

### Saída de dados

1. Programe um método *exibirDados()* para cada uma das classes **Livro**, **Aluno**, **Material** e **Pessoa** programadas na aula passada. Vamos adotar esse método como padrão para mostrar na tela o valor de cada atributo do objeto. Exemplo, para a classe **Material**:

Código: 1234  
Nome: Caderno  
Preço: R\$ 6.50  
Saldo: 12

### Programando uma classe de teste

- **Atenção:** essa classe de teste não tem atributos e só tem um método (*main*), dentro do qual se codificam todas as operações que são pedidas.

2. No mesmo projeto onde está a classe **Aluno**, crie uma classe de nome **TestaAluno** e preencha-a com o código abaixo. A seguir, complete a programação do método *main*, programando o que pede cada comentário:

```
public class TestaAluno{
    public static void main(String[] args){
        // Instanciar um objeto Aluno com dados de sua escolha:

        // Instanciar outro Aluno:

        // Calcular a média de cada um dos alunos, chamando o método que
        // calcula a média:

        // Exibir na tela os dados dos dois alunos, chamando o método
        // exibeDados() para cada um deles:

    }
}
```

Execute a classe de teste chamando o método *main*.

### Praticando métodos não void

3. Se ainda não tiver, coloque métodos de acesso (*get*) nas classes **Livro**, **Aluno** e **Material**.

4. Repita o exercício 2, porém, em vez de chamar os métodos *exibirDados()*, programe no código da classe de teste a saída dos dados dos dois alunos no formato abaixo:

Relatório de notas				
-----				
Nome do aluno		Notas		Média
-----				
xxxxxxxxxxxxxxxxxx	xx.xx	xx.xx	xx.xx	xx.xx
xxxxxxxxxxxxxxxxxx	xx.xx	xx.xx	xx.xx	xx.xx
Média dos dois alunos:      xx.xx				

**Obs.** use nomes de alunos com a mesma quantidade de caracteres, para proporcionar a aparência regular das colunas do relatório.

5. Programe a classe **Data** com os seguintes membros e faça testes:

Atributos privados inteiros:

- dia
- mês
- ano

Construtor: recebe 3 parâmetros para inicializar o dia, o mês e o ano da data

Métodos:

- + *obterDataPadrao* – método sem parâmetros, que deve retornar a data no formato padrão dia/mês/ano, como um String. Por exemplo, 25/3/2016.
- + um método *get* para cada atributo



### Atributo de tipo objeto – Associação de classes

6. Programe a classe **Atleta**, com os seguintes membros e faça testes:

Atributos privados (somente estes):

- nome
- altura - double
- data de nascimento – do tipo **Data**

Construtor: com 3 parâmetros. Um deles é um objeto do tipo **Data**

Métodos:

- + *set* – um para cada atributo
- + *get* – um para cada atributo
- + *exibirDados* – deve exibir os valores dos atributos no seguinte formato:

```
Dados do atleta
Nome: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Altura: x.xx m
Data de nascimento: xx/xx/xxxx
```

Quando uma classe tem atributo cujo tipo é outra classe, dizemos que há uma **associação** das duas classes.

Complete o construtor abaixo e inclua-o na classe **Atleta**:

```
public Atleta (String umNome, double umaAltura, int umDia, int umMes, int umAno) {
```

7. Escreva a classe de teste **TestaAtleta** para executar os seguintes passos:

- Instanciar um objeto **Data** com a sua data de nascimento e atribuir à uma variável **d**
- Instanciar um objeto **Atleta**, com seu nome, sua altura e a data que está em **d**, chamando o primeiro construtor da classe.
- Instanciar um segundo objeto **Atleta**, com dados inventados, chamando o segundo construtor da classe.
- Exibir os dados dos dois atletas, chamando o método **exibirDados** para cada um deles.
- Instanciar um terceiro objeto **Atleta**, com o segundo construtor, tendo o nome do segundo atleta, a altura do primeiro e a data de nascimento igual a do segundo atleta, mas do ano seguinte. **Atenção:** não usar constantes para valores dos parâmetros, mas chamar gets.
- Exibir os dados do terceiro atleta.