

Linguagem de Programação C++

Universidade Federal de Santa Catarina

Departamento de Engenharia Elétrica, CTC

Prof. Eduardo Augusto Bezerra

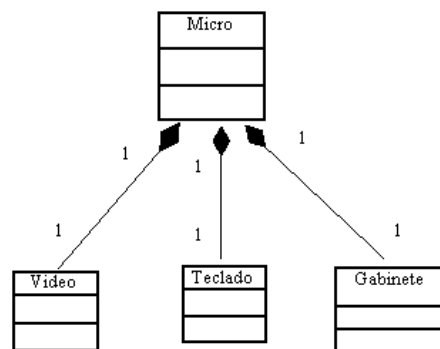
Profa. Cristiane Woszezenki

Composição (ou agregação de composição)

Agregações de composição possibilitam definir relacionamentos do tipo "é parte de". No exemplo a seguir, retirado do apostila do "Curso de C++ por Alceu Heinke Frigeri - Bernardo Copstein - Carlos Eduardo Pereira", é explorado um microcomputador composto de gabinete, teclado e vídeo. Cada uma dessas partes é um objeto individual. Agregadas, formam um objeto mais complexo. A definição de agregações em "C++" é feita pela definição de atributos que são objetos. Esses podem ser definidos diretamente na classe ou alocados dinamicamente a partir de atributos que são ponteiros. O que se torna relevante nesses casos é saber como se comportam as funções construtoras e como passar os parâmetros para as mesmas.

[Siga esse link para o código fonte de exemplos de agregação.](#)

Exemplo 1 – Um microcomputador com video, teclado e gabinete



```

/*****
***** Saida fornecida pelo programa:
*****
bezerra@sirius:~/public_html/tmp$ g++ a.cpp
bezerra@sirius:~/public_html/tmp$ ./a.out

Descricao do micro:Micro1
Marca do video: 486 DX 66
Descricao do teclado: Teclado comum - 128 teclas
Tipo da cpu: Samsung SyncMaster3
  
```

```

Descricao do micro:Micro2
Marca do video: 386 DX 25
Descricao do teclado: Teclado comum - 128 teclas
Tipo da cpu: Nec MultiSync 3D
  
```

```

bezerra@sirius:~/public_html/tmp$
*****/
  
```

```

#include <iostream>
using namespace std;
#include <string>

class video {
    string marca;
public:
    video(string);
    video(void);
    void mostra(void);
};
class teclado{
    string descricao;
public:
    teclado(string);
    teclado(void);
  
```

```

        void mostra(void);
};
class gabinete{
    string cpu;
public:
    gabinete(string);
    gabinete(void);
    void mostra(void);
};

class micro{
    string nome;
    video vid;
    teclado tec;
    gabinete gab;
public:
    micro(string, string, string);
    void mostra(void);
};

//----- Implementacao da classe video
video::video(){
}

video::video(string marca_video){
    marca = marca_video;
}

void video::mostra(){
    cout <<"Marca do video: " << marca << endl;
}

//----- Implementacao da classe teclado
teclado::teclado(string tipo){
    descricao = tipo;
}

teclado::teclado(){
}

void teclado::mostra(){
    cout <<"Descricao do teclado: " << descricao << endl;
}

//----- Implementacao da classe gabinete
gabinete::gabinete(string tipo){
    cpu = tipo;
}

gabinete::gabinete(){
}

void gabinete::mostra(){
    cout <<"Tipo da cpu: " << cpu << endl;
}

// Implementacao da classe micro
micro::micro(string n, string marca_video, string tipo_cpu):tec("Teclado comum - 128 teclas"), vid(marca_video), gab(tipo_cpu){
    nome = n;
}

void micro::mostra(){
    cout <<"\nDescricao do micro:" << nome << endl;
    vid.mostra();
    tec.mostra();
    gab.mostra();
    cout << endl;
}

// Rotina principal
int main(){
    micro micro1("Micro1", "486 DX 66", "Samsung SyncMaster3");
    micro micro2("Micro2", "386 DX 25", "Nec MultiSync 3D");
    micro1.mostra();
    micro2.mostra();
    return 0;
}

```

Exemplo 2 – Exemplo 1 reescrito com ponteiros e destrutores

```

/*****
***** Saida fornecida pelo programa
*****

bezerra@sirius:~/public_html/tmp$ g++ a.cpp
bezerra@sirius:~/public_html/tmp$ ./a.out

Descricao do micro: Micro1
Marca do video: Samsung SyncMaster3
Descricao do teclado: Teclado comum - 128 teclas
Tipo da cpu: 486 DX 66

Descricao do micro: Micro2
Marca do video: Nec MultiSync 3D
Descricao do teclado: Teclado comum - 128 teclas
Tipo da cpu: 386 DX 25

Destruiu o video

```

```

Destruiu gabinete
Destruiu micro: Micro1
Destruiu teclado
Destruiu o video
Destruiu gabinete
Destruiu micro: Micro2
Destruiu teclado
bezerra@sirius:~/public_html/tmp$

```

```

*****
*****
*****/

```

```

#include <iostream>
using namespace std;
#include <string>

class video {
    string marca;
public:
    video(string m) {
        marca = m;
    }
    ~video() {
        cout << "Destruiu o video" << endl;
    }
    void imp(void);
};

class teclado{
    string descricao;
public:
    teclado(string descr) {
        descricao = descr;
    }
    ~teclado() {
        cout << "Destruiu teclado" << endl;
    }
    void imp(void);
};

class gabinete{
    string cpu;
public:
    gabinete(string c) {
        cpu = c;
    }
    ~gabinete() {
        cout << "Destruiu gabinete" << endl;
    }
    void imp(void);
};

class micro{
    string nome;
    video *vid;
    teclado t;
    gabinete *gab;
public:
    micro(string nome, string tipo_cpu, string marca_video);
    ~micro(void);
    void imp(void);
};

// Implementacao da classe video
void video::imp() {
    cout << "Marca do video: " << marca << endl;
}

// Implementacao da classe teclado
void teclado::imp() {
    cout << "Descricao do teclado: " << descricao << endl;
}

// Implementacao da classe gabinete
void gabinete::imp() {
    cout << "Tipo da cpu: " << cpu << endl;
}

// Implementacao da classe micro
micro::micro(string n, string tipo_cpu, string marca_video):t("Teclado comum - 128 teclas") {
    nome = n;
    vid = new video(marca_video);
    gab = new gabinete(tipo_cpu);
}

micro::~micro() {
    delete vid;
    delete gab;
    cout << "Destruiu micro: " << nome << endl;
}

void micro::imp() {
    cout << endl << "Descricao do micro: " << nome << endl;
    vid->imp();
    t.imp();
    gab->imp();
    cout << endl;
}

// Rotina principal
int main() {
    micro *micro1;
    micro micro2("Micro2", "386 DX 25", "Nec MultiSync 3D");
    micro1 = new micro("Micro1", "486 DX 66", "Samsung SyncMaster3");
    micro1->imp();
}

```

```
    micro2.imp();  
    delete micro1;  
    return 0;  
}
```