

# Linguagem de Programação 2

## **5b: Programação por Refinamentos Sucessivos**

**Prof. Flávio José Mendes Coelho**

# Programação por refinamentos sucessivos

- É uma técnica de programação.
- Também denominada abordagem ***Top-down***
  1. Escreva uma função para resolver um problema geral.
  2. Quebre o problema em subproblemas mais simples e, na função, escreva uma função para cada subproblema.
  3. Dentro de cada função realize o passo 2, até que a função escrita tenha somente comandos simples e não chame outras funções.

# Exemplo

## **Problema: Limpar casa.**

Criar uma função principal para o problema geral de **limpar uma casa**.

```
int main() {  
    ???  
    return 0;  
}
```

## **Problema: Limpar casa**

(como este problema pode ser quebrado em subproblemas mais simples?)

- Subproblema: Limpar lado de dentro da casa.
- Subproblema: Limpar lado de fora da casa.

```
int main() {  
    limparDentro();  
    limparFora();  
    return 0;  
}
```

```
limparDentro() {???
```

```
int main() {  
    limparDentro();  
    limparFora();  
    return 0;  
}
```

## Problema: Limpar casa

- Subproblema: Limpar lado de dentro (**como...?**).
  - Limpar quartos.
  - Lavar louça.
- Subproblema: Limpar lado de fora. (fica para depois!)

```
limparDentro() {  
    limparQuartos();  
    lavarLouça();  
}
```

```
int main() {  
    limparDentro();  
    limparFora();  
    return 0;  
}
```



```
limparQuartos() {???
```

```
limparDentro() {  
    limparQuartos();  
    lavarLouça();  
}
```

```
int main() {...}
```

## Problema: Limpar casa

- Subproblema: Limpar lado de dentro.
  - Subproblema: Limpar quartos (como...?).  
Limpar quarto 1, quarto 2, etc.
    - Subproblema: Limpar um quarto.
  - Lavar louça.
- Subproblema: Limpar lado de fora.

```
limparQuartos() {  
    for (int i = 1; i <= NumQuartos; i++) {  
        limparQuarto();  
    }  
}  
limparDentro() {  
    limparQuartos();  
    lavarLouça();  
}  
int main() {...}
```

```
limparQuarto() {???
```

```
limparQuartos() {  
    for (int i = 1; i <= NumQuartos; i++) {  
        limparQuarto();  
    }
```

```
}
```

```
limparDentro() {...}
```

```
int main() {...}
```

## Problema: Limpar casa

- Subproblema: Limpar lado de dentro.
  - Subproblema: Limpar quartos.  
Limpar quarto 1, quarto 2, etc.
  - Subproblema: Limpar um quarto (como...?).
    - Subproblema: varrer.
    - Subproblema: passar desinfetante.
  - Lavar louça.
- Subproblema: Limpar lado de fora.

```
limparQuarto() {  
    varrer();  
    passarDesinfetante ();  
}  
limparQuartos() {  
    for (int i = 1; i <= NumQuartos; i++) {  
        limparQuarto();  
    }  
}  
limparDentro() {...}  
int main() {...}
```

```
varrer() {???
```

```
limparQuarto() {
```

```
    varrer();
```

```
    passarDesinfetante ();
```

```
}
```

```
limparQuartos() {
```

```
    for (int i = 1; i <= NumQuartos; i++) {
```

```
        limparQuarto();
```

```
    }
```

```
}
```

```
limparDentro() {...}
```

```
int main() {...}
```

## Problema: Limpar casa

- Subproblema: Limpar lado de dentro.
  - Subproblema: Limpar quartos.  
Limpar quarto 1, quarto 2, etc.
  - Subproblema: Limpar um quarto.
    - Subproblema: varrer (como...?).  
» Segurar a vassoura e varrer.
    - Subproblema: passar desinfetante.
    - Lavar louça.
- Subproblema: Limpar lado de fora.



```
varrer() {  
    segurar a vassoura e varrer;  
}
```

```
limparQuarto() {  
    varrer();  
    passarDesinfetante();  
}
```

```
limparQuartos() {...}
```

```
limparDentro() {...}
```

```
int main() {...}
```

varrer() {...}

passarDesinfetante() {???

limparQuarto() {  
 varrer();  
 passarDesinfetante();  
}

limparQuartos() {...}

limparDentro() {...}

int main() {...}

## Problema: Limpar casa

- Subproblema: Limpar lado de dentro.
  - Subproblema: Limpar quartos.  
Limpar quarto 1, quarto 2, etc.
  - Subproblema: Limpar um quarto.
    - Subproblema: varrer.
      - » Segurar a vassoura e varrer.
    - Subproblema: passar desinfetante (como...?).
      - » Molhar chão com desinfetante e passar pano.
    - Lavar louça.
- Subproblema: Limpar lado de fora.

```
varrer() {...}  
passarDesinfetante() {  
    Molhar chão com desinfetante e passar pano;  
}  
limparQuarto() {  
    varrer();  
    passarDesinfetante();  
}  
limparQuartos() {...}  
limparDentro() {...}  
int main() {...}
```

```
varrer() {...}  
passarDesinfetante() {  
    Molhar chão com desinfetante e passar pano;  
}  
limparQuarto() {  
    varrer();  
    passarDesinfetante();  
}  
limparQuartos() {...}  
limparDentro() {...}  
int main() {...}
```

```
varrer() {...}  
passarDesinfetante() {...}  
limparQuarto() {  
    varrer();  
    passarDesinfetante();  
}  
limparQuartos() {...}  
limparDentro() {...}  
int main() {...}
```

```
varrer() {...}  
passarDesinfetante() {...}  
limparQuarto() {...}  
limparQuartos() {  
    for (int i = 1; i <= NumQuartos; i++) {  
        limparQuarto();  
    }  
}  
limparDentro() {...}  
int main() {...}
```

```
varrer() {...}  
passarDesinfetante() {...}  
limparQuarto() {...}  
limparQuartos() {...}  
limparDentro() {  
    limparQuartos();  
    lavarLouça();  
}  
int main() {...}
```



...

lavarLouça() {???

limparDentro() {  
 limparQuartos();  
 lavarLouça();

}

int main() {...}

## Problema: Limpar casa

- Subproblema: Limpar lado de dentro.
  - Subproblema: Limpar quartos.  
Limpar quarto 1, quarto 2, etc.
  - Subproblema: Limpar um quarto.
    - Subproblema: varrer.
      - » Segurar a vassoura e varrer.
    - Subproblema: passar desinfetante.
      - » Molhar chão com desinfetante e passar pano.
  - Lavar louça (como...?).
    - enxaguar, passar detergente na louças,  
enxaguar, enxugar, guardar.
- Subproblema: Limpar lado de fora.

```
lavarLouça() {  
    enxaguar();  
    passar detergente;  
    enxaguar();  
    enxugar();  
    guardar();  
}  
limparDentro() {  
    limparQuartos();  
    lavarLouça();  
}
```

```
enxaguar() {???
```

```
enxugar() {???
```

```
guardar() {???
```

```
lavarLouça() {
```

```
    enxaguar();
```

```
    passar detergente;
```

```
    enxaguar();
```

```
    enxugar();
```

```
    guardar();
```

```
}
```

```
limparDentro() {
```

```
    limparQuartos();
```

```
    lavarLouça();
```

```
}
```

```
enxaguar() {molhar a louça com água;}
enxugar() {secar a louça com pano seco e limpo;}
guardar() {guardar cada louça em uma armário;}
lavarLouça() {
    enxaguar();
    passar detergente;
    enxaguar();
    enxugar();
    guardar();
}
limparDentro() {
    limparQuartos();
    lavarLouça();
}
```

```
enxaguar() {...}  
enxugar() {...}  
guardar() {...}  
lavarLouça() {  
    enxaguar();  
    passar detergente;  
    enxaguar();  
    enxugar();  
    guardar();  
}  
limparDentro() {  
    limparQuartos();  
    lavarLouça();  
}
```

```
enxaguar() {...}  
enxugar() {...}  
guardar() {...}  
lavarLouça() {...}  
limparDentro() {  
    limparQuartos();  
    lavarLouça();  
}
```

```
varrer() {...}  
passarDesinfetante() {...}  
limparQuarto() {...}  
limparQuartos() {...}  
enxaguar() {...}  
enxugar() {...}  
guardar() {...}  
lavarLouça() {...}  
limparDentro() {  
    limparQuartos();  
    lavarLouça();  
}
```



varrer() {...}  
passarDesinfetante() {...}  
limparQuarto() {...}  
limparQuartos() {...}  
enxaguar() {...}  
enxugar() {...}  
guardar() {...}  
lavarLouça() {...}  
limparDentro() {...}

```
varrer() {...}  
passarDesinfetante() {...}  
limparQuarto() {...}  
limparQuartos() {...}  
enxaguar() {...}  
enxugar() {...}  
guardar() {...}  
lavarLouça() {...}  
limparDentro() {...}  
int main() {  
    limparDentro();  
    limparFora();  
    return 0;  
}
```

```
limparFora() {???
```

```
int main() {  
    limparDentro();  
    limparFora();  
    return 0;  
}
```

## Problema: Limpar casa

- Subproblema: Limpar lado de dentro.
  - Subproblema: Limpar quartos.  
Limpar quarto 1, quarto 2, etc.
  - Subproblema: Limpar um quarto.
    - Subproblema: varrer.
      - » Segurar a vassoura e varrer.
    - Subproblema: passar desinfetante.
      - » Molhar chão com desinfetante e passar pano.
  - Lavar louça.
    - enxaguar, passar detergente na louças, enxaguar, enxugar, guardar.
- Subproblema: Limpar lado de fora (**como...?**).
  - **Aguar jardim.**
  - **Lavar pátio.**

```
limparFora() {  
    aguarJardim();  
    lavarPatio();  
}
```

```
int main() {  
    limparDentro();  
    limparFora();  
    return 0;  
}
```

```
aguarJardim() {???
```

```
limparFora() {  
    aguarJardim();  
    lavarPatio();  
}
```

```
int main() {  
    limparDentro();  
    limparFora();  
    return 0;  
}
```

```
aguarJardim() {  
    molhar com o esguicho todas as plantas, grama  
    e árvores do jardim;  
}
```

```
limparFora() {  
    aguarJardim();  
    lavarPatio();  
}
```

```
int main() {  
    limparDentro();  
    limparFora();  
    return 0;  
}
```

```
aguarJardim() {  
    molhar com o esguicho todas as plantas, grama  
    e árvores do jardim;  
}  
lavarPatio() {???  
limparFora() {  
    aguarJardim();  
    lavarPatio();  
}  
int main() {  
    limparDentro();  
    limparFora();  
    return 0;  
}
```



```
aguarJardim() {  
    molhar com o esguicho todas as plantas, grama  
    e árvores do jardim;  
}  
lavarPatio() {  
    molhar pátio e varrer a sujeira;  
}  
limparFora() {  
    aguarJardim();  
    lavarPatio();  
}  
  
int main() { imparDentro(); limparFora(); return 0;}
```

```
aguarJardim() {...}  
lavarPatio() {...}  
limparFora() {  
    aguarJardim();  
    lavarPatio();  
}
```

```
int main() { imparDentro(); limparFora(); return 0;}
```

```
aguarJardim() {...}  
lavarPatio() {...}  
limparFora() {...}  
int main() { imparDentro(); limparFora(); return 0;}
```

```
varrer() {...}  
passarDesinfetante() {...}  
limparQuarto() {...}  
limparQuartos() {...}  
enxaguar() {...}  
enxugar() {...}  
guardar() {...}  
lavarLouça() {...}  
limparDentro() {...}  
aguarJardim() {...}  
lavarPatio() {...}  
limparFora() {...}  
int main() { limparDentro(); limparFora(); return 0;}
```

# Exemplo

- **Problema:** calcular a média dos valores reais de um vetor

1. Escreva uma função para resolver um problema.

```
main() {  
    int tam = 5  
    float notas[tam];  
    ??  
}
```

# Programação por refinamentos sucessivos

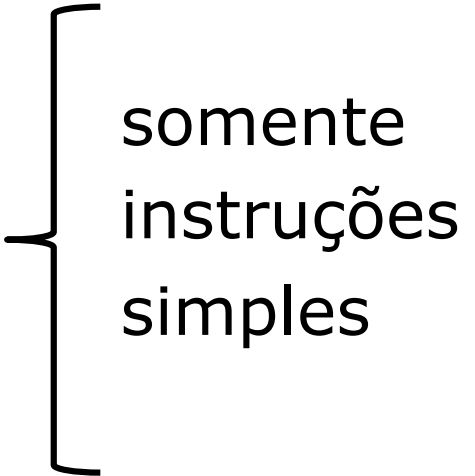
- Problema: calcular a média dos valores reais de um vetor
2. Na função, quebre o problema em subproblemas e escreva uma função para cada subproblema.

```
main() {  
    const int tam = 5;  
    float notas[tam];  
    lerVetor(notas, tam);  
    int media = calcularMedia(notas, tam);  
    mostrarMedia(notas, tam, media);  
}
```

# Programação por refinamentos sucessivos

- Problema: calcular a média dos valores reais de um vetor
3. Dentro de cada função realize o passo 2, até que a função possua somente instruções simples.

```
void lerVetor(float notas[], int tam) {  
    for (int i = 0; i < tam; i++) {  
        scanf("%f", &nota[i]);  
    }  
}
```



somente  
instruções  
simples

# Programação por refinamentos sucessivos

- Problema: calcular a média dos valores reais de um vetor
3. Dentro de cada função realize o passo 2, até que a função possua somente instruções simples.

```
float calcularMedia(float notas[], int tam) {  
    float soma = 0.0f;  
    // como somar tam números?  
}
```



# Programação por refinamentos sucessivos

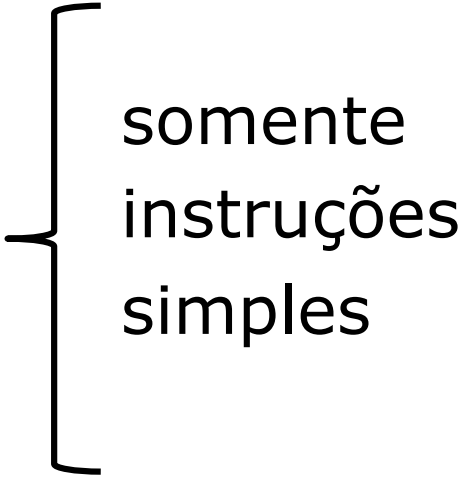
- Problema: calcular a média dos valores reais de um vetor
3. Dentro de cada função realize o passo 2, até que a função possua somente instruções simples.

```
float calcularMedia(float notas[], int tam) {  
    float soma = 0.0f;  
    soma = somarNotas(notas, tam);  
    float media = soma / tam;  
    return media;  
}
```

# Programação por refinamentos sucessivos

- Problema: calcular a média dos valores reais de um vetor
3. Dentro de cada função realize o passo 2, até que a função possua somente instruções simples.

```
float somarNotas (float notas[], int tam) {  
    float soma = 0.0f;  
    for (int i = 0; i < tam; i++) {  
        soma += notas[i];  
    }  
    return soma;  
}
```

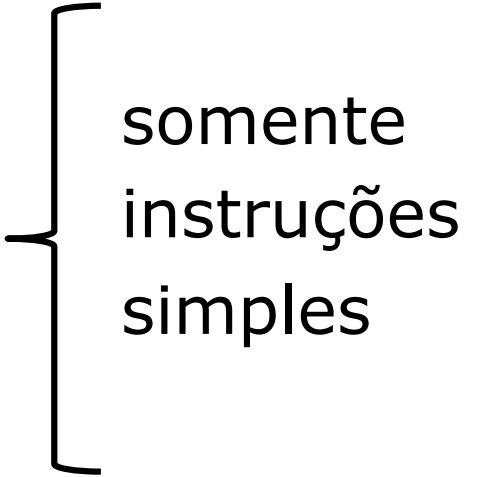


somente  
instruções  
simples

# Programação por refinamentos sucessivos

- Problema: calcular a média dos valores reais de um vetor
3. Dentro de cada função realize o passo 2, até que a função possua somente instruções simples.

```
void mostrarMedia (... notas[], ... tam, ... media) {  
    printf("%d\n", "notas:");  
    for (int i = 0; i < tam; i++) {  
        printf("%d", notas[i] ",");  
    }  
    printf("%d\n", media);  
}
```



somente  
instruções  
simples

# Escopo de variáveis

- Escopo de uma variável
  - Determina a visibilidade de um variável e seu tempo de vida
- Variáveis têm escopo global ou local

# Variáveis globais

- São visíveis por qualquer função do programa
- Devem ser usadas com **cuidado**, pois qualquer função pode mudar o valor de uma variável global

```
int a;
```

```
void incrementa() {  
    a++;  
    printf("incrementa():%d\n", a);  
}
```

```
main() {  
    a++;  
    incrementa();  
    printf("main():%d\n", a);  
}
```

# Variáveis locais

- São visíveis somente dentro da função onde são declaradas

```
void incrementa() {  
    b++; // erro: b só existe na função main()  
    printf("incrementa():%d\n", b);  
}
```

```
main() {  
    int b = 0;  
    incrementa();  
    printf("main():%d\n", b);  
}
```

# Variáveis locais

- Somente podem ser visíveis em outra função por meio de passagem de parâmetros

```
void incrementa(int b) {  
    b++;  
    printf("incrementa():%d\n", b);  
}
```

```
main() {  
    int b = 0;  
    incrementa(b);  
    printf("main():%d\n", b);  
}
```

# Bibliografia

- Deitel, H. M., Deitel, P. J. C - Como Programar. 6a. ed. Pearson, 2011.