



Se você está aqui e não faz a mínima idéia de como começar, faça esta trilha tomando este ponto de partida.
Ao longo da jornada você vai entender todas as necessidades e principalmente por que você está estudando tudo isso!

Trilha Algoritmos/Lógica e Java

O objetivo deste guia é tornar sua rotina de estudos mais auto-suficiente possível para que você obtenha constância e evolua de forma consistente.

Lembre-se: não nos tornamos DEVs do dia para a noite. É um processo que exige e exigirá de você estudo permanente. O hábito é tão ou mais importante que a intensidade do momento de estudos.

Aqui você não terá apenas a sequência de tópicos, mas direcionamentos para exercícios e práticas deliberadas (repetições nas quais você sabe o que está evoluindo a cada passo)

As vezes será necessário repassar as aulas mais de uma vez, ok?

Por que estou aqui? → Como eu começo?

Antes de começar, veja estas 2 aulas só para você se ambientar... A idéia é que a gente comece a praticar Lógica usando uma ferramenta de apoio, beleza?

O que é programação, o que é um programa e qual seu propósito?
(Entrada -> Processamento -> Saída)

Entenda o que são variáveis, como armazenar variáveis e como exibir seus valores na tela

Vamos dar utilidade aos nossos programas, vamos fazer cálculos aritméticos com as variáveis

Conhecemos mais uma ferramenta "elegante" para alguns tipos de aninhamentos

Decisões começam a ficar mais complexas e com mais alternativas de testes

Ok, agora nosso modelo E->P->S começa a ter decisões

Aqui você já pode realizar exercícios disponíveis neste link aqui. Concentre-se apenas no tema "Seqüência de Instruções" (exercícios de 1002 a 1020)

dica importante: é fundamental que você saiba alguns aspectos relacionados a cálculos, como porcentagem, regra de 3, etc

Aqui você também já tem uma lista imensa de exercícios para fazer (alguns simples, outros bastante trabalhosos)
Concentre nos exercícios do tema "Decisões" (do 1035 ao 1052)

Agora veremos o real motivo de programar: fazer coisas repetitivas
Para isso temos 3 ferramentas (enquanto/faça/para)

Legal, mas como juntar tudo?
Vamos entender?

E os algoritmos clássicos de média, soma, maior, menor?
Temos também!

concluiu?

Nesta etapa os exercícios aglutinam todo o conhecimento acumulado até aqui.
Não há fórmula única para resolver.
A seção dos exercícios é a "Repetições" (do 1060 ao 1164)

Uma observação: você vai perceber que ao longo da inclusão das ferramentas de decisão e repetições o nosso modelo de Entrada, processamento e Saída fica cada vez mais conciso e difícil de diferenciar cada componente. Você entra e/ou saídas atreladas a códigos de repetição e decisão. Então, no início, você consegue diferenciar as linhas de código e percebe "aqui é entrada, aqui é processamento, aqui é saída". À medida que os códigos ficam mais complexos, é a sua lógica que vai lhe permitir entender e diferenciar as etapas



Próxima Trilha

Fundamentos de Java. Vamos poder revisar todos estes elementos de lógica em uma linguagem mais robusta e também aprender novos conceitos, como Orientação a Objetos.



isi.flix

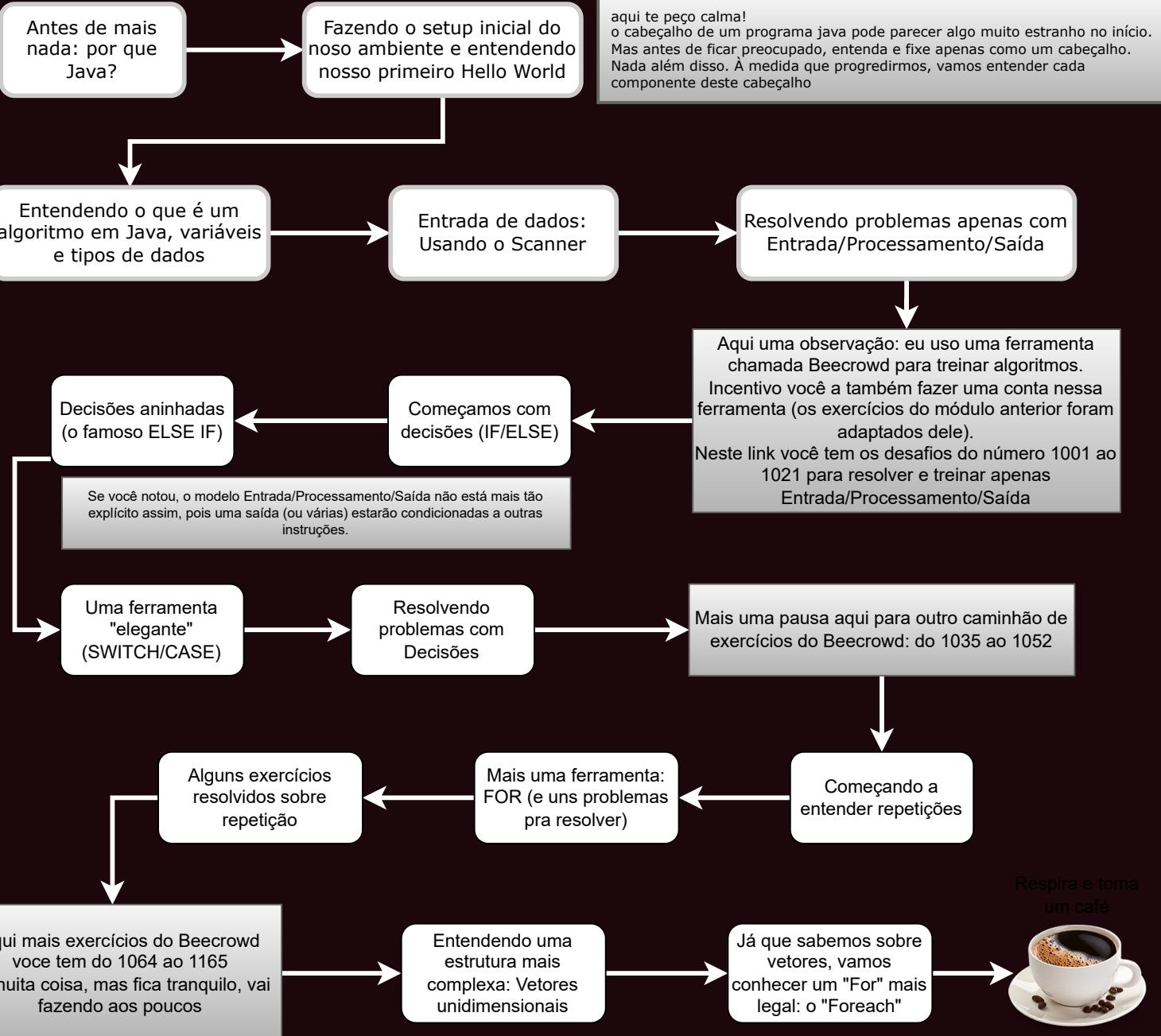


Aqui um pequeno "disclaimer" antes de começarmos. Por que Java?
Todo mundo adora criticar o Java, mas poucos sabem realmente que "quando a coisa aperta", é o Java que salva boa parte dos projetos (Netflix que o diga).
Bom, mas tem um vídeo sobre isso que eu gostaria que você assistisse para que enxergue um pouco meu ponto de vista (sem precisar chegar nas questões de quais empresas usam e porque).
Mas Java vai te dar muito (mas muito mesmo) subsídio para entender os principais conceitos de Orientação a Objetos que precisaremos daqui pra frente, ok? Então bora que a coisa tá só começando!



e agora?

Vamos dividir o módulo de JAVA em 2 grandes partes: A primeira vai ser uma "revisão" de lógica de programação (sim aquela vista anteriormente) porém agora com uma linguagem mais robusta (o Java) e depois toda parte avançada de Orientação a Objetos, beleza? Então bora!



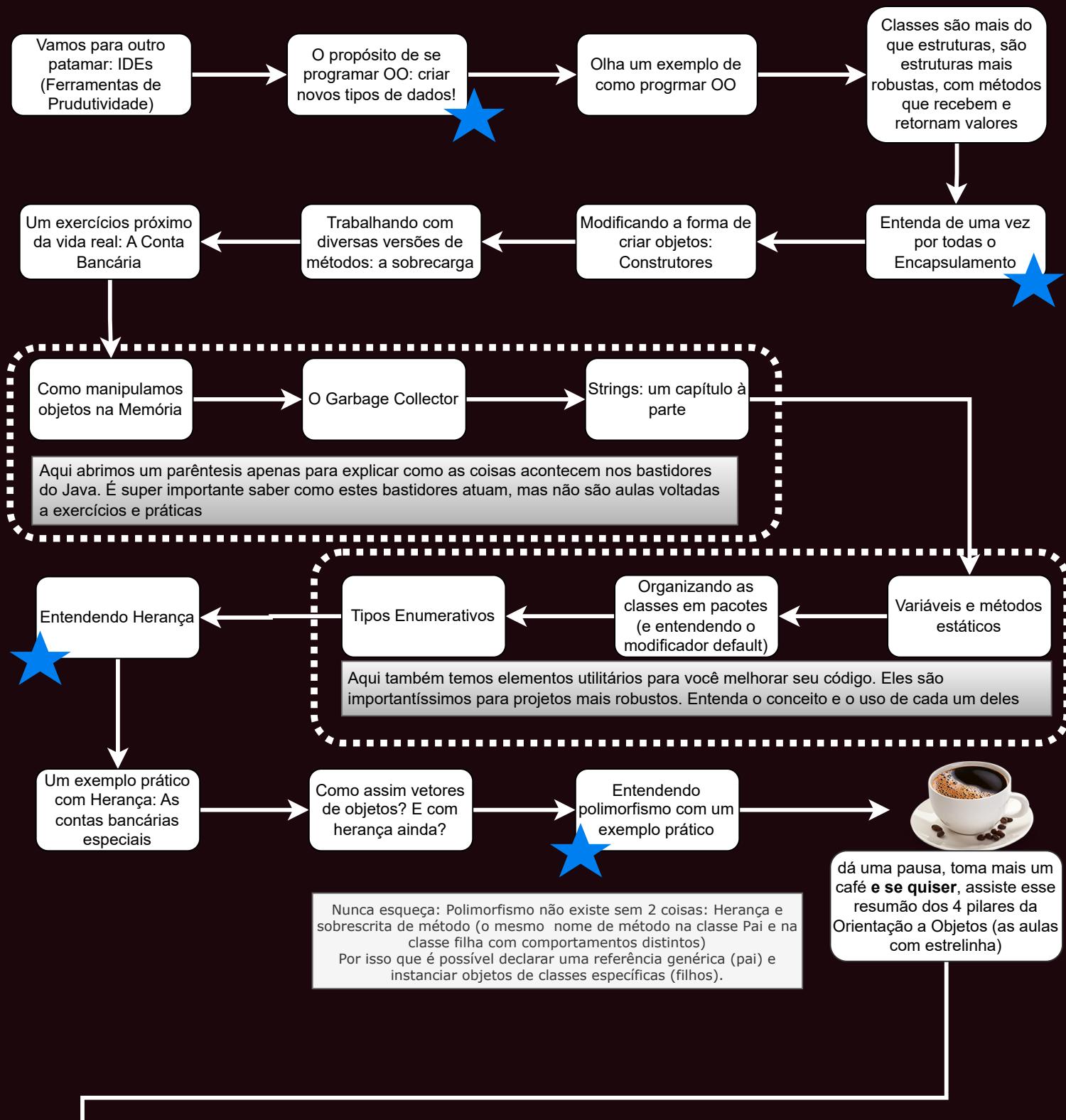
isi.flix



Olha só: pela cara do nosso amigo aqui do lado, já deu pra perceber que programar Java é muito mais legal do que parece, certo?

A idéia agora é começarmos a aprofundar os estudos em Orientação a Objetos, pois esse é o grande mérito que popularizou o Java (e que o Java também contribuiu de forma absurda para popularizar esta forma de desenvolver software). Então vamos começar a entender as coisas como elas realmente são

Note que algumas aulas estão marcadas com uma estrela. Elas são aulas que definem os principais pilares do Estudo de Orientação a Objetos (OO). Dê uma atenção especial a estes materiais, ok?



isi.flix

Aqui entramos em uma outra seara (mas ainda dentro do tema de herança e polimorfismo) que acredito que as coisas podem se encaixar (e muito). Se eu puder compartilhar uma experiência pessoal, meu entendimento de polimorfismo melhorou muito ao aprender classes abstratas. Dá uma conferida

Entenda o poder das classes abstratas para você poder referenciar de forma genérica sua hierarquia de classes

A Folha de Pagamento usando classes abstratas facilita demais a compreensão. Confere aí

Interfaces são ferramentas poderosas quando o quesito é "Desacoplamento". Lá na frente, na construção de APIs você vai usar muito isso!

Vamos fazer um exemplo que envolve classes abstratas e interfaces pra juntar tudão

Interfaces para dar a um objeto múltiplas "caras". Aqui você pode considerar o que muitos chamam de "herança múltipla"

Uma observação: A frequência de criação de classes que implementam múltiplas interfaces é algo que ainda é pequena (pra quem está no começo dos estudos). Está começando a se popularizar graças ao SOLID (mas à frente neste guia você terá um conteúdo sobre isso)

Agora entender Exceções é algo que vai levar seu código a outro patamar!

E como fazemos para criar nossas próprias Exceptions? Quais tipos? Verificadas ou Não Verificadas?

Dois recursos poderosos para exceptions: MultiCatch e o Try-With-Resources

Vamos dar uma espiada na API do Java? E ver algumas coisas que podem ser usadas?

Cansou de fazer FOR em Listas? Que tal usar STREAMS?

Vamos ver uns exemplos massa com Lambda

O divisor de águas do Java 8 em diante: Lambdas

Fechando uma parte do assunto: Arquivos (tanto texto quanto binários)

Café 1: Quer saber um pouco mais de SOLID? Dá uma olhada aqui



Café 2: Quer saber mais da API de Coleções do Java e de Streams? Olha aqui

isi.flix



Isidrooooooooooooo
acabou?

Calma!!! Ainda não necessariamente. Chegamos ao final de alguns dos principais fundamentos de Java? Sim
Isso significa que seu estudo terminou? Jamais!

Inclusive vou te indicar 4 materiais bacanas pra você evoluir um pouco mais sobre a linguagem e também sobre como
organizar as classes para o seu código ficar bem mais organizado e fácil de manter

Conheça a API Reflection do Java e
seja capaz de Hackear qualquer
objeto que você manipular

Conheça os principais padrões
de projeto que visam organizar
responsabilidades das classes
que você desenvolver visando
facilidade de manutenção e
evolução do seu software

Padrões Criacionais

Padrões Estruturais

Padrões
Comportamentais