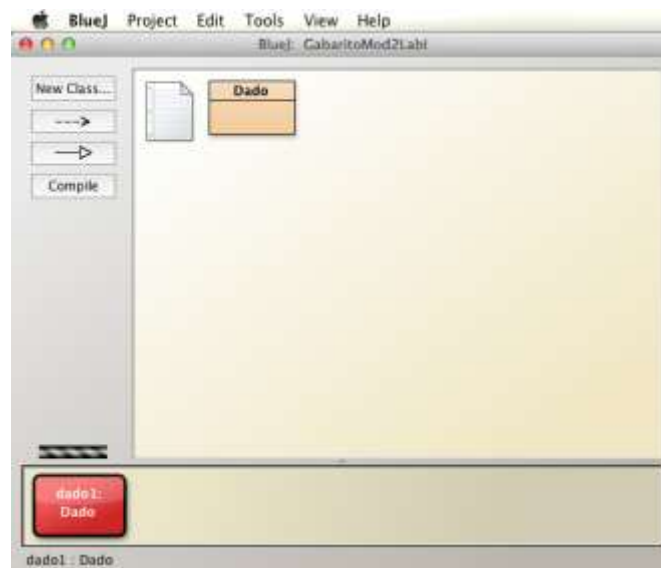
    /**
     * constructor de objetos da classe Dado
     */
    public Dado (int nlados)
    {
        lados = nlados;
    }

    // método que simula o lançamento de um dado

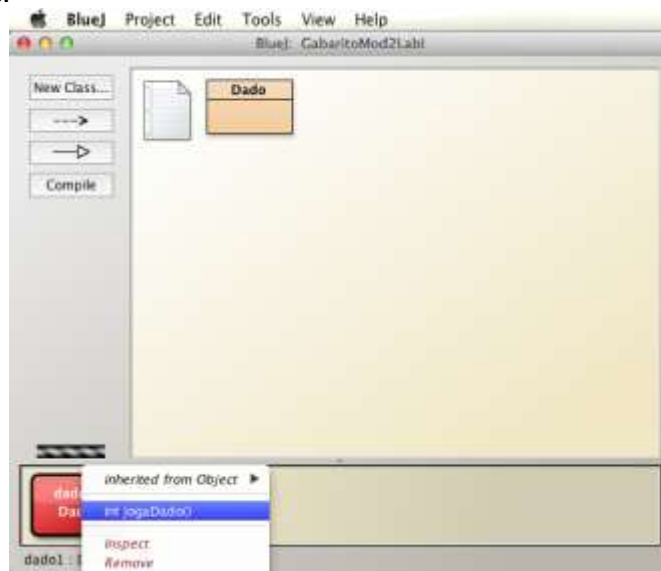
    public int jogaDado()
    {
        return (int) (Math.random ( ) * lados + 1);
    }
}
```

3. No BlueJ:
 - Crie um objeto da classe Dado (com 6 lados), clicando no construtor:

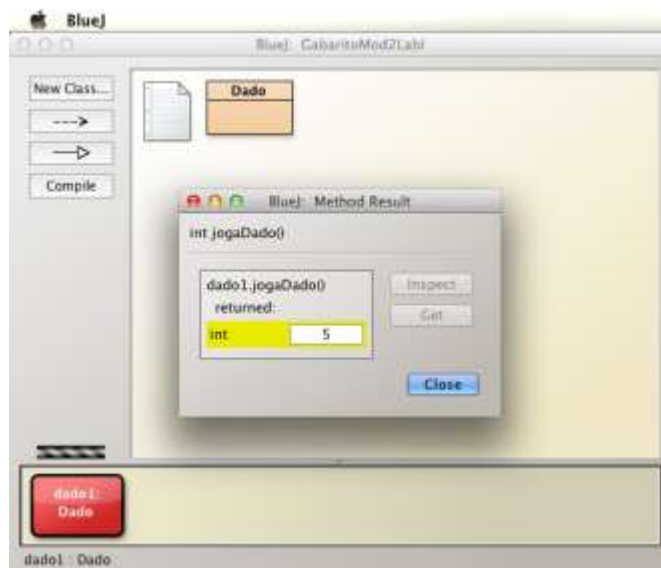




- Jogue o dado, clicando no método jogaDado várias vezes e observando os resultados.



Primeira Jogada = retorno lado 5:



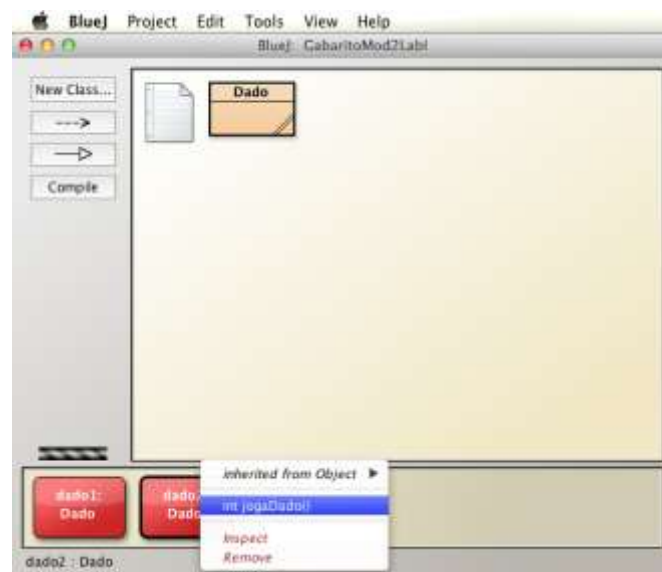
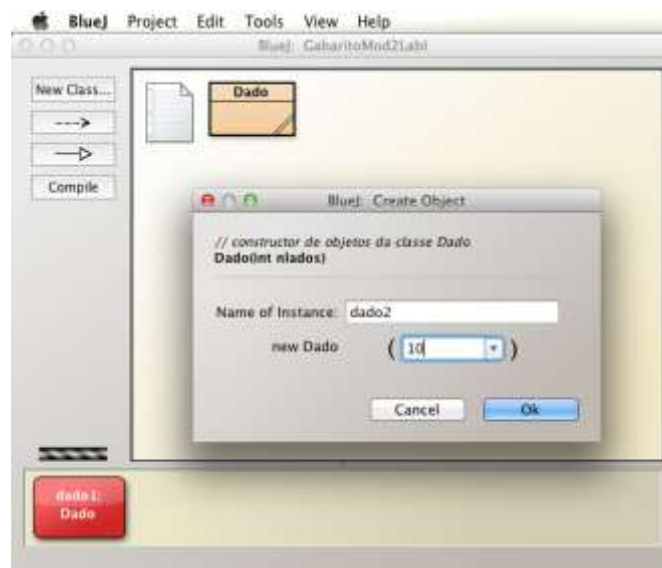
Segunda Jogada, retorno lado 6:

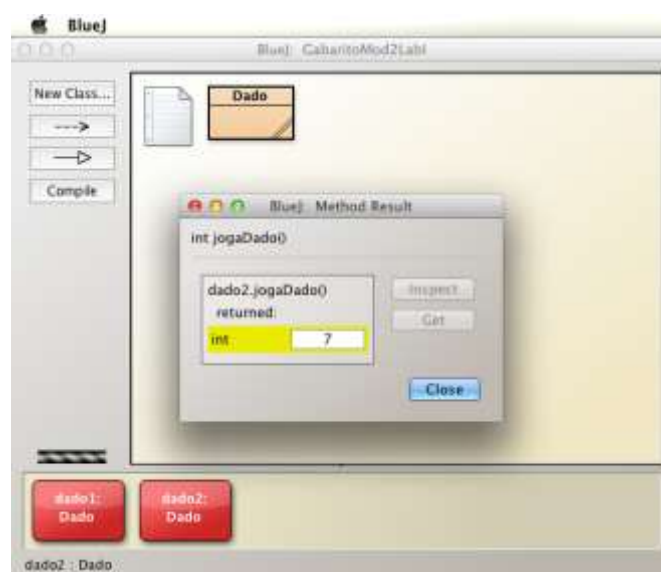
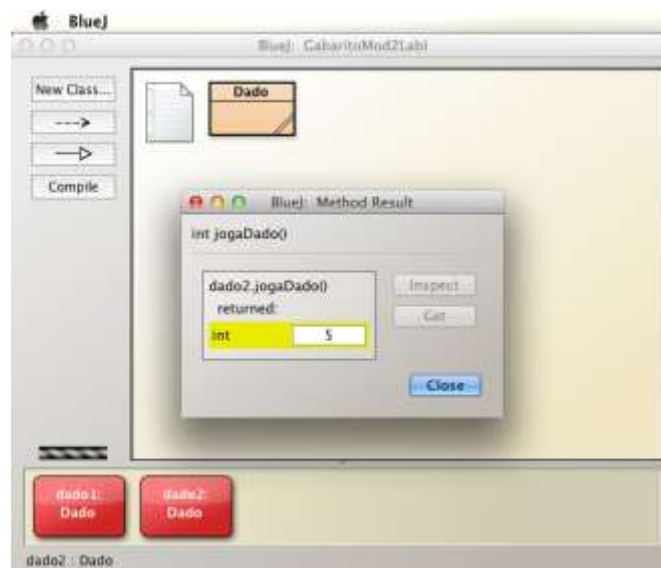
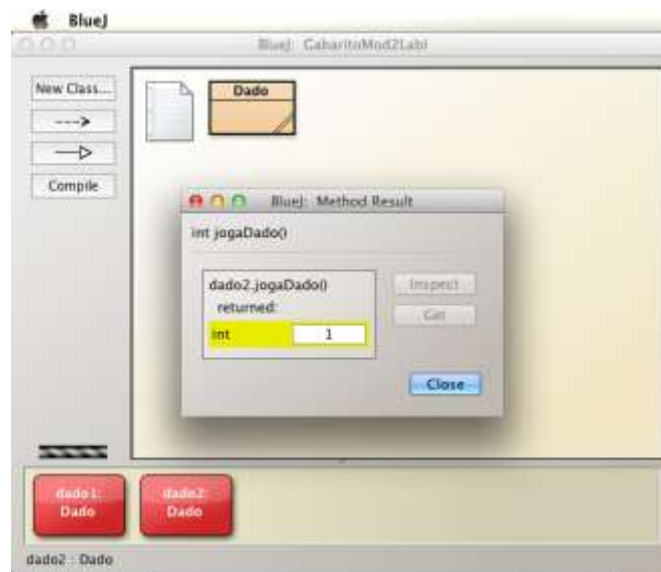


Terceira Jogada, retorno lado 4:



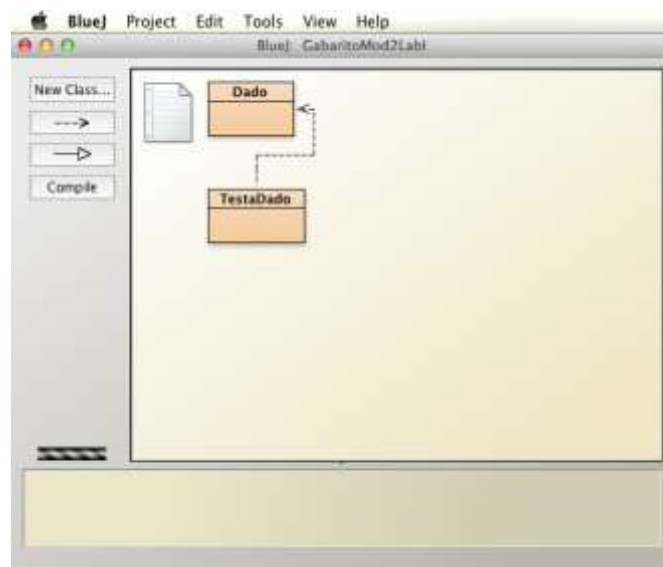
- Crie outro dado de dez lados e jogue várias vezes o dado, observando os resultados.



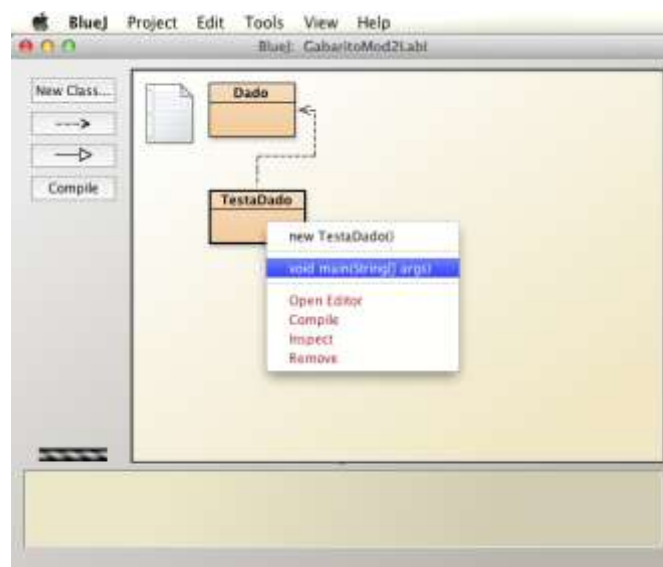


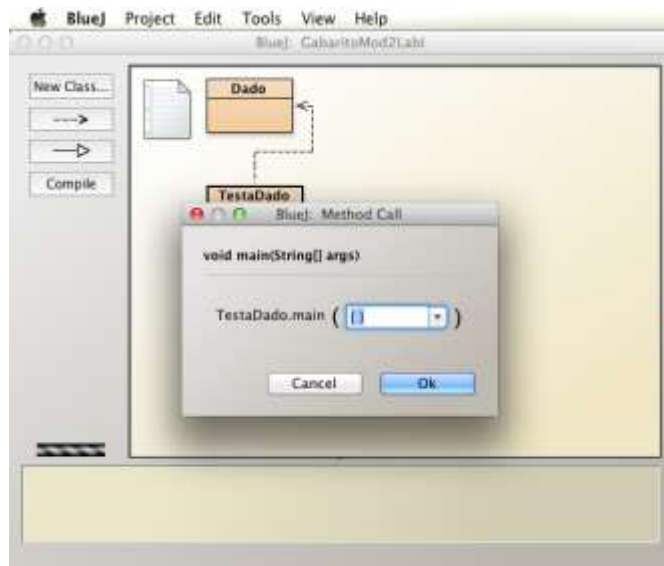
4. No exercício anterior, testamos a classe **Dado** usando as facilidades que o ambiente integrado de desenvolvimento blueJ oferece. Outra forma de testar a classe **Dado** é através de outra classe, isto é através de uma classe de teste. Faremos isto agora: crie a classe **TestaDado** com o seguinte código (copie exatamente como está abaixo) e a compile:

```
public class TestaDado
{
    // método principal
    public static void main (String[] args)
    {
        Dado d = new Dado(6);
        System.out.println("Lado do dado = " + d.jogaDado());
    }
} //fim da classe de teste
```



5. Pelo blueJ, execute o método main (clcando com o botão direito do mouse no ícone da classe **TestaDado**, clique na assinatura do método main). Observe o resultado. Repita esta operação várias vezes e tente entender.





Ao clicar na assinatura do método main serão executados os comandos que estão implementados no corpo deste método, no caso a criação de um objeto dado com 6 lados:

```
Dado d = new Dado(6);
/*
```

Antes do símbolo “=” temos o nome: **Dado** que se refere ao tipo do objeto que será criado. Este tipo possui o nome da classe que será chamada para criar o objeto.

Depois de informar o tipo do objeto, é necessário dar um nome para o objeto que será criado em memória. No exemplo acima o nome do objeto será: “**d**”.

Depois do símbolo “=” temos a palavra reservada: **new**. A execução do comando new permitirá a criação do objeto Dado e o vínculo deste objeto que foi criado em memória com a letra “d”, nome do objeto neste exemplo.

Quando o comando `new Dado(6)` for executado, o método construtor da classe `Dado` será chamado, veja a imagem abaixo:

```
public class Dado
{
    // atributos
    private int lados;

    /**
     * constructor de objetos da classe Dado
     */
    public Dado (int nlados)
    {
        lados = nlados;
    }

    // método que simula o lançamento de um dado
    public int jogaDado()
    {
        return (int) (Math.random ( ) * lados + 1);
    }
}
```

Este método será chamado e, o parâmetro: nlados que está dentro dos parênteses receberá o número 6.

*/

6. Crie a classe **Livro** conforme a descrição e o diagrama UML abaixo. Lembre-se: uma classe é constituída de membros (atributos e métodos).

descrição

Para cada livro deve ser armazenado o seu título e uma indicação se o livro está emprestado ou não (esta indicação deve ser do tipo boolean). Além de um construtor (método especial, com o mesmo nome da classe), a classe deve oferecer as seguintes funcionalidades:

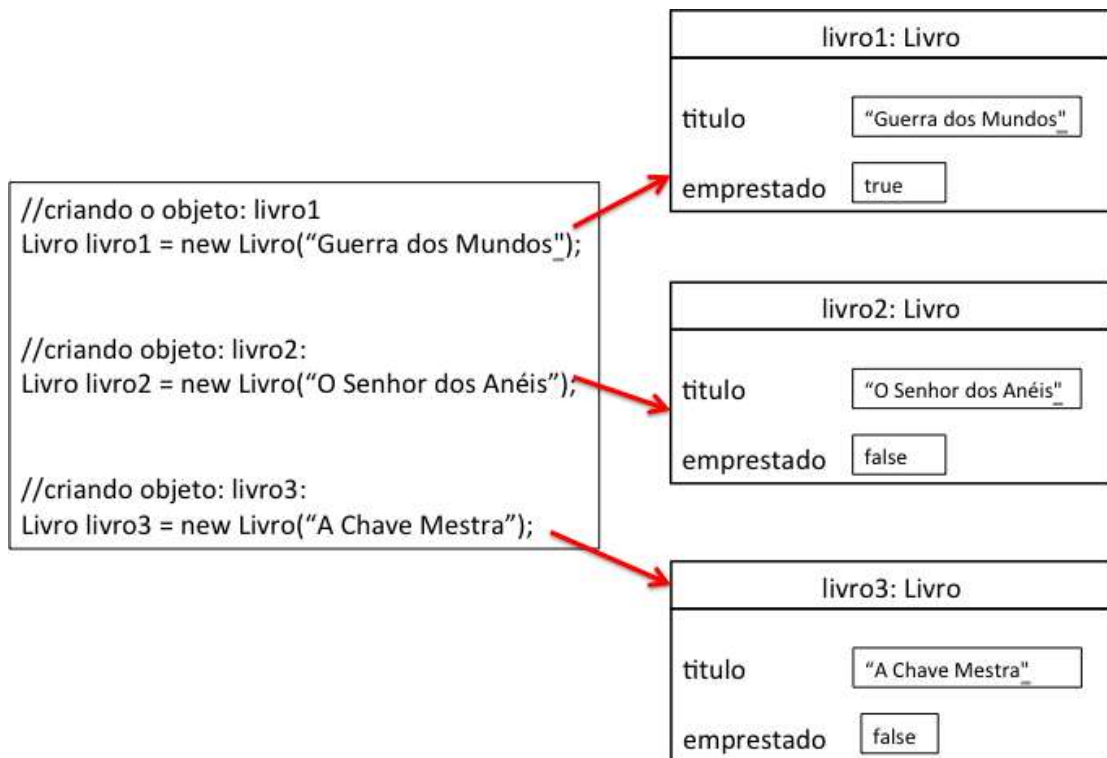
- ◆ empréstimo do livro
- ◆ devolução do livro
- ◆ exibição dos dados do livro na tela
- ◆ acesso aos atributos (métodos get)

Diagrama da classe



Verifique o conteúdo da classe no projeto do BlueJ disponibilizado.

7. Cada objeto é uma instância da classe. Com base no diagrama de objetos abaixo, crie uma classe de teste para exemplificar a criação dos 3 objetos apresentados no diagrama.



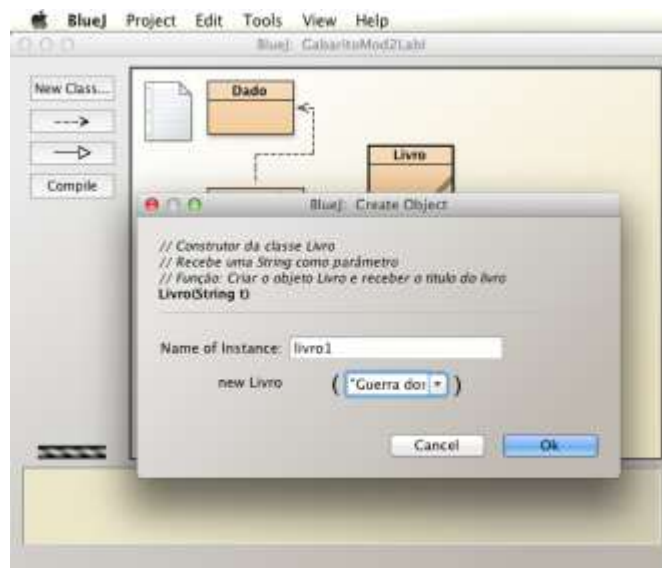
Verifique o conteúdo da classe no projeto do BlueJ disponibilizado.

8. Toda a classe tem um (ou mais) método especial que é o construtor. A responsabilidade de um construtor é configurar (inicializar) os campos do objeto quando este é criado (instanciado). O construtor é automaticamente acionado quando o operador **new** for executado. Copie o construtor da classe Livro para dentro da sua classe. Compile-a novamente e então crie (instancie) os mesmos objetos que você exemplificou na questão 7, executando (basta clicar) no construtor no BlueJ. Inspecione os objetos para ver seu estado interno (isto é, os valores armazenados em seus atributos).

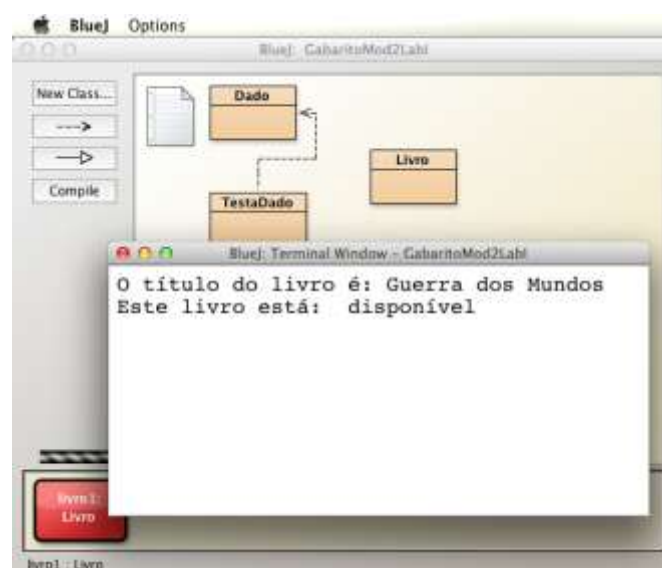
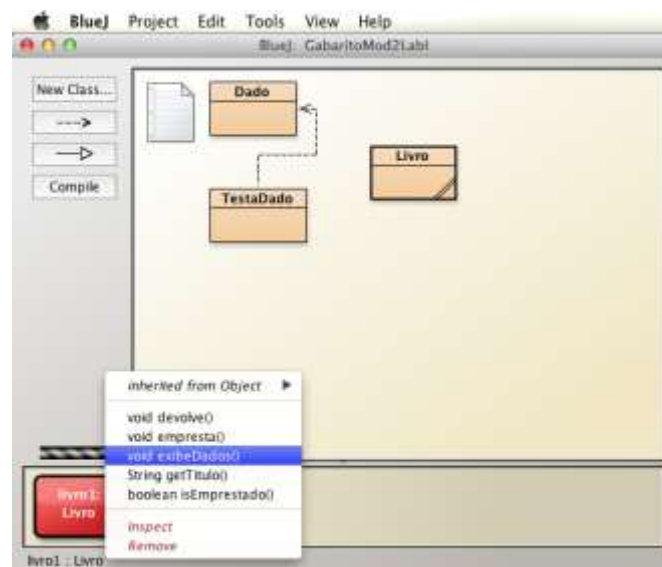
```
// construtor de objetos da classe Livro
public Livro(String t)
{
    titulo = t;
    emprestado = false;
}
```

Verifique o conteúdo da classe no projeto do BlueJ disponibilizado junto com o gabarito.

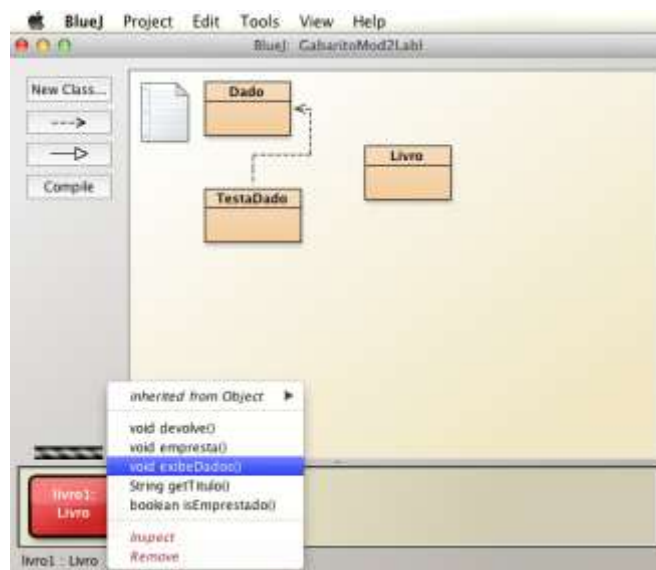
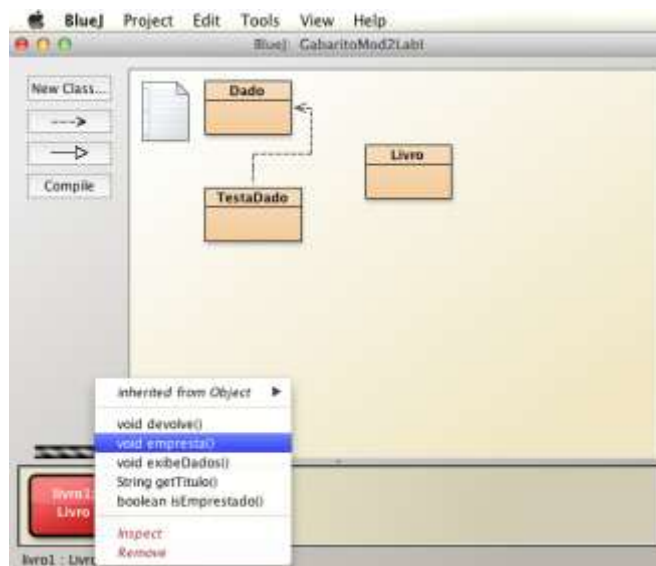
Primeiro Livro Criado:

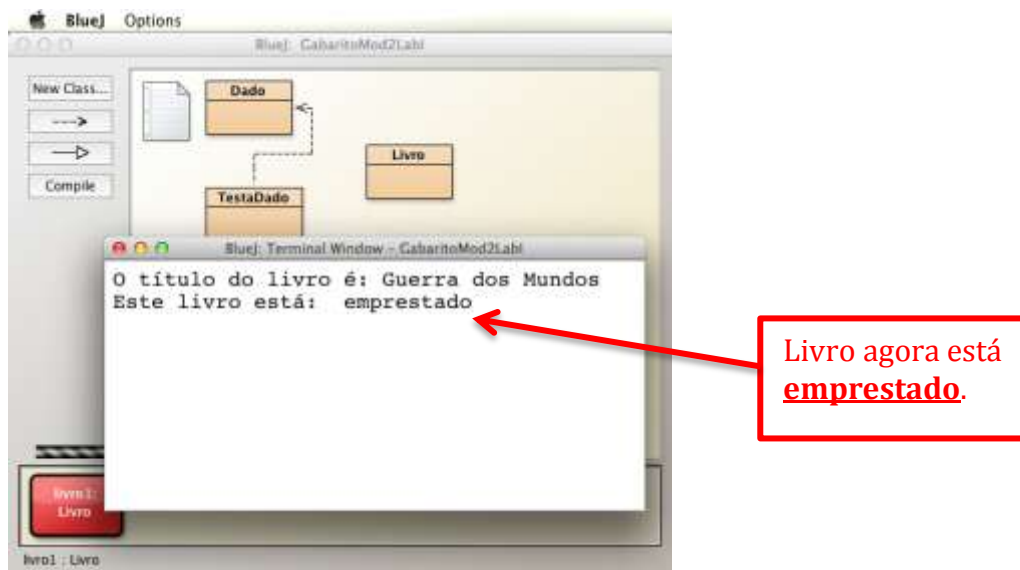


Verifica os valores dos atributos após a criação do objeto:

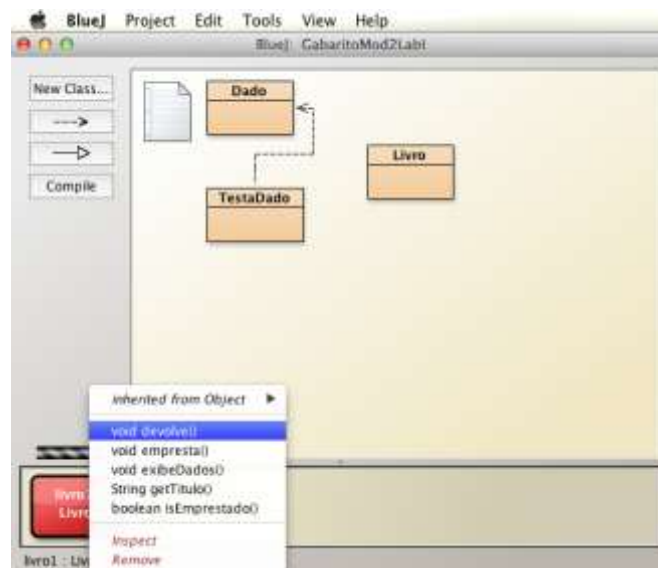


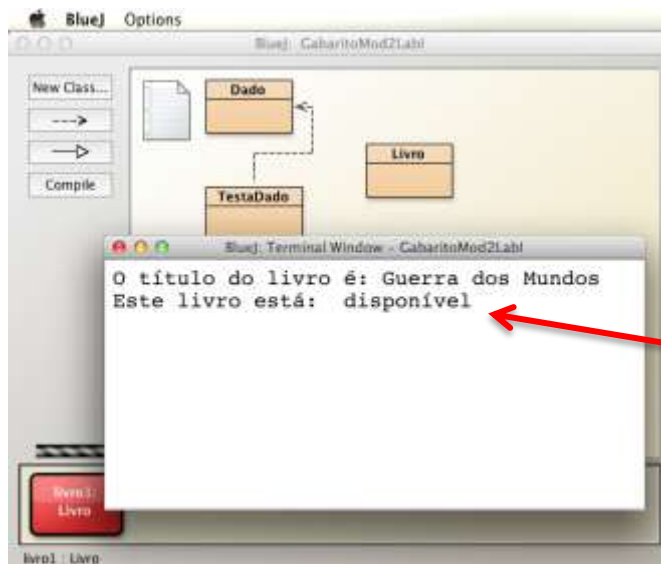
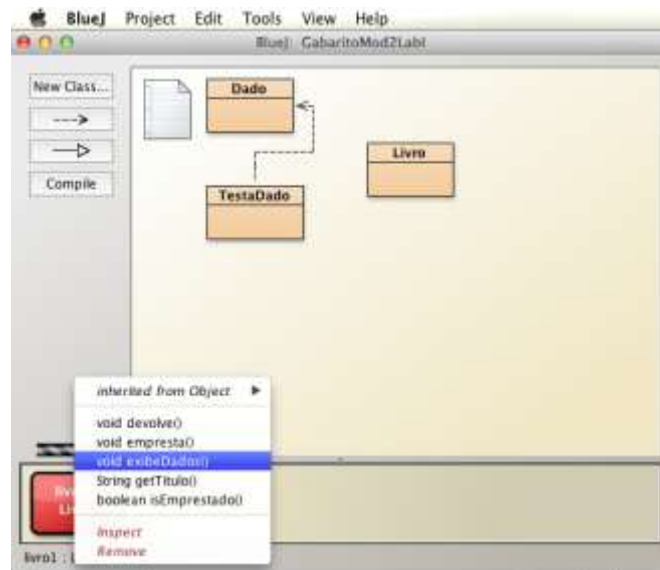
Chama o método `empresta()` e verifica novamente os valores dos atributos depois da execução deste método:





Chama o método `devolve()` e verifica novamente os valores dos atributos depois da execução deste método:





Livro agora está disponível novamente.

9. Copie os outros métodos da classe Livro (listados abaixo) para dentro da sua classe. Compile-a novamente e então crie (instancie), executando o construtor no BlueJ os mesmos objetos que você exemplificou na questão 7. Execute os métodos e inspecione os objetos para observar as mudanças em seu estado interno (isto é, os valores armazenados em seus atributos).

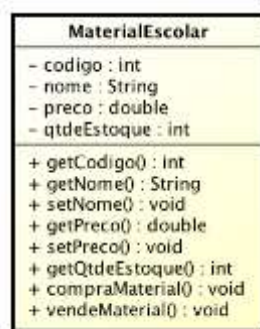
Verifique o conteúdo da classe no projeto do BlueJ disponibilizado junto com o gabarito.

10. As classes se relacionam (olhe as ligações entre os diagramas no BlueJ). Vamos implementar, dentro do mesmo projeto, uma classe de teste de nome **TestaLivro**, sem atributos, contendo apenas o método principal (public static void main (String[] args)) a partir do qual inicia a execução de um programa java.

- a) Abaixo do comentário, escreva o comando correspondente. Compile e execute a classe, observando os resultados, na tela de saída.
- b) Acrescente, livremente, outros comandos para criar novos objetos e manipular os objetos criados e compile e execute novamente a classe.

Verifique o conteúdo da classe no projeto do BlueJ disponibilizado junto com o gabarito.

11. Programe, em Java, a classe **MaterialEscolar**. Cada material tem código (inteiro), nome, preço e quantidade em estoque. A classe deve ter um construtor que configura os atributos do objeto do tipo **MaterialEscolar**. As funcionalidades que devem ser oferecidas pela classe são:
- a. métodos de acesso (get) para obter nome, código, preço e quantidade em estoque do material escolar;
 - b. alterar o nome do material escolar;
 - c. alterar o preço do material escolar;
 - d. adicionar material ao estoque (método que será chamado quando a loja;
 - e. comprar material;
 - f. vender material (diminuindo o estoque).

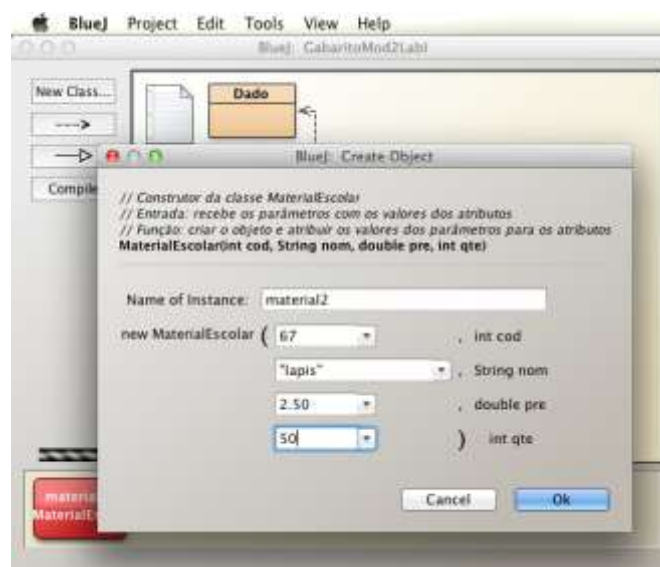
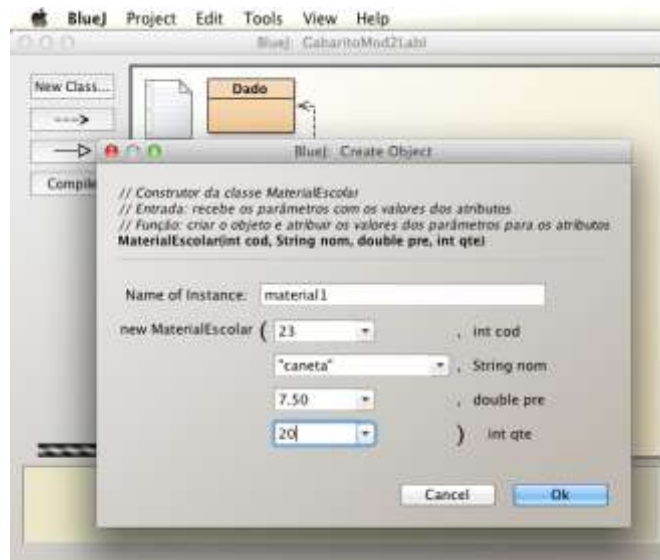


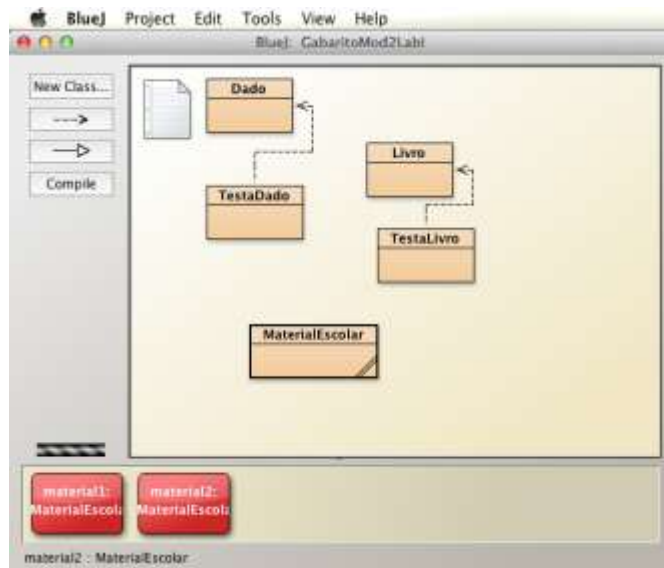
powered by antah

Verifique o conteúdo da classe no projeto do BlueJ disponibilizado junto com o gabarito.

12. Após compilar a classe **MaterialEscolar**, teste-a no blueJ, clicando adequadamente nos ícones para realizar as operações abaixo. Neste exercício você deverá compreender tudo o que está acontecendo na memória (estado interno do objeto) e relacionar com o que foi abordado na atividade programação I. Se necessário releia partes do livro e/ou envie dúvidas para o fórum e peça explicações. Realize as operações, nesta ordem e responda às perguntas, observando que as questões que perguntam “Quem?” estão se referindo à alguma parte da classe.
- a. crie (instancie), no bluej, os dois objetos especificados abaixo:

MaterialEscolar	MaterialEscolar
código: 23	código: 67
nome: "caneta"	nome: "lapis"
preço: 7.50	preço: 2.50
saldo: 20	saldo: 50

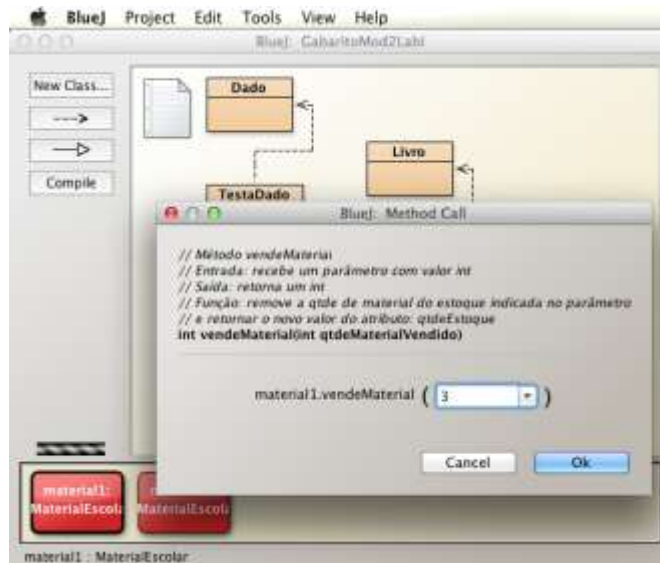




- b. inspecione os objetos criados e observe os valores dos atributos. “Quem” colocou estes valores nos atributos dos objetos?

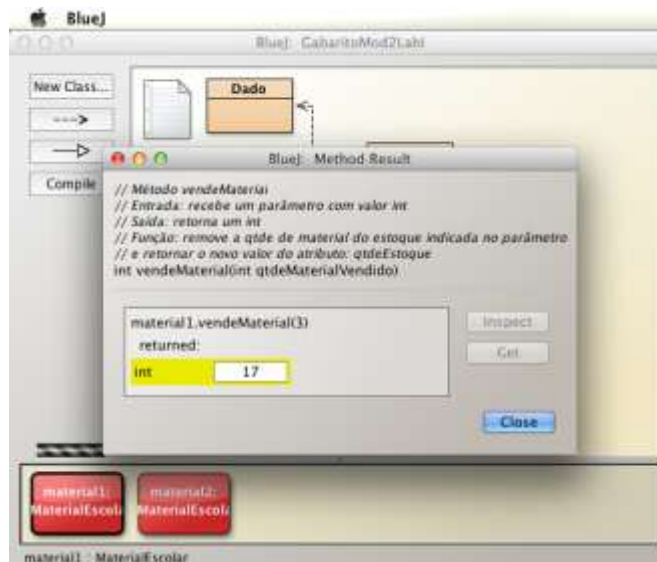
Resposta: o método construtor.

- c. aplique o método que vende material escolar, para realizar a venda de 3 canetas.

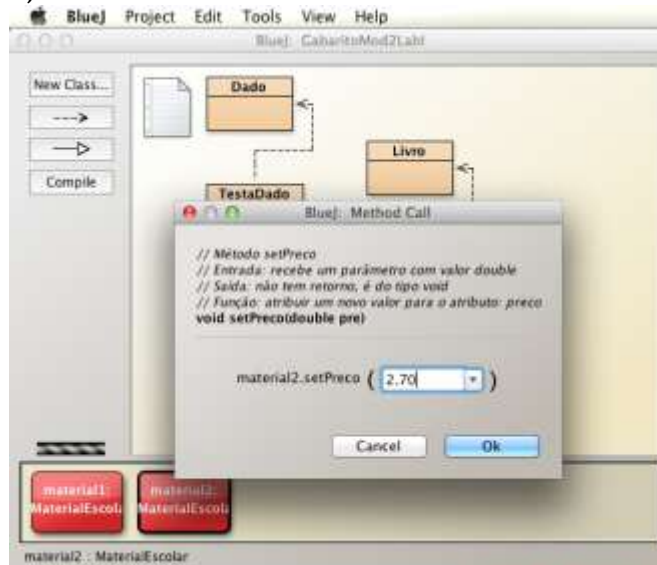


- d. inspecione o objeto sobre o qual foi efetuada a venda e verifique a quantidade em estoque. O que houve? “Quem” foi o responsável?

Resposta: Qtde= 17. O método vendeMaterial() diminuiu o valor do estoque.



- e. aplique o método que altera preço para alterar o preço dos lápis para 2,70

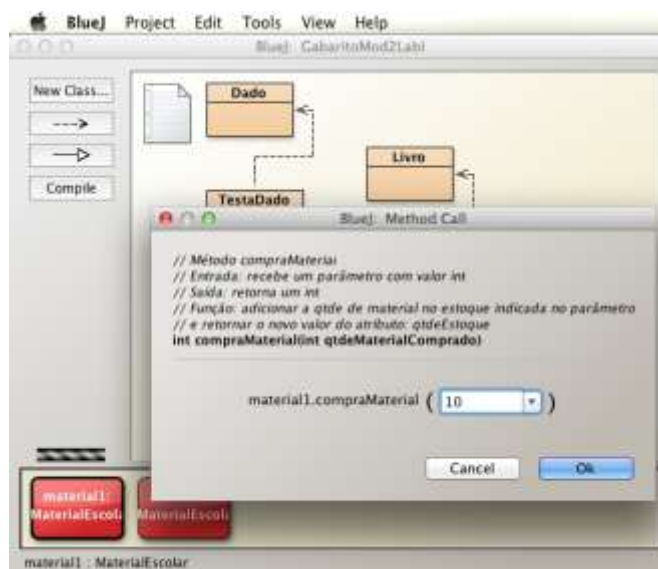


- f. inspecione o objeto sobre o qual foi feita a alteração e verifique o preço. O que houve? “Quem” foi o responsável?

Resposta: Novo Preço= 2.70. O método setPreco() alterou o preço do lápis criado.



- g. aplique o método que compra material escolar, para realizar a compra de 10 canetas;



- h. inspecione o objeto sobre o qual foi efetuada a compra e verifique a quantidade em estoque. O que houve? “Quem” foi o responsável?

Resposta: Nova Qtde em Estoque= 30. O método compraMaterial() adicionou mais 10 canetas no estoque.



13. Programe uma classe de teste (**TesteMaterialEscolar**), contendo apenas o método principal (public static void main(String[] args)), conforme especificações abaixo e execute-a no blueJ, clicando no método main. As operações abaixo devem ser colocadas nesta ordem, dentro do método main:

- usando o operador **new**, criar (instanciar) um objeto da classe **MaterialEscolar** com código 23, nome "caneta", preço 7,50 e quantidade 20.
- usando o operador **new**, criar (instanciar) outro objeto da classe **MaterialEscolar** com código 67, nome "lapis", preço 2,50 e quantidade 50.
- usando comandos **System.out.println**, exibir todos os dados do material escolar 23, na tela.
- chamar o método adequado para vender 3 canetas.
- chamar o método adequado para alterar o preço dos lápis para 2,70
- chamar o método adequado para comprar (adicionando ao estoque) 10 canetas
- usando comandos **System.out.println**, exibir todos os dados dos dois materiais na tela, com títulos adequados.

Verifique o conteúdo da classe de Teste no projeto do BlueJ disponibilizado junto com o gabarito.