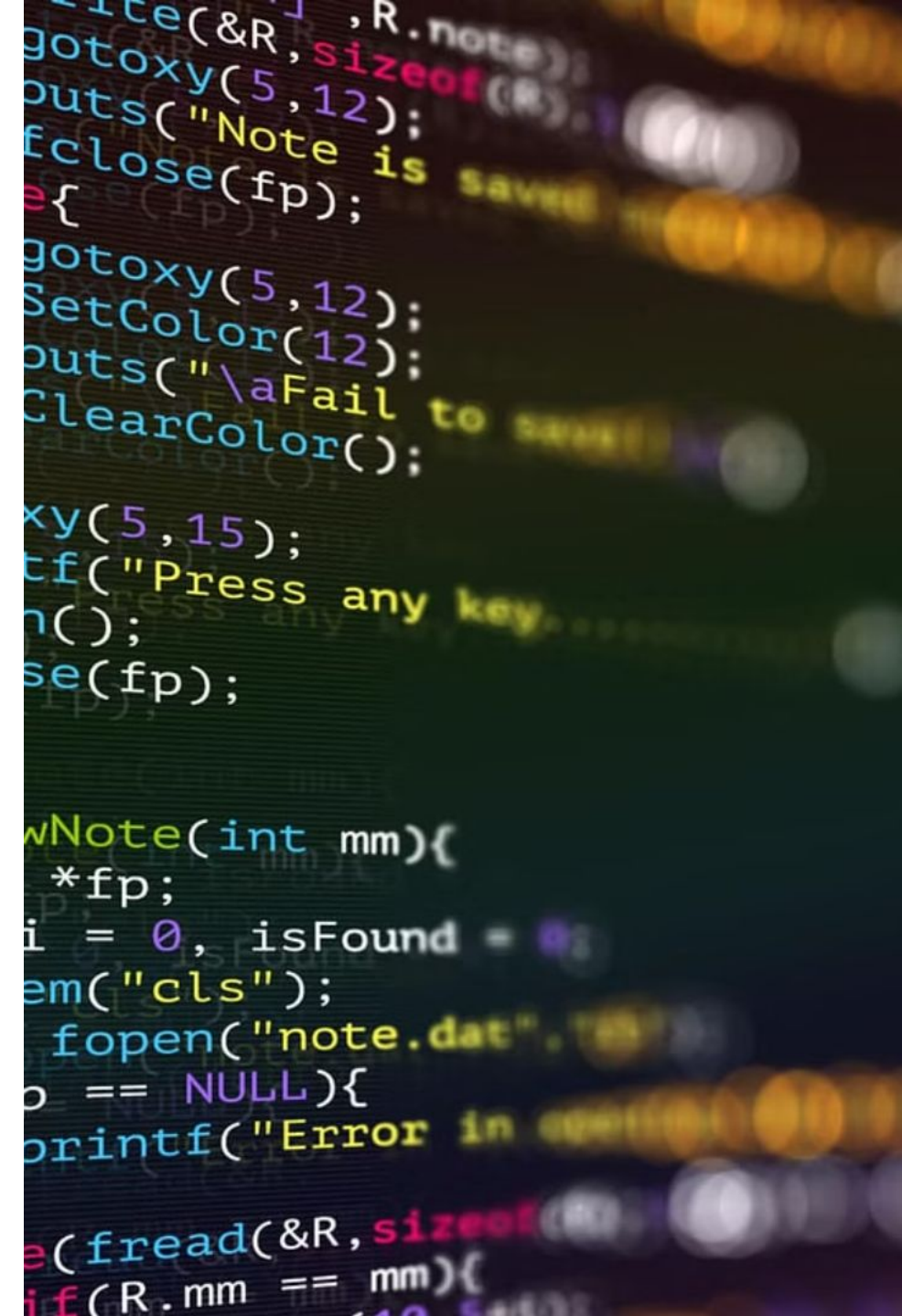


Estruturas condicionais e de repetição em C++

Disciplina: Programação Aplicada

Prof. Rodrigo Maciel

A blurred image of C++ code, likely from a text editor or IDE, showing various functions and control structures. The code is color-coded, with keywords in blue, identifiers in green, and literals in yellow. The background is dark, and the text is slightly out of focus, creating a bokeh effect with some light spots.

```
...ite(&R, sizeof(R).note);  
gotoxy(5, 12);  
puts("Note is saved");  
fclose(fp);  
...  
gotoxy(5, 12);  
SetColor(12);  
puts("\aFail to save");  
ClearColor();  
...  
xy(5, 15);  
tf("Press any key...");  
n();  
se(fp);  
...  
wNote(int mm){  
    *fp;  
    i = 0, isFound = 0;  
    em("cls");  
    fopen("note.dat", "a");  
    o == NULL){  
        printf("Error in opening file");  
    }  
    e(fread(&R, sizeof(R).note, 1, fp));  
    if(R.mm == mm){  
        ...  
    }  
}
```

Estruturas condicionais

- A estrutura condicional de decisão simples **serve para escolher um comando para execução, caso a condição for verdadeira;**
- A estrutura condicional ou de decisão composta serve para escolher um entre dois blocos de comandos alternativos;
- Na linguagem C++ pode ser utilizado o comando **if** ou **case**.

Estrutura condicional: if

- A instrução **IF** é a estrutura de teste mais básica e é encontrada em todas as linguagens (com sintaxes diferentes);
- Ela permite executar uma série de instruções caso uma condição se realize.

```
if (condição é verdadeira) {  
    executa instruções;  
}
```

Lógica com a condição if

Lógica com a Condição AND:

```
if ((condição1) && (condição2)) {  
    só entra se ambas as condições forem True  
}
```

Lógica com a Condição OR:

```
if ((condição1) || (condição2)) {  
    só entra se uma das condições forem True  
}
```

If Else

- O if/else ou se/senão, testa a primeira condição, se esta não for verdadeira, pula diretamente para o else.

```
if (condição é verdadeira) {  
    executa instruções;  
}  
else {  
    outra série de instruções  
}
```

Exemplo 1

- Comparação de dois valores:

```
int main() {  
    // Declaração de variáveis  
    double num1, num2;  
  
    // Solicita ao usuário que insira os dois números  
    std::cout << "Digite o primeiro número: ";  
    std::cin >> num1;  
    std::cout << "Digite o segundo número: ";  
    std::cin >> num2;  
  
    // Compara os números e imprime o maior  
    if (num1 > num2) {  
        std::cout << "O maior número é: " << num1 << std::endl;  
    } else if (num2 > num1) {  
        std::cout << "O maior número é: " << num2 << std::endl;  
    } else {  
        std::cout << "Os números são iguais." << std::endl;  
    }  
  
    return 0;  
}
```

Estrutura condicional: switch-case

- Também conhecida como estrutura de seleção múltipla, é composta por uma série de seleções simples encadeadas, onde:
 - Todas as seleções nas decisões são de igualdade;
 - Todas as condições comparam uma mesma expressão a uma constante;
 - Todas as constantes são consideradas do tipo inteiro ou caractere.

A instrução Switch Case

Ao encontrar o comando switch-case, o computador avalia a expressão e indica dentro do parênteses que segue a palavra **switch** e executa apenas o comando associado ao valor da constante tem o valor igual da expressão.

```
switch (condição de teste) {  
    case 0: instruções;  
        break;  
    case 1: instruções;  
        break;  
    default: intruções;  
        break;  
}
```


Outras considerações

- Tanto o comando ***break*** quanto o ***default***, são opcionais:
 - Se um caso é selecionado para a execução e não existe um break no seu final, então após a sua execução, a execução continua no caso seguinte;
 - Se o caso default é omitido, e nenhum dos demais casos tem constante igual ao valor da expressão, então nada é feito e a execução do programa prossegue no primeiro comando após o bloco switch-case.

Exemplo 2

- Ao selecionar o número será informado o dia da semana.

```
#include <iostream>

int main() {
    // Declaracao de variavel
    int dia;
    // Solicita ao usuario que insira um numero
    std::cout << "Digite um numero (0-6) para descobrir o dia da semana: ";
    std::cin >> dia;
    // Estrutura switch para determinar o dia da semana
    switch(dia) {
        case 0:
            std::cout << "Domingo" << std::endl;
            break;
        case 1:
            std::cout << "Segunda-feira" << std::endl;
            break;
        case 2:
            std::cout << "Terca-feira" << std::endl;
            break;
        case 3:
            std::cout << "Quarta-feira" << std::endl;
            break;
        case 4:
            std::cout << "Quinta-feira" << std::endl;
            break;
        case 5:
            std::cout << "Sexta-feira" << std::endl;
            break;
        case 6:
            std::cout << "Sabado" << std::endl;
            break;
        default:
            std::cout << "Numero invalido! Por favor, insira um numero entre 0 e 6." << std::endl;
            break;
    }
    return 0;
}
```

Estruturas de Repetição

- As estruturas de repetição nada mais são do que laços que se mantêm ativos até que uma condição seja obedecida;
- Em C++ há basicamente três modelos de laços de repetição:
 - For;
 - While;
 - Do/While.

Estrutura de Repetição: **For**

- Geralmente utilizado quando existe um término definido, deve ser iniciado um valor de variável, a condição de início e o valor do incremento;

for (variável de inicialização; condição; incremento)

Exemplo estrutura For

```
#include <iostream>

int main() {
    // Estrutura for para imprimir os valores de 0 a 9
    for (int i = 0; i < 10; ++i) {
        std::cout << i << std::endl;
    }

    return 0;
}
```

Estrutura de repetição: **While**

- O While testa uma condição que pode ter término definido ou não.
- Ele testa o código inicial e executa o bloco, caso a função se torne falsa ele sai da repetição.

While (Condição)

```
{  
    Comandos;  
}
```

Exemplo da estrutura While

```
#include <iostream>

int main() {
    // Declaração da variável de controle
    int i = 0;

    // Estrutura while para imprimir os valores de 0 a 9
    while (i < 10) {
        std::cout << i << std::endl;
        i++; // Incrementa a variável de controle
    }

    return 0;
}
```

Laço Do While

- Tem praticamente a mesma definição do While, com a diferença que primeiro executa e somente testa a condição no final.

```
Do  
{  
    Comandos;  
} while(condição);
```


Exemplo da estrutura

```
#include <iostream>

int main() {
    // Declaração da variável de controle
    int i = 0;

    // Estrutura do-while para imprimir os valores de 0 a 9
    do {
        std::cout << i << std::endl;
        i++; // Incrementa a variável de controle
    } while (i < 10);

    return 0;
}
```

Atividade Avaliativa

Resolver os seguintes exercícios das respectivas listas:

- Exercício 07 da Lista 01
- Exercício 10 da Lista 02
- Exercício 01 da Lista 03
- Exercício 06 da Lista 04

Postar no AVA, Aula 02, até as 22h de hoje.

Obs: Coloque todos os códigos em uma pasta compactada (.zip ou .rar)