

JAVA

para iniciantes

```
def filter(s: Set, p: Int => Boolean): Set = {
  def forall(s: Set, p: Int => Boolean): Boolean = {
    def iter(a: Int): Boolean = {
      if (contains(p,a) && !contains(s,a)) false
      else if (a > 1000) true
      else iter(a + 1)
    }
    iter(-1000)
  }
  def exists(s: Set, p: Int => Boolean): Boolean = {
    (x: Int) => !(forall(s, !p(x)))
  }
  def map(s: Set, f: Int => Int): Set = {
    (b: Int) => exists(s, (c: Int) => f(c) = b)
  }
}
```



Paulo Balmer

JAVA

para iniciantes

Isenção de Responsabilidade

Todo o conteúdo presente neste e-book são provenientes das minhas experiências pessoais com o aprendizado da linguagem de programação Java ao longo de vários anos de estudos e também ensino. Apesar do meu esforço para garantir a precisão e a mais alta qualidade dessas informações e entenda que todas as técnicas e métodos aqui ensinados sejam altamente eficientes para qualquer estudante de programação desde que seguidos conforme instruídos, nenhum dos métodos ou informações foi cientificamente testado ou comprovado, e eu não me responsabilizo por erros ou omissões. Além disso, fatores como a sua situação e/ou condição particular também pode não se adequar aos métodos e técnicas ensinados neste e-book. Assim, você deverá utilizar e ajustar as informações deste guia de acordo com sua situação e necessidades.

Todos os nomes de marcas, produtos e serviços mencionados neste guia são propriedades de seus respectivos donos e são usados somente como referência. Além disso, em nenhum momento neste guia há a intenção de difamar, desrespeitar, insultar, humilhar ou menosprezar qualquer pessoa, cargo ou instituição. Caso qualquer parte deste e-book seja interpretada dessa maneira, eu gostaria de deixar claro que não houve intenção nenhuma de minha parte em fazer isso. Caso você acredite que alguma parte deste e-book seja de alguma forma desrespeitosa ou indevida e deva ser removida ou alterada, você pode entrar em contato diretamente comigo através do e-mail paulo@academiadaprogramacao.com.br.

JAVA

para iniciantes

Sobre o Autor

Eu me chamo Paulo Roberto Balmer, e há mais de 20 anos tenho contato direto, estudando, trabalhando e também ensinando programação. Tive meu primeiro contato com desenvolvimento de software no ano de 1997 quando comecei a pesquisar e desenvolver sistemas, na época apenas por interesse em saber como era o processo de criação desses programas. Em 2003 ingressei no curso de graduação em Tecnologia em Processamento de Dados pela Faculdade de Ciências Sociais Aplicadas de Cascavel, PR o qual obtive diploma no de 2005. Continuando minha jornada de aprendizado, fiz uma especialização em 2009 em Desenvolvimento de Software e Novas Tecnologias pela Universidade do Oeste do Paraná, a qual abordou de uma forma muito intensa Java e Orientação a Objetos. Tenho experiência como professor de disciplinas de programação para curso Técnico em Informática de nível médio e também no curso superior de Análise e Desenvolvimento de Sistemas, onde pude obter vasta experiência na área de Ciência da Computação, com ênfase em Linguagens de Programação Orientada a Objetos, atuando principalmente nos seguintes temas: ensino, algoritmos, programação, banco de dados, web e sistemas distribuídos e também na didática de ensinar esses assuntos. Atualmente sou diretor proprietário da empresa Balmer Sistemas, e também compartilho através da Academia da Programação meus conhecimentos com quem tem interesse em aprender programação, ou aperfeiçoar seus conhecimentos nesta área.

JAVA

para iniciantes

Para quem esse e-book é indicado

Eu já fui professor universitário, e já tive a oportunidade de testar em sala de aula com alunos ao vivo, por diversos anos a metodologia de ensino que abordarei nesse e-book. Nestas turmas, pude observar a variedade de perfis de alunos diferentes. Alguns já trabalhavam com desenvolvimento de software, alguns no setor de suporte de empresas de desenvolvimento, outros só conheciam a programação estruturada, e por fim uns não tinham contato nenhum com programação, quiçá com computador.

Posso afirmar, que se você estiver disposto a aprender os conceitos que te levarão a no futuro ser um programador que vai poder tirar o máximo de proveito de toda a linguagem Java esse e-book é pra você.

Independente da sua faixa etária, nível de conhecimento em programação, ou dificuldade de aprendizagem. Tento durante a transcrição do texto ser o mais claro e objetivo possível. Nem todos os meu alunos aprenderam o que ensinei, e é evidente que existe uma variável que deve ser levada em consideração que é seu nível de interesse e insistência em aprender algo.

Índice

1. Introdução.....	6
2. Minha decepção com o Java.....	8
3. Se preparando para começar a aprender.....	9
4. Caminho facilitado.....	10
5. Aprender Orientação a Objetos.....	11
6. Programação Orientada a Objetos.....	11
7. Escolha uma IDE de desenvolvimento.....	25
8. Classes, Atributos, Métodos e Objetos.....	26
9. Instruções de Controle.....	26
10. Coleções, Arrays e ArrayLists.....	27
11. Instruções de Repetição.....	27
12. Exceções.....	27
13. Manipulação de Arquivos.....	28
14. Threads.....	28
15. Aplicativos Gráficos Swing.....	29
16. Java para Web.....	29
17. Considerações Finais.....	30

1. Introdução

A linguagem de programação Java não é fácil, mas é uma tecnologia que após aprendida, poderá te ajudar a tirar o máximo de proveito de seus recursos.

Seja muito bem vindo e obrigado por fazer o download deste e-book. Em suas mãos, está o melhor conteúdo que poderá lhe auxiliar a trilhar um caminho para aprender a programar em Java. Eu não falo isso só porque acredito neste conteúdo e na técnica ensinada aqui, mas porque acredito em você, acredito no seu potencial que pode ser explorado, e que se você seguir as orientações apresentadas aqui, obterá sucesso no aprendizado dessa linguagem de programação. Eu já fui professor nas mais diversas instituições de ensino, e em muitas vezes, é impressionante ver a quantidade de professores que simplesmente não acreditam que seus alunos conseguirão aprender a programar. A expressão que “os alunos são fracos”, ou que “não existe muito tempo para aprender” são muito freqüentes nesse meio. “Já ensinei diversas vezes a mesma coisa, e não tem como eles aprenderem.”. Você já parou para perceber que todos os cursos prometem um método único de ensino? E que na verdade, seja em uma faculdade, em um cursinho, ou em algum curso pela internet, todos acabam utilizando métodos muito parecidos para ensinar.

Me lembro bem do cenário quando fiz a minha graduação: quando iniciei a graduação, nós estávamos em 120 alunos divididos em duas turmas, e ao final do curso, três anos após, apenas 30 e poucos alunos se formaram, sendo alguns de turmas anteriores.

Mas eu me pergunto: o que leva a esse alto índice de desistência e/ou reprovação? Seja qual for a resposta a essa pergunta, essa desistência com certeza gera na mente da pessoa que tentou pensamentos do tipo “não levo jeito para programação”, “não tenho dom pra isso”, “sou muito burro”, “eu não consigo entender, é muito complexo pra mim”. Definitivamente, hoje, no cenário atual, uma pequena parcela de pessoas que se dispõe a estudar programação consegue de fato obter sucesso, geralmente mais por conta do esforço pessoal do que do curso em si. O que posso dizer é que neste e-book você encontrará informações valiosas, pois eu aprendi a

programar sozinho, sem fazer nenhum curso, e em uma época que nem acesso a internet eu tinha. Quando fiz 18 anos entrei em uma faculdade de Processamento de Dados, com conhecimentos intermediários de programação. E com esse e-book, a minha missão, é lhe ajudar a tornar o seu desejo de se tornar um programador Java realidade.

A idéia com esse e-book não é de lhe ensinar Java propriamente dito, mas lhe dar o caminho por quais conceitos começar e qual seqüência de assuntos seguir.

Resolvi escrever este e-book por ver que em diversos fóruns por muitas vezes pessoas não sabiam por onde começar os estudos, justamente pela amplitude de conceitos e recursos que essa linguagem possui. É fato também que a profissão de desenvolvedor de sistemas hoje é uma profissão em média muito bem remunerada em nosso país, e ter domínio completo de Java por exemplo, pode ser um passo importante para alavancar a sua carreira como profissional, ou então até como eu fiz, empreender, desenvolvendo o seu software e iniciando o seu próprio negócio.

2. Minha decepção com o Java

Eu não sei quanto a você que está lendo este e-book, mas na minha primeira experiência pessoal, ao ter contato com programação, aprendi a programação funcional. Para mim se tornou muito natural, pensar que o programa iniciava sua execução na primeira linha, e de uma forma seqüencial ia executando instrução por instrução até encerrar.

Quando tentei aprender Java e orientação a objetos pela primeira vez, acabei não conseguindo. A verdade é que os conceitos não são tão complicados assim, principalmente agora que entendo e conheço a fundo cada um dos conceitos. Hoje, olho para trás, e imagino que um dos fatores foi falta de tempo, pois tentaram me ensinar Orientação a Objetos e o básico de Java em apenas uma disciplina de faculdade no último ano de um curso de tecnologia.

Gostaria de deixar claro que a culpa não era de ninguém, pois a verdade é que na época, o Java e Orientação a Objetos estavam começando a entrar no mercado, e a instituição mudou o curso uma turma após a minha, e inclui orientação a objetos de uma forma mais intensa na grade do curso, desde o primeiro ano.

É muito provável que se você hoje faz um curso de graduação de Análise e Desenvolvimento de Sistemas, você tenha contato com Orientação a Objetos desde o primeiro período. E talvez você já tenha tido contato com ela, e tenha ficado assustado, por isso está lendo este e-book. Então, gostaria de lhe encorajar e animar, pois apesar dessa minha primeira experiência não ter sido a melhor, chegou um momento em que pude aprender definitivamente todos esses conceitos. E de fato só consegui criar algum programa útil usando Java - que é a linguagem na qual eu desenvolvo - após isso ter acontecido.

O objetivo deste e-book é te ajudar, compartilhando com você os conceitos, da maneira como entendo ser a forma mais didática para se aprender Java e Orientação a Objetos. Espero realmente que você passe a entender. Mas não tenho como lhe garantir, porque em todos os anos que ministrei aula, em todas as turmas, sempre existiram alunos que de uma forma ou de outra acabaram não aprendendo os conceitos que eu ensinava.

Hoje eu trabalho como desenvolvedor de software, e utilizo Java como principal linguagem de programação. E a verdade é que aprendi utilizando os métodos e conceitos que passarei a apresentar à você a partir do próximo capítulo para facilitar a minha vida como programador, bem como otimizar a criação de códigos. Por isso sei da importância desse material para quem quer aprender essa poderosa linguagem de programação.

Te convido agora, a embarcar comigo nessa jornada, para entender esses conceitos e desmistificar mitos sobre a dificuldade em se utilizar esse tipo de programação.

3. Se preparando para começar a aprender

Sim! Você precisa já ter conhecimentos básicos para iniciar o aprendizado em Java. Se você já é programador em outras linguagens, com certeza você já possui habilidades com raciocínio lógico. Este é um ponto muito importante para quem quer ser programador, antes de aprender qualquer conceito. E eu aprendi que o raciocínio lógico pode ser exercitado, e melhorado com esse exercício. É como um músculo. Quanto mais você exercitar, mais forte ficará. Existem exercícios de raciocínio lógico para que você exercite isso em você. Esse é um pré-requisito para quem quer ser programador seja na área que for. Além disso, se você já é programador já tem idéia de algumas instruções como instruções de controle (if, else...), e repetição (while, for..).

Isso é bom sim, mas por outro lado, se você está começando a aprender a programar agora, e quer ir de cara programar em Java, não existe nada que te impeça disso também. O fato é que você precisa antes de partir para o Java propriamente dito, entender de algoritmos. É importante você saber como um programa de computador funciona, e um bom curso de algoritmo vai te ajudar a esclarecer isso para você.

Uma das grandes vantagens do Java é que ele é uma linguagem de programação multi-plataforma, e com isso é possível que você a utilize em diversos sistemas operacionais como Windows, Linux e/ou MacOS. Então, independente do sistema operacional que você possui afinidade, você pode utilizar Java, e o mais interessante é que uma vez desenvolvida uma aplica-

ção, você poderá executar a mesma compilação dela em qualquer um desses sistemas operacionais, graças a Máquina Virtual do Java sobre a qual os softwares que você criará são executados.

Com esses itens resolvidos, você estará apto a começar a aprender Java, seguindo o que eu chamo nesse e-book de caminho facilitado de aprendizagem.

4. Caminho facilitado

Aprender Java pode ser complexo, pois são muitos conceitos, muitas estruturas para conhecer e saber onde e quando usar. Mas isso é um bom sinal, pois uma linguagem com tantas estruturas, é também uma linguagem com muito recurso a ser utilizado. Seja o que você precisar fazer, desde escrever uma palavra na tela do seu computador até manipular um equipamento através de uma porta de comunicação do seu computador, enviar uma mensagem pela internet, executar tarefas complexas, tudo é possível com as bibliotecas existentes hoje no mercado.

Costumo dizer que é tanta coisa que chega a ser um mar de conhecimento. Mas para desfrutar desse mar, você precisa aprender a nadar nele. E o objetivo desse e-book como já expus anteriormente, não é te ensinar de fato, mas te mostrar uma ordem pela qual entendo que seria melhor que você aprendesse essa linguagem.

Um aprendizado guiado, te facilitará no aprendizado, pois você aprenderá conceitos de forma gradativa, e irá conforme o tempo passa, fazer com que esses conceitos fiquem cada vez mais fixados em sua mente.

Uma das maneiras que você pode adotar para estudar Java é ir acompanhando esse e-book, e realizando pesquisas na internet, em outros e-books, ou então, em um curso online onde você possa estudar na ordem que desejar os assuntos, e ir conhecendo os recursos conforme esse e-book for avançando a cada capítulo.

Meu objetivo em cada capítulo será basicamente te dar uma explicação breve do significado de cada tópico, e te explicar o que você deve aprender, e porque deve aprender.

Então, a partir deste ponto, iniciaremos a nossa jornada nesse aprendizado. Espero que ao utilizar este e-book, você possa guiar seus estudos de Java de uma forma produtiva e eficiente. Então, vamos lá!

5. Aprender Orientação a Objetos

Programação Orientada a Objetos (também conhecida pela sua sigla POO) é um modelo de análise, projeto e programação de software baseado na composição e interação entre diversas unidades chamadas de ‘objetos’.

Não é possível você querer aprender uma linguagem de programação que é em sua essência orientada a objetos, sem antes entender esses conceitos. Na prática, você melhorará o entendimento desses conceitos quando estiver programando, mas se aventurar a começar a escrever códigos em Java sem antes saber o que são objetos, classes, herança, atributos, métodos, pode ser algo que te atrapalhe, ou até te impeça de saber o porque você está fazendo determinadas etapas na programação.

Existem bons livros para que você possa aprender Orientação a Objetos, mas esses livros costumam ser muito completos e complexos, e dão poucos exemplos práticos. Pensando nisso, e sabendo da importância desse assunto, e também por ter tido uma excelente experiência quando aprendi Orientação a Objetos, resolvi somente sobre este assunto incluir um capítulo que abordará esses conceitos, para que você possa ser um excelente programador Java.

6. Programação Orientada a Objetos

Existem ao todo 4 paradigmas de programação, além da Programação Orientada a Objetos, temos a programação imperativa, funcional e a lógica. Com o objetivo de te fazer aproveitar ao máximo o seu tempo para aprender o que realmente importa para você agora, focaremos neste e-book somente no Paradigma de Programação Orientada a Objetos. Este paradigma tam-

bém é muito conhecido como POO, e a partir de agora utilizaremos essa sigla sempre que nos referirmos a esse termo.

O POO trata-se de um modelo de análise, projeto e programação de programas de computador, baseado na criação e comunicação entre diversas unidades chamadas de objetos. Existem ferramentas pelas quais é possível modelar um programa de computador neste paradigma, projetar um software e muitas linguagens de programação, senão a maioria, são multi-paradigma com suporte à POO. Alguns exemplos são Java, C#, Python, Object Pascal, Ruby entre outras.

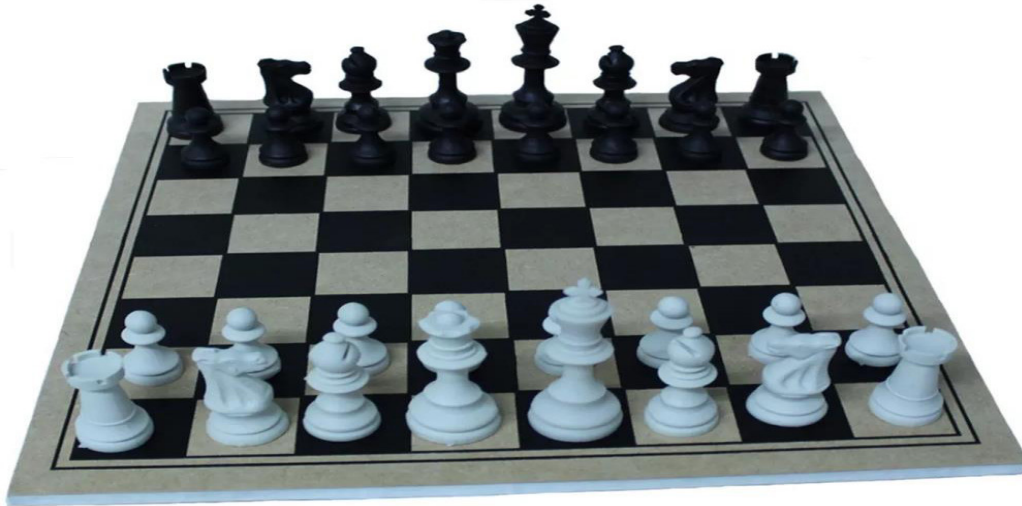
Ao final deste e-book encontra-se um glossário com todos os termos que envolvem esse paradigma, e uma explicação técnica do que significa. Porém, posso lhe garantir, que somente com essa explicação, sem um exemplo prático correto, você não entenderá o significado de cada tema.

Se olharmos na internet existem muitas maneiras de se ensinar orientação a objeto, mas no intuito de lhe ajudar a aprender de uma forma mais eficaz, irei utilizar exatamente o mesmo exemplo que foi utilizado quando eu de fato aprendi esses conceitos. Para mim funcionou, espero, sinceramente que para você, possa também servir. Decidi fazer algumas adaptações na ordem em que os conceitos geralmente são apresentados, e imagino que isso seja um grande diferencial para que você entenda os conceitos.

ESTUDO DE CASO: JOGO DE XADREZ

Neste e-book, utilizaremos somente um estudo de caso para exemplificar os conceitos apresentados. Talvez isso possa lhe parecer pobre da minha parte, mas posso lhe garantir que ao final deste e-book, você será encorajado a me enviar um email agradecendo a minha decisão de não divagar por outros exemplos. Confie em mim, em quem já sofreu para aprender orientação a objetos.

Para poder lhe apresentar os conceitos e explicar cada um, utilizaremos então um Jogo de Xadrez. Gostaria que você observasse com atenção este Jogo de Xadrez abaixo, pois é ele que será nosso suporte a partir de agora.



CONCEITOS BÁSICOS

Escolhi dividir os conceitos em dois níveis, para que você possa ter um intervalo proposital entre um e outro. Ao final deste quarto capítulo, irei sugerir alguns exercícios que recomendo fortemente que você realize. Será muito bom para o seu aprendizado.

CLASSE

Desde muito novo eu sempre me perguntei como as coisas são feitas. Esse inclusive foi o motivo pelo qual eu ingressei na programação. E você? já parou para pensar como um jogo de xadrez, igual ao da foto que lhe apresentei é fabricado?

Para facilitar, neste momento não vamos pensar no jogo todo, mas apenas nas peças. Para se criar essas peças de Xadrez que são, nosso exemplo, de plástico, utiliza-se um molde parecido com a imagem abaixo:



Neste momento te convido a se concentrar neste molde. Ele possui algumas funções entre as quais podemos destacar:

- **Ele define o formato, tamanho e peso da peça**
- **A partir de apenas um molde, podemos fazer quantas peças precisarmos.**
- **Esse molde foi desenvolvido por alguém, que o criou, pensando na peça, pois a partir do molde as peças seriam criadas.**

Na POO, esse molde é chamado de classe. Quando criamos uma classe, fazemos ela pensando na peça. E a partir dessa classe, podemos criar quantas peças precisarmos.

É a classe que vai definir o que vai ser o objeto e como ele vai ser, mas ela não é o objeto propriamente dito.

Não é possível falar mais sobre a Classe antes de falarmos do próximo conceito. Por isso, retomarei na sequência a explicação de uma forma mais completa.

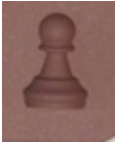





OBJETO

No jogo de xadrez, temos 32 peças que são únicas. Apesar de termos algumas peças idênticas, como por exemplo 8 peões iguais da mesma cor, não podemos dizer que temos 1 peão, e sim 8, isso é óbvio!

Quando trabalhamos com POO, a princípio não conseguimos fazer nada só com o molde, a não ser que a partir do molde nós criemos peças, que seriam então os nossos objetos. (Na verdade, existe uma possibilidade com o molde, mas para não te confundir, falaremos sobre isso mais a frente).

Cada peça é um objeto único, individual. Mas todos foram feitos a partir do mesmo molde. Então, seguindo essa analogia, nós conseguimos criar vários objetos a partir de uma mesma classe.

Toda vez que criamos um objeto a partir de uma classe, dizemos que estamos criando uma instância, que seria a existência desse objeto. No nosso exemplo, temos 16 instâncias de peças de xadrez do tipo peão, criados a partir de uma única classe (um único molde) e 4 instâncias de peças de xadrez do tipo torre com apenas uma classe (um modelo).

Classe	Objetos
	 
	 

Na POO podemos criar quantos objetos quisermos a partir de uma classe.

ATRIBUTO

Se você observar a tabela logo acima, analisando a linha dos peões, vai perceber logo de cara que alguns peões são diferentes uns dos outros. Um exemplo é a cor: 8 deles são brancos e 8 são azuis. Então, sabemos que cada peça dessas é um objeto, e que eles possuem cores diferentes.

Mas o que seria a cor neste caso? A cor é uma característica dessa peça, desse objeto. Então, apesar de serem objetos que são da mesma classe (do mesmo tipo), cada um deles é único e possui as suas próprias características.

Além disso, podemos pensar na peça em jogo. Cada uma dessas peças fica localizada em uma posição no tabuleiro, correto? Então a localização da peça também é uma característica dela. E se a peça for eliminada pelo adversário? Percebemos que a peça pode estar em jogo ou não. Tudo isso são características dessa peça, ou podemos dizer que são atributos desses objetos.

Na classe definimos que os objetos terão essas características, e quando criamos o objeto, cada um pode ter uma característica diferente, e essas características, podem mudar, e mudam, durante a execução de um programa.

Com este conceito, podemos iniciar a formular a nossa classe Peça:

Classe: Peça de Xadrez	
Cor	Branco/Preto
Em Jogo	Sim/Não
Linha no Tabuleiro	1 a 8
Coluna no Tabuleiro	1 a 8

MÉTODOS/COMPORTAMENTOS

Um jogo de Xadrez não seria um jogo se as peças não pudessem ser mudadas de lugar, não é mesmo? Então, nós podemos dizer que as peças desse jogo possuem o comportamento de poderem se mover, ou então serem movidas por alguém. Outra possibilidade é que uma peça pode eliminar outra.

A esses comportamentos damos o nome de método. Então um objeto pode ter métodos, eles podem executar esses métodos. Mas para que os objetos possuam esse método e saibam como executar a função, precisamos criar esse método e definir o que ele fará na classe.

Mas se em uma linha de comando eu chamar um método dizendo que uma torre branca deve se movimentar, eu preciso informar qual será o destino dessa peça. Se a peça for eliminar uma peça, ela precisa saber qual a peça deve eliminar.

Então quando criamos a classe, precisamos definir que será necessário informar esses parâmetros quando criamos esses métodos.

A nossa classe ficaria dessa forma:

Classe: Peça de Xadrez	
Cor	Branco/Preto
Em Jogo	Sim/Não
Linha no Tabuleiro	1 a 8
Coluna no Tabuleiro	1 a 8
+ Movimentar (Nova Linha, Nova Coluna)	
+ Eliminar Outra Peça (Peça a ser eliminada)	

ABSTRAÇÃO

Perdão por só te falar isso no final deste capítulo, mas entendi que esse seria o melhor momento para lhe falar sobre a abstração, pois se tivesse falado antes, você focaria na abstração e acabaria deixando de lado os conceitos que relacionei. Entretanto a partir de agora não é mais possível continuarmos sem que você entenda de fato o que é abstração.

Tudo que eu fiz desde que comecei a falar com você, desde que lhe mostrei o tabuleiro de Xadrez até este momento, foi abstrair da vida real - no caso de um jogo de xadrez - informações para que possam ser modeladas em um sistema que poderá ser programado orientado a objetos.

A técnica de abstrair essas informações precisa ser exercitada! Só assim você conseguirá modelar suas classes e objetos de maneira tal que elas sejam eficientes e você possa fazer com que sua aplicação seja bem organizada e fácil de programar.

Ao final deste capítulo, disponibilizarei alguns exercícios sobre abstração. E aconselho fortemente que você antes de prosseguir com seus estudos, faça-os, e refaça-os, até que consiga chegar no nível de abstração mais próximo da realidade.

Uma dica que posso dar, é que você deve abstrair os dados pensando no objeto da vida real, mas também não pode desconsiderar as funções que o seu sistema que será programado irá executar.

Por exemplo, não é intuitivo dizer que a peça está em uma posição de linha e coluna no tabuleiro. Mas se eu penso que preciso programar isso, e fazer com que elas se movimentem, logo lembro que precisarei trabalhar com coordenadas (posições mesmo), e que preciso dessa informação armazenada em cada objeto. Por isso, coloco esses atributos nas classes para que o objeto possam ser controlados dessa maneira.

EXERCÍCIOS DE FIXAÇÃO - LISTA 1

1. Relacione quatro exemplos de objetos do mundo real que de alguma forma se relacionem entre si.
2. Relacione os atributos que você consegue abstrair dos quatro objetos relacionados anteriormente.
3. Quais são os comportamentos desses objetos? Defina em cada objeto os seus devidos comportamentos.
4. Escreva quais são as semelhanças que sempre serão iguais em todos os objetos, se é que elas existem.
5. Escreva as diferenças entre objetos. O que um faz, e o outro não. O que é diferente tanto nos atributos como nos comportamentos. (Esse exercício é para lhe preparar para o nosso próximo capítulo).

CONCEITOS AVANÇADOS

Decidi separar os conceitos em básicos e avançados, mas apenas por questões didáticas. Mas a verdade é que todos esses conceitos são conceitos de POO. Nenhum é mais fácil ou difícil, e nem mais ou menos importante. Todos os conceitos são de muita importância.

Agora apresentarei dois conceitos dos quais utilizo constantemente em meus códigos, que fazem com que eu economize tempo, reaproveite muito código, e ganhe tempo no desenvolvimento.

HERANÇA

Neste momento eu gostaria de te instigar a fazer algumas análises e sinceramente responder as seguintes perguntas:

- Quantas peças existem no tabuleiro de xadrez?
- São todas iguais?
- Quais são as diferenças de uma para outra?
- Quais são as semelhanças?

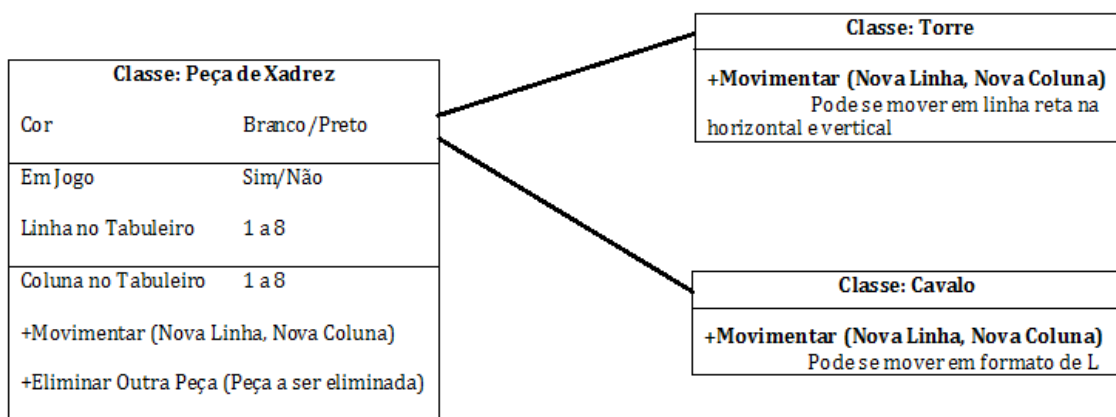
Bom, todas as 36 peças de Xadrez, são peças, correto? A verdade é que elas possuem algumas semelhanças, algumas diferenças, e algumas variações entre elas.

Vamos comparar duas peças: O cavalo e a torre. Não sou um especialista em xadrez, mas qualquer pessoa sabe que ambas as peças podem executar movimentos, e isso é algo que elas tem em comum, mas o tipo de movimento que um cavalo pode fazer é diferente do movimento que a torre pode executar. A maneira como a peça se movimenta deve ser especificada no método ou comportamento **movimentar(Nova Linha, Nova Coluna)**.

Bem, se eu sei que os tipos de movimentos permitidos para cada peça são diferentes, e que esses movimentos devem ser escritos dentro da classe que define a peça, então eu teria que fazer uma classe para a Torre e uma para o Cavalo, correto? Definitivamente não! Veja bem: Ambas são Peças de Xadrez. Elas possuem características em comum. E somente algumas diferenças em determinados comportamentos e atributos.

Logo, a POO fornece um recurso para que possamos programar isso de forma intuitiva, evitando que tenhamos que reescrever o mesmo código mais de uma vez. Esse recurso que a POO oferece se chamada herança. Quando falo em herança, logo vem a mente herdar algo de alguém. Você talvez tenha pensado em dinheiro, mas gostaria que focasse por exemplo na herança genética que você recebeu de seus pais quando nasceu.

Existem determinadas coisas em você que são como são em você, porque elas assim também são no seu pai. Assim como em POO podemos ter uma classe que herda de outra as suas características. Quando a herança acontece, ela herda tudo da classe pai, mas podemos mudar algumas coisas específicas na classe filha. No nosso caso ficaria assim:



Analisando a imagem acima, percebemos que possuímos uma definição genérica de peça de xadrez que possui todos os elementos em comum entre todas as peças que podem existir.

Ao observar as classes Torre e Cavalo, percebemos que nelas apenas existe a especificação de como deve ser realizado o movimento de cada objeto que for criado a partir dessas classes. Mas é importante observar que como essas classes possuem uma herança da classe Peça de Xadrez, tudo que existe na classe pai que é a Peça de Xadrez existe automaticamente nas classes filhas.

Então, se eu criar dois objetos do tipo cavalo, a partir dessa classe cavalo, esses dois objetos possuirão os atributos em jogo, cor, posição, e etc.. e todos os comportamentos também. Somente o comportamento movimentar que será realizado conforme está escrito na peça filha, porque ele foi reescrito nessa classe.

Então, se eu percebo que todas as peças de xadrez possuem um atributo novo que eu não havia percebido antes, eu insiro esse atributo na classe pai, e todas as filhas possuirão esse atributo automaticamente.

QUANDO UTILIZAR HERANÇA

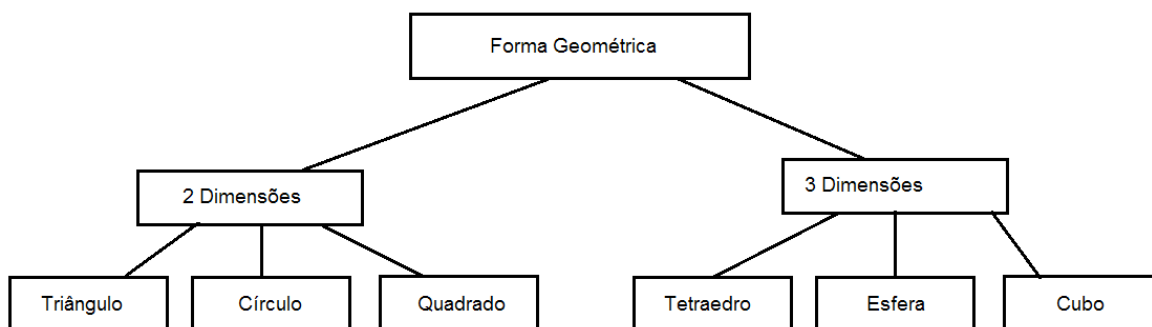
Geralmente o conceito de herança será utilizado em casos onde você vai perceber elementos que são em geral iguais, mas possuem variações de um determinado tipo para outro.

Alguns exemplos de classe pai e filhas poderiam ser:

Classe Pai	Classes Filhas
Conta Bancária	Conta Corrente, Conta Poupança, Aplicação
Forma Geométrica	Círculo, Triângulo, Retângulo, Esfera, Cubo
Pessoa	Pessoa Física, Pessoa Jurídica

HERANÇA EM MAIS E UM NÍVEL

Vale lhe explicar que a herança pode acontecer em quantos níveis forem necessários. Como por exemplo você é filho do seu pai, e neto do seu avô, assim também podemos ter classes herdando umas das outras quantos níveis forem necessários. Por exemplo:



HERANÇA MÚLTIPLA

Em POO existe um conceito de herança múltipla, onde uma classe herda de mais de uma classe pai. Mas é preciso cuidado, pois algumas linguagens de programas não implementam essa função da POO. O Java por exemplo não permite fazer isso.

POLIMORFISMO

Voltemos ao nosso jogo de Xadrez. Sabemos então, que se programarmos esse jogo, teremos diversas classes como Torre, Peão, Cavalo, etc. Então, deveríamos também ter uma classe chamada Tabuleiro, que precisará ter as 64 posições, e no qual também poderemos dar a opção de fazer uma jogada.

Para programar na classe Tabuleiro o comportamento jogada, nós precisamos passar alguns parâmetros para esse comportamento, como por exemplo a peça que vai ser jogada, e também a nova posição dela.

Quando vamos definir esse comportamento, nós precisamos dizer a Classe tabuleiro que quando esse comportamento for invocado um objeto precisa ser passado para ele, e a posição linha e coluna que seriam dois números. Como nós utilizamos a herança, e todas as nossas peças do nosso jogo herdam da classe PeçaDeXadrez, nós podemos dizer ao método jogada do tabuleiro que o que será passado quando o comportamento for acionado é uma PeçaDeXadrez, e ele vai aceitar que você passe um objeto Peão, um objeto Torre, ou qualquer objeto que tenha sido criado a partir de uma das classes que herdam de PeçaDeXadrez.

A verdade é que como ele sabe que o objeto que está sendo informado foi criado a partir de uma classe que herda a classe da qual ele espera o parâmetro, então ele transforma esse objeto filho na classe pai, e ele aceita que esse objeto seja passado como parâmetro.

A esse poder de informar um objeto do tipo Torre a um método que espera PeçaDeXadrez é dado o nome de polimorfismo. Ele é muito útil, pois não é necessário você programar vários comportamentos do tipo:

- `Tabuleiro.jogada(Torre, NovaLinha, NovaColuna);`
- `Tabuleiro.jogada(Peão, NovaLinha, NovaColuna);`
- `Tabuleiro.jogada(Cavalo, NovaLinha, NovaColuna);`

Ao invés disso, que seria um retrabalho e também pouco intuitivo, você pode se utilizar da reutilização de código, e criar o método da seguinte forma:

- **`Tabuleiro.jogada(PeçaDeXadrez, NovaLinha, NovaColuna);`**

Se você analisar, esse método em destaque acima espera um objeto do tipo `PeçaDeXadrez`, e uma Torre além de ser um objeto do tipo `Torre`, é também uma `PeçaDeXadrez` devido a herança, e assim a linguagem de programação vai aceitar esse objeto sem nenhum problema.

EXERCÍCIOS DE FIXAÇÃO - LISTA 2

1. Analise como a herança pode ajudar na reutilização de código, economizar tempo durante a criação de um software e evitar erros.

2. O mundo das formas é muito mais abrangente que as formas incluídas na hierarquia tratada no item 5.3 deste capítulo. Anote todas as formas que você pode imaginar, e transforme-as em uma hierarquia mais completa do que a apresentada, com o maior número possível de níveis.

7. Escolha uma IDE de desenvolvimento

Me lembro bem das minhas primeiras aulas de Java, onde o meu professor me obrigava a escrever o código no Bloco de Notas, e depois salvar e ir pelo Prompt de Comando do Windows e compilar e executar manualmente. Eu entendo que o motivo dessa escolha por parte dele foi para que os alunos fossem obrigados a escrever o código, visto que se você utiliza uma IDE de desenvolvimento, muitas vezes os códigos já vem pré escritos para nós.

Mas sinceramente, eu gostaria de te incentivar a escolher uma IDE de desenvolvimento para iniciar a escrever códigos em Java. Um dos principais motivos é que ela é muito mais intuitiva. Ela te mostra em tempo real os erros que você está cometendo enquanto está digitando o código, completa automaticamente o código para você quando você solicita, organiza melhor o seu projeto quando este possui vários arquivos, testa, compila e executa seu código, lhe fazendo em resumo ganhar muito tempo.

Mas se você quer um conselho, gostaria de lhe encorajar, a no início, utilizar a IDE sim, mas como um editor de textos. Não se beneficie dos códigos que são gerados automaticamente pela IDE. Se você quer aprender realmente, e principalmente saber o que está fazendo, utilize uma IDE, e quando criar uma classe, selecione todo o conteúdo que vem digitado no arquivo novo, e o exclua. O processo de entender como as coisas funcionam te ajudará muito, e evidentemente você deve escolher se quer realmente aprender o que está fazendo ou não. Isso fará toda a diferença no futuro.

Dito isso, posso lhe sugerir duas IDE's de desenvolvimento que são muito boas. Eu particularmente utilizo o Netbeans, do qual pode ser feito o download em seu site oficial www.netbeans.org. Outra IDE que é amplamente utilizada é o www.eclipse.org. Não creio que exista uma IDE melhor que a outra entre essas duas opções, mas entendo que a que você escolher para aprender, acabará utilizando em seguida na sua carreira.

Além da IDE, você precisará ter instalado em seu computador - na verdade você precisa instalar antes que a IDE - uma compilação JDK do Java. Esse JDK pertence atualmente a Oracle e pode ser baixado no site oficial www.oracle.com/technetwork/pt/java/javase/downloads/.

Neste momento não foque muito em aprender os recursos da IDE. Lembre-se você só estará utilizando ela como um editor de textos melhorado.

8. Classes, Atributos, Métodos e Objetos

No Java, todo código é escrito obrigatoriamente dentro de uma Classe. Por isso, você precisa entender como que é a sintaxe para se definir uma classe em java. Essas classes possuem atributos, e esses atributos são de algum determinado tipo, que podem ser outras classes que você cria ou então já existentes na linguagem, ou então tipos primitivos, os quais também te aconselho a aprender agora.

Neste início aprenda também sobre a classe Scanner para poder interagir com o usuário, como ler números pelo teclado ou coisas do tipo. E também será necessário que você aprenda a escrever valores na tela do seu computador, e é na tela do prompt de comando mesmo. Nada de interface visual por enquanto.

Você precisa aprender também como criar objetos a partir de uma classe, como criar instâncias de objetos novos, e utilizar eles. Sugiro que ao menos você crie um projeto com duas classes, e que uma utilize a outra através da criação de um objeto e posterior invocação de um método seu.

9. Instruções de Controle

Se você já é programador, ou não é, mas já estou algoritmo como sugeri no início do e-book, você deve saber da importância e de quanto utilizamos as instruções de controle em programas de computador.

Aprender a sintaxe dessas instruções de controle é importante, e pode ser feito neste momento. O Java conta com instruções de controle como (if), (if... else...) e (switch.. case).

10. Coleções, Arrays e ArrayLists

Toda linguagem de programação fornece o recurso de se trabalhar com vetores. Mas o Java possui classes específicas para que você possa criar coleções complexas, as quais por si só terão funções muito interessantes embutidas nelas, como ordenação, inserção e remoção de itens, por exemplo.

É possível também criar um array simples, de tamanho pré-definido, e o utilizar no código. Tanto as coleções como os arrays podem ser de objetos, e somente os arrays simples podem ser de tipos primitivos.

11. Instruções de Repetição

Assim como em outras linguagens, fazer uma tarefa repetidas vezes até que uma condição seja satisfeita é possível no Java. O interessante é que como o Java trabalha com coleções, existem estruturas de repetição nas quais é possível varrer toda a coleção.

Essas não são instruções comuns de se ver em uma linguagem de programação funcional por exemplo, e é um recurso que facilita a codificação e deve ser utilizado na programação Java.

12. Exceções

Um dos maiores erros quando se começa a programar em Java é não tratar as exceções. Eu mesmo cometi esse erro, e demorei a entender como as exceções funcionavam, e como eu deveria tratá-las. A verdade é que enquanto eu não aprendi esse conceito, e não ajustei os códigos que já havia desenvolvido e distribuído aos meus clientes, por muitas vezes eles me ligavam

falando que simplesmente um botão no sistema não funcionava. Quando eu ia analisar o código, era uma exceção não tratada que estava sendo acionada.

13. Manipulação de Arquivos

Acessar, Ler, Gravar, Criar, Excluir e manipular seja da forma que for arquivos em disco é algo que é importante e por muitas vezes necessário. Você aprenderá a fazer com que o seu aplicativo Java se conecte a um banco de dados, mas por exemplo, as configurações de acesso a esse banco de dados podem ser salvas em um arquivo no computador, o que facilita a parametrização da sua aplicação. O Java possui classes próprias para esses fins. Várias Classes, e várias formas de fazer isso. E em muitos momentos você irá se deparar com essa situação onde existem várias formas de fazer algo. Sugiro que você faça da maneira que for mais confortável para você, e/ou a equipe que está desenvolvendo com você se este for o caso.

14. Threads

A thread é uma linha de execução. Na prática, quando teu código está sendo executado, é como se o cursor de execução seguisse um caminho, e em cada momento ele está em um local no seu código. Mas o Java é uma linguagem Multi-Thread, e isso significa que ele em teoria pode ter vários cursores de execução. Por exemplo, em um software enquanto o usuário digita os cadastros em uma tela do seu sistema, uma outra Thread pode executar uma tarefa em segundo plano em sua aplicação. Quando trabalhamos com aplicações gráficas ou web, e sistemas executam tarefas demoradas, ou que podem ter um tempo de execução muito longo, é comum iniciar um Thread nova para essas tarefas.

15. Aplicativos Gráficos Swing

Neste ponto, você já está familiarizado com a codificação na linguagem Java, e é possível começar a trabalhar com elementos gráficos caso deseje se aprofundar no desenvolvimento para aplicativos desktop. Hoje em dia, muitas das aplicações tem sido transportadas para a nuvem, e na verdade, talvez programar para desktop não seja o seu foco de aprendizado, se você achar que talvez seja mais relevante, pode focar em aprender Java para a Web ao invés de Java Desktop com Swing.

16. Java para Web

Atualmente as aplicações voltadas para a web estão em alta. O Java possui desde muito tempo atrás recursos robustos para desenvolvimento web. Entre os recursos que você deve estudar para desenvolver aplicativos web com java são Servlets, Páginas JSP, e JSF.

Existem diversos frameworks para agilizar seu desenvolvimento para web, e é importante estudar esses frameworks que são ótimos para agilizar o desenvolvimento. Um detalhe importante a ressaltar é que com exceção do capítulo 15, todos os demais tópicos apresentados neste e-book podem e serão utilizados para o desenvolvimento web com Java.

17. Considerações Finais

Todas as informações apresentadas neste e-book foram dispostas de uma maneira a facilitar o aprendizado e a obtenção de conhecimento na programação utilizando a linguagem de programação Java.

Espero ter aberto a sua mente para a compreensão do que tentei expor durante essas páginas. E te aconselho fortemente a continuar explorando esses conceitos, para que você possa se aperfeiçoar, e se tornar um programador melhor a cada linha de código escrita.

Gostaria de te convidar a acessar o meu site **www.academiadaprogramacao.com.br** pois lá você terá a sua disposição diversos materiais gratuitos para estudo e aprendizado de programação.