

06. Modelos com Sequências

Princípios de Engenharia de Software (Texto em Elaboração)

Italo S. Vega

italo@pucsp.br

Faculdade de Estudos Interdisciplinares (FACEI)



PUC-SP

Pontifícia Universidade Católica de São Paulo



2022 Italo S. Vega

Sumário

| | | |
|----------|---|-----------|
| 6 | Modelos com Sequências | 4 |
| 6.1 | Sequências | 4 |
| 6.2 | Inversa de uma Relação | 9 |
| 6.3 | Funções e Expressões- λ | 11 |
| 6.4 | Resumo | 14 |
| 6.5 | Exercícios | 15 |
| | Referências | 17 |



Leslie Lamport, *Specifying Systems* — Thinking clearly is hard; we can use all the help we can get.

6 Modelos com Sequências



— Consigo **definir** a composição de funções, **observar** a sua implementação em Python e **construir** soluções compondo funções.

6.1 Sequências

Observando o seu modelo de alfabeto, Fubã conclui que se trata de uma função. Por algum motivo, ele se lembrou de uma estrutura conhecida por sequência:

Sequência: um conjunto ordenado de valores do mesmo tipo.

Representaremos sequências de símbolos conforme a seguinte notação. Dois parênteses angulados delimitarão os elementos de uma sequência, separados por vírgulas, como em $\langle a, b, c \rangle$. Fubã, então, utilizou uma sequência para modelar o alfabeto da Cifra de César.



— Construirei uma lógica partindo de um alfabeto de símbolos modelado como a sequência A :

$$A \triangleq \langle \square, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, \zeta \rangle$$

Conforme a notação, o par de parênteses angulados $\langle \dots \rangle$ indica uma sequência de valores. Por conseguinte, A corresponde ao conjunto contendo elementos como $1 \mapsto \square$ e $2 \mapsto a$. A abreviação possibilita usarmos o nome A no lugar do próprio conjunto. Fubã sabe que esta é apenas uma forma reduzida de representar a função:

$$A \triangleq \{1 \mapsto \square, 2 \mapsto a, 3 \mapsto b, 4 \mapsto c, \dots, 28 \mapsto \zeta\}$$



— Sequências são conjuntos!

Devido às características da Cifra de César, a ordem dos símbolos no alfabeto é importante e a noção de sequência captura esta ideia. O primeiro elemento está na posição de número 1, o segundo, na posição 2 e assim por diante. É possível que um elemento ocupe outras posições, pois sequências admitem repetições. No problema do Fubã, entretanto, isso não ocorre. A presença do símbolo $A(1) = A\ 1 = A_1 = \square$ é única na sequência A .

Cardinalidade Espec aproveita para lembrar que a quantidade de elementos em uma sequência será representada por “#” seguido do nome da sequência.

Tipos Outra consideração do Espec refere-se ao tipo da variável A .



— O tipo da variável A depende de outro tipo: SÍMBOLO.

Assim, Espec introduz o tipo SÍMBOLO, constituído pelos valores de um conjunto enumerado:

$$\text{SÍMBOLO} \triangleq \{\square, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, \zeta\}$$

Espec pretende marcar a diferença entre o conjunto que define o tipo SÍMBOLO e o tipo do valor atribuído à variável A . Conforme a definição de SÍMBOLO, é verdade que os valores a e m têm tipo SÍMBOLO, escrevendo-se:

$$a, m : \text{SÍMBOLO}$$

A variável A , por sua vez, possui o seguinte tipo funcional:

$$A : \text{seq SÍMBOLO}$$

Ou seja:

$$\langle \square, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, \zeta \rangle : \text{seq SÍMBOLO}$$



“Melhor entender valores do tipo sequência... Depois eu retomo este assunto...”

Pergunta (Modelagem com Sequências). *Contexto* Considere as sequências:

- I) $n = \langle 2, 4, 6, 6 \rangle$
- II) $m = \langle a, b, d, b, d \rangle$
- III) $t = \langle \text{FUBA}, \text{ESPEC}, \text{PROFE} \rangle$

Enunciado Assinale a alternativa contendo uma afirmação verdadeira:

- 1. $m_3 = b$.
- 2. $t_{n_1} = \text{FUBA}$.
- 3. $\#t = \#m = 3$.

4. $(n_2 = 4) \wedge (\#n = 4)$.

A partir da modelagem do alfabeto como a sequência A , as seguintes afirmações tornam-se verdadeiras na lógica do Fubã:

1. **dom** $A = 1..28$
2. $\{a, c.m\} \subset \mathbf{ran} A$
3. $\#A = 28$
4. $A(2) = A\ 2 = A_2 = a$

A notação $1..28$ apenas abrevia o conjunto $\{1, 2, 3, \dots, 28\}$.

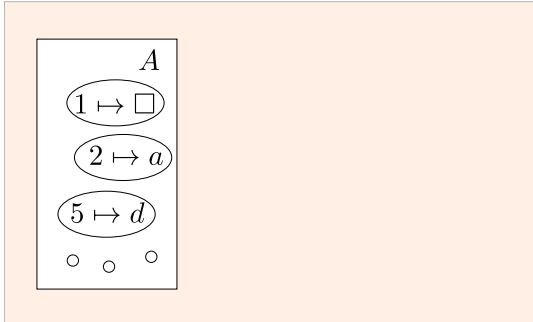
Visualização de Sequências Para visualizar uma sequência na forma de um diagrama, Fubã decide destacar dois elementos:

- um retângulo para a sequência e
- diversas ovas para destacar elementos de interesse, desenhadas na parte interna do retângulo.

Como um exemplo, ele retoma a representação do alfabeto, modelado como a sequência A :

$$A \triangleq \langle \square, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, \zeta \rangle$$

Em A , encontramos 28 elementos, mas Fubã resolve destacar apenas três no diagrama que, ao ser inspecionado, revela que $A(5) = d$, por exemplo.



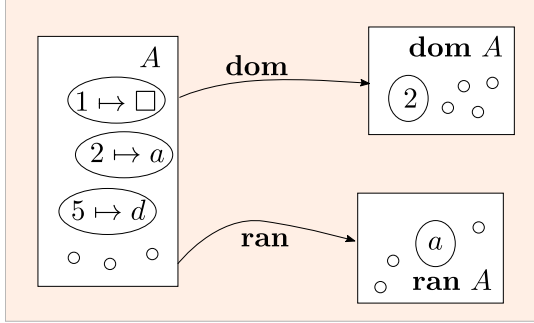
Domínio e imagem de uma sequência Como sequências também são funções, as projeções **dom** e **ran** podem ser aplicadas sobre elas. A projeção **dom** A produz o conjunto dos índices de A :

$$\mathbf{dom} A = \{ i : \mathbb{N} \mid (i \mapsto x) \in A \bullet i \} = \{1, 2, 3, \dots, 28\} = 1..28$$

Os elementos de **ran** A formam o conjunto-imagem da sequência A , constituído pelos segundos elementos de cada par:

$$\begin{aligned}\mathbf{ran} A &= \{ i : \mathbb{N} \mid (i \mapsto x) \in A \bullet x \} \\ &= \{ \square, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, x, c \}\end{aligned}$$

Para visualizar as projeções $\mathbf{dom} A$ e $\mathbf{ran} A$ Fubã elabora o diagrama mostrado a seguir.



Pergunta (DNA e Sequências). *Contexto* Estudos recentes na biologia sintética resultaram em ferramentas para a construção de moléculas complexas de DNA, aspecto importante em muitos projetos de biologia molecular e biotecnologia¹.

Cadeias de DNA resultam de uma combinação das microbases $Bases \triangleq \{A, C, G, T\}$. Um experimento envolveu uma cadeia DNA com o trecho modelado pela sequência:

$$DNA_1 = \langle G, A, T, C, T, C, C, C \rangle = \langle GATCTCCC \rangle$$

Enunciado Assinale a alternativa contendo a afirmação **falsa**:

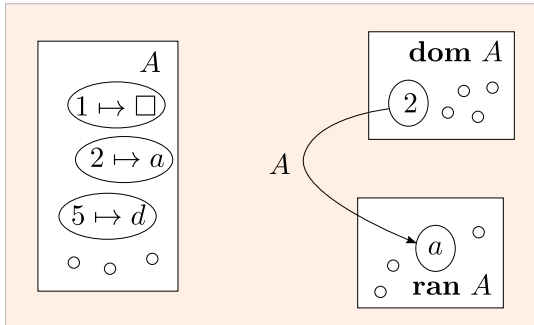
1. $\mathbf{dom} DNA_1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$.
2. $\mathbf{ran} DNA_1 = Bases$.
3. $(2 \mapsto A) \in DNA_1$.
4. $\#DNA_1 = 7$.

Aplicação de sequência Quando aplicarmos uma sequência a um argumento do domínio, obteremos a sua imagem. Qual a imagem de 3 sob a sequência $\langle 2, 4, 6, 8, 10 \rangle$? “Devo me lembrar do conjunto representado pela sequência e encontrar o par cujo primeiro elemento é 3”, pensa Fubã. Ele escreve:

$$\langle 2, 4, 6, 8, 10 \rangle (3) = \{1 \mapsto 2, 2 \mapsto 4, 3 \mapsto 6, 4 \mapsto 8, 5 \mapsto 10\}(3) = 6$$

¹URL: <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-015-0544-x>. Acesso em 06/04/2022.

O seguinte diagrama ilustra o caso da aplicação da sequência A no argumento 2: $A(2) = A\ 2 = A_2 = a$.



Pergunta (DNA e Sequências em Python). *Contexto* Cadeias de DNA resultam de uma combinação das microbases $Bases \triangleq \{A, C, G, T\}$. Um experimento envolveu uma cadeia DNA com o trecho modelado pela sequência:

$$DNA_1 = \langle G, A, T, C, T, C, C, C \rangle = \langle GATCTCCC \rangle$$

Fubã decidiu implementar a sequência DNA_1 usando uma estrutura do tipo `list` de Python:

```
# CENÁRIO de implementação de um DNA por "list"
def dom(lst) :
    return [i+1 for i in range(len(lst))]

def ran(lst) :
    return lst

// Arrays como sequências
DNA_1 = ['G', 'A', 'T', 'C', 'T', 'C', 'C', 'C']
print (dom (DNA_1) == [1, 2, 3, 4, 5, 6, 7, 8]) # <-- (A)
print (ran (DNA_1) == ['G', 'A', 'T', 'C', 'T', 'C', 'C', 'C']) # <-- (B)
print (DNA_1[2] == 'G') # <-- (C)
print (len (DNA_1) == 8) # <-- (D)
```

Enunciado Assinale a alternativa contendo a afirmação **falsa**:

1. (A)
2. (B)
3. (C)
4. (D)

6.2 Inversa de uma Relação

Ao modelar o alfabeto da cifra de César como uma sequência, Fubã tem acesso aos símbolos a partir dos seus índices. E como recuperar o índice a partir de um símbolo?



— Você precisa do conceito de relação inversa!

Relação inversa Fubã já havia esquecido das funções inversas (Ince, 1992, p. 119). Primeiro ele recupera a noção de relação como um conjunto de pares ordenados. Em seguida, caso o elemento $(x, y) \in r$, sendo r uma relação, a sua inversa, indicada por r^\sim , contém o elemento (y, x) :

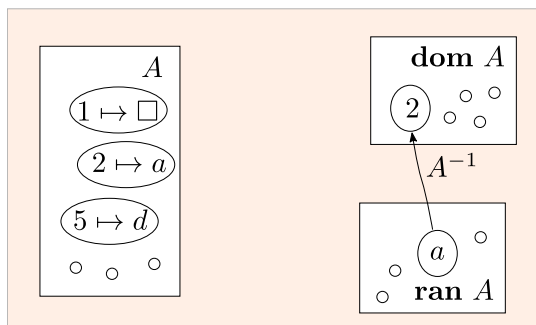
$$(y, x) \in r^\sim$$

A relação inversa é o conjunto de pares ordenados obtido quando se reverte cada par da relação original. O conjunto resultante pode ser ou não uma função. No particular caso de uma função f cuja inversa também é uma função, indicaremos esta última por f^{-1} .



— A função inversa troca de lugar os componentes de um par ordenado.

No específico caso da sequência A , temos uma função bijetora. A sua inversa, neste caso, também é uma função, como podemos observar no seguinte diagrama. A aplicação da sequência A no número 2 produz a letra a . A função inversa A^{-1} aplicada na letra a produz o número 2:



$$A \triangleq \{1 \mapsto \square, 2 \mapsto a, \dots, 28 \mapsto \zeta\}$$

$$A^{-1} = \{\square \mapsto 1, a \mapsto 2, \dots, \zeta \mapsto 28\}$$

Pergunta (Modelagem com Sequências). *Contexto* Considere uma sequência de atividades realizadas pelos estudantes em um curso de computação:

$$\text{atv} \triangleq \{1 \mapsto A_{11}, 2 \mapsto A_{12}, 3 \mapsto A_{21}, 4 \mapsto A_{22}\}$$

Enunciado Assinale a alternativa contendo a afirmação verdadeira:

1. $\text{atv}(2) = A_{21}$
 2. $\text{atv}^{-1}(2) = A_{12}$
 3. $\# \text{atv} = \text{atv}^{-1}(A_{22})$
 4. $\text{atv}^{-1}(A_{11}) + \text{atv}^{-1}(A_{12}) = \perp$
-

Pergunta (Inversas em Python). *Contexto* Considere a sequência:

$$s \triangleq \langle a, b, c, d \rangle$$

Considerando-se as apropriadas restrições em tempo de execução, s e a sua inversa s^{-1} podem ser implementadas pelos seguintes métodos em Python:

```
# CENÁRIO de implementação da sequência
# s = <a,b,c,d>
def s (i) :
    return chr (i+96)
def s_inv (x) :
    return ord(x) - 96
```

Exemplos de aplicações destas funções seguem:

```
# CENÁRIO de aplicações
print (s (1))
print (s_inv ('d') == 4)
```

Enunciado Assinale a alternativa contendo a afirmação verdadeira:

1. $\text{s_inv}('d') == \text{s_inv}('c') - 1$
 2. $\llbracket \#s \rrbracket_{\text{Python}} = \text{s_inv}('d')$
 3. $\text{s}(1) + \text{s}(2) == 3$
 4. $\text{s}(3) == 'd'$
-

Com a ajuda dos colegas, Fubã chega na seguinte proposta para a implementação de A^{-1} , a função inversa do modelo do alfabeto A no caso da cifragem de César:

```

def A_inv(x) :
    dom = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
           11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
           21, 22, 23, 24, 25, 26, 27, 28]
    ran = [' ', 'a', 'b', 'c', 'd', 'e', 'f',
           'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o',
           'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'ç']
    return { ran[i]: dom[i] for i in range (len (dom)) } [x]

```

Pergunta (Propriedades da Inversa). *Contexto* Fubã modelou o alfabeto do seu estudo da Cifra de César pela sequência A :

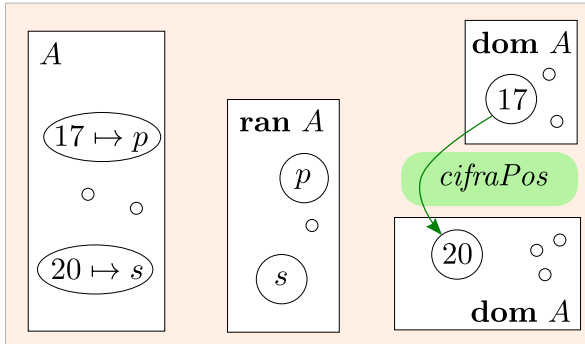
$$A \triangleq \langle \square, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, \varsigma \rangle$$

Enunciado Assinale a alternativa contendo uma propriedade **falsa** da inversa de A :

1. $A^{-1}(\square) = A^{-1} \square = 1$
 2. $\text{dom } A^{-1} = \text{ran } A$
 3. $\text{ran } A^{-1} = 1..28$
 4. $A^{-1}(a) = A(a)$
-

6.3 Funções e Expressões- λ

Retornando ao desenvolvimento de um programa para implementar a Cifra de César, Fubã começa a elaborar uma regra para a função *cifraPos*. A ideia é mapear um índice de símbolo em outro, considerando um deslocamento de cifragem:



Um tanto apressado, Fubã propõe uma definição para a função que calcula a posição cifrada de um símbolo. Espec sugere, no entanto, que Fubã dê o seguinte nome para a sua função: *cifraPos*_{Erro}.



“Não sei porque ele sugeriu isso, mas vamos lá...”

$$\text{cifraPos}_{\text{Erro}} \triangleq \lambda i : \mathbb{Z} \mid i \in \text{ran } A \bullet i + 3$$

Uma inesperada visita do Ocara surpreende a dupla de colegas. Ocara não é muito conhecido pela qualidade dos programas que desenvolve. Em geral, eles não funcionam.



— Implemento o método `cifraPosErro` usando três linhas de código!

```
# CENÁRIO de implementação da função "cifraPosErro"
def cifraPosErro (x) :
    return x+3
```

Pergunta (Modelos Inválidos em Python). *Contexto* Segundo o Ocara, o código de implementação da lógica de cifragem de um símbolo envolve os seguintes métodos:

```
# CENÁRIO de cifragem logicamente ERRADA de um símbolo
def A (i):
    if (i == 1) :
        return ' '
    if (i == 28) :
        return 'ç'
    return chr (i + 95)

def cifraPosErro (i) :
    return (i + 3)

def A_inv (x) :
    if x == ' ' :
        return 1
    if x == 'ç' :
        return 28
    return ord(x) - 95
```

Enunciado Assinale a alternativa contendo a afirmação **logicamente absurda**:

```
# CENÁRIO de uso dos métodos de cifragem de letras
A_inv ('a') == 2           # <-- (1)
cifraPosErro (2) == 2     # <-- (2)
cifraPosErro (28) == 31   # <-- (3)
A (2) == 'a'              # <-- (4)
```

Casos de uma Função Enquanto Ocara procura o erro por análise do seu código, Fubã retorna ao nível lógico da sua definição. Após refletir por algum tempo, ele entendeu o seu erro na definição $cifraPos_{Erro}$. Tal função opera sobre o alfabeto:

$$A \triangleq \langle \square, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, \zeta \rangle$$

Usando a notação para regras de funções definidas por partes, Fubã escreve o domínio e a imagem da $cifraPos$ corrigida:

$$cifraPos \in (\mathbf{dom} A \rightarrow \mathbf{dom} A)$$

Como o valor do deslocamento é igual a 3 e a maior posição de A é 28, até a posição 25 basta somar 3. Assim, se a posição de cifragem i estiver entre 1 e $(\#A - 3)$, ou seja, $i \in 1..25$, a regra deverá mapeá-la em $i + 3$.



— Vou reescrever a definição.

Tipos enumerados Fubã formaliza o tipo $Char$ dos símbolos que constituem um alfabeto enumerando os seus valores:

$$\begin{aligned} Char \triangleq & \square \mid a \mid b \mid c \mid d \mid e \mid f \mid g \mid h \mid i \mid j \mid k \mid l \mid m \mid \\ & n \mid o \mid p \mid q \mid r \mid s \mid t \mid u \mid v \mid w \mid x \mid y \mid z \mid \zeta \end{aligned}$$

O sinal \mid introduz os possíveis valores de um tipo de dado enumerado. Neste caso, os valores a e m são um dos valores do tipo $Char$. Tal afirmação escreve-se como:

$$a, m : Char$$

Partes de uma função Revendo a sua definição da $cifraPos$, Fubã estabelece um alfabeto s como primeiro parâmetro e um índice i a ser cifrado, como segundo:

$$cifraPos_1 \triangleq \{ s : \mathbf{seq}(CHAR), i : \mathbb{Z} \mid (i \in \mathbf{dom} s) \wedge (i + 3 \leq \#s) \bullet (s, i) \mapsto (i + 3) \}$$

Caso contrário, a imagem será calculada pela expressão $(x + 3) - \#A$:

$$cifraPos_2 \triangleq \{ s : \mathbf{seq}(CHAR), i : \mathbb{Z} \mid (i \in \mathbf{dom} s) \wedge (i + 3 > \#s) \bullet (s, i) \mapsto (i + 3) - \#s \}$$



— Sensacional! A definição da $cifraPos$ envolve dois casos:

$$cifraPos \triangleq cifraPos_1 \cup cifraPos_2$$

Fubã decide aplicar essa função no valor 25, considerando o alfabeto A , para recapitular a computação que será feita pelo interpretador Python:

$$\begin{aligned} cifraPos(A, 25) &= cifraPos_1(A, 25) \\ &= \{ s : seq(Char), i : \mathbb{Z} \mid (i \in \mathbf{dom} \ s) \wedge (i + 3 \leq \#s) \bullet \\ &\quad (s, i) \mapsto (i + 3) \} (A, 25) \\ &= \{ A : seq(Char), 25 : \mathbb{Z} \mid (25 \in 1..28) \wedge (25 + 3 \leq 28) \bullet \\ &\quad (A, 25) \mapsto (25 + 3) \} (A, 25) \\ &= \{ A : seq(Char), 25 : \mathbb{Z} \mid \mathbf{true} \wedge \mathbf{true} \bullet (A, 25) \mapsto 28 \} (A, 25) \\ &= \{ (A, 25) \mapsto 28 \} (A, 25) = 28 \end{aligned}$$

Pergunta (Cifra da Posição em Python). *Contexto* Fubã corrigiu o método `cifraPos` apoiando-se na lógica da função `cifraPos`:

```
# CENÁRIO de implementação do método "cifraPos"
def cifraPos(s, i) :
    return (x+3) if 1 <= i <= len(s) and (x+3) <= len(s) else ((x+3) - len(s))
# aplicação de "cifraPos"
A = [' ', 'a', 'b', 'c', 'd', 'e', 'f',
     'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o',
     'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'ç']
```

Enunciado Assinale a alternativa contendo a afirmação **falsa**:

```
cifraPos(A, 1) == 4) # <-- (A)
cifraPos(A, 2) == 5) # <-- (B)
cifraPos(A, 25) == 28) # <-- (C)
cifraPos(A, 28) == 2) # <-- (D)
```

6.4 Resumo

1. Composição de funções produz funções mais complexas.
2. Representa-se a composição pelo símbolo \circ .
3. Funções descrevem conjuntos de objetos e podem ser combinadas, formando cadeias de associações.
4. A recursão é uma técnica de programação que se baseia em uma definição realizada sobre si mesma.

6.5 Exercícios

6.5.1 Elementos de Programação

Enunciado Considere as seguintes afirmações (Deitel & Deitel, 2012, p. 66):

- I) Operadores aritméticos são avaliados da esquerda para a direita em Python.
- II) Uma expressão aritmética válida em Python, sem parênteses, é avaliada da esquerda para direita.
- III) Os seguintes identificadores são válidos em Python: `3g` e `h22`.

Enunciado Assinale a alternativa contendo apenas afirmações verdadeiras:

- 1. I e II.
- 2. I e III.
- 3. II e III.
- 4. I, II e III.

6.5.2 Polinômios e Atribuições

Contexto Considere o polinômio $ax^3 + 7$ que, ao ser avaliado, produz o valor da variável y (Deitel & Deitel, 2012, p. 66).

Enunciado Assinale a alternativa contendo uma implementação correta dessa intenção:

- 1. `y = a * x * x * (x + 7)`
- 2. `y = a * x * (x * x + 7)`
- 3. `y = a * (x * x * x + 7)`
- 4. `y = (a * x * x * x + 7)`

6.5.3 Crescimento da População Mundial

Contexto Utilize serviços da Internet para determinar a população mundial p em 2020 e a taxa de crescimento populacional t (Deitel & Deitel, 2012, p. 70).

Enunciado

- 1. Defina uma função que estime a população mundial após um, dois, três, quatro e cinco anos.
- 2. Implemente a função do item anterior em Python.

6.5.4 Pegada de Carbono [**]

Contexto Alguns cientistas acreditam que as emissões de carbono contribuem significativamente para o aquecimento global (Deitel & Deitel, 2012, p. 34). Organizações e indivíduos estão cada vez mais preocupados com suas “pegadas de carbono”. Uma das calculadoras de pegada de carbono encontra-se disponível em <https://www.iniciativaverde.org.br/calculadora>.

Enunciado

1. Teste essas calculadoras para determinar sua pegada de carbono.
2. Pesquise fórmulas para calcular as pegadas de carbono.
3. Implemente uma das fórmulas do item anterior em Python.

6.5.5 Sequências em Python

Os seguintes elementos do modelo Cifra de César correspondem ao alfabeto e ao seu comprimento:

$$A \triangleq \langle \square, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, \zeta \rangle$$

$$n = \#A$$

Utilizando um valor do tipo `str` e outro do tipo `int` em Python, implemente estes elementos. Qual o código resultante?

6.5.6 Validade de Modelos

Descreva o comportamento do computador como resultado da chamada:

```
cifraCadeia( " " )
```

Sugira o que deve ser feito em relação a tal comportamento.

6.5.7 Relações Inversas

Considere a sequência $s \triangleq \langle A, B, E, E \rangle$.

1. Enumere os elementos da inversa s^\sim .
2. Mostre que s^\sim não é uma função.

Referências

- Deitel, P., & Deitel, H. (2012). *Java How to Program* (9º ed). Prentice Hall. Recuperado de <http://www.deitel.com>
- Ince, D. C. (1992). *An Introduction to Discrete Mathematics, Formal System Specification, and Z* (2º ed). Oxford University Press.