

```
;;=====;;
;; How to Design Programs, 2ª edição                                ;;
;; https://htdp.org                                              ;;
;;-----;;
;; Capítulos: Preface                                           ;;
;;           Prologue                                           ;;
;;-----;;
;; Resumo de estudo por: Abrantes Araújo Silva Filho           ;;
;;           abrantesasf@gmail.com                             ;;
;;=====;;
```

O foco do livro é em:

- Hábitos de BOA PROGRAMAÇÃO
- DESIGN SISTEMÁTICO de programas
- Programas SISTEMATICAMENTE DESENHADOS

Frase que resume o foco acima:

"O DESIGN de programas --- mas NÃO a programação --- tem a mesma função na educação liberal do que habilidades matemáticas e linguísticas."

O que é o DESIGN SISTEMÁTICO de programas? É o conjunto de 2 conceitos chaves: "receitas de design" e "refinamento iterativo".

- RECEITAS DE DESIGN
 - Para funções individuais:
 - 6 passos básicos: Receita de Design de Função (RDF)
 - Para programas completos:
 - GUI programs
 - Batch programs
- REFINAMENTO ITERATIVO
 - Lida com problemas complexos e multifacetados através da eliminação de detalhes não essenciais e o desenvolvimento de uma solução para o problema central; novas iterações (repetições) são realizadas para refinar a solução até que ela seja completa.

Divisão do livro:

- Parte I : conceitos fundamentais do design sistemático
- Intermezzo 1: sintaxe e semântica
- Parte II : dados compostos com número arbitrário
- Intermezzo 2: quotation e anti-quotation
- Parte III : abstração
- Intermezzo 3: escopo léxico e abstração
- Parte IV : refinamento iterativo
- Intermezzo 4: números decimais, natureza dos números
- Parte V : performance de programas
- Intermezzo 5: custo da computação
- Parte VI : acumuladores

6 Passos básicos: Receita de Design de Função (RDF)

1) DA ANÁLISE DO PROBLEMA À DEFINIÇÃO DOS DADOS

-> Analise um problema, tipicamente escrito em palavras:

Identifique a informação que deve ser representada e como ela é representada na linguagem de programação escolhida. Formule definições de dados e ilustre com exemplos.

2) ASSINATURA, DECLARAÇÃO DE PROPÓSITO, CABEÇALHO

-> Extraia e expresse sua essência, de modo abstrato:

Declare que tipo de dados a função desejada consome e produz. Formule uma resposta concisa para a questão: "a função calcula o quê?". Defina um esboço para a assinatura.

3) EXEMPLOS FUNCIONAIS:

-> Ilustre a essência com exemplos:

Trabalhe/defina exemplos que ilustram o propósito da função.

4) MODELO FUNCIONAL:

-> Estabeleça esboços e planos baseados na análise:

Traduza as definições de dados em um esboço da função.

5) DEFINIÇÃO DA FUNÇÃO:

-> Termine a função e avalie os resultados esperados:

Preencha as lacunas no modelo da função. Explore a declaração do propósito da função e os exemplos funcionais.

6) TESTE:

-> Revise o produto à luz de checagens e testes que falharam:

Articule os exemplos como testes e garanta que a função passa em todos eles. Descubra erros. Os testes complementam os exemplos e ajudam outros a ler e entender a definição da função em caso de necessidade (e essa necessidade sempre ocorre em qualquer programa).

Preparação:

- Instale a versão mais nova da Racket:

<https://racket-lang.org/>

- Use o HelpDesk! É a documentação HTML instalada junto com a Racket.

- Use o Steeper!

- Siga o "Prologue"!