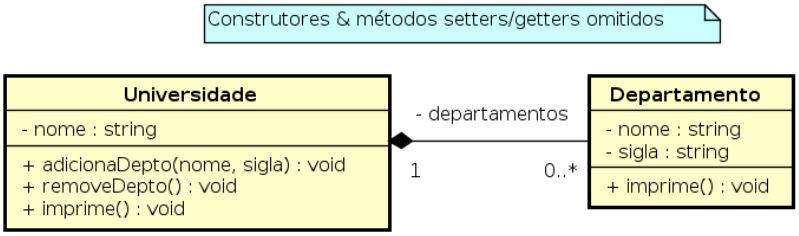
Programação Orientada a Objetos

Prof. Delano M. Beder

Roteiro 10 – Composição entre classes (Universidade e Departamento)



1. Crie um projeto (C++) denominado UniversidadeDepartamento

2. Nova classe C++ denominada Departamento.
Dois arquivos: Departamento.h (Cabeçalho) e Departamento.cpp (Código-fonte)

2.1 Arquivo Departamento.h

```
#ifndef DEPARTAMENTO_H
#define DEPARTAMENTO_H

#include <string>
using namespace std;

class Universidade;

class Departamento {
public:
  Departamento(string nome, string sigla, Universidade* universidade);
  virtual ~Departamento();
  string getNome() const;
  void setNome(string nome);
  string getSigla() const;
  void setSigla(string sigla);
  Universidade* getUniversidade() const;
  void setUniversidade(Universidade* universidade);
  void imprime() const;
private:
  string nome;
  string sigla;
  Universidade* universidade;
};

#endif /* DEPARTAMENTO_H */
```

2.2 Arquivo Departamento.cpp

```
#include "Departamento.h"
#include "Universidade.h"

#include <iostream>
using namespace std;

Departamento::Departamento(string nome, string sigla, Universidade* universidade) :
nome(nome), sigla(sigla), universidade(universidade) {
  cout << "[Construindo departamento " << sigla << "]" << endl;
}

Departamento::~Departamento() {
  cout << "[Destruindo departamento " << sigla << "]" << endl;
}

string Departamento::getNome() const {
  return nome;
}

void Departamento::setNome(string nome) {
  this->nome = nome;
}

string Departamento::getSigla() const {
  return sigla;
```

```
}

void Departamento::setSigla(string sigla) {
  this->sigla = sigla;
}

Universidade* Departamento::getUniversidade() const {
  return universidade;
}

void Departamento::setUniversidade(Universidade* universidade) {
  this->universidade = universidade;
}

void Departamento::imprime() const {
  cout << "Departamento " << nome << " de sigla " << sigla;
  cout << " (" << universidade->getNome() << ")" << endl;
}
```

3. Nova classe C++ denominada Universidade
Dois arquivos: Universidade.h (Cabeçalho) e Universidade .cpp (Código-fonte)

3.1 Arquivo Universidade.h
Utilizando vetor de ponteiros para a classe Departamento

```
#ifndef UNIVERSIDADE_H
#define UNIVERSIDADE_H

#include <iostream>
#include "Departamento.h"

using namespace std;

class Universidade {
public:
  Universidade(string nome);
  virtual ~Universidade();
  string getNome() const;
  void setNome(string nome);
  void adicionaDepartamento(string nome, string sigla);
  void removeDepartamento(string sigla);
  void imprime() const;
private:
  void realoca(int tam);
  string nome;
  Departamento** departamentos; // array dinâmico que pode ser realocado (tamanho pode aumentar ou diminuir)
  int qtde; // qtde atual de departamentos
  int max;  // qtde máxima que o array "departamentos" pode armazenar
};

#endif /* UNIVERSIDADE_H */
```

3.2 Arquivo denominado Universidade.cpp

```
#include "Universidade.h"

Universidade::Universidade(string nome) :
nome(nome) {
  cout << "[Construindo " << nome << "]" << endl;
  qtde = 0;
  max = 2;
  departamentos = new Departamento*[max];
}

Universidade::~Universidade() {
  for (int i = 0; i < qtde; i++) {
    delete departamentos[i];
  }
  delete departamentos;
  cout << "[Destruindo " << nome << "]" << endl;
}

string Universidade::getNome() const {
  return nome;
}

void Universidade::setNome(string nome) {
  this->nome = nome;
}
```

```
void Universidade::realoca(int tam) {

  // cria um novo array com mudanÇa no tamanho do array

  max = tam;

  Departamento** aux = new Departamento*[max];

  for (int i = 0; i < qtde; i++) {
    aux[i] = departamentos[i];
  }

  delete departamentos; // liberando o antigo array
  departamentos = aux;
}

void Universidade::adicionaDepartamento(string nome, string sigla) {
  if (qtde == max) {
    realoca(max + 5);
  }
  departamentos[qtde++] = new Departamento(nome, sigla, this);
}

void Universidade::removeDepartamento(string sigla) {
  bool found = false;
  int i = 0;

  while (i < qtde && !found) {
    if (departamentos[i]->getSigla().compare(sigla) == 0) {
      found = true;
    } else {
      i++;
    }
  }

  if (found) {
    delete departamentos[i];

    while (i < qtde) {
      departamentos[i] = departamentos[i + 1];
      i++;
    }

    qtde = qtde - 1;

    if (qtde == max - 5) {
      realoca(max - 5);
    }
  }
}

void Universidade::imprime() const {
  cout << "Departamentos da " << nome << endl;
  for (int i = 0; i < qtde; i++) {
    departamentos[i]->imprime();
  }
}
```

4. Arquivo main.cpp

```
#include "Universidade.h"
int main() {

  Universidade *ufscar = new Universidade("UFSCar");
  cout << "--------------------------------" << endl;
  ufscar->adicionaDepartamento("ComputaÇão", "DC");
  ufscar->adicionaDepartamento("Matemática", "DM");
  ufscar->adicionaDepartamento("Física", "DF");
  cout << "--------------------------------" << endl;
  ufscar->imprime();
  cout << "--------------------------------" << endl;
  ufscar->removeDepartamento("DM");
  cout << "--------------------------------" << endl;
  ufscar->imprime();
  cout << "--------------------------------" << endl;
  delete ufscar;
  return 0;
}
```

5. Compile e execute (verifique a saída impressa)

6. Fim