

# Unidade I:

## Conceitos Básicos - Recursividade

Prof. Max do Val Machado



**PUC Minas**

Instituto de Ciências Exatas e Informática  
Curso de Ciência da Computação

Era uma vez dois gatinhos: o grapete e o repete. O grapete morreu, quem ficou?

- Definição: Um método é dito recursivo quando ele faz chamada a si próprio
- Um método recursivo normalmente apresenta duas características básicas:
  - Chamada recursiva
  - Condição de parada
- Qual é a condição de parada da nossa estória?

- Identifique as chamadas recursivas e condições de parada

```
int fat (int n){  
    int resp;  
    if (n == 1){  
        resp = 1;  
    } else {  
        resp = n * fat (n - 1);  
    }  
    return resp;  
}
```

```
int fib (int n){  
    int resp;  
    if (n == 0 || n == 1){  
        resp = 1;  
    } else {  
        resp = fib (n - 1) + fib(n - 2);  
    }  
    return resp;  
}
```

- Identifique as chamadas recursivas e condições de parada

```
int fat (int n){  
    int resp;  
    if (n == 1){  
        resp = 1;  
    } else {  
        resp = n * fat (n - 1);  
    }  
    return resp;  
}
```

Chamadas recursivas

```
int fib (int n){  
    int resp;  
    if (n == 0 || n == 1){  
        resp = 1;  
    } else {  
        resp = fib (n - 1) + fib(n - 2);  
    }  
    return resp;  
}
```

- Identifique as chamadas recursivas e condições de parada

```
int fat (int n){  
    int resp;  
    if (n == 1){  
        resp = 1;  
    } else {  
        resp = n * fat(n - 1);  
    }  
    return resp;  
}
```

A cada chamada recursiva,  
o  $n$  se aproxima do último valor

Condições de parada

```
int fib (int n){  
    int resp;  
    if (n == 0 || n == 1){  
        resp = 1;  
    } else {  
        resp = fib(n - 1) + fib(n - 2);  
    }  
    return resp;  
}
```

# Exemplo de Método Iterativo vs Recursivo


- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

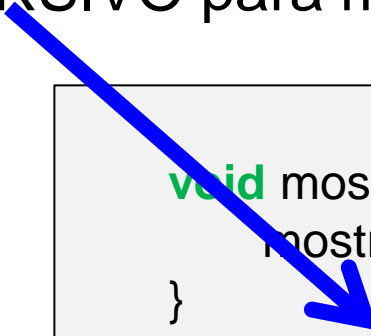
```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3



```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```



```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```



# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

Tela

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

Tela

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

i	0
---	---

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

Tela

true

i	0
---	---

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

Tela

0

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

i

0

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

Tela

0

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

i

1

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

Tela

0

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

i

1

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

Tela
0

true

i	1
---	---

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {
    for (int i = 0; i < 4; i = i + 1) {
        System.out.println(i);
    }
}
```

```
void mostrar () {
    mostrar (0);
}
void mostrar (int i) {
    if (i < 4) {
        System.out.println(i);
        mostrar (i + 1);
    }
}
```

Tela
0
1

i	1
---	---



# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

Tela
0
1

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

i	2
---	---

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

Tela
0
1

i	2
---	---

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

Tela
0
1

true

i	2
---	---

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

Tela
0
1
2

i	2
---	---

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

Tela
0
1
2

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

i	3
---	---

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

Tela
0
1
2

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

i	3
---	---

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

Tela
0
1
2

true

i	3
---	---

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

Tela

0

1

2

3

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

i

3



# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

Tela
0
1
2
3

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

i	4
---	---

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

Tela
0
1
2
3

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

i	4
---	---

# Exemplo de Método Iterativo vs Recursivo

- Faça métodos ITERATIVO e RECURSIVO para mostrar os números 0 à 3

```
void mostrar () {  
    for (int i = 0; i < 4; i = i + 1) {  
        System.out.println(i);  
    }  
}
```

```
void mostrar () {  
    mostrar (0);  
}  
void mostrar (int i) {  
    if (i < 4) {  
        System.out.println(i);  
        mostrar (i + 1);  
    }  
}
```

Tela
0
1
2
3

false

i	4
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){
    printRecursivo(2);
}

void printRecursivo(int i){
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    System.out.println(i);
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){
    printRecursivo(2);
}

void printRecursivo(int i){
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    System.out.println(i);
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

i	
---	--

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

i	
---	--

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

i	2
---	---



## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

2

i	2
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

**true**

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

2

i	2
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

2

i	2
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

2

i	1
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela
2
1

i	1
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

**true**

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

2  
1

i	1
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela
2
1

i	1
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

2  
1

i	0
---	---



## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela
2
1
0

i	0
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

**false**

É fácil compreender a impressão do 2, 1, 0 e 0. Contudo, o 1 e 2 no final ... ☹

Tela

2  
1  
0

i	0
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

E agora José?

Tela

2  
1  
0  
0

i	0
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

Agora, estamos no print do zero e retornaremos para o do um onde i vale 1 e chamou o print do zero

Tela
2
1
0
0

i	0
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

Agora, estamos no print do zero e retornaremos para o do um onde i vale 1 e chamou o print do zero

Tela
2
1
0
0

i	1
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

Agora, retornaremos para o print do dois

Tela

2  
1  
0  
0  
1

i	1
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

Agora, retornaremos para o print do dois

Tela

2  
1  
0  
0  
1

i	2
---	---

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

Agora, retornaremos para o primeiro print

Tela

2  
1  
0  
0  
1  
2

i	2
---	---



## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

Agora, retornaremos para o primeiro print

Tela

2  
1  
0  
0  
1  
2

i	
---	--

# Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    System.out.println("1o – inicio");  
    segundo();  
    System.out.println("1o – fim");  
}  
  
void segundo(){  
    System.out.println("2o – inicio e fim");  
}  
  
void main (){  
    System.out.println("main – inicio");  
    primeiro();  
    System.out.println("main – fim");  
}
```

# Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    System.out.println("1o – início");  
    segundo();  
    System.out.println("1o – fim");  
}  
  
void segundo(){  
    System.out.println("2o – início e fim");  
}  
  
void main (){  
    System.out.println("main – início");  
    primeiro();  
    System.out.println("main – fim");  
}
```

TELA

# Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    System.out.println("1o – início");  
    segundo();  
    System.out.println("1o – fim");  
}  
  
void segundo(){  
    System.out.println("2o – início e fim");  
}  
  
void main (){  
    System.out.println("main – início");  
    primeiro();  
    System.out.println("main – fim");  
}
```

TELA

main – início

# Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    System.out.println("1o – início");  
    segundo();  
    System.out.println("1o – fim");  
}  
  
void segundo(){  
    System.out.println("2o – início e fim");  
}  
  
void main (){  
    System.out.println("main – início");  
    primeiro();  
    System.out.println("main – fim");  
}
```

TELA

main – início

# Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    System.out.println("1o – início");  
    segundo();  
    System.out.println("1o – fim");  
}  
  
void segundo(){  
    System.out.println("2o – início e fim");  
}  
  
void main (){  
    System.out.println("main – início");  
    primeiro();  
    System.out.println("main – fim");  
}
```

TELA

main – início

# Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    System.out.println("1º – início");  
    segundo();  
    System.out.println("1º – fim");  
}  
  
void segundo(){  
    System.out.println("2º – início e fim");  
}  
  
void main (){  
    System.out.println("main – início");  
    primeiro();  
    System.out.println("main – fim");  
}
```

TELA

main – início

1º – início

# Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    System.out.println("1º – início");  
    segundo();  
    System.out.println("1º – fim");  
}  
  
void segundo(){  
    System.out.println("2º – início e fim");  
}  
  
void main (){  
    System.out.println("main – início");  
    primeiro();  
    System.out.println("main – fim");  
}
```

TELA

main – início

1º – início



# Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    System.out.println("1º – início");  
    segundo();  
    System.out.println("1º – fim");  
}  
  
void segundo(){  
    System.out.println("2º – início e fim");  
}  
  
void main (){  
    System.out.println("main – início");  
    primeiro();  
    System.out.println("main – fim");  
}
```

TELA

main – início

1º – início

# Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    System.out.println("1º – início");  
    segundo();  
    System.out.println("1º – fim");  
}  
  
void segundo(){  
    System.out.println("2º – início e fim");  
}  
  
void main (){  
    System.out.println("main – início");  
    primeiro();  
    System.out.println("main – fim");  
}
```

TELA

main – início

1º – início

2º – início e fim

# Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    System.out.println("1º – início");  
    segundo();  
    System.out.println("1º – fim");  
}  
  
void segundo(){  
    System.out.println("2º – início e fim");  
}  
  
void main (){  
    System.out.println("main – início");  
    primeiro();  
    System.out.println("main – fim");  
}
```

TELA

main – início

1º – início

2º – início e fim

# Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    System.out.println("1º – início");  
    segundo();  
    System.out.println("1º – fim");  
}  
  
void segundo(){  
    System.out.println("2º – início e fim");  
}  
  
void main (){  
    System.out.println("main – início");  
    primeiro();  
    System.out.println("main – fim");  
}
```

## TELA

main – início  
1º – início  
2º – início e fim  
1º – fim

# Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    System.out.println("1o – início");  
    segundo();  
    System.out.println("1o – fim");  
}  
  
void segundo(){  
    System.out.println("2o – início e fim");  
}  
  
void main (){  
    System.out.println("main – início");  
    primeiro();  
    System.out.println("main – fim");  
}
```

## TELA

main – início  
1º – início  
2º – início e fim  
1º – fim

# Exemplo de Método Iterativo

- O que o programa iterativo abaixo mostra na tela?

```
void primeiro(){  
    System.out.println("1º – início");  
    segundo();  
    System.out.println("1º – fim");  
}  
  
void segundo(){  
    System.out.println("2º – início e fim");  
}  
  
void main (){  
    System.out.println("main – início");  
    primeiro();  
    System.out.println("main – fim");  
}
```

## TELA

main – início  
1º – início  
2º – início e fim  
1º – fim  
main – fim

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

## Exercício

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

Temos como se cada chamada recursiva fosse um método diferente!!!



## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

## Exercício - Reavaliando

● Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
(1) void printRecursivo(){  
(2)     printRecursivo(2);  
}
```

```
(3) void printRecursivo(int i){ // i (2)  
(4)     System.out.println(i);  
(5)     if (i > 0){  
(6)         printRecursivo(i - 1);  
        }  
        System.out.println(i);  
}
```

```
(7) void printRecursivo(int i){ // i (1)  
(8)     System.out.println(i);  
(9)     if (i > 0){  
(10)        printRecursivo(i - 1);  
        }  
        System.out.println(i);  
}
```

```
(11) void printRecursivo(int i){ // i (0)  
(12)     System.out.println(i);  
(13)     if (i > 0){  
        printRecursivo(i - 1);  
        }  
(14)     System.out.println(i);  
}
```

## Exercício - Reavaliando

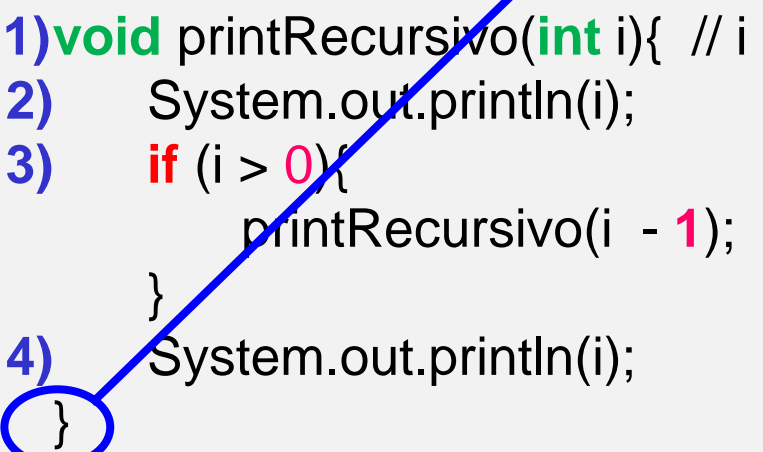
- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
(1) void printRecursivo(){  
(2)     printRecursivo(2);  
}
```

```
(3) void printRecursivo(int i){ // i (2)  
(4)     System.out.println(i);  
(5)     if (i > 0){  
(6)         printRecursivo(i - 1);  
        }  
        System.out.println(i);  
}
```

```
(7) void printRecursivo(int i){ // i (1)  
(8)     System.out.println(i);  
(9)     if (i > 0){  
(10)        printRecursivo(i - 1); (15)  
        } (16)  
        System.out.println(i); (17)  
}
```

```
(11) void printRecursivo(int i){ // i (0)  
(12)     System.out.println(i);  
(13)     if (i > 0){  
        printRecursivo(i - 1);  
    }  
(14)     System.out.println(i);  
    }
```



## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
(1) void printRecursivo(){  
(2)     printRecursivo(2);  
}
```

```
(3) void printRecursivo(int i){ // i (2)  
(4)     System.out.println(i);  
(5)     if (i > 0){  
(6)         printRecursivo(i - 1); (18)  
        } (19)  
        System.out.println(i); (20)  
}
```

```
(7) void printRecursivo(int i){ // i (1)  
(8)     System.out.println(i);  
(9)     if (i > 0){  
(10)         printRecursivo(i - 1); (15)  
        } (16)  
        System.out.println(i); (17)  
}
```

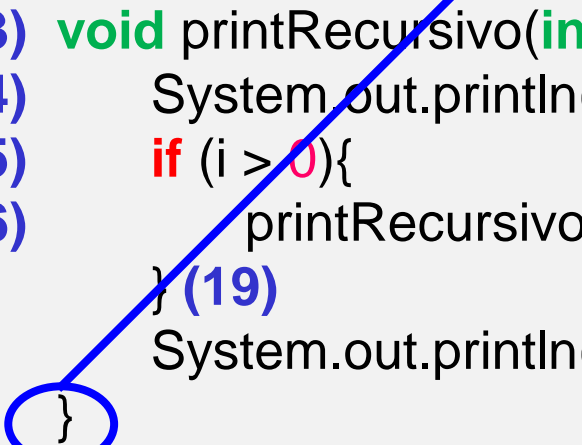
```
(11) void printRecursivo(int i){ // i (0)  
(12)     System.out.println(i);  
(13)     if (i > 0){  
        printRecursivo(i - 1);  
    }  
(14)     System.out.println(i);  
}
```

## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
(1) void printRecursivo(){
(2)     printRecursivo(2); (21)
}
```

```
(3) void printRecursivo(int i){ // i (2)
(4)     System.out.println(i);
(5)     if (i > 0){
(6)         printRecursivo(i - 1); (18)
            } (19)
        System.out.println(i); (20)
    }
```



```
(7) void printRecursivo(int i){ // i (1)
(8)     System.out.println(i);
(9)     if (i > 0){
(10)         printRecursivo(i - 1); (15)
            } (16)
        System.out.println(i); (17)
    }
```

```
(11) void printRecursivo(int i){ // i (0)
(12)     System.out.println(i);
(13)     if (i > 0){
            printRecursivo(i - 1);
        }
(14)     System.out.println(i);
    }
```

## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
(1) void printRecursivo(){
(2)     printRecursivo(2); (21)
    }
```

```
(3) void printRecursivo(int i){ // i (2)
(4)     System.out.println(i);
(5)     if (i > 0){
(6)         printRecursivo(i - 1); (18)
    } (19)
    System.out.println(i); (20)
}
```

```
(7) void printRecursivo(int i){ // i (1)
(8)     System.out.println(i);
(9)     if (i > 0){
(10)         printRecursivo(i - 1); (15)
    } (16)
    System.out.println(i); (17)
}
```

```
(11) void printRecursivo(int i){ // i (0)
(12)     System.out.println(i);
(13)     if (i > 0){
        printRecursivo(i - 1);
    }
(14)     System.out.println(i);
}
```

# Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

# Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```


```
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```



## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```



```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```



```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```



```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

**true**


```
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```



```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1); (a)  
    }  
    System.out.println(i);  
}
```

Vamos para o print do um,  
contudo, depois, voltaremos para (a)


```
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```


## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?


```
void printRecursivo(){  
    printRecursivo(2);  
}
```



```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1); (a)  
    }  
    System.out.println(i);  
}
```



```
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```




```
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```


## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?


```
void printRecursivo(){  
    printRecursivo(2);  
}
```



```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1); (a)  
    }  
    System.out.println(i);  
}
```



```
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```



```
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

# Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){
    printRecursivo(2);
}
```

```
void printRecursivo(int i){ // i (2)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1); (a)
    }
    System.out.println(i);
}
```

```
void printRecursivo(int i){ // i (1)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    System.out.println(i); true
}
```

```
void printRecursivo(int i){ // i (0)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    System.out.println(i);
}
```

## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){
    printRecursivo(2);
}
```

```
void printRecursivo(int i){ // i (2)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1); (a)
    }
    System.out.println(i);
}
```

Vamos para o print do zero,  
contudo, depois, voltaremos para (b)

```
void printRecursivo(int i){ // i (1)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1); (b)
    }
    System.out.println(i);
}
```

```
void printRecursivo(int i){ // i (0)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    System.out.println(i);
}
```



# Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){
    printRecursivo(2);
}
```

```
void printRecursivo(int i){ // i (2)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1); (a)
    }
    System.out.println(i);
}
```

```
void printRecursivo(int i){ // i (1)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1); (b)
    }
    System.out.println(i);
}
```

```
void printRecursivo(int i){ // i (0)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    System.out.println(i);
}
```

# Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){
    printRecursivo(2);
}
```

```
void printRecursivo(int i){ // i (2)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1); (a)
    }
    System.out.println(i);
}
```

```
void printRecursivo(int i){ // i (1)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1); (b)
    }
    System.out.println(i);
}
```

```
void printRecursivo(int i){ // i (0)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    System.out.println(i);
}
```

## Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, 0, 1 e 2?

```
void printRecursivo(){
    printRecursivo(2);
}
```

```
void printRecursivo(int i){ // i (2)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1); (a)
    }
    System.out.println(i);
}
```

```
void printRecursivo(int i){ // i (1)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1); (b)
    }
    System.out.println(i);
}
```

```
void printRecursivo(int i){ // i (0)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    System.out.println(i); false
}
```

# Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){
    printRecursivo(2);
}
```

```
void printRecursivo(int i){ // i (2)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1); (a)
    }
    System.out.println(i);
}
```

```
void printRecursivo(int i){ // i (1)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1); (b)
    }
    System.out.println(i);
}
```

```
void printRecursivo(int i){ // i (0)
    System.out.println(i);
    if (i > 0){
        printRecursivo(i - 1);
    }
    System.out.println(i);
}
```

## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1); (a)  
    }  
    System.out.println(i);  
}
```

Voltando para (b)

```
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1); (b)  
    }  
    System.out.println(i);  
}
```

```
void printRecursivo(int i){ // i (0)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

## Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, **0**, **1** e **2**?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1); (a)  
    }  
    System.out.println(i);  
}
```

Voltando para (b)

```
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1); (b)  
    }  
    System.out.println(i);  
}
```

## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```



```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1); (a)  
    }  
    System.out.println(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

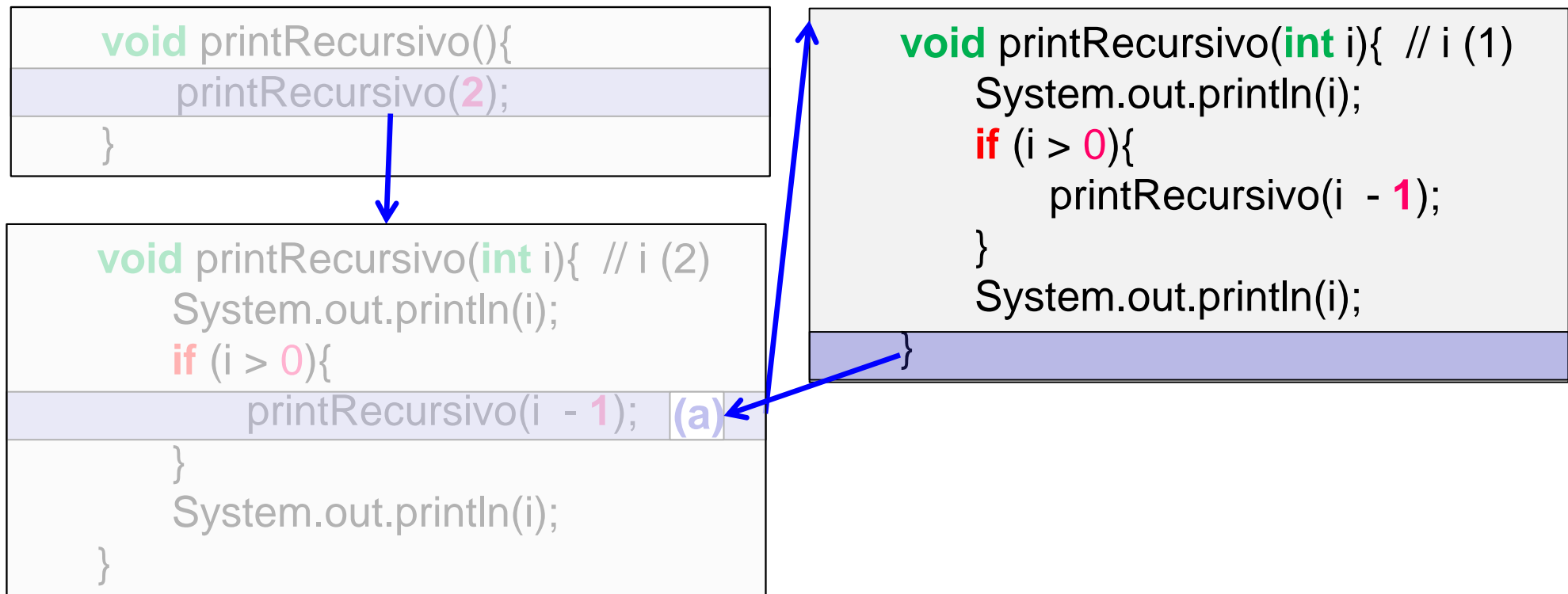
```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1); (a)  
    }  
    System.out.println(i);  
}
```

```
void printRecursivo(int i){ // i (1)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```



## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?



Voltando para (a)

## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```




```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1); (a)  
    }  
    System.out.println(i);  
}
```

Voltando para (a)

## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```




```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```



```
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

## Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, **0**, **1** e **2**?

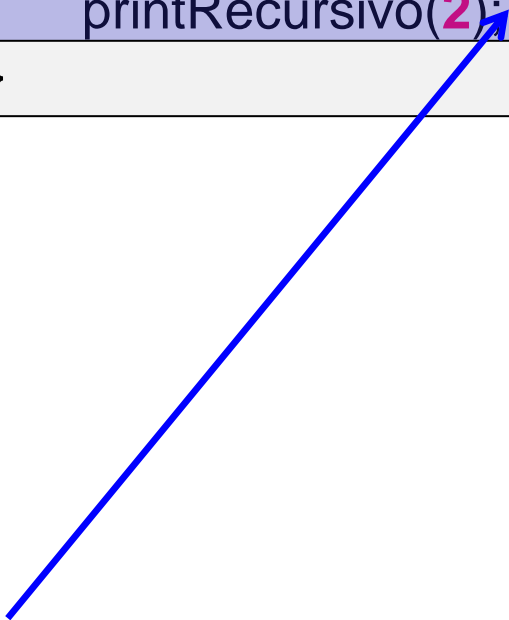
```
void printRecursivo(){  
    printRecursivo(2);  
}  
  
void printRecursivo(int i){ // i (2)  
    System.out.println(i);  
    if (i > 0){  
        printRecursivo(i - 1);  
    }  
    System.out.println(i);  
}
```

Voltando para **(primeiro)**

## Exercício - Reavaliando

- Por que o código abaixo imprime 2, 1, 0, 0, 1 e 2?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```



# Exercício - Reavaliando

- Por que o código abaixo imprime **2**, **1**, **0**, **0**, **1** e **2**?

```
void printRecursivo(){  
    printRecursivo(2);  
}
```

# Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$



# Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

# Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

$$\text{Fat}(5) = 5 * \text{Fat}(4)$$

# Exemplo: Fatorial Recursivo

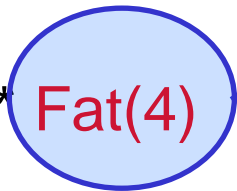
- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

$$\text{Fat}(4) = 4 * \text{Fat}(3)$$

- Qual é o valor do fatorial de 5?

$$\text{Fat}(5) = 5 * \text{Fat}(4)$$



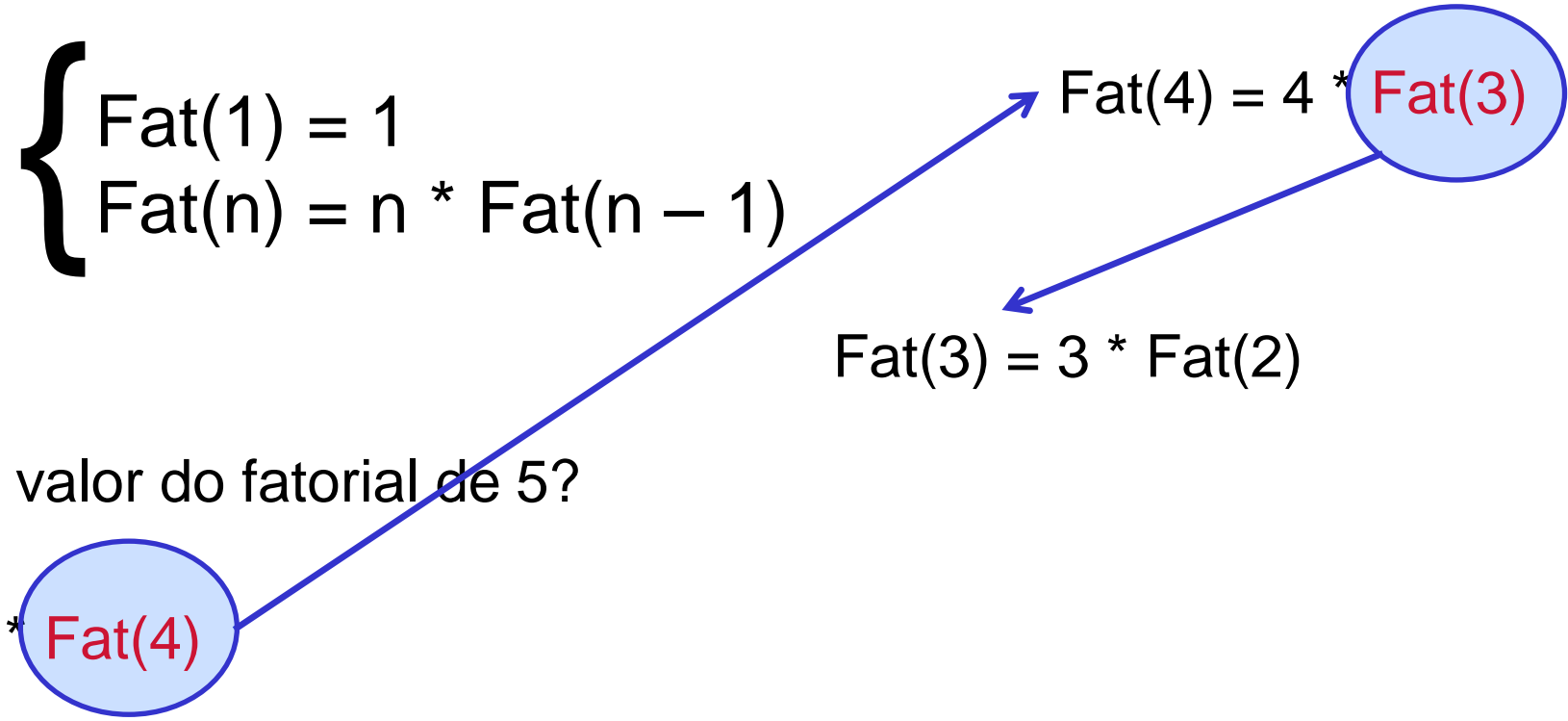
# Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

$$\text{Fat}(5) = 5 * \text{Fat}(4)$$

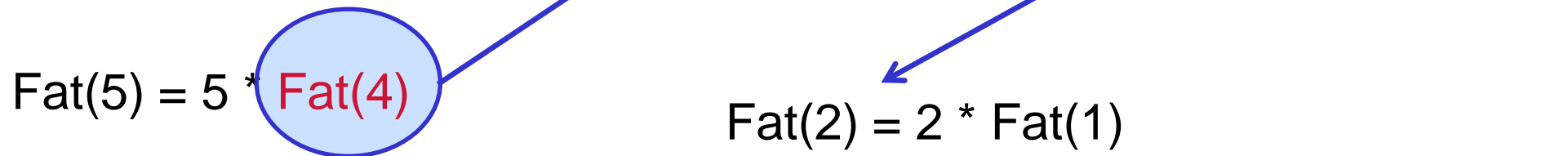
$$\text{Fat}(4) = 4 * \text{Fat}(3)$$
$$\text{Fat}(3) = 3 * \text{Fat}(2)$$


# Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

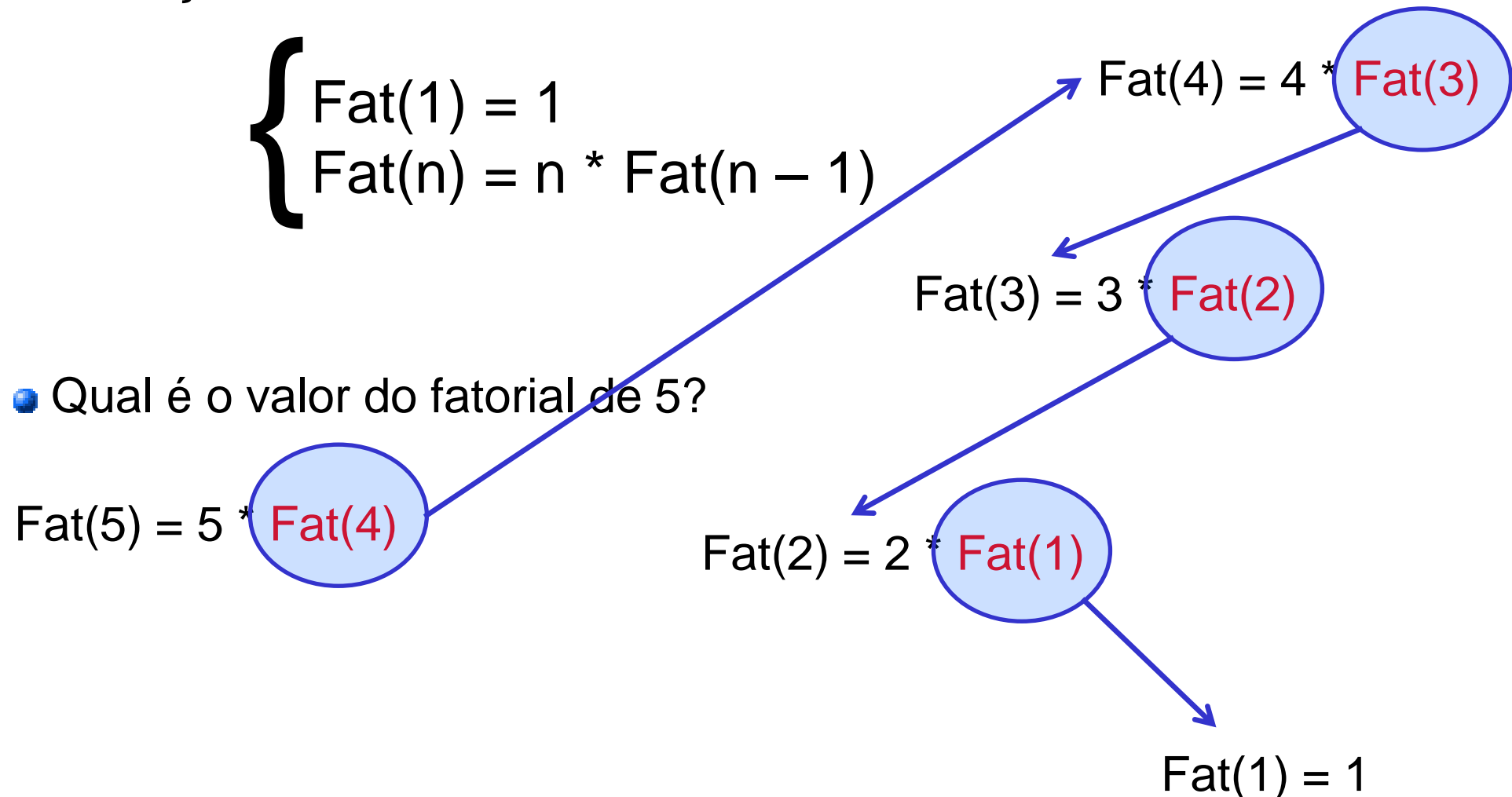


# Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

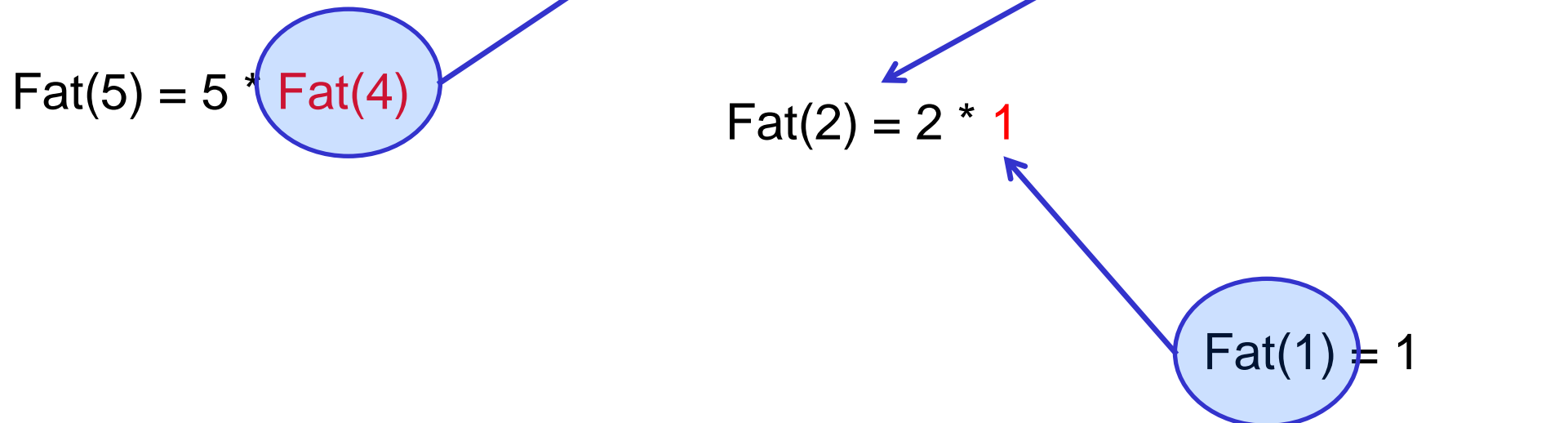


# Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

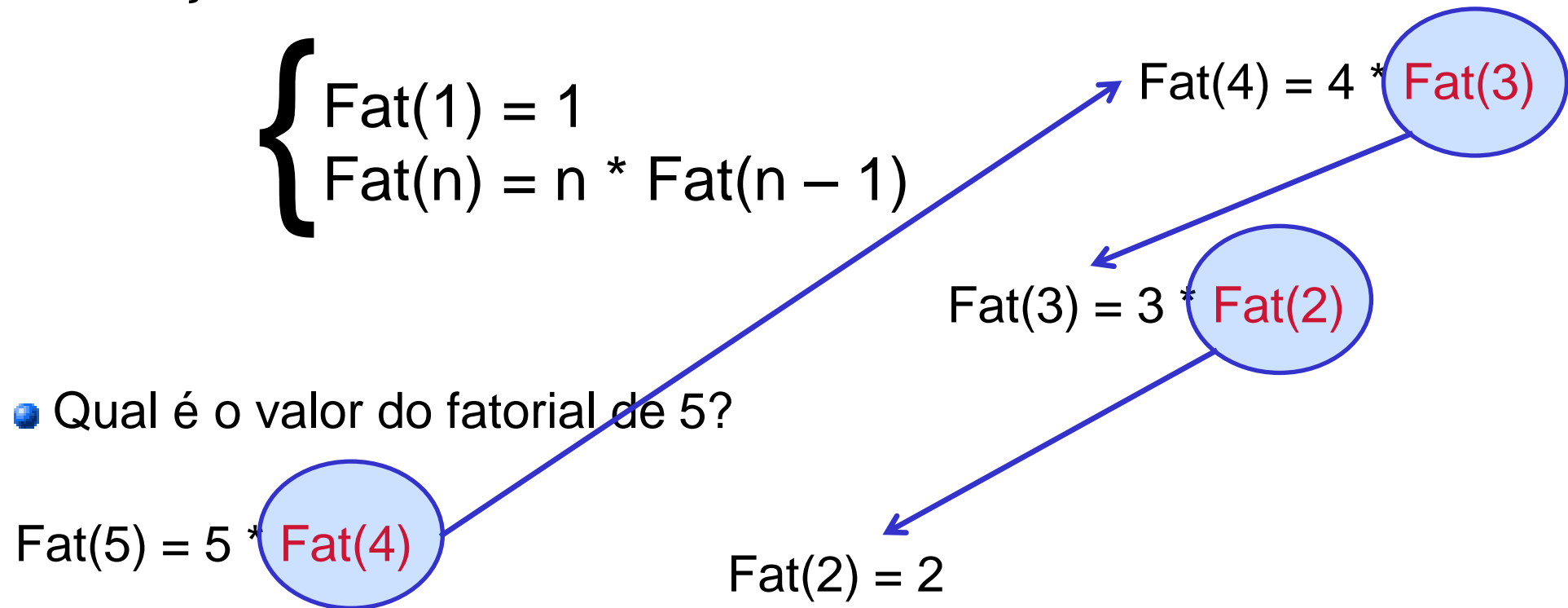


# Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?



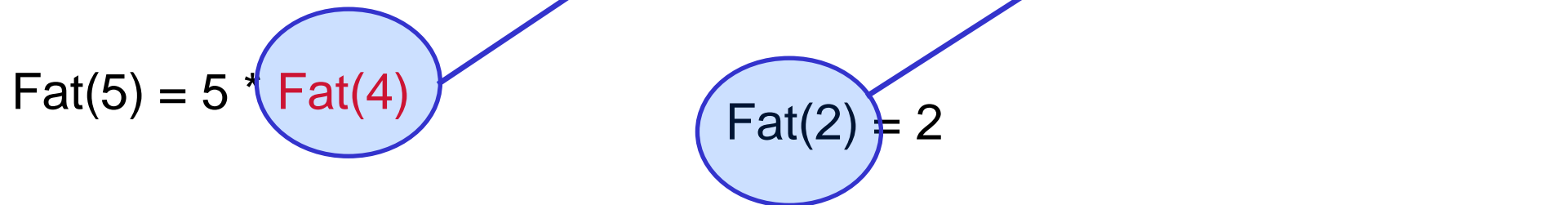


# Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?



# Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?



# Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

$\text{Fat}(5) = 5 * \text{Fat}(4)$

$\text{Fat}(3) = 6$

$\text{Fat}(4) = 4 * 6$

# Exemplo: Fatorial Recursivo

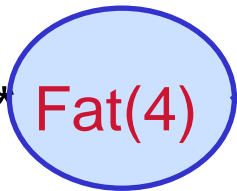
- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

$$\text{Fat}(4) = 24$$

- Qual é o valor do fatorial de 5?

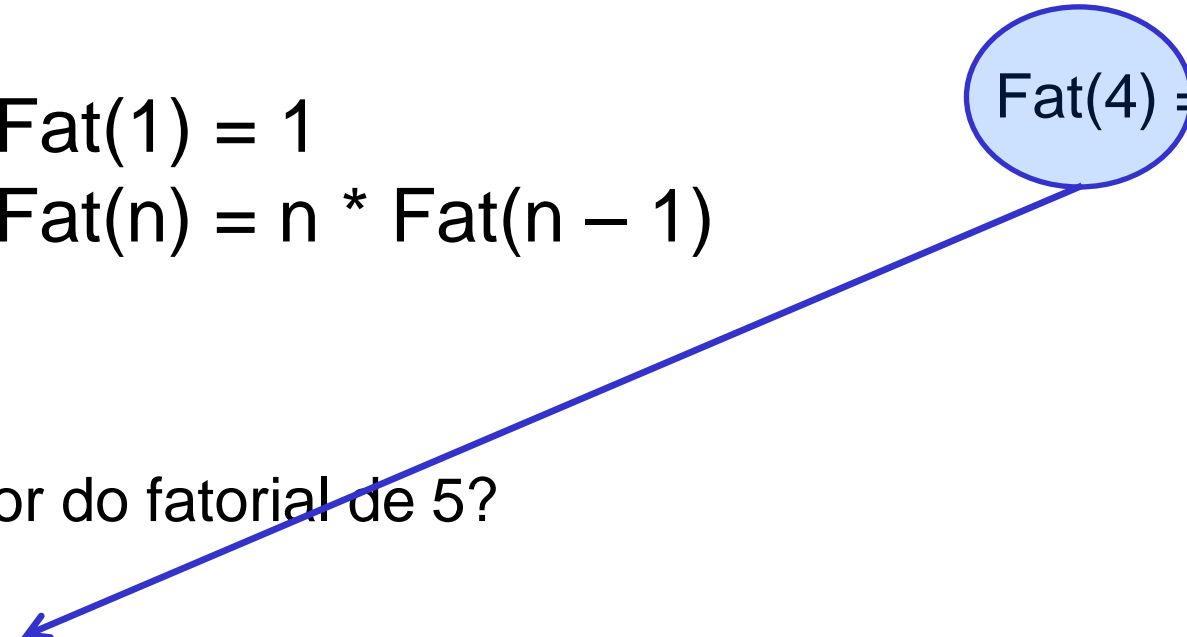
$$\text{Fat}(5) = 5 * \text{Fat}(4)$$



# Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$


$$\text{Fat}(4) = 24$$

- Qual é o valor do fatorial de 5?

$$\text{Fat}(5) = 5 * 24$$

# Exemplo: Fatorial Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fat}(1) = 1 \\ \text{Fat}(n) = n * \text{Fat}(n - 1) \end{cases}$$

- Qual é o valor do fatorial de 5?

$\text{Fat}(5) = 120$

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){  
    int resp;  
    if (n == 1){  
        resp = 1;  
    } else {  
        resp = n * fatorial(n - 1);  
    }  
    return resp;  
}  
void main(){  
    int valor = fatorial(5);  
    System.out.println(valor);  
}
```

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){  
    int resp;  
    if (n == 1){  
        resp = 1;  
    } else {  
        resp = n * fatorial(n - 1);  
    }  
    return resp;  
}
```

```
void main(){  
    int valor = fatorial(5);  
    System.out.println(valor);  
}
```



# Exemplo: Fatorial Recursivo

```
int fatorial (int n){  
    int resp;  
    if (n == 1){  
        resp = 1;  
    } else {  
        resp = n * fatorial(n - 1);  
    }  
    return resp;  
}  
void main(){  
    int valor = fatorial(5);  
    System.out.println(valor);  
}
```

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}
void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

$$\text{fatorial}(5) = 5 * \text{fatorial}(4)$$

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) =



# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) =





# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) =

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) =



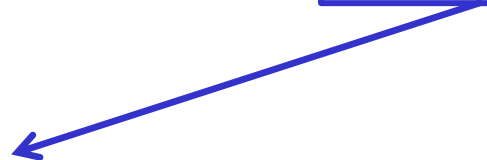
# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)



# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) =

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

←  
fatorial (4) = 4 \* fatorial (3)

←  
fatorial (3) =

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) =

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) =

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* fatorial (2)



# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* fatorial (2)

fatorial (2) =

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* fatorial (2)

fatorial (2) =

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* fatorial (2)

fatorial (2) =

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* fatorial (2)

fatorial (2) =

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* fatorial (2)

fatorial (2) = 2 \* fatorial (1)

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (1)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* fatorial (2)

fatorial (2) = 2 \* fatorial (1)

fatorial (1) =

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (1)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* fatorial (2)

fatorial (2) = 2 \* fatorial (1)

fatorial (1) =

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (1)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* fatorial (2)

fatorial (2) = 2 \* fatorial (1)

fatorial (1) =



# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (1)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* fatorial (2)

fatorial (2) = 2 \* fatorial (1)

fatorial (1) = 1

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (1)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* fatorial (2)

fatorial (2) = 2 \* fatorial (1)

fatorial (1) = 1

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* fatorial (2)

fatorial (2) = 2 \* 1

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* fatorial (2)

fatorial (2) = 2

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (2)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* fatorial (2)

fatorial (2) = 2

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 3 \* 2

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 6

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (3)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* fatorial (3)

fatorial (3) = 6



# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 4 \* 6



# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 24



# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (4)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 5 \* fatorial (4)

fatorial (4) = 24



# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

$$\text{fatorial}(5) = 5 * \boxed{24}$$

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 120

# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 120



# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```

fatorial (5) = 120



# Exemplo: Fatorial Recursivo

```
int fatorial (int n){ // n (5)
    int resp;
    if (n == 1){
        resp = 1;
    } else {
        resp = n * fatorial(n - 1);
    }
    return resp;
}

void main(){
    int valor = fatorial(5);
    System.out.println(valor);
}
```



# Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?

# Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?      `Fibonacci(4)`

# Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{cases}$$

- Qual é o Fibonacci de 4?

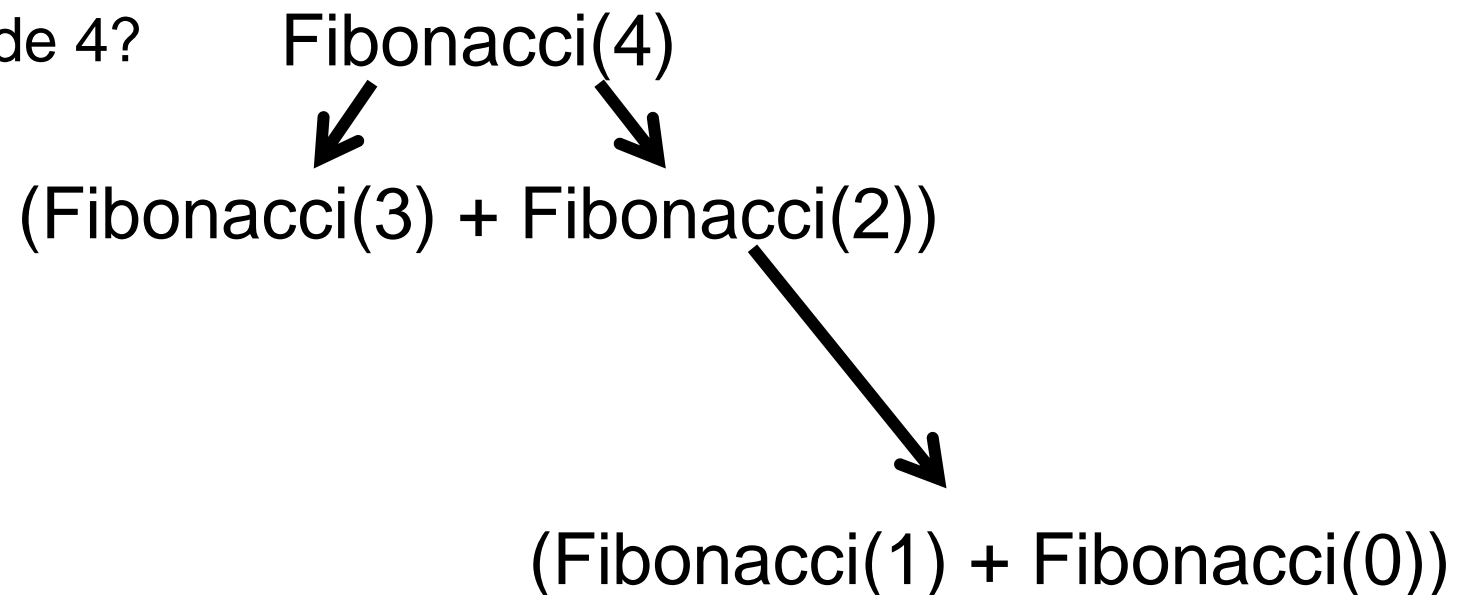
$$\begin{array}{c} \text{Fibonacci}(4) \\ \swarrow \quad \searrow \\ (\text{Fibonacci}(3) + \text{Fibonacci}(2)) \end{array}$$

# Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{cases}$$

- Qual é o Fibonacci de 4?

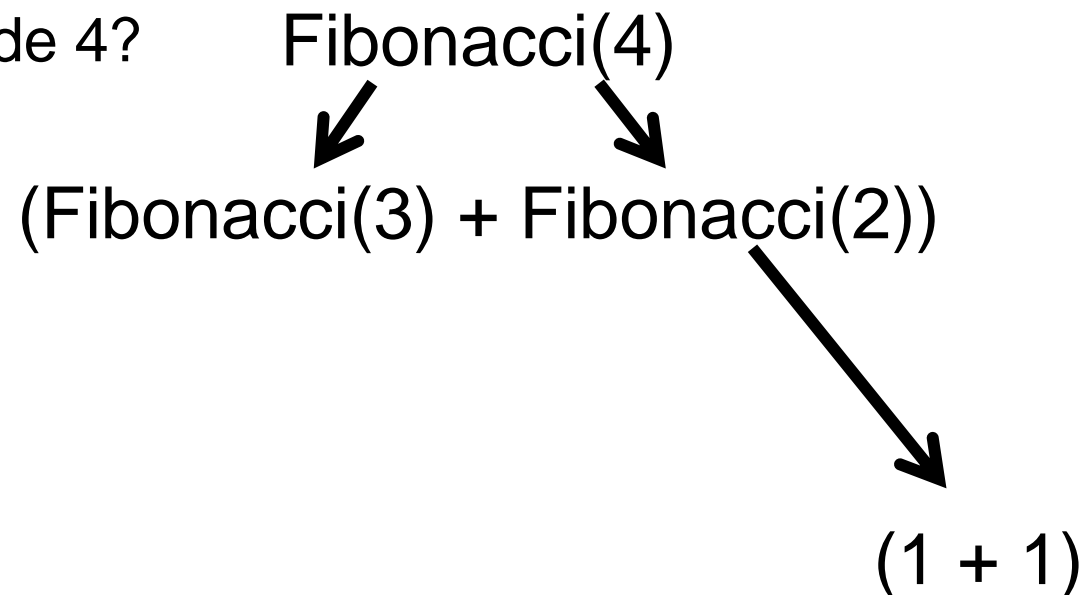


# Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{cases}$$

- Qual é o Fibonacci de 4?



# Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{cases}$$

- Qual é o Fibonacci de 4?

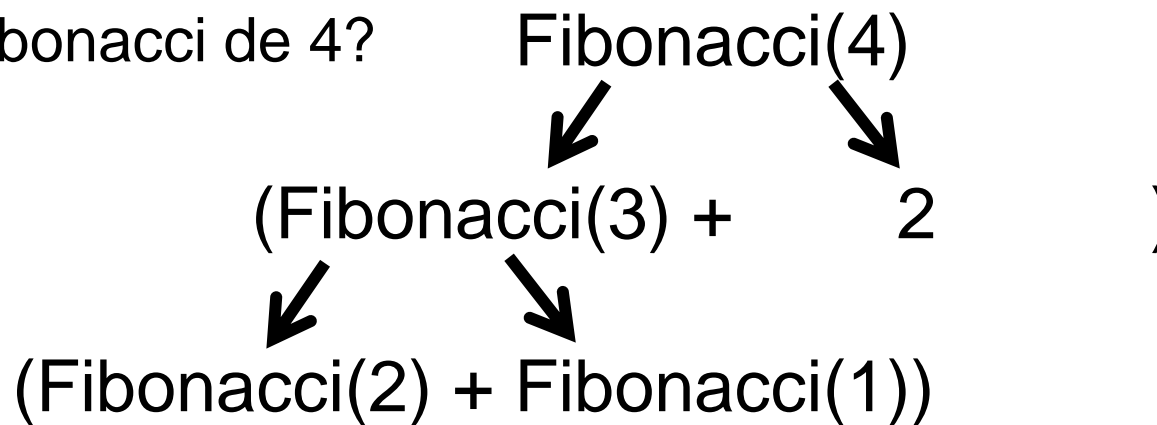
$$\begin{array}{c} \text{Fibonacci}(4) \\ \swarrow \quad \searrow \\ (\text{Fibonacci}(3) + \quad 2 \quad ) \end{array}$$

# Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{cases}$$

- Qual é o Fibonacci de 4?

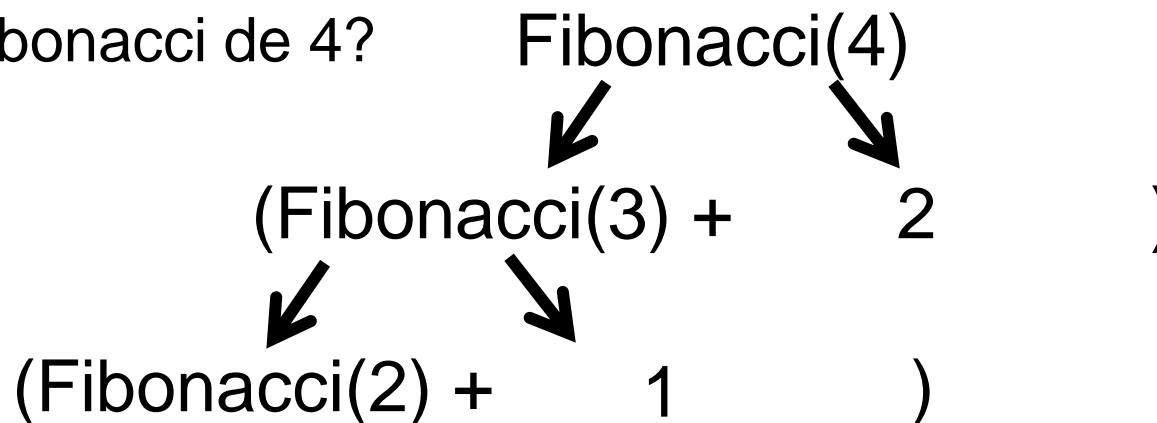


# Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{cases}$$

- Qual é o Fibonacci de 4?



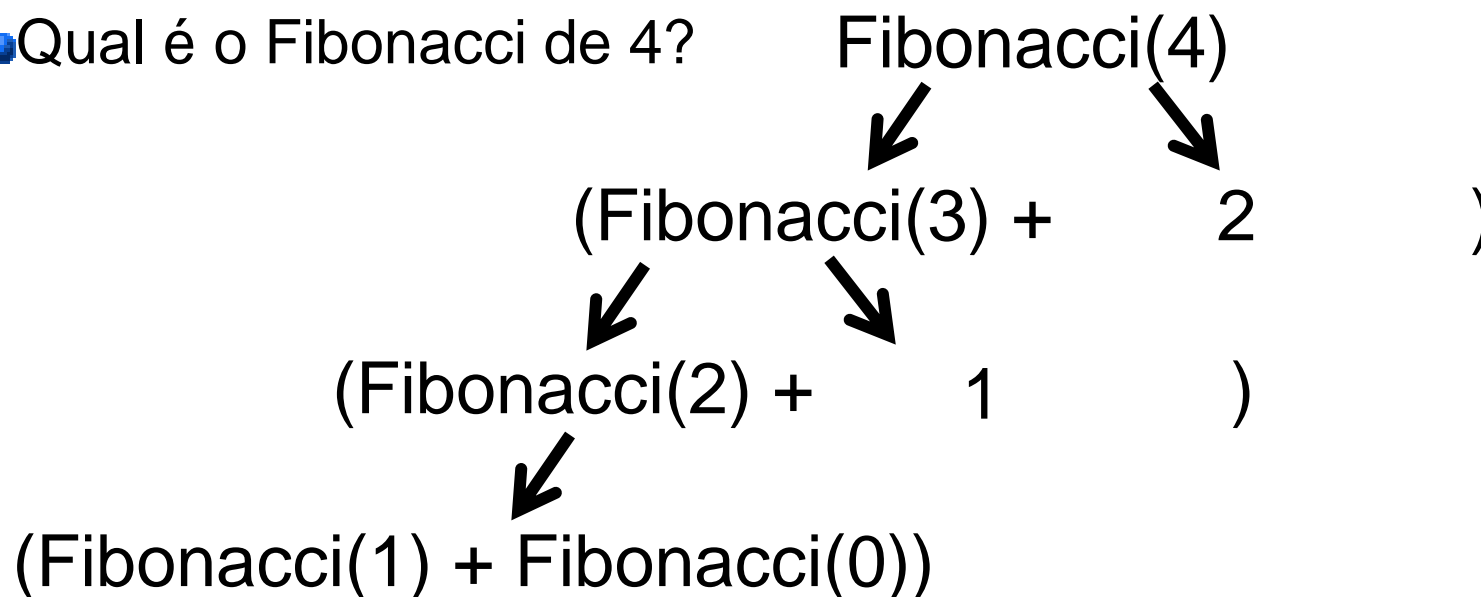


# Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{cases}$$

- Qual é o Fibonacci de 4?

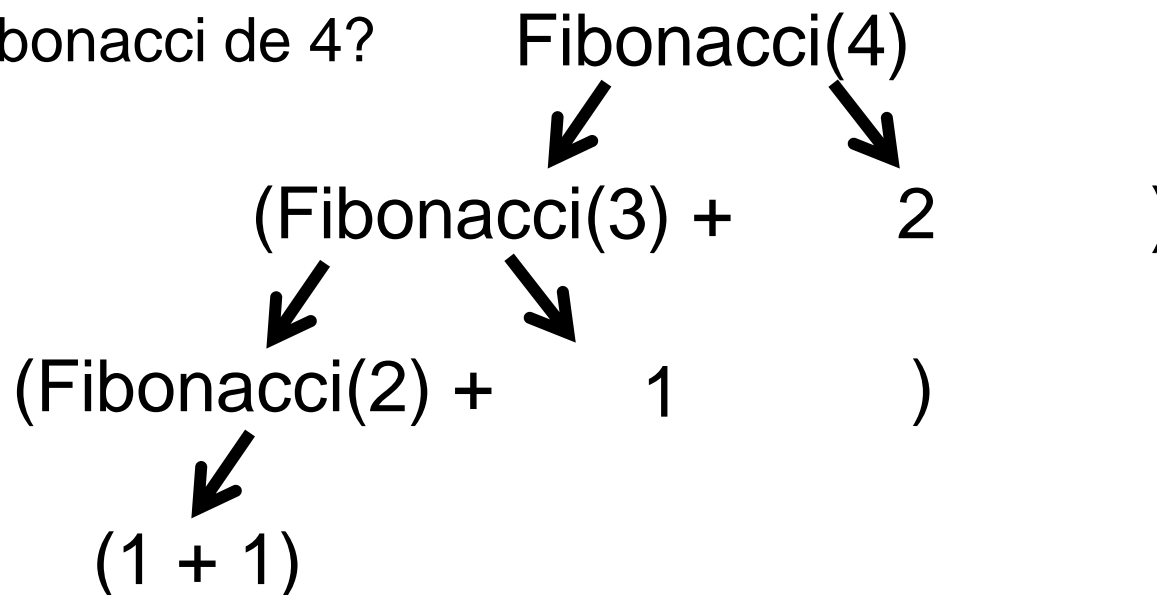


# Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{cases}$$

- Qual é o Fibonacci de 4?

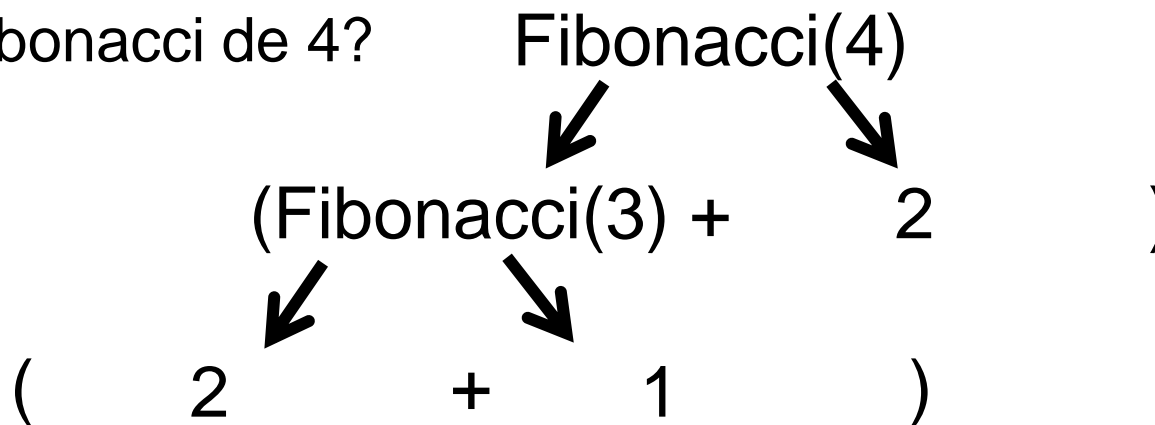


# Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{cases}$$

- Qual é o Fibonacci de 4?



# Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\begin{cases} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{cases}$$

- Qual é o Fibonacci de 4?

Fibonacci(4)

( 3 + 2 )

# Exemplo: Fibonacci Recursivo

- Definição do fatorial é recursiva:

$$\left\{ \begin{array}{l} \text{Fib}(0) = 1 \\ \text{Fib}(1) = 1 \\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \end{array} \right.$$

- Qual é o Fibonacci de 4?

5

# Exemplo: Fibonacci Recursivo

```
int fibonacci (int n){  
    int resp;  
    if (n == 0 || n == 1){  
        resp = 1;  
    } else {  
        resp = fibonacci(n - 1) + fibonacci(n - 2);  
    }  
    return resp;  
}  
  
void main(){  
    int valor = fibonacci(4);  
    System.out.println(valor);  
}
```

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas

$$5 \times 13 = \underbrace{5 + 5 + \dots + 5}_{13 \text{ vezes}}$$

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas

```
int multiplicacao (int a, int b){  
    int resp = 0;  
  
    if (b > 0){  
        resp = a + multiplicacao(a, b - 1);  
    }  
  
    return resp;  
}  
void main (...){  
    multiplicacao(4, 3);  
}
```



## Exercício

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas (**outra resposta**)

```
int multiplicacao (int a, int b, int i){  
    int resp = 0;  
    if (i < b){  
        resp = a + multiplicacao(a, b, i + 1);  
    }  
    return resp;  
}  
  
int multiplicacao (int a, int b){  
    return multiplicacao(a, b, 0);  
}  
  
void main (...){  
    multiplicacao(4, 3);  
}
```

## Exercício

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas (**outra resposta**)

```
int multiplicacao (int a, int b, int i){  
    int resp = 0;  
  
    if (i < b){  
        resp = a + multiplicacao(a, b, i + 1);  
    }  
  
    return resp;  
}  
  
int multiplicacao (int a, int b){  
    return multiplicacao(a, b, 0);  
}
```

```
int multiplicacao (int a, int b){  
    int resp = 0;  
  
    for (int i = 0; i < b; i = i + 1){  
        resp += a;  
    }  
  
    return resp;  
}
```

## Exercício

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas (**outra resposta**)

```
int multiplicacao (int a, int b, int i){  
    int resp = 0;  
  
    if (i < b){  
        resp = a + multiplicacao(a, b, i + 1);  
    }  
  
    return resp;  
}  
  
int multiplicacao (int a, int b){  
    return multiplicacao(a, b, 0);  
}
```

```
int multiplicacao (int a, int b){  
    int resp = 0;  
  
    for (int i = 0; i < b; i = i + 1){  
        resp += a;  
    }  
  
    return resp;  
}
```

## Exercício

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas (**outra resposta**)

```
int multiplicacao (int a, int b, int i){  
    int resp = 0;  
  
    if (i < b){  
        resp = a + multiplicacao(a, b, i + 1);  
    }  
  
    return resp;  
}  
  
int multiplicacao (int a, int b){  
    return multiplicacao(a, b, 0);  
}
```

```
int multiplicacao (int a, int b){  
    int resp = 0;  
  
    for (int i = 0; i < b; i = i + 1){  
        resp += a;  
    }  
  
    return resp;  
}
```

## Exercício

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas (**outra resposta**)

```
int multiplicacao (int a, int b, int i){  
    int resp = 0;  
  
    if (i < b){  
        resp = a + multiplicacao(a, b, i + 1);  
    }  
  
    return resp;  
}  
  
int multiplicacao (int a, int b){  
    return multiplicacao(a, b, 0);  
}
```

```
int multiplicacao (int a, int b){  
    int resp = 0;  
  
    for (int i = 0; i < b; i = i + 1){  
        resp += a;  
    }  
  
    return resp;  
}
```

## Exercício

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas (**outra resposta**)

```
int multiplicacao (int a, int b, int i){  
    int resp = 0;  
  
    if (i < b){  
        resp = a + multiplicacao(a, b, i + 1);  
    }  
  
    return resp;  
}  
  
int multiplicacao (int a, int b){  
    return multiplicacao(a, b, 0);  
}
```

```
int multiplicacao (int a, int b){  
    int resp = 0;  
  
    for (int i = 0; i < b; i = i + 1){  
        resp += a;  
    }  
  
    return resp;  
}
```

## Exercício

- Faça um método recursivo que receba dois números inteiros e retorne a multiplicação do primeiro pelo segundo fazendo somas (**outra resposta**)

```
int multiplicacao (int a, int b, int i){  
    int resp = 0;  
  
    if (i < b){  
        resp = a + multiplicacao(a, b, i + 1);  
    }  
  
    return resp;  
}  
  
int multiplicacao (int a, int b){  
    return multiplicacao(a, b, 0);  
}
```

```
int multiplicacao (int a, int b){  
    int resp = 0;  
  
    for (int i = 0; i < b; i = i + 1){  
        resp += a;  
    }  
  
    return resp;  
}
```

## Exercício

- Faça um método recursivo que receba um *array* de números inteiros e um número inteiro *n* indicando o tamanho do *array* e retorne o maior elemento



## Exercício

- Faça um método recursivo que receba um *array* de números inteiros e um número inteiro *n* indicando o tamanho do *array* e retorne o maior elemento

```
int maior (int vet[], int n){  
    return maior (vet, n, 0);  
}  
int maior (int vet[], int n, int i){  
    int resp;  
    if (i == n - 1){  
        resp = vet[n - 1];  
    } else {  
        resp = maior(vet, n, i + 1);  
        if (resp < vet[i]){  
            resp = vet[i];  
        }  
    }  
    return resp;  
}
```

- Faça um método recursivo que receba um *array* de caracteres e retorne um valor booleano indicando se esse é um palíndromo

- Faça um método recursivo que receba um *array* de caracteres e retorne um valor booleano indicando se esse é um palíndromo

```
bool isPalindromo(char string[]){  
    return isPalindromo(string, 0);  
}  
bool isPalindromo(char string[], int i){  
    bool resp;  
    if (i > tamanho(string) / 2){  
        resp = true;  
    } else if (string[i] != string[tamanho(string) - 1 - i]){  
        resp = false;  
    } else {  
        resp = isPalindromo(string, i + 1);  
    }  
    return resp;  
}
```

- Faça um método recursivo que receba um *array* de caracteres e retorne um número inteiro indicando a quantidade de vogais do mesmo

## Exercício

- Faça um método recursivo que receba um *array* de caracteres e retorne um número inteiro indicando a quantidade de vogais do mesmo

```
char toUpper(char c){  
    return (c >= 'a' && c <= 'z') ? (char)(c - 32) : c ;  
}
```

- Faça um método recursivo que receba um *array* de caracteres e retorne um número inteiro indicando a quantidade de vogais do mesmo

```
int contMaiusculo (String s){  
    return contMaiusculo (s, 0);  
}  
int contMaiusculo (String s, int i){  
    int cont = 0;  
  
    if (i < s.length()){  
        if (isUpper(s.charAt(i)) == true){  
            cont++;  
        }  
        cont += contMaiusculo (s, i + 1);  
    }  
    return cont;  
}
```

- Faça um método recursivo que receba um string e retorne um número inteiro indicando a quantidade de caracteres NOT vogal AND NOT consoante maiúscula da string recebida como parâmetro
- Faça um método recursivo que receba um *array* de inteiros e os ordene

## Exercício

- Faça um método recursivo para cada um dos problemas abaixo

$$\begin{cases} T(0) = 1 \\ T(1) = 2 \\ T(n) = T(n-1) * T(n-2) - T(n-1) \end{cases}$$

$$\begin{cases} T(0) = 1 \\ T(n) = T(n-1)^2 \end{cases}$$



# Considerações

- Todo programa iterativo pode ser feito de forma recursiva e vice-versa
  - Algumas vezes é mais “fácil” fazer um programa de forma recursiva!!!
- O conceito de recursividade é fundamental na computação e na matemática (por exemplo, número naturais, fatorial e outros)
- A recursividade pode ser direta ou indireta (A chama B que chama A)

# Considerações

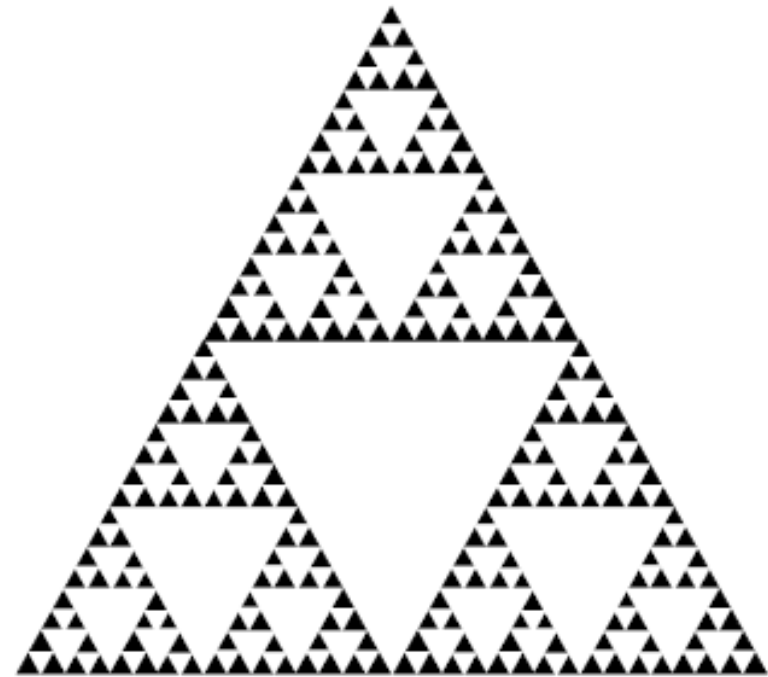
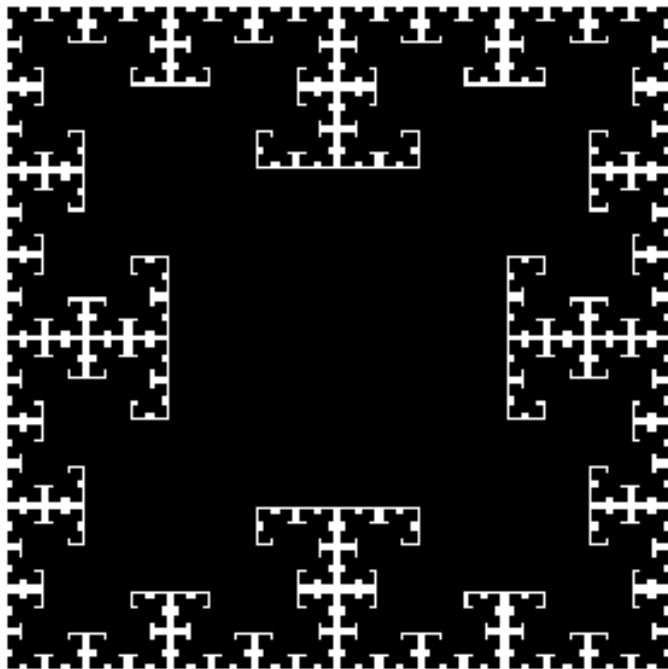
- O SO usa uma pilha para armazenar o estado corrente do programa antes de cada chamada não terminada e quando uma chamada termina, o SO recupera o estado armazenado na pilha
  - As variáveis locais são recriadas para cada chamada recursiva
- Por que na prática é importante manter um nível “limitado” de chamadas recursivas?

# Considerações

- Um dos principais exemplos sobre recursividade apresentados na literatura (e neste material também!!!) é o Fibonacci
- Qual é a sua opinião sobre as implementações recursiva e iterativa do Fibonacci?

# Considerações

- Outro exemplo de recursividade são os fractais, pequenos padrões geométricos que ao serem repetidos diversas vezes de forma recursiva criam desenhos mais sofisticadas



## Exercício

- Pesquisar e implementar uma solução recursiva para o problema das Torres de Hanói, dado o número de pinos

