
LISTA DE EXERCÍCIOS 1

TI323 - TÓPICOS EM SISTEMAS DE COMPUTAÇÃO

Primeiro Semestre de 2024

Guilherme Macedo

Exercício 1

Qual o retorno deste programa para `funcao([83, 41, 5, 1, 59, 97], 2)`?

```
1      public static int[] funcao(int[] A, int i) {
2          A[1] = 17;
3          A[i / 2] = 9;
4          A[2 * i - 1] = 95;
5          A[i - 1] = A[5] / 2;
6          A[3] = A[i];
7          A[i + 1] = A[i] + A[i - 1];
8          A[A[2] - 2] = 78;
9          A[A[i] - 1] = A[1] * A[i] / 5;
10         A[A[2] mod 2 + 2] = A[i + 6 / 2] - A[i - 1 * 2];
11         return A;
12     }
```

Exercício 2

Qual o retorno deste programa para `funcao(81)`?

```

1      public static int funcao(int n) {
2          int p = 1, r = n;
3          while (p + 1 < r)
4          {
5              int q = (p + r) / 2;
6              if (Math.pow(q, 2) <= n) {
7                  p = q;
8              }
9              else {
10                 r = q;
11             }
12         }
13         return p;
14     }

```

Exercício 3

O programa abaixo recebe um arranjo A de n números inteiros e o rearranja de modo que seus elementos, ao final, estejam ordenados de forma decrescente. Contudo, este programa possui alguns erros de lógica. Encontre-os e corrija-os.

```

1      public static int[] ordena(int[] A) {
2          int i, j, chave;
3          for (i = 1; i < A.Length; i++)
4          {
5              chave = A[i];
6              j = i - 1;
7              while (j >= 0 && A[j] > chave) {
8                  A[j + 1] = A[j];
9                  j = j - 1;
10             }

```

```

11         A[j + 1] = chave;
12     }
13     return A;
14 }

```

Exercício 4

Observe o programa abaixo e responda:

```

1     private static int busca1(int[] A, int k) {
2         for (int i = 0; i < A.Length; i++) {
3             if (A[i] == k) {
4                 return i;
5             }
6         }
7         return -1;
8     }
9
10    private static int busca2(int[] A, int k) {
11        int inicio = 0, fim = A.Length;
12        while (inicio < fim)
13        {
14            int meio = (inicio + fim) / 2;
15            if (A[meio] == k) {
16                return meio;
17            }
18            if (A[meio] < k) {
19                inicio = meio + 1;
20            }
21            else {
22                fim = meio - 1;

```

```

23         }
24     }
25     return -1;
26 }
27
28 public static void main(String[] args) {
29     int[] A = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19};
30     int valor1 = busca1(A, 13);
31     int valor2 = busca2(A, 13);
32     Console.WriteLine(valor1 + " - " + valor2);
33 }

```

- (a) Qual é o resultado da impressão da linha 30?
- (b) `busca1` é “melhor” do que `busca2`? Justifique sua resposta.

Exercício 5

Escreva um programa em **Java** que embaralhe um arranjo A de n inteiros.

Exercício 6

Escreva um programa em **Java** que encontre dois elementos de um arranjo A de n inteiros, distintos entre si, que somados seja igual a um determinado inteiro k .

Exercício 7

Escreva um programa em **Java** que remova os elementos duplicados de um arranjo A de n cadeias de caracteres.

Exercício 8

Escreva um programa em **Java** que organize um arranjo A de n inteiros de modo que todos os inteiros negativos apareçam antes de todos os inteiros positivos.

Exercício 9

Escreva um programa em **Java** que obtenha o elemento majoritário de um arranjo A de n inteiros. Um elemento majoritário é um elemento que aparece mais de $n/2$ vezes.

Exercício 10

Escreva um programa em **Java** que encontre um elemento específico em um arranjo A de n inteiros usando a busca por interpolação.

Exercício 11

Escreva um programa em **Java** que represente o digrama de classes da Figura 1.

Exercício 13

Escreva um programa em **Java** que represente o digrama de classes da Figura 2. Suponha que um pedido não tenha mais de 30 produtos ao mesmo tempo.

Exercício 14

Escreva um programa em **Java** que represente o digrama de classes da Figura 3.

Exercício 15

Escreva um programa em **Java** que represente o digrama de classes da Figura 4.

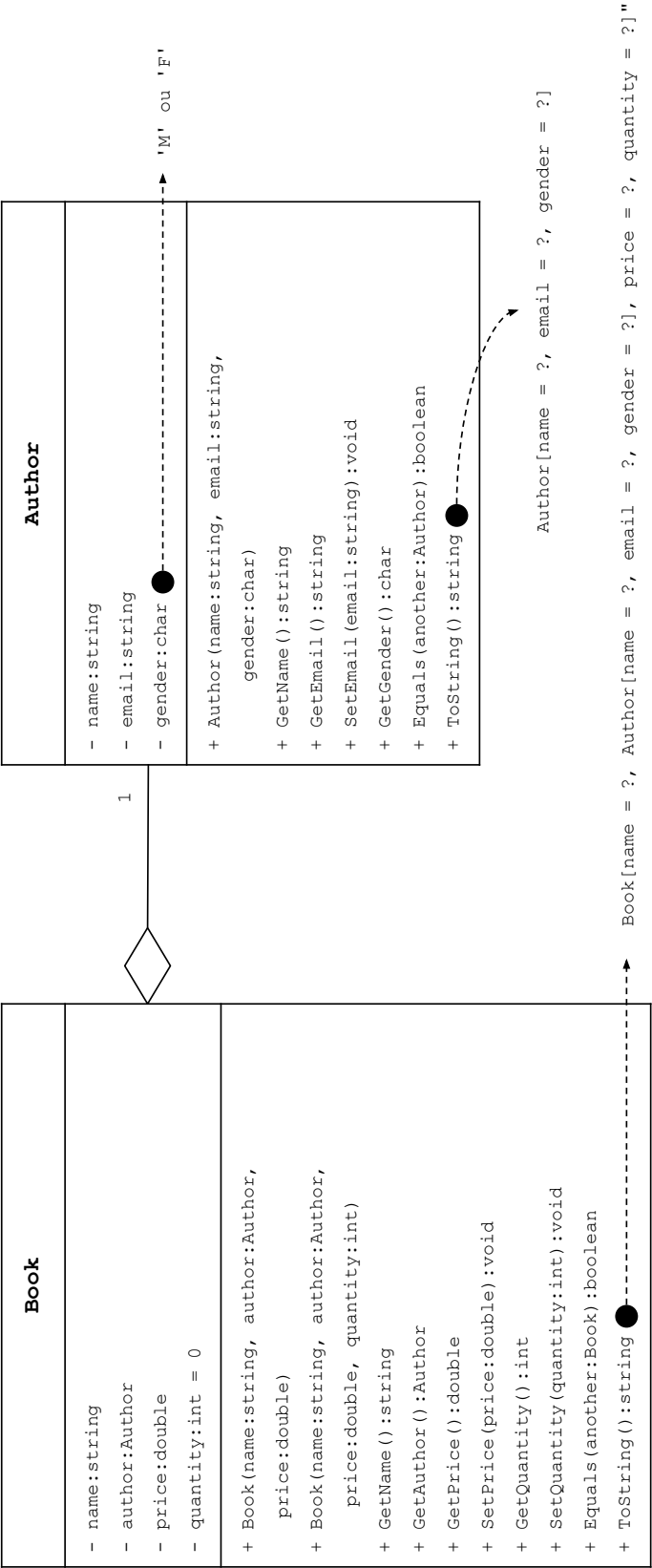


Figura 1

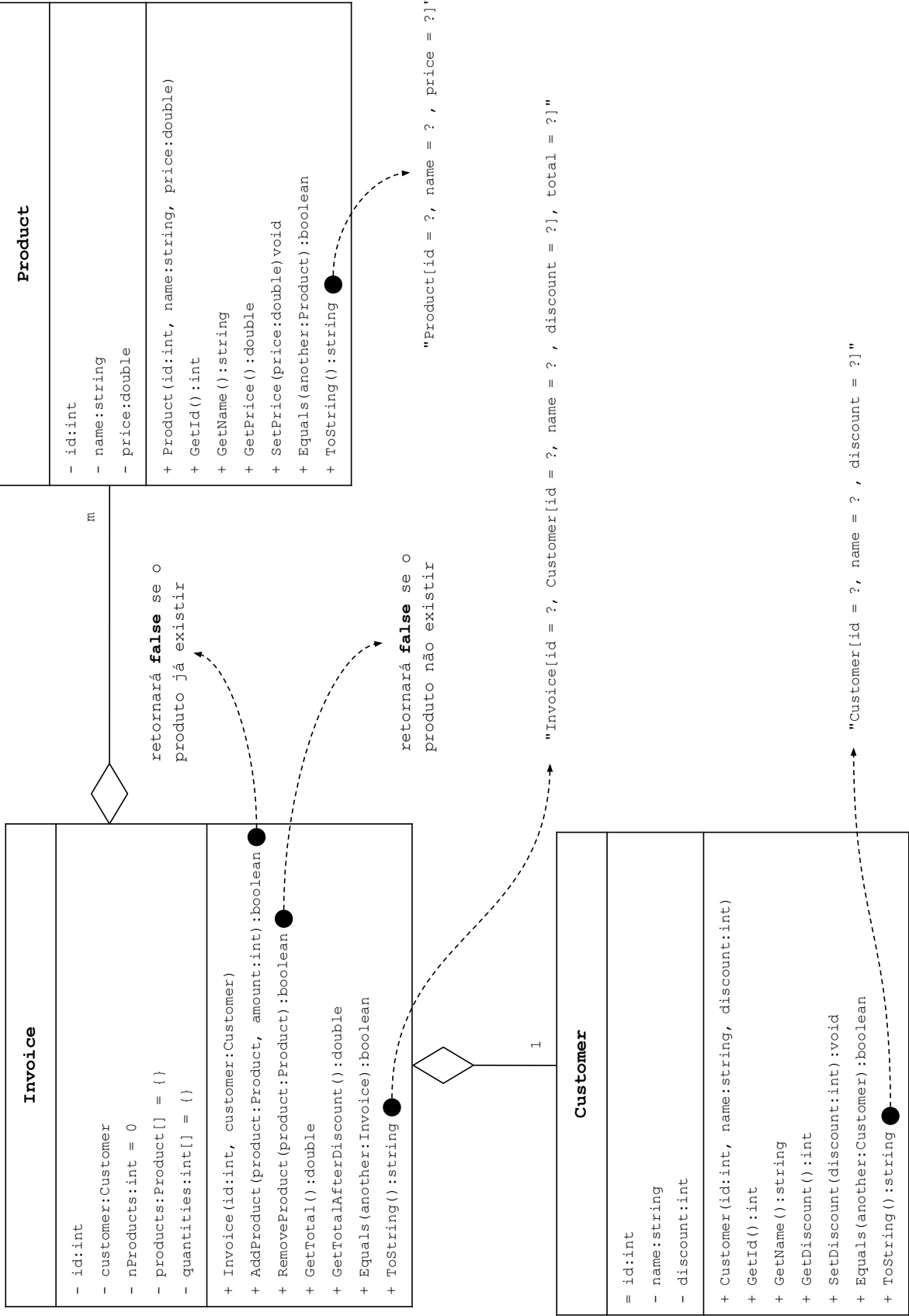
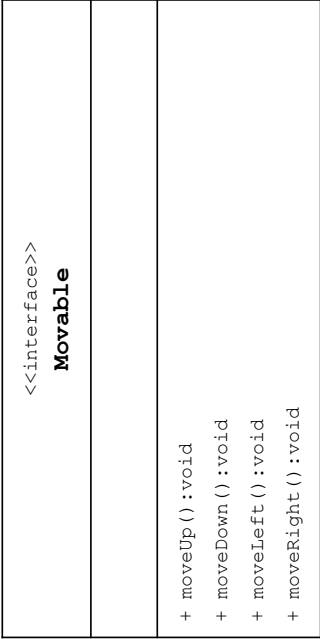


Figura 2

```
moveUp:    y += speed
moveDown:  y -= speed
moveLeft:  x -= speed
moveRight: x += speed
```



8

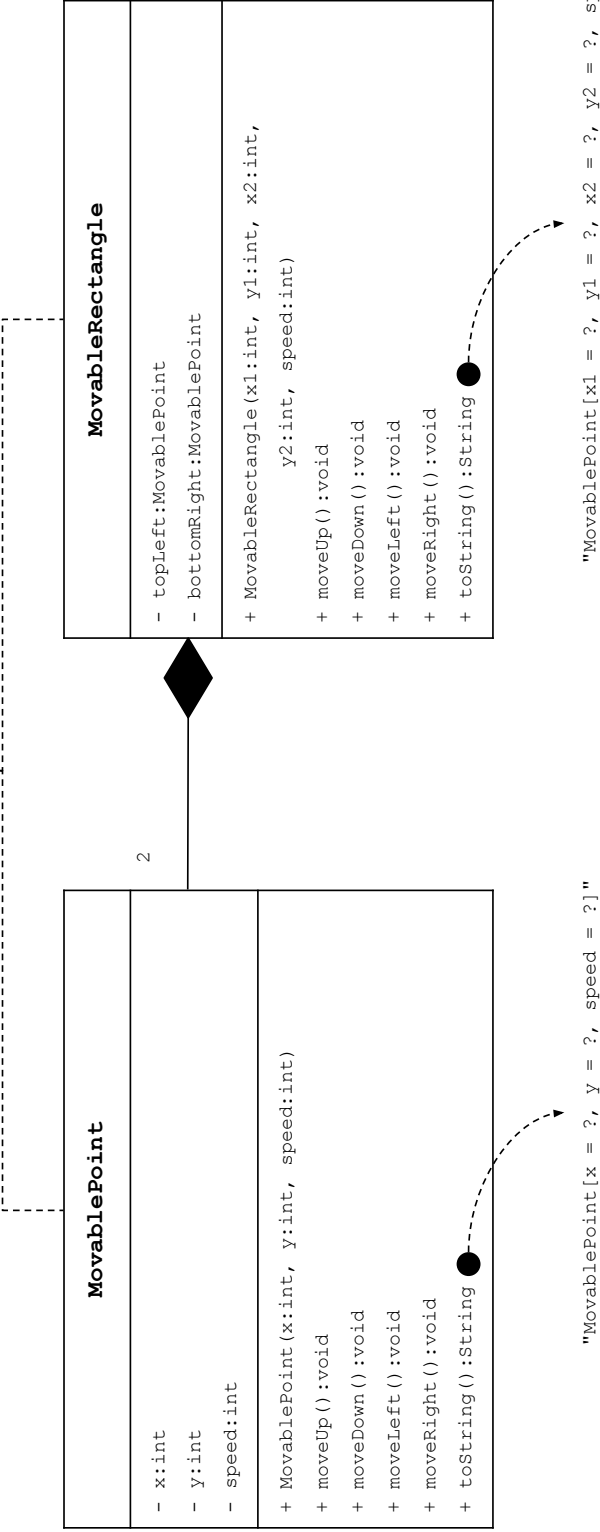


Figura 3

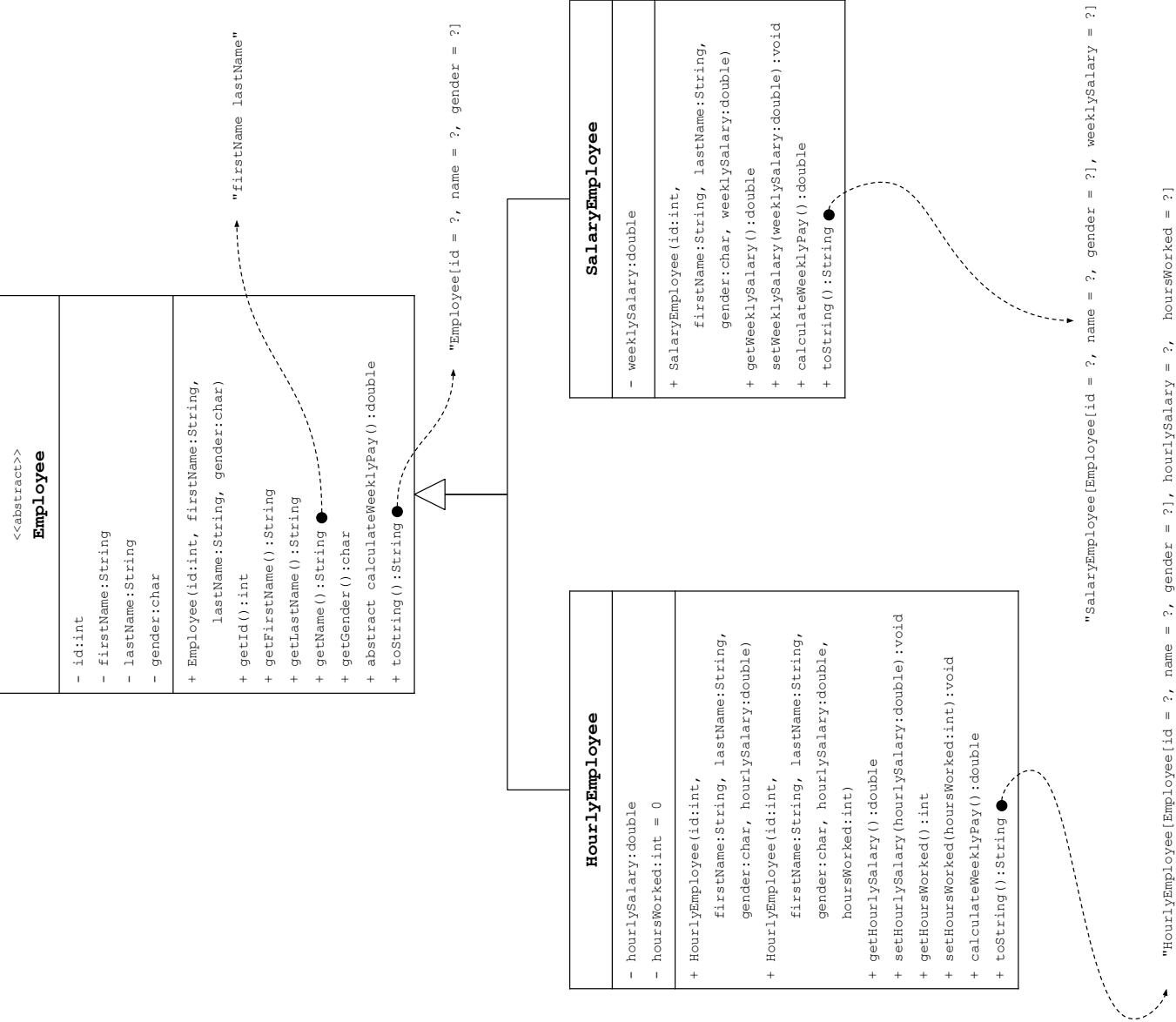


Figura 4