

Roadmap Detalhado de Java: Básico ao Intermediário (1 Ano - 2h/dia)

Foco: Fortalecer lógica de programação e orientação a objetos com Java.



Plano Mensal Detalhado

Cada mês tem - 60 horas de estudo (2h/dia × 30 dias).



Método de Estudo Recomendado:

20% Teoria (vídeos, livros, documentação)

60% Prática (exercícios, projetos, desafios)

20% Revisão (anotações, flashcards, refatoração)

Use o GitHub para versionar seus projetos.



Mês 1-2: Lógica de Programação e Sintaxe Java

Tempo: 2 meses (120 horas)

Objetivo: Dominar algoritmos e estruturas básicas.

Conteúdo:

- ✓ Variáveis, tipos de dados e operadores (20h)
- ✓ Estruturas condicionais (if-else, switch) (20h)
- ✓ Loops (for, while, do-while) (20h)
- ✓ Arrays e manipulação de dados (20h)
- ✓ Métodos simples (parâmetros, retorno) (20h)
- ✓ Exercícios de lógica (20h)

Projetos:

- ◆ Calculadora no Terminal
- ◆ Jogo da Forca

Como Estudar:

Pratique no HackerRank ou Beecrowd (30 min/dia).

Revise conceitos com flashcards (ex: Anki).

Mês 3-4: Orientação a Objetos (Fundamentos)

Tempo: 2 meses (120 horas)

Objetivo: Entender classes, objetos, herança e polimorfismo.

Conteúdo:

- ✓ Classes vs Objetos (20h)
- ✓ Encapsulamento (private, getters/setters) (20h)
- ✓ Construtores e sobrecarga (20h)
- ✓ Herança e polimorfismo (extends, @Override) (30h)
- ✓ Exercícios de OO (30h)

Projetos:

- ◆ Sistema de Cadastro de Pessoas
- ◆ Simulador de Animais (herança)

Como Estudar:

Desenhe diagramas UML simples.

Refatore projetos antigos aplicando OO.

Mês 5-6: Estruturas de Dados e Tratamento de Erros

Tempo: 2 meses (120 horas)

Objetivo: Aprender Collections, Exceções e Generics.

Conteúdo:

- ✓ Listas (ArrayList, LinkedList) (20h)
- ✓ Conjuntos e Mapas (HashSet, HashMap) (20h)
- ✓ Exceções (try-catch, throws) (20h)
- ✓ Generics (List<T>, métodos genéricos) (20h)
- ✓ Exercícios com Collections (40h)

Projetos:

- ◆ Agenda de Contatos (CRUD em memória)
- ◆ Sistema de Estoque com Validações

Como Estudar:

Compare estruturas (ex: ArrayList vs LinkedList).

Use Debug para entender fluxos de exceção.

✦ **Mês 7-8: Programação Funcional e Arquivos**

Tempo: 2 meses (120 horas)

Objetivo: Dominar Lambda, Streams e manipulação de arquivos.

Conteúdo:

- ✓ Lambda Expressions (20h)
- ✓ Streams API (filter, map, reduce) (30h)
- ✓ Leitura/Escrita de arquivos (File, BufferedReader) (20h)
- ✓ Serialização JSON (Gson/Jackson) (20h)
- ✓ Exercícios de processamento de dados (30h)

Projetos:

- ◆ Filtro de Dados de Vendas (Streams)
- ◆ Exportador de Dados para JSON

Como Estudar:

Converta loops tradicionais para Streams.

Trabalhe com datasets reais (ex: CSV de produtos).

✦ **Mês 9-10: Threads e Banco de Dados (JDBC)**

Tempo: 2 meses (120 horas)

Objetivo: Aprender concorrência e SQL básico.

Conteúdo:

- ✓ Threads (Runnable, synchronized) (30h)
- ✓ SQL (SELECT, INSERT, JOIN) (30h)
- ✓ JDBC (Connection, PreparedStatement) (30h)
- ✓ CRUD com MySQL/PostgreSQL (30h)

Projetos:

- ◆ Sistema de Biblioteca com Banco de Dados
- ◆ Simulador de Download Multithread

Como Estudar:

Use Docker para subir um banco local.

Teste concorrência com tarefas longas (ex: processamento de imagens).

Mês 11: Testes Unitários (JUnit 5)

Tempo: 1 mês (60 horas)

Objetivo: Escrever testes automatizados.

Conteúdo:

✓ JUnit 5 (@Test, assertions) (20h)

✓ Testes para métodos e classes (20h)

✓ Mocking básico com Mockito (20h)

Projeto:

- ◆ Testes para o Sistema de Biblioteca

Como Estudar:

Pratique TDD (escreva testes antes do código).

Refatore projetos antigos adicionando testes.

Mês 12: Projeto Final e Revisão

Tempo: 1 mês (60 horas)

Objetivo: Consolidar conhecimentos em um projeto completo.

Atividades:

✓ Projeto CRUD Full (terminal + banco + testes) (40h)

✓ Revisão de tópicos fracos (20h)

Projeto Final:

- ◆ Sistema de Vendas

Cadastro de clientes/produtos

Registro de vendas com persistência

Relatórios em JSON

Como Estudar:

Documente seu projeto no GitHub (README.md).

Peça feedback em fóruns (ex: Reddit r/learnjava).

Recursos Recomendados

Livros: "Java: Como Programar" (Deitel), "Effective Java" (Bloch).

Cursos: Loiane Groner (Udemy), Java MOOC (Helsinki).

Pratique: Beecrowd, HackerRank, Codewars.

Checklist de Progresso

Ao final de cada mês, verifique:

Dominei os conceitos teóricos?

Completei os projetos propostos?

Refatorei código antigo?

Revisei anotações?

Com esse plano, em 1 ano você estará pronto para desafios profissionais em Java! 🚀