

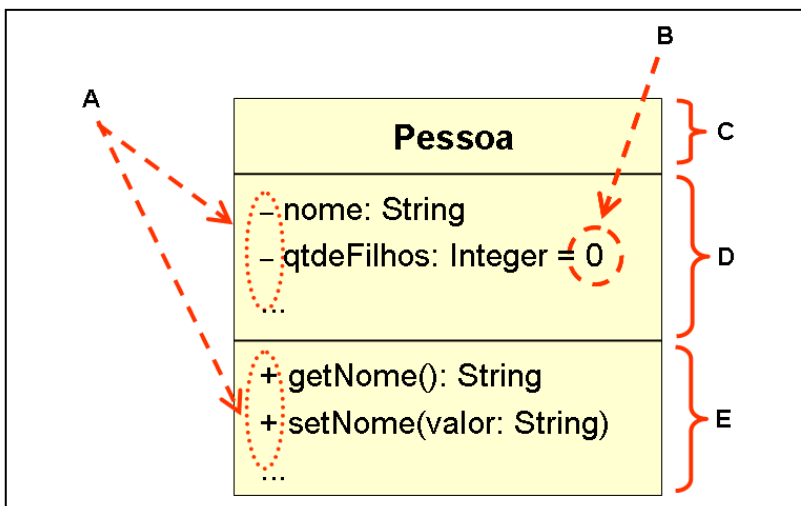
## Roteiro 01 – Exercícios de fixação

Com base no que vimos até agora e no conhecimento adquirido no semestre anterior, você deve ser capaz de responder às perguntas abaixo.

1. O que devemos fazer no processo de abstração?
2. Por que a classe **Person** é uma abstração?
3. Qual a mudança de paradigma da abordagem estruturada para a orientada a objetos?
4. Qual a diferença entre uma classe e um objeto?
5. O que é encapsulamento?
6. Como identificar classes e objetos necessários para resolver um problema?
7. O que é coesão?
8. Por que uma classe deve ter alta coesão?
9. Por que as operações de uma classe devem ter alta coesão?
10. Qual a relação entre abstração, classe e coesão?
11. Qual a diferença entre um objeto e a instância de uma classe?
12. O que significa um objeto ter uma identidade única?
13. O que representa a palavra reservada `this` em Java?
14. Como explicar o estado de um objeto?
15. Objetos com estados iguais representam um mesmo objeto?
16. Em quais situações o estado de um objeto é modificado?
17. O que é um objeto imutável? Adicione um exemplo em Java, demonstrando o conceito.
18. Quando devemos utilizar variáveis globais?
19. Por que os atributos devem ser privados?
20. Qual a relação de atributos privados com encapsulamento e ocultamento de informação (*information hiding*)?
21. Quando um atributo deve ser público?
22. O que define o comportamento de um objeto?
23. Todas as operações devem ser públicas (explique)?
24. Quando uma operação pode ser privada?
25. O que é a assinatura de uma operação?
26. O que você entende por interface no contexto de uma classe?
27. Qual a diferença entre criar um objeto e instanciar um objeto?
28. Para quem devemos pedir a criação de um novo objeto?
29. Para quem devemos pedir a destruição de um objeto (pense de modo geral, sem considerar especificamente a linguagem Java)?

## Roteiro 01 – Exercícios de fixação

30. Na linguagem Java, como os objetos são destruídos?
31. Como o Garbage Collector sabe que ele pode destruir um objeto?
32. Qual a diferença entre um compilador e um interpretador?
33. Qual a diferença entre código de máquina e bytecode?
34. Qual a função da Máquina Virtual Java (JVM)?
35. Qual a diferença entre um arquivo “.java” e “.class”?
36. O que é UML?
37. Qual a utilidade da UML na orientação a objetos?
38. Qual a definição de um pacote (*package*) e para que ele serve em UML e Java?
39. Qual a diferença entre operação e método?
40. O que é uma operação de classe (operação estática)?
41. Considerando o diagrama (UML) abaixo, identifique e defina cada um dos elementos indicados pelas letras (A, B, C, D e E). Explique cada elemento (e, quando for o caso, as informações contidas no elemento) no contexto da figura.



42. Como uma operação de classe (estática) é representada na UML?
43. Adicione a operação “+ **getAge(): int**” na classe **Person**. Esta operação retorna a idade (em anos) de uma pessoa. Desenvolva o método correspondente em Java (veja a classe de teste fornecida para mais detalhes sobre como implementá-la). Você saberia dizer porque a operação poderia ter a propriedade UML **isQuery** marcada (não é obrigatório)?
44. Considere que você precisa implementar um programa em Java para permitir o gerenciamento de um cadastro de pessoas (considerando a classe **Person** trabalhada neste roteiro). O programa precisa apenas permitir a inclusão de pessoas e oferecer uma funcionalidade para listar as pessoas que foram cadastradas até agora. Quais classes seriam necessárias? Onde as pessoas ficariam armazenadas? Em qual das classes definidas? Em qual classe estaria a entrada de dados?
45. Modele as classes em UML e implemente o programa que você descreveu na questão anterior.

## Roteiro 01 – Exercícios complementares

O objetivo desta atividade é que você exercite seus conhecimentos de programação e, ao mesmo tempo, fique familiarizado com a linguagem Java.

1. Entre no link <http://codingbat.com/java> e navegue entre as centenas de opções de exercícios. Não fique limitado ao que é visto em sala. Para programar bem, é necessário aplicar na prática.
2. Links com exercícios, tutoriais e atividades em Java e NetBeans:
  - a. <https://projecteuler.net/>
  - b. <http://docs.oracle.com/javase/tutorial/>
  - c. <http://www.java67.com/2013/01/10-programming-questions-and-exercises.html>
  - d. <http://www.homeandlearn.co.uk/java/java.html>
  - e. <http://www.javapractices.com>
  - f. <http://javafree.uol.com.br/>
  - g. [http://www.java.com/pt\\_BR/](http://www.java.com/pt_BR/)
  - h. <http://www.guj.com.br/>
  - i. Java Tutorial (Oracle) em <http://docs.oracle.com/javase/tutorial/index.html>
  - j. Java API Specification (Oracle) em <http://docs.oracle.com/javase/8/docs/api/overviewsummary.html>
  - k. Tutorial rápido para NetBeans em <https://NetBeans.org/kb/docs/java/quickstart.html>