

1. Anteriormente, 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, dentro do mesmo projeto onde está a classe Dado, a classe **TestaDado** com o seguinte código (copie exatamente como está abaixo) e compile-a.

a) As classes se relacionam (olhe as ligações entre os diagramas no BlueJ).

b) Analise o código e faça um “teste de mesa” mostrando o que acontecerá nas duas unidades do computador: memória e saída (tela); quando o método main for executado.

c) Após a compilação, execute o método main e entenda o que foi realizado.

```
public class TestaDado {  
  
    // método principal  
    public static void main (String[] args)  {  
  
        //declaração de variáveis  
        int resultado;  
        Dado d;  
  
        //instanciação de objeto  
        d = new Dado(6);  
  
        //chamada de método  
        resultado = d.jogaDado();  
  
        //exibição na tela  
        System.out.println("O dado foi jogado e deu: " + resultado);  
  
    } //fim do método main  
} //fim da classe de teste
```

2. Após ler o material sobre a linguagem de modelagem UML, tente fazer um esboço do diagrama da classe **Livro** conforme a descrição abaixo. Para fazer o diagrama completo seria necessário muito conteúdo que ainda não foi desenvolvido. Então após a tentativa olhe o diagrama pronto e não se preocupe com o que ainda não foi visto em aula.

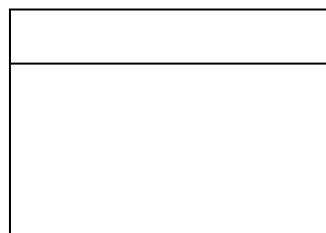
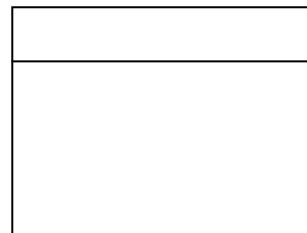
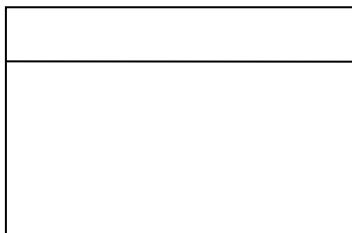
descrição

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)

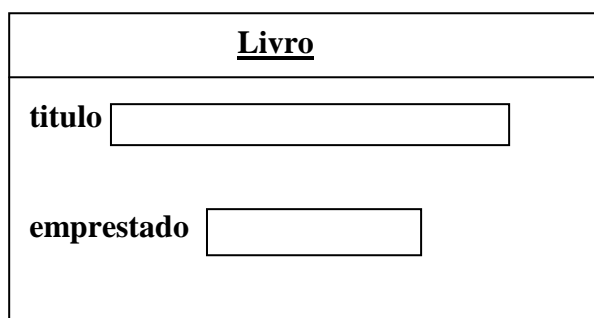


3. Cada objeto é uma instância da classe. Faça os diagramas de objetos exemplificando pelo menos três objetos da classe Livro.



4. Uma classe é constituída de membros (atributos e métodos). Na primeira aula de programação você aprendeu a especificar os atributos em java. Abra o BlueJ, crie um projeto e a classe Livro. Abra o editor e escreva em Java, **somente**, os **atributos** da classe Livro. **Para cada método coloque apenas um comentário** indicando o que o método deverá realizar. Compile a classe. Crie um objeto, inspecione e preencha o diagrama abaixo mostrando como ficou.

Obs. comentário é uma descrição escrita em português que será ignorada pelo compilador. Uma das formas de indicar que é uma linha de comentário é começar a linha com //.



5. Toda a classe tem um (ou mais) 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 do quadro o construtor da classe Livro para dentro da sua classe. Compile-a novamente e então crie (instancie), executando (basta clicar) o construtor no BlueJ, os mesmos objetos que você exemplificou na questão 3. Inspecione os objetos para ver seu estado interno (isto é, os valores armazenados em seus atributos).
6. Copie do quadro os outros métodos da classe livro 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 3. Execute os métodos e inspecione os objetos para observar as mudanças em seu estado interno (isto é, os valores armazenados em seus atributos).
7. As classes se relacionam (olhe as ligações entre os diagramas no BlueJ). Vamos implementar, dentro do mesmo projeto, a classe **TesteLivro**, 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.
- b) Acrescente outros comandos para criar e/ou manipular os objetos criados e compile e execute novamente a classe.

```

public class TesteLivro {
    public static void main (String[] args) {

        // Declarar duas variáveis objeto, do tipo Livro

        //Usando o operador new, criar um livro e atribuir a referência para a variável adequada

        /**
         * Usando o operador new, instanciar outro livro e
         * atribuir a referência para a variável adequada.
         */

        /**
         * Chamar o método para exibir na tela os dados dos dois livros
         * criados com cabeçalhos “Primeiro Livro” e “Segundo Livro”
         */

        // Chamar o método para emprestar o primeiro livro

        /**
         * Chamar o método para Exibir na tela os dados do primeiro livro
         * com o cabeçalho “Primeiro livro após empréstimo”
         */

        // acrescente outros comandos .....
    }
}

```

8. Faça o **diagrama UML** e programe, **em Java**, a classe **MaterialEscolar**. Cada material tem código (inteiro), nome, preço e quantidade em estoque. A classe deve ter um construtor (*que será automaticamente ativado pelo operador new*) que configura os atributos do objeto do tipo **MaterialEscolar**. As funcionalidades que devem ser oferecidas pela classe são:
- adicionar material ao estoque (método que será chamado quando a loja comprar material)
 - vender material (diminuindo o estoque)
 - alterar o nome do material escolar
 - alterar o preço do material escolar
 - métodos de acesso (get) para obter nome, código, preço e quantidade em estoque do material escolar.

Obs. se os métodos de acesso e operadores aritméticos ainda não foram vistos na aula de progI, então copie do quadro os métodos prontos.

9. Após compilar a classe **MaterialEscolar**, ela pode ser executada, pela JVM?
Por quê?

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, na tela) e relacionar com o que foi visto em progI. Se necessário releia partes das notas de aula e/ou chame o professor e peça explicações. Realize as operações, nesta ordem e responda as perguntas, observando que as questões que perguntam “*Quem?*” estão se referindo à alguma parte da classe.:

a) No blueJ, crie (instancie) os dois objetos especificados abaixo:

<u>MaterialEscolar</u>
código: 23 nome: “caneta” preço: 7.50 qtEstoque: 20

<u>MaterialEscolar</u>
código: 67 nome: “lapis” preço: 2.50 qtEstoque: 50

b) No blueJ, inspecione os objetos criados e observe os valores dos atributos. “*Quem*” colocou estes valores nos atributos dos objetos?

c) No blueJ, aplique o método que vende material escolar, para realizar a venda de 3 canetas;

d) No blueJ, inspecione o objeto sobre o qual foi efetuada a venda e verifique a quantidade em estoque. O que houve? “*Quem*” foi o responsável?

e) No blueJ, aplique o método que altera preço para alterar o preço dos lápis para 2,70

f) No blueJ, inspecione o objeto sobre o qual foi feita a alteração e verifique o preço. O que houve? “*Quem*” foi o responsável?

g) No blueJ, aplique o método que compra material escolar, para realizar a compra de 10 canetas;

h) No blueJ, inspecione o objeto sobre o qual foi efetuada a compra e verifique a quantidade em estoque. O que houve? “*Quem*” foi o responsável?

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