

OO – Engenharia Eletrônica

Orientação a Objetos
-
Programação em C++

4^a Slides: Listas de Objetos

Prof. Jean Marcelo SIMÃO – DAELN / UTFPR

Relembrando: Qual tem sido nosso exemplo para estudos?

Um sistema acadêmico!

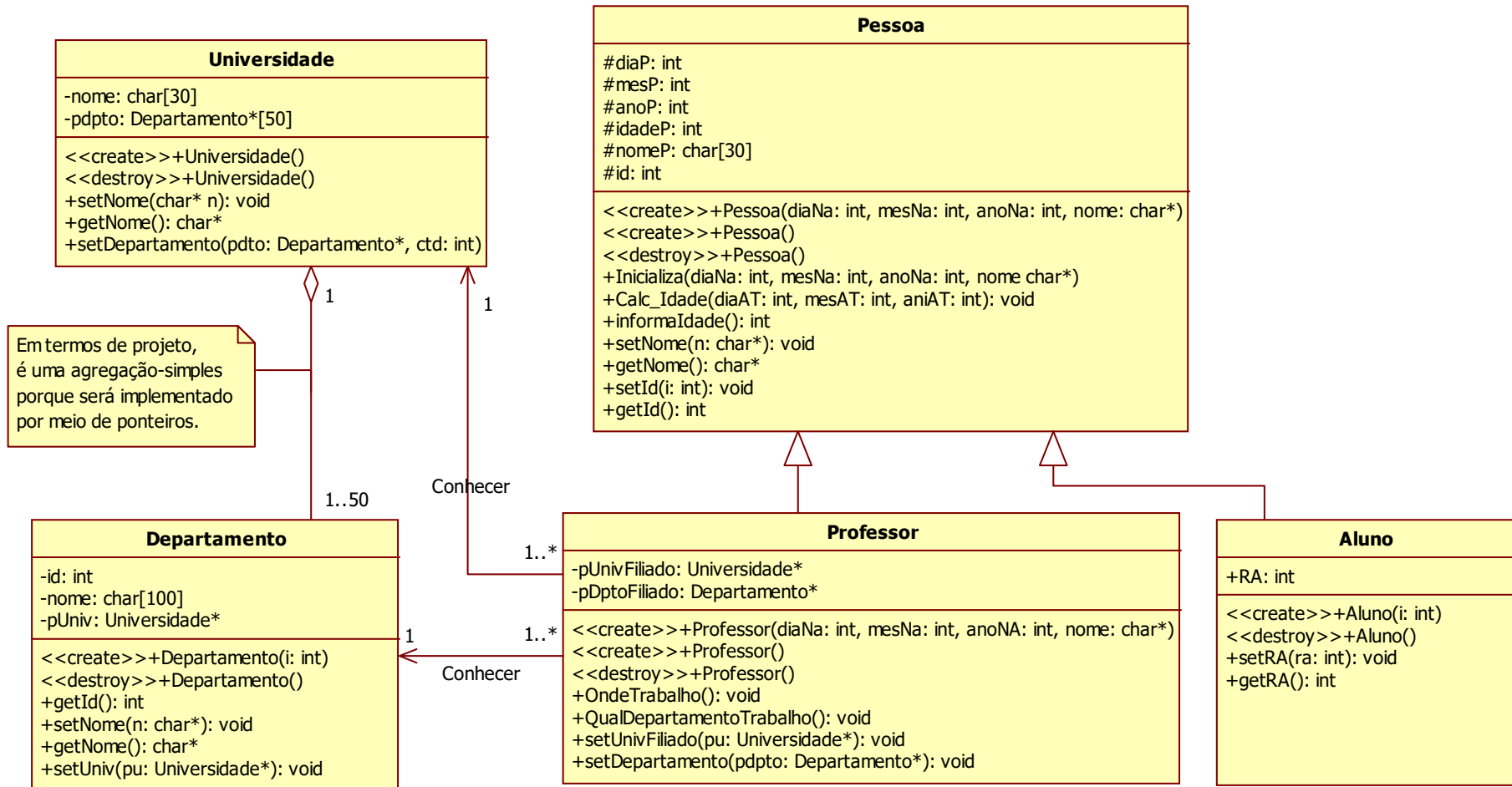
Definindo Requisitos Funcionais (e Gerais) do Sistema Acadêmico

- Registrar no sistema um conjunto de Universidades.
- Registrar no sistema um conjunto de Departamentos relacionados a Universidades.
- Registrar no sistema um conjunto de Disciplinas relacionados a Departamentos.
- Registrar um conjunto de Professores relacionados a entidades pertinentes como Departamentos e Disciplinas.
- Registrar um conjunto de Alunos relacionados a entidades pertinentes como Departamentos e Disciplinas.
- . . .

Obs.: Estes são requisitos funcionais (e gerais) que serão expandidos e refinados (inclusive em termos técnicos) nos enunciados de exercícios. Isto porque estamos no âmbito de uma disciplina aprendendo conceitos progressivamente. Em um sistema real, os requisitos deveriam ser definidos e estabelecidos o quanto antes (dentro do ciclo de engenharia de software).

Sistema Acadêmico

Diagrama de Classes de Projeto de um Sistema Acadêmico.

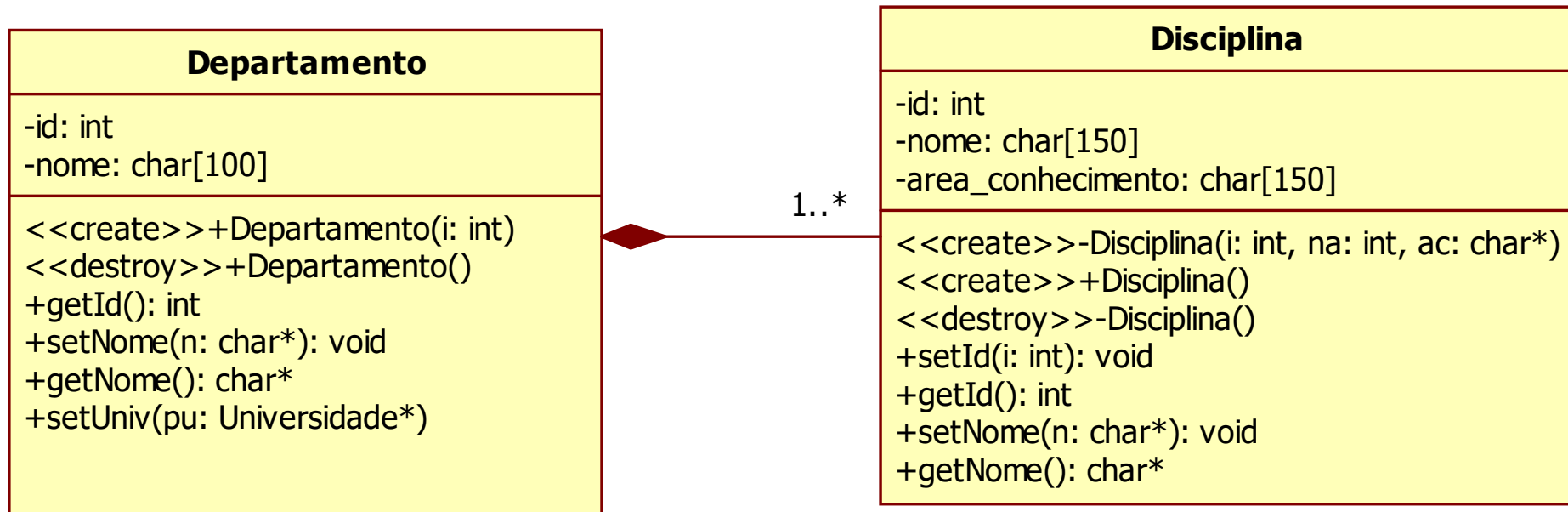


Exercício 1A

Baseando-se nos requisitos funcionais (sejam gerais ou específicos), pode-se derivar requisitos técnicos, como estes abaixo que compõem o exercício 1A.

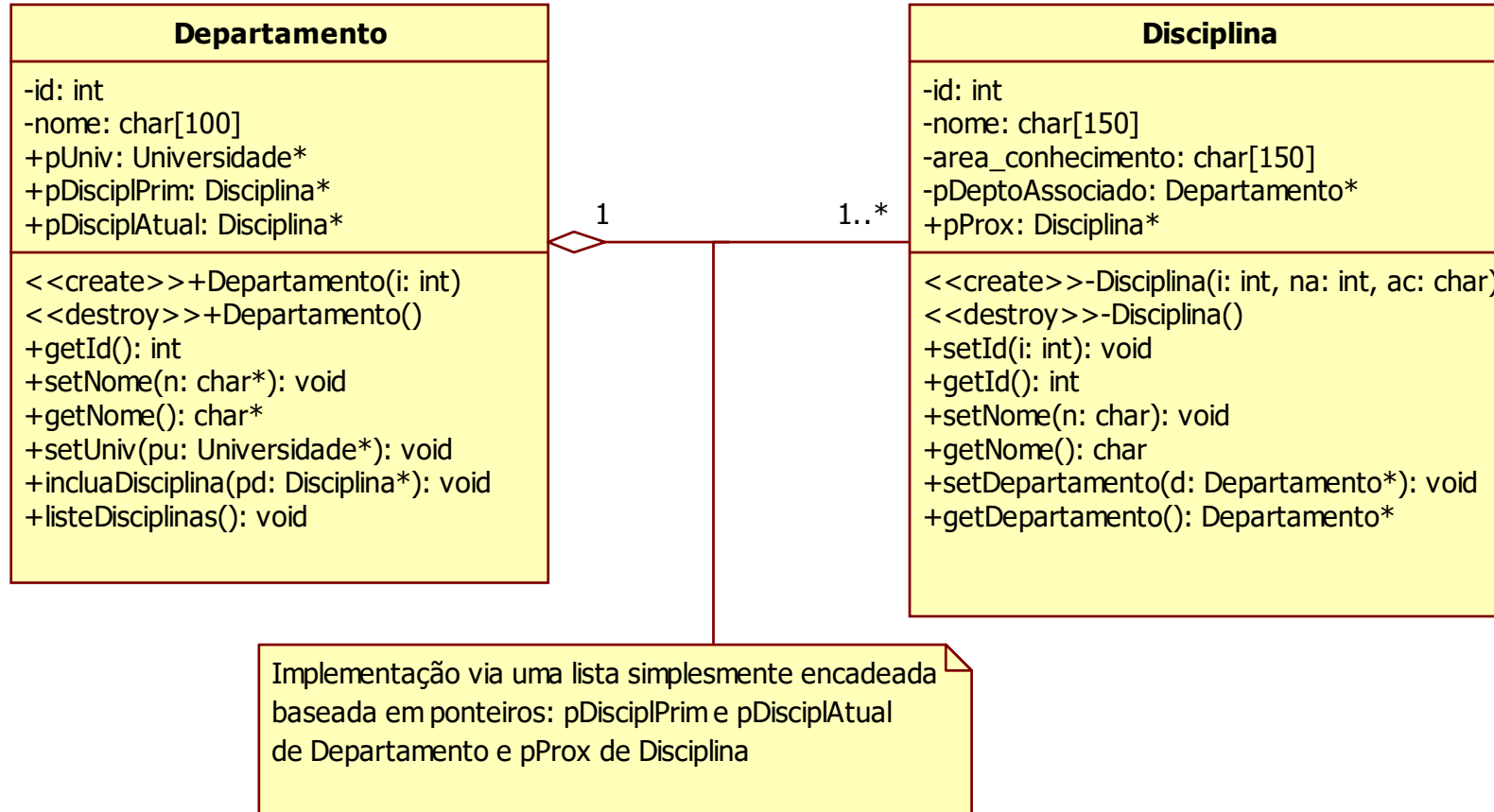
- Crie uma classe *Disciplina*.**
- Associe cada Disciplina a um Departamento.**
- Cada Departamento deve ser capaz de armazenar uma lista de disciplinas.**
- Esta lista deverá ser simplesmente encadeada.**

Diagrama de Classes – Análise.



Do ponto de vista da análise, é uma **COMPOSIÇÃO** (ou agregação-forte).

Diagrama de Classes – Projeto.



Do ponto de vista do projeto, é uma agregação simples (ou fraca).

'Correção de Exercício'

```
#ifndef _DISCIPLINA_H_  
#define _DISCIPLINA_H_  
#include "Departamento.h"
```

```
class Disciplina  
{  
private:  
    int id;  
    char nome[150];  
    char area_conhecimento[150];  
  
    Departamento* pDeptoAssociado;  
  
public:  
    Disciplina ( char* ac = "" );  
    ~Disciplina ( );  
  
    void setId(int n);  
    int getId();  
  
    void setNome ( char* n );  
    char* getNome ( );  
  
    void setDepartamento ( Departamento* pdpto );  
    Departamento* getDepartamento();  
  
};  
#endif
```

Crie uma classe *Disciplina*

Associe cada *Disciplina* a um *Departamento*.

'Correção de Exercício'

```
#ifndef _DEPARTAMENTO_H_
#define _DEPARTAMENTO_H_
class Universidade;
class Disciplina;

class Departamento
{
private:
    char nome[100];
    // Associação para com uma Universidade.
    Universidade* pUniv;
    // Associação para com várias Disciplinas.
    Disciplina *pDisciplPrim;
    Disciplina *pDisciplAtual;
public:
    Departamento ( );
    ~Departamento ( );

    void setNome ( char* n );
    char* getNome ( );
    void setUniversidade ( Universidade* pu );
    Universidade* getUniversidade ( );

    void incluadisciplina ( Disciplina* pd );
    void listeDisciplinas ( );

};
#endif
```

Cada Departamento deve ser capaz de armazenar uma lista de disciplinas

Esta lista deverá ser simplesmente encadeada.

```
#ifndef _DISCIPLINA_H_
#define _DISCIPLINA_H_
#include "Aluno.h"
#include "Departamento.h"

class Disciplina
{
private:
    int id;
    char nome [ 150 ];
    char area_conhecimento [ 150 ];
    Departamento* pDeptoAssociado;
public:
    Disciplina ( char* ac = "" );
    ~Disciplina ( );
    Disciplina* pProx;

    void setid ( int n );
    int getid ( );
    void setNome ( char* n );
    char* getNome ( );

    void setDepartamento ( Departamento* pdpto );
    Departamento* getDepartamento ( );

};
#endif
```

'Correção de Exercício'

```

#ifndef _DEPARTAMENTO_H_
#define _DEPARTAMENTO_H_
class Universidade;
class Disciplina;

class Departamento
{
private:
    char nome[130];

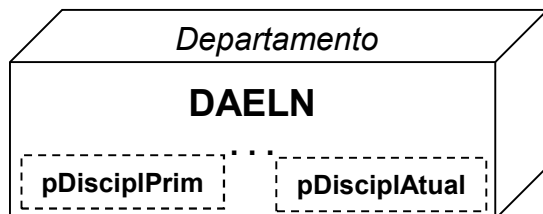
    Universidade* pUniv;

    Disciplina *pDisciplPrim;
    Disciplina *pDisciplAtual;
public:
    Departamento ( );
    ~Departamento ( );

    void setNome ( char* n );
    char* getNome ( );
    void setUniversidade ( Universidade* pu );
    Universidade* getUniversidade ( );

    void incluuaDisciplina ( Disciplina* pd );
    void listeDisciplinas ( );
};
#endif

```



```

#ifndef _DISCIPLINA_H_
#define _DISCIPLINA_H_
#include "Aluno.h"
#include "Departamento.h"

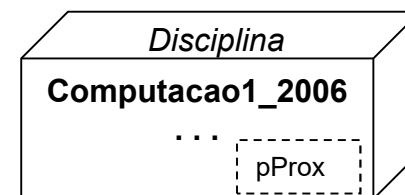
class Disciplina
{
private:
    int id;
    char nome[150];
    char area_conhecimento[150];
    Departamento* pDeptoAssociado;
public:
    Disciplina ( char* ac = "" );
    ~Disciplina ( );

    Disciplina* pProx;

    void setId ( int n );
    int getId ( );
    void setNome ( char* n );
    char* getNome ( );

    void setDepartamento ( Departamento* pdpto );
    Departamento* getDepartamento ( );
};
#endif

```



```

#include "Departamento.h"
#include "Universidade.h"
#include "Disciplina.h"
#include <stdio.h>

Departamento::Departamento ( )
{
    pDisciplPrim = NULL;
    pDisciplAtual = NULL;
}

Departamento::~~Departamento ( )
{
    pDisciplPrim = NULL;
    pDisciplAtual = NULL;
}

void Departamento::setNome ( char* n )
{
    strcpy ( nome, n );
}

char* Departamento::getNome ( )
{
    return nome;
}

void Departamento::setUniversidade ( Universidade* pu )
{
    pUniv = pu;
}

Universidade* Departamento::getUniversidade ( )
{
    return pUniv;
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
    if ( pDisciplPrim == NULL )
    {
        pDisciplPrim = pd;
        pDisciplAtual = pd;
    }
    else
    {
        pDisciplAtual->pProx = pd;
        pDisciplAtual = pd;
    }
}

void Departamento::listeDisciplinas ( )
{
    Disciplina* pAux;

    pAux = pDisciplPrim;

    while ( pAux != NULL )
    {
        printf (" A disciplina %s pertence ao Departamento %s \n",
            pAux->getNome(), nome );

        pAux = pAux->pProx;
    }
}

```

```

#include "Disciplina.h"
#include <stdio.h>
#include <string.h>

Disciplina::Disciplina ( char* ac )
{
    pDeptoAssociado = NULL;
    pProx           = NULL;

    strcpy (area_conhecimento, ac);
}

Disciplina::~Disciplina ( )
{
    pDeptoAssociado = NULL;
    pProx           = NULL;
}

void Disciplina::setId ( int n )
{
    id = n;
}

int Disciplina::getId ( )
{
    return id;
}

void Disciplina::setNome (char* n)
{
    strcpy ( nome, n );
}

char* Disciplina::getNome ( )
{
    return nome;
}

```

```

void Disciplina::setDepartamento ( Departamento* pdpto )
{
    pDeptoAssociado = pdpto;
}

Departamento* Disciplina::getDepartamento ( )
{
    return pDeptoAssociado;
}

```

1a Versão

'Correção de Exercício'

```
#ifndef _PRINCIPAL_H_
#define _PRINCIPAL_H_

#include "Professor.h"
#include "Universidade.h"
#include "Aluno.h"
#include "Disciplina.h"

class Principal
{
private:
    // Universidades
    Universidade UTFPR;
    Universidade Princeton;
    Universidade Cambridge;

    // Departamentos
    Departamento DAELN;
    Departamento MatematicaUTFPR;
    Departamento FisicaUTFPR;

    Departamento MatematicaPrinceton;
    Departamento FisicaPrinceton;

    Departamento MatematicaCambridge;
    Departamento FisicaCambridge;

    ...

    // Professores
    Professor Simao;
    Professor Einstein;
    Professor Newton;
```

```
    // Disciplinas
    Disciplina Computacao1_2006;
    Disciplina Introd_Alg_2007;
    Disciplina Computacao2_2007;
    Disciplina Metodos2_2007;

    int diaAtual;
    int mesAtual;
    int anoAtual;

public:

    Principal ( );

    // Inicializações...
    void Inicializa ( );
    void InicializaUniversidades ( );
    void InicializaDepartamentos ( );
    void InicializaProfessores ( );
    void InicializaDisciplinas ( );

    void Executar ( );

    void CalcIdadeProfs ( );
    void UnivOndeProfsTrabalham ( );
    void DepOndeProfsTrabalham ( );
    void ListeDiscDeptos ( );

};

#endif
```

Organização

```

#include "Principal.h"
#include <stdio.h>

Principal::Principal ( )
{
    // leitura da data atual
    ...

    Inicializa();
}

void Principal::Inicializa ( )
{
    InicializaUnivesidades ( );
    InicializaDepartamentos ( );
    InicializaProfessores ( );
    InicializaDisciplinas ( );
}

void Principal::InicializaUnivesidades ( )
{
    // Registro do(s) nome(s) da(s) universidade(s)
    UTFPR.setNome ( "UTFPR" );
    Princeton.setNome ( "Princeton" );
    Cambridge.setNome ( "Cambridge" );
}

void Principal::InicializaDepartamentos ( )
{
    // Registro do(s) nome(s) do(s) departamento(s)
    DAELN.setNome ( "Eletronica UTFPR" );
    MatematicaUTFPR.setNome ( "Matematica UTFPR" );
    FisicaUTFPR.setNome ( "Fisica UTFPR" );
    MatematicaPrinceton.setNome ( "Matematica Princeton" );
    FisicaPrinceton.setNome ( "Fisica Pirnceton" );
    MatematicaCambridge.setNome ( "Matematica Cambridge" );
    FisicaCambridge.setNome ( "Fisica Cambridge" );

    ...
}

```

```

void Principal::InicializaProfessores ( )
{

    // Inicialização do(s) objeto(s) da classe Professor
    Simao.Inicializa ( 3, 10, 1976, "Jean Simão" );
    Einstein.Inicializa ( 14, 3, 1879, "Albert Einstein" );
    Newton.Inicializa ( 4, 1, 1643, "Isaac Newton" );

    // "Filiação" a universidade.
    Simao.setUnivFiliado ( &UTFPR );
    Einstein.setUnivFiliado ( &Princeton );
    Newton.setUnivFiliado ( &Cambridge );

    // "Filiação" ao departamento.
    Simao.setDepartamento ( &DAELN );
    Einstein.setDepartamento ( &FisicaPrinceton );
    Newton.setDepartamento ( &MatematicaCambridge );

}

```

```

void Principal::InicializaDisciplinas ( )
{
    Computacao1_2006.setNome ( "Computacao I 2006" );
    Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
    Computacao2_2007.setNome ( "Computao II" );
    Metodos2_2007.setNome ( "Métodos II" );

    DAELN.incluaDisciplina ( &Computacao1_2006);
    DAELN.incluaDisciplina ( & Introd_Alg_2007);
    DAELN.incluaDisciplina ( &Computacao2_2007);
    DAELN.incluaDisciplina ( &Metodos2_2007);
    ...
}

void Principal::CalcIdadeProfs ( )
{
    // Cálculo da idade.
    Simao.Calc_Idade ( diaAtual, mesAtual, anoAtual );
    Einstein.Calc_Idade ( diaAtual, mesAtual, anoAtual );
    Newton.Calc_Idade ( diaAtual, mesAtual, anoAtual );
    printf ( "\n" );
}

void Principal::UnivOndeProfsTrabalham ( )
{
    // Universidade que a Pessoa trabalha.
    Simao.OndeTrabalho ( );
    Einstein.OndeTrabalho ( );
    Newton.OndeTrabalho ( );
    printf ( "\n" );
}

void Principal::DepOndeProfsTrabalham ( )
{
    // Departamento que a Pessoa trabalha.
    Simao.QualDepartamentoTrabalho ( );
    Einstein.QualDepartamentoTrabalho ( );
    Newton.QualDepartamentoTrabalho ( );
    printf ( "\n" );
}

```

```

void Principal::ListeDiscDeptos ( )
{
    DAELN.listeDisciplinas ( );
    printf ( "\n" );
}

void Principal::Executar ( )
{
    CalcIdadeProfs ( );
    UnivOndeProfsTrabalham ( );
    DepOndeProfsTrabalham ( );
    ListeDepPorUniv ( );
    // Metodos2.listeAlunos();
    ListeDiscDeptos ( );
}

```

1a Versão

```
void Principal::InicializaDisciplinas ( )
{
    Computacao1_2006.setNome ( "Computacao I 2006" );
    Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
    Computacao2_2007.setNome ( "Computao II" );
    Metodos2_2007.setNome ( "Métodos II" );

    DAELN.incluaDisciplina ( &Computacao1_2006);
    DAELN.incluaDisciplina ( &Introd_Alg_2007);
    DAELN.incluaDisciplina ( &Computacao2_2007);
    DAELN.incluaDisciplina ( &Metodos2_2007);

    Computacao1_2006.setDepartamento ( &DAELN );
    Introd_Alg_2007.setDepartamento ( &DAELN );
    Computacao2_2007.setDepartamento ( &DAELN );
    Metodos2_2007.setDepartamento ( &DAELN );
}
...
```

1a Versão



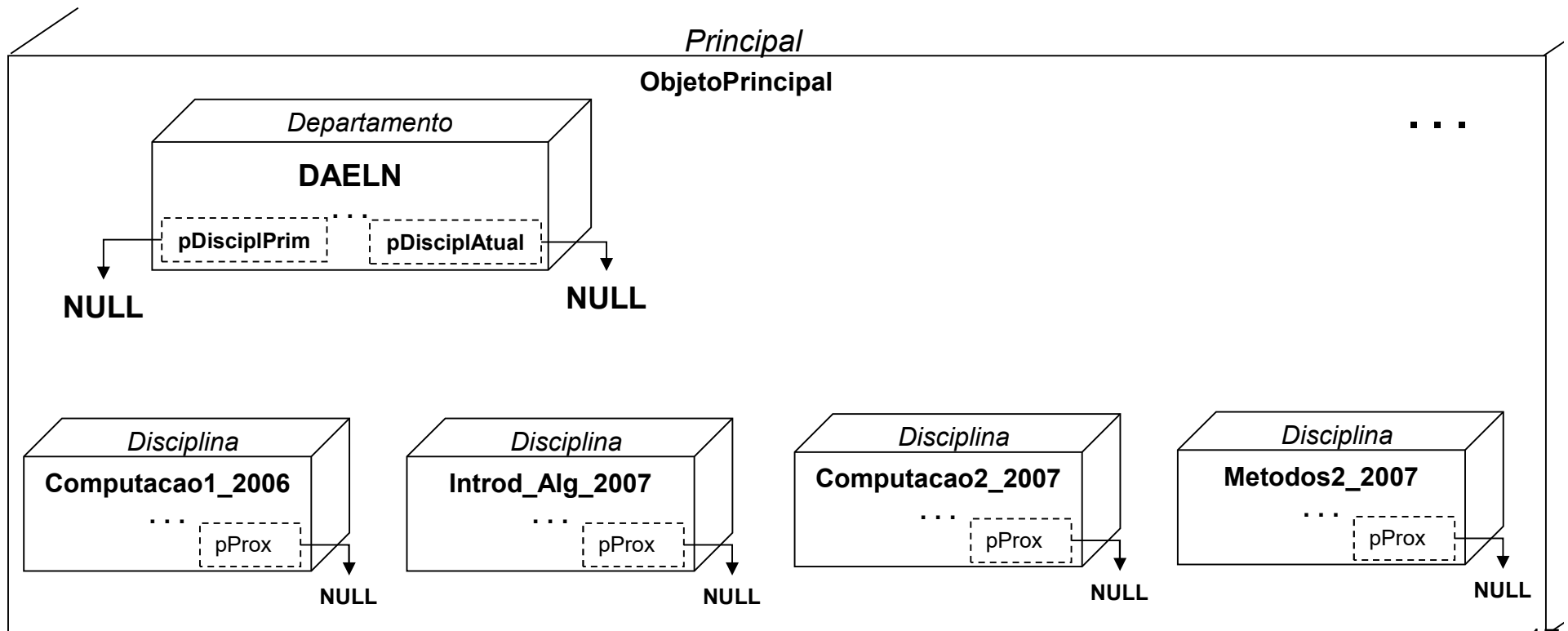
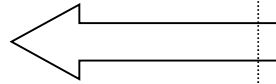

```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  DAELN.incluaDisciplina ( &Computacao1_2006);
  DAELN.incluaDisciplina ( &Introd_Alg_2007);
  DAELN.incluaDisciplina ( &Computacao2_2007);
  DAELN.incluaDisciplina ( &Metodos2_2007);

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```



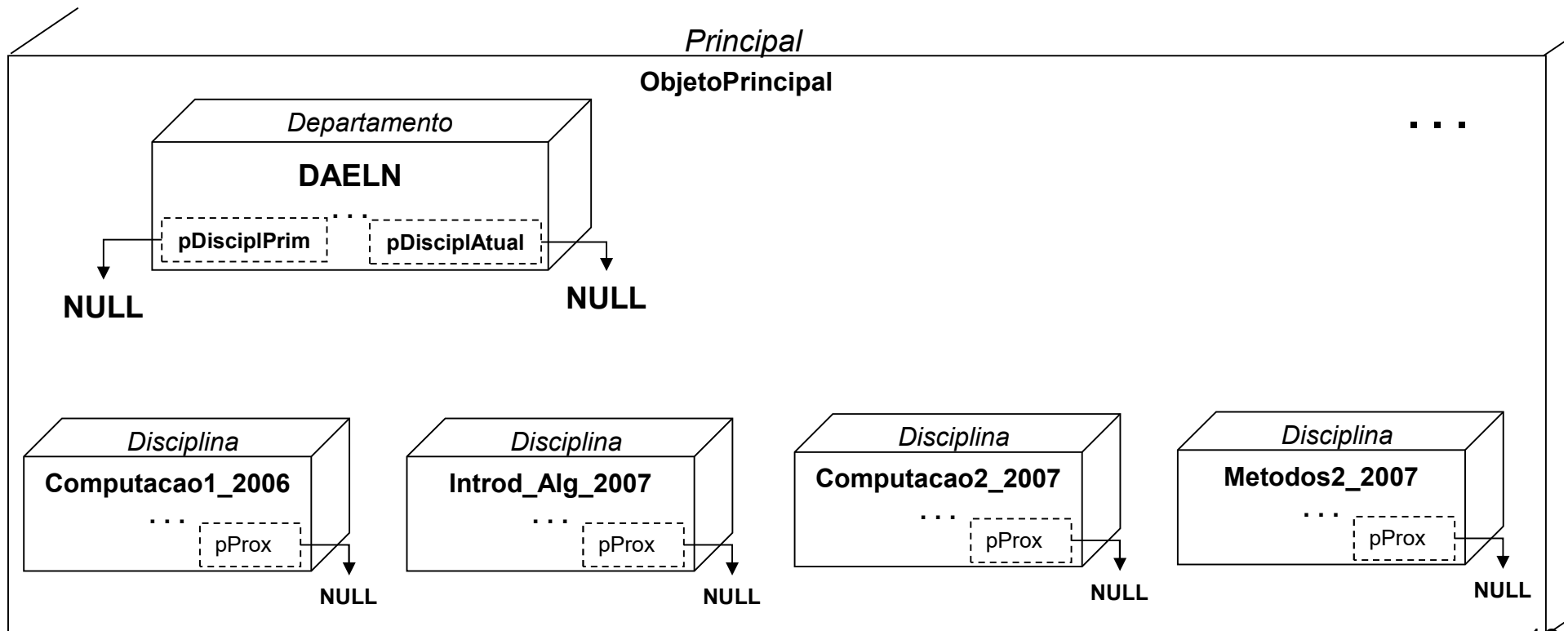
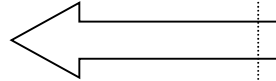
```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  DAELN.incluaDisciplina ( &Computacao1_2006);
  DAELN.incluaDisciplina ( & Introd_Alg_2007);
  DAELN.incluaDisciplina ( &Computacao2_2007);
  DAELN.incluaDisciplina ( &Metodos2_2007);

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```



```

void Principal::InicializaDisciplinas ( )
{
    Computacao1_2006.setNome ( "Computacao I 2006" );
    Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
    Computacao2_2007.setNome ( "Computao II" );
    Metodos2_2007.setNome ( "Métodos II" );

    DAELN.incluaDisciplina ( &Computacao1_2006);
    DAELN.incluaDisciplina ( & Introd_Alg_2007);
    DAELN.incluaDisciplina ( &Computacao2_2007);
    DAELN.incluaDisciplina ( &Metodos2_2007);

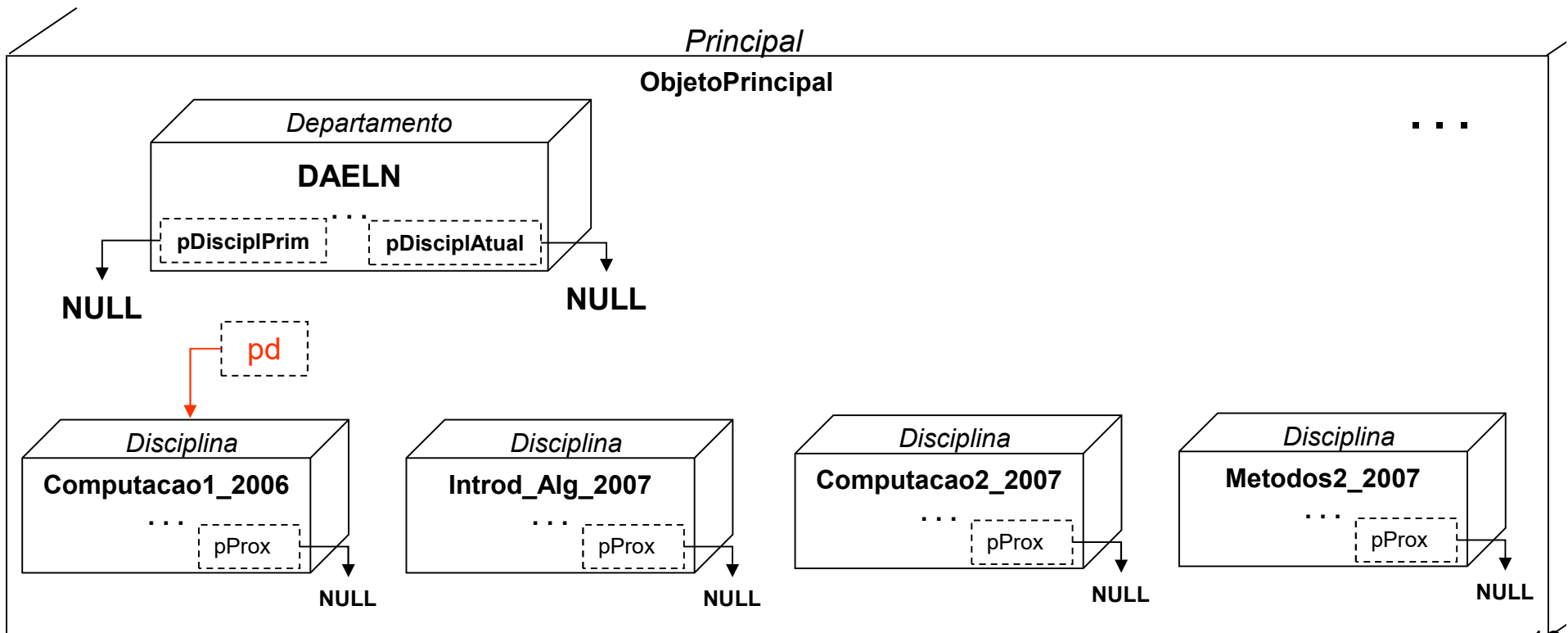
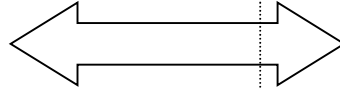
    Computacao1_2006.setDepartamento ( &DAELN );
    Introd_Alg_2007.setDepartamento ( &DAELN );
    Computacao2_2007.setDepartamento ( &DAELN );
    Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
    if ( pDisciplPrim == NULL )
    {
        pDisciplPrim = pd;
        pDisciplAtual = pd;
    }
    else
    {
        pDisciplAtual->pProx = pd;
        pDisciplAtual = pd;
    }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  DAELN.incluaDisciplina ( &Computacao1_2006);
  DAELN.incluaDisciplina ( & Introd_Alg_2007);
  DAELN.incluaDisciplina ( &Computacao2_2007);
  DAELN.incluaDisciplina ( &Metodos2_2007);

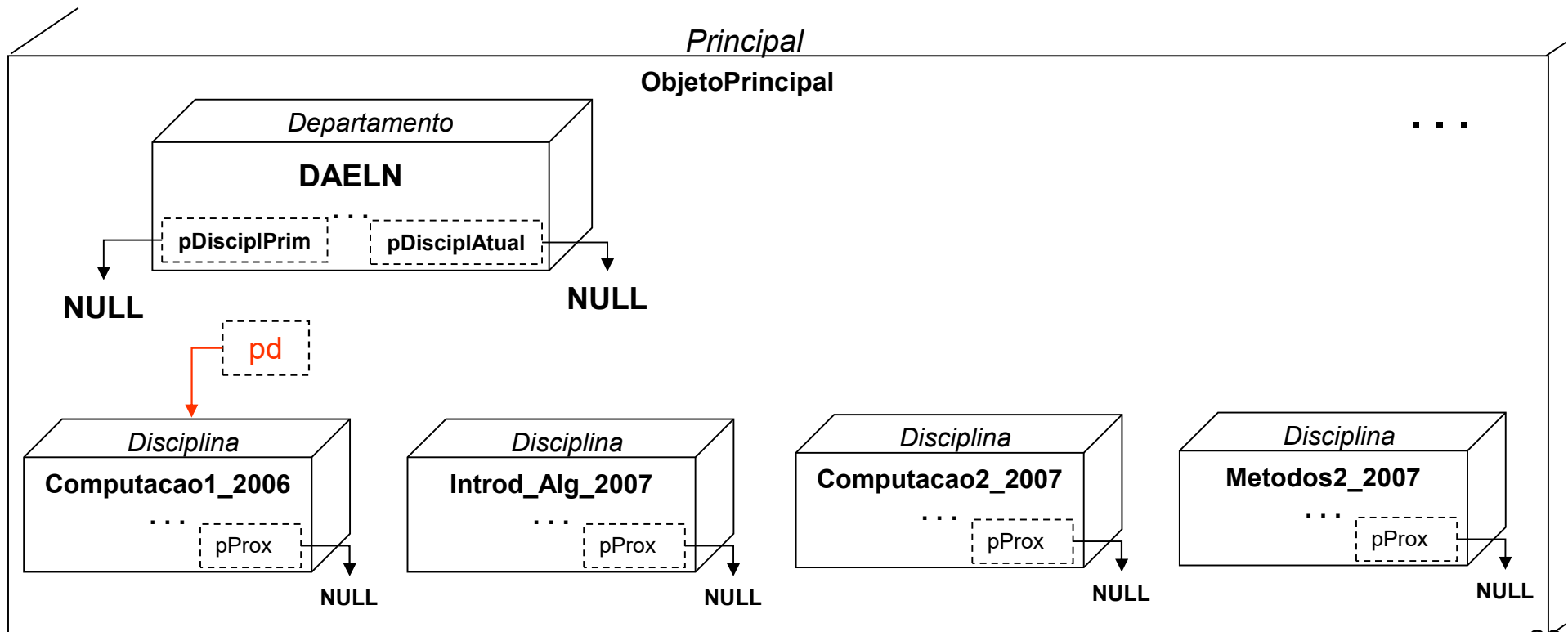
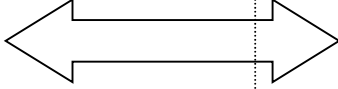
  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  DAELN.incluaDisciplina ( &Computacao1_2006);
  DAELN.incluaDisciplina ( & Introd_Alg_2007);
  DAELN.incluaDisciplina ( &Computacao2_2007);
  DAELN.incluaDisciplina ( &Metodos2_2007);

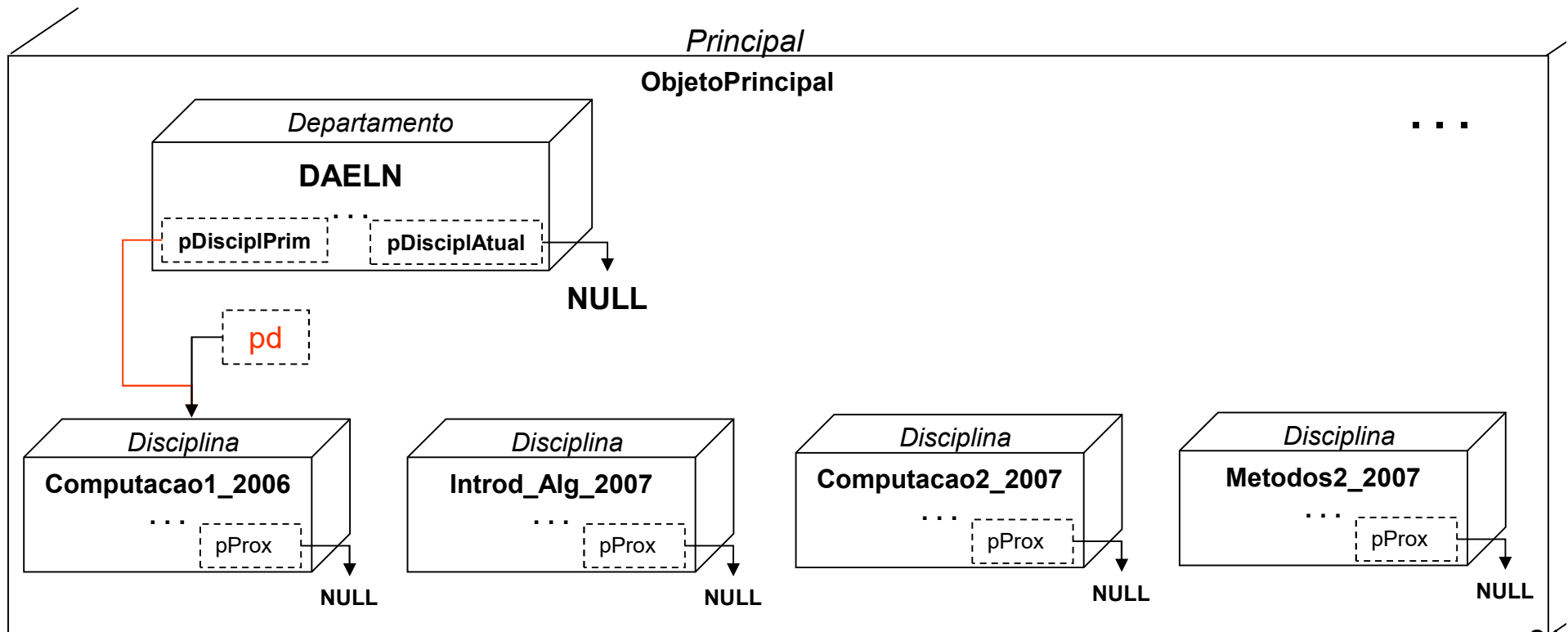
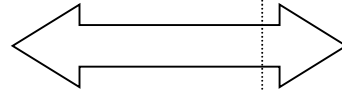
  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  DAELN.incluaDisciplina ( &Computacao1_2006);
  DAELN.incluaDisciplina ( & Introd_Alg_2007);
  DAELN.incluaDisciplina ( &Computacao2_2007);
  DAELN.incluaDisciplina ( &Metodos2_2007);

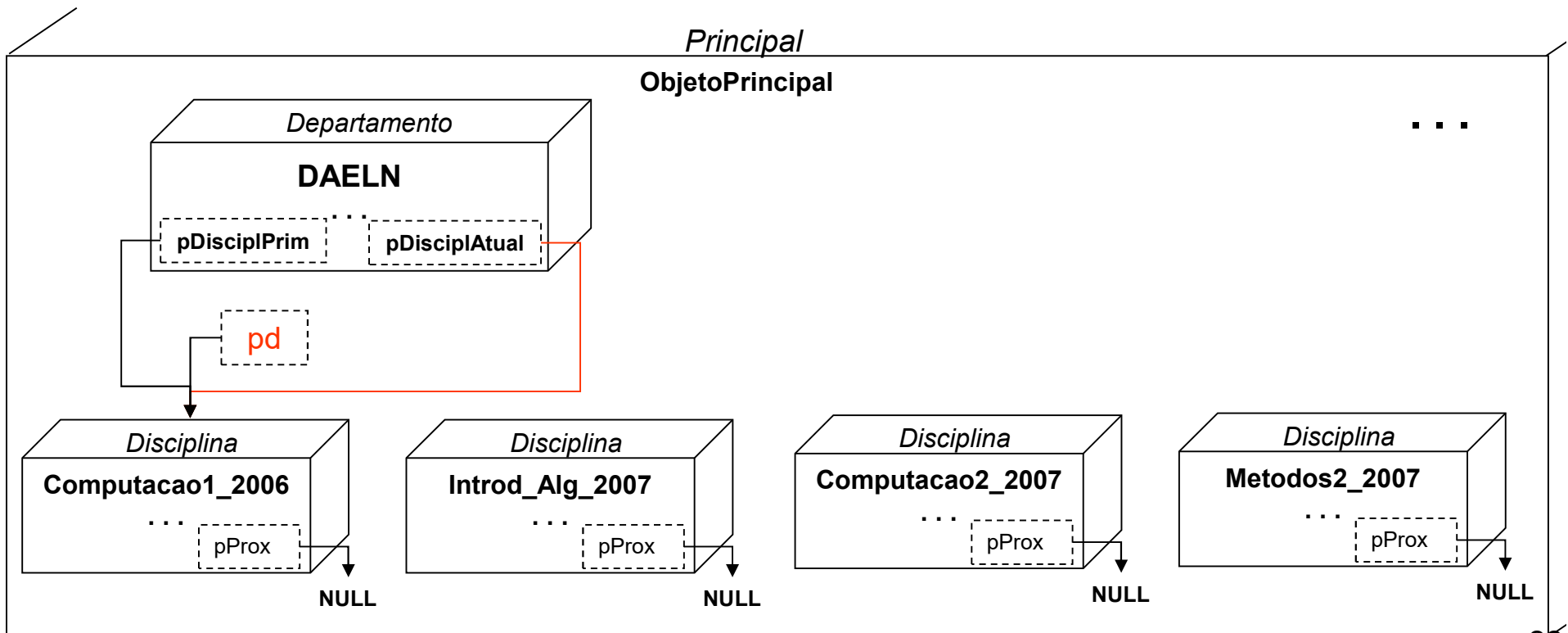
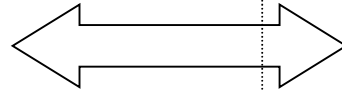
  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  DAELN.incluaDisciplina ( &Computacao1_2006);
  DAELN.incluaDisciplina ( & Introd_Alg_2007);
  DAELN.incluaDisciplina ( &Computacao2_2007);
  DAELN.incluaDisciplina ( &Metodos2_2007);

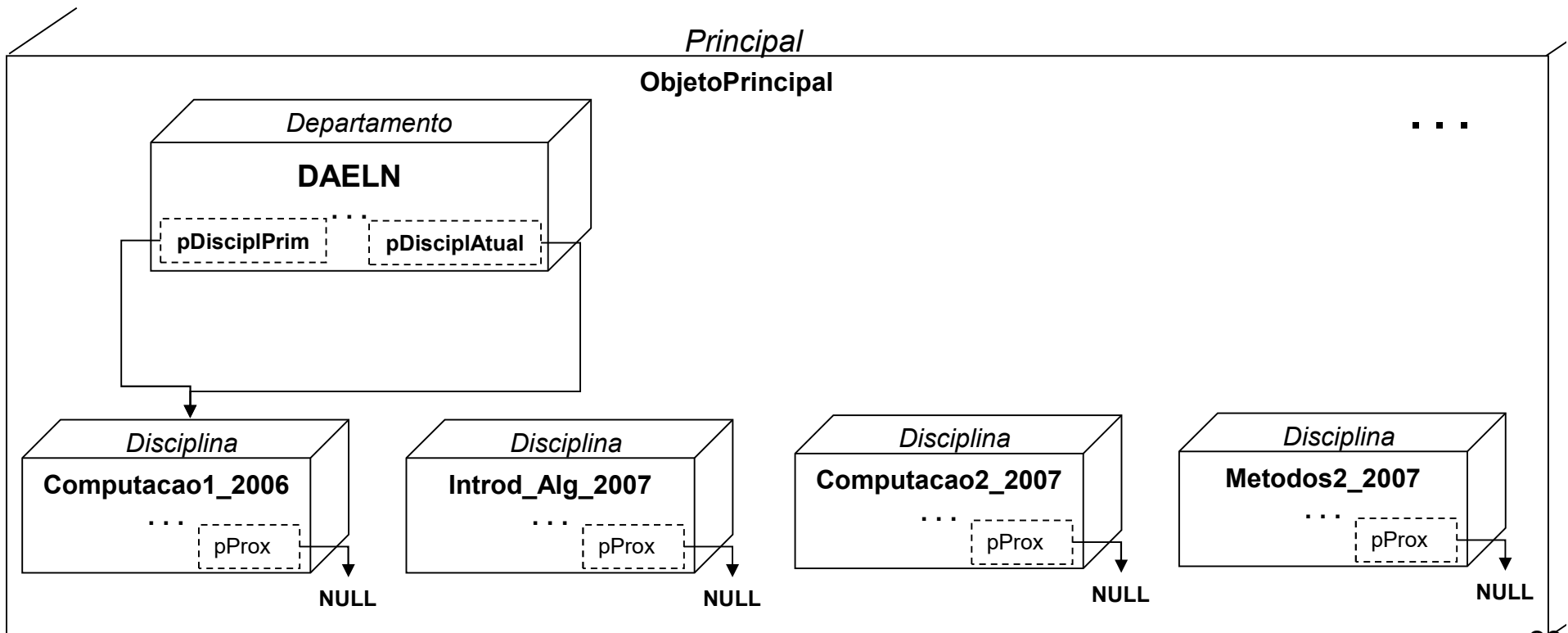
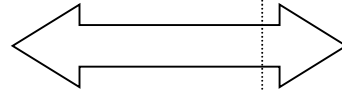
  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



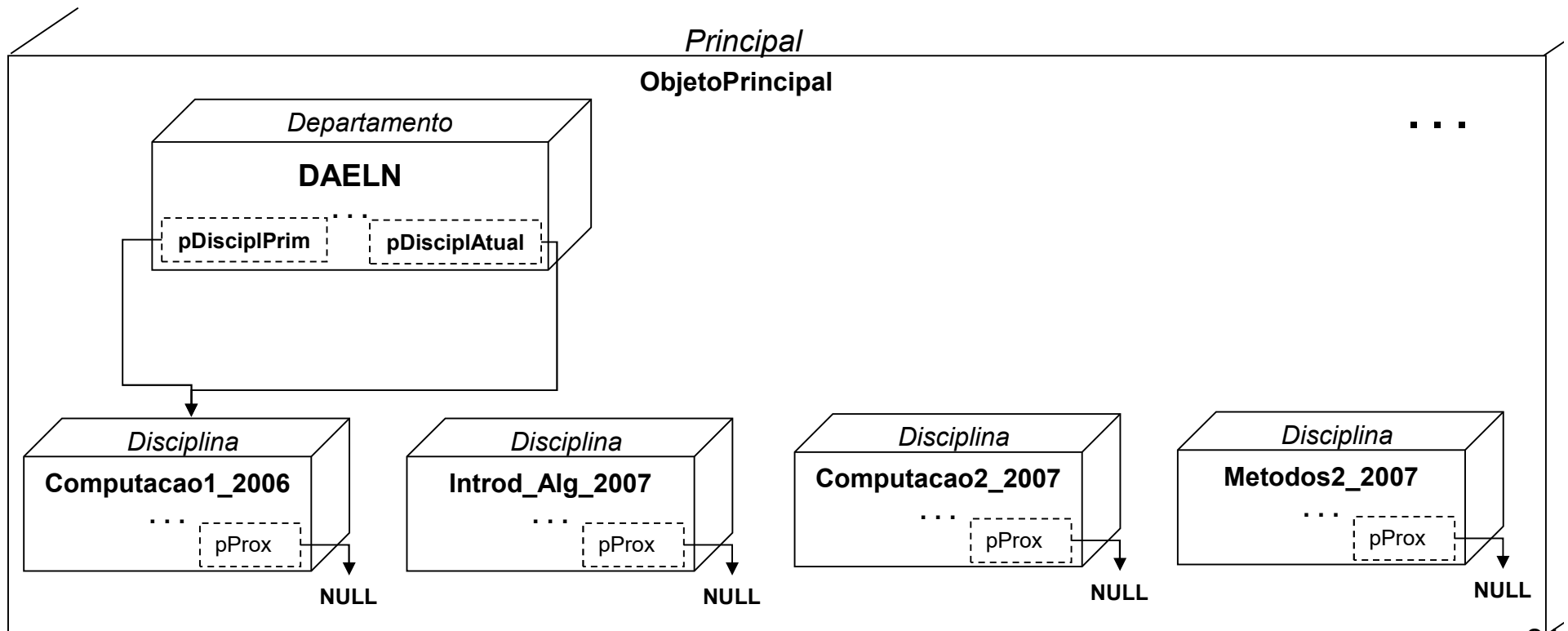
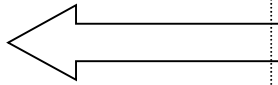
```

void Principal::InicializaDisciplinas ( )
{
    Computacao1_2006.setNome ( "Computacao I 2006" );
    Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
    Computacao2_2007.setNome ( "Computao II" );
    Metodos2_2007.setNome ( "Métodos II" );

    DAELN.incluaDisciplina ( &Computacao1_2006);
    DAELN.incluaDisciplina ( &Introd_Alg_2007);
    DAELN.incluaDisciplina ( &Computacao2_2007);
    DAELN.incluaDisciplina ( &Metodos2_2007);

    Computacao1_2006.setDepartamento ( &DAELN );
    Introd_Alg_2007.setDepartamento ( &DAELN );
    Computacao2_2007.setDepartamento ( &DAELN );
    Metodos2_2007.setDepartamento ( &DAELN );
}

```




```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  DAELN.incluaDisciplina ( &Computacao1_2006);
  DAELN.incluaDisciplina ( &Introd_Alg_2007);
  DAELN.incluaDisciplina ( &Computacao2_2007);
  DAELN.incluaDisciplina ( &Metodos2_2007);

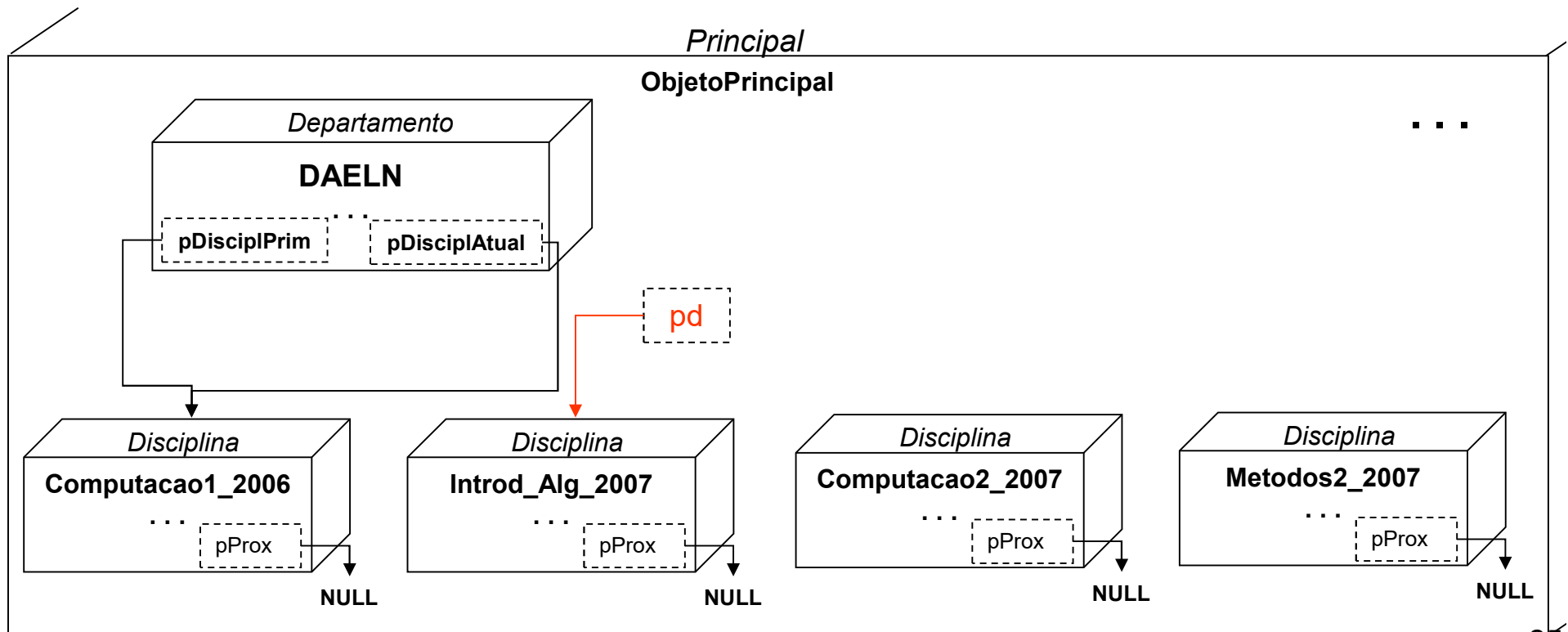
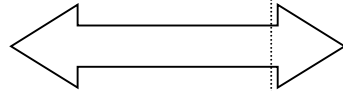
  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
    Computacao1_2006.setNome ( "Computacao I 2006" );
    Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
    Computacao2_2007.setNome ( "Computao II" );
    Metodos2_2007.setNome ( "Métodos II" );

    DAELN.incluaDisciplina ( &Computacao1_2006);
    DAELN.incluaDisciplina ( &Introd_Alg_2007);
    DAELN.incluaDisciplina ( &Computacao2_2007);
    DAELN.incluaDisciplina ( &Metodos2_2007);

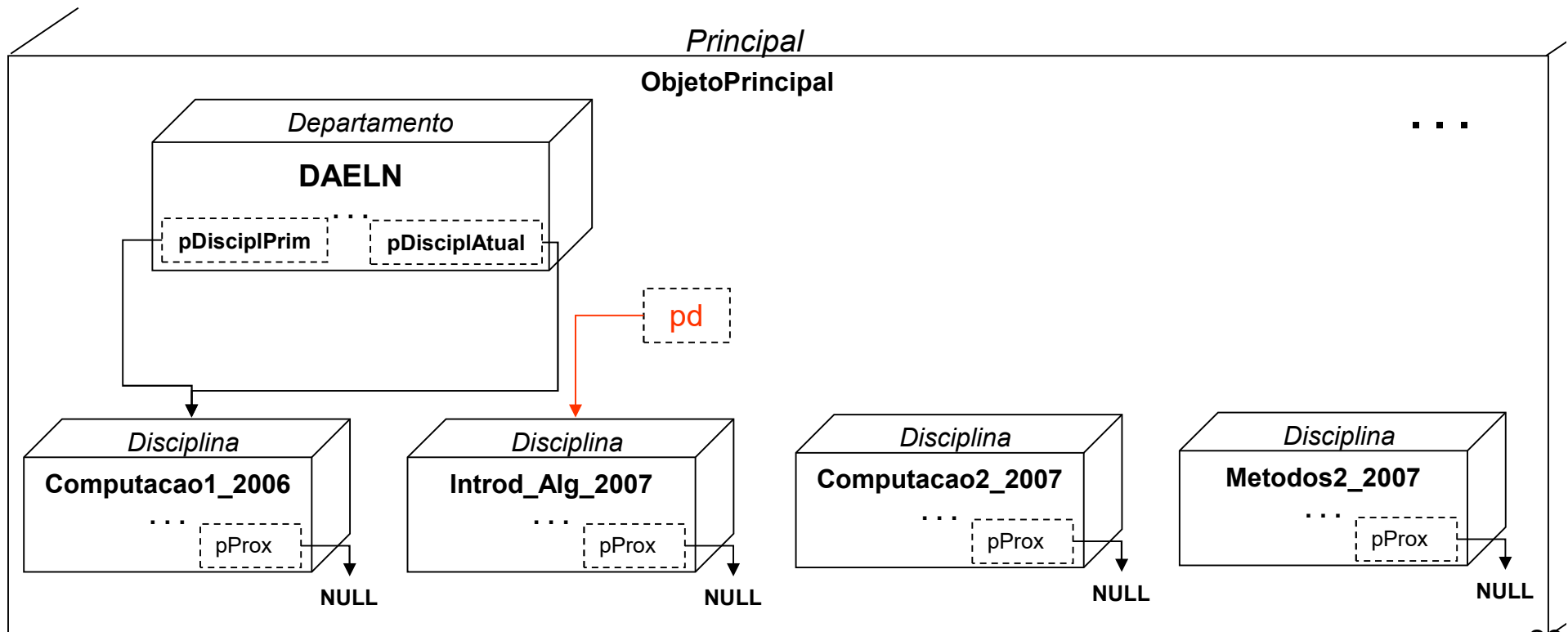
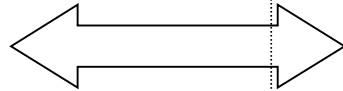
    Computacao1_2006.setDepartamento ( &DAELN );
    Introd_Alg_2007.setDepartamento ( &DAELN );
    Computacao2_2007.setDepartamento ( &DAELN );
    Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
    if ( pDisciplPrim == NULL )
    {
        pDisciplPrim = pd;
        pDisciplAtual = pd;
    }
    else
    {
        pDisciplAtual->pProx = pd;
        pDisciplAtual = pd;
    }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  DAELN.incluaDisciplina ( &Computacao1_2006);
  DAELN.incluaDisciplina ( &Introd_Alg_2007);
  DAELN.incluaDisciplina ( &Computacao2_2007);
  DAELN.incluaDisciplina ( &Metodos2_2007);

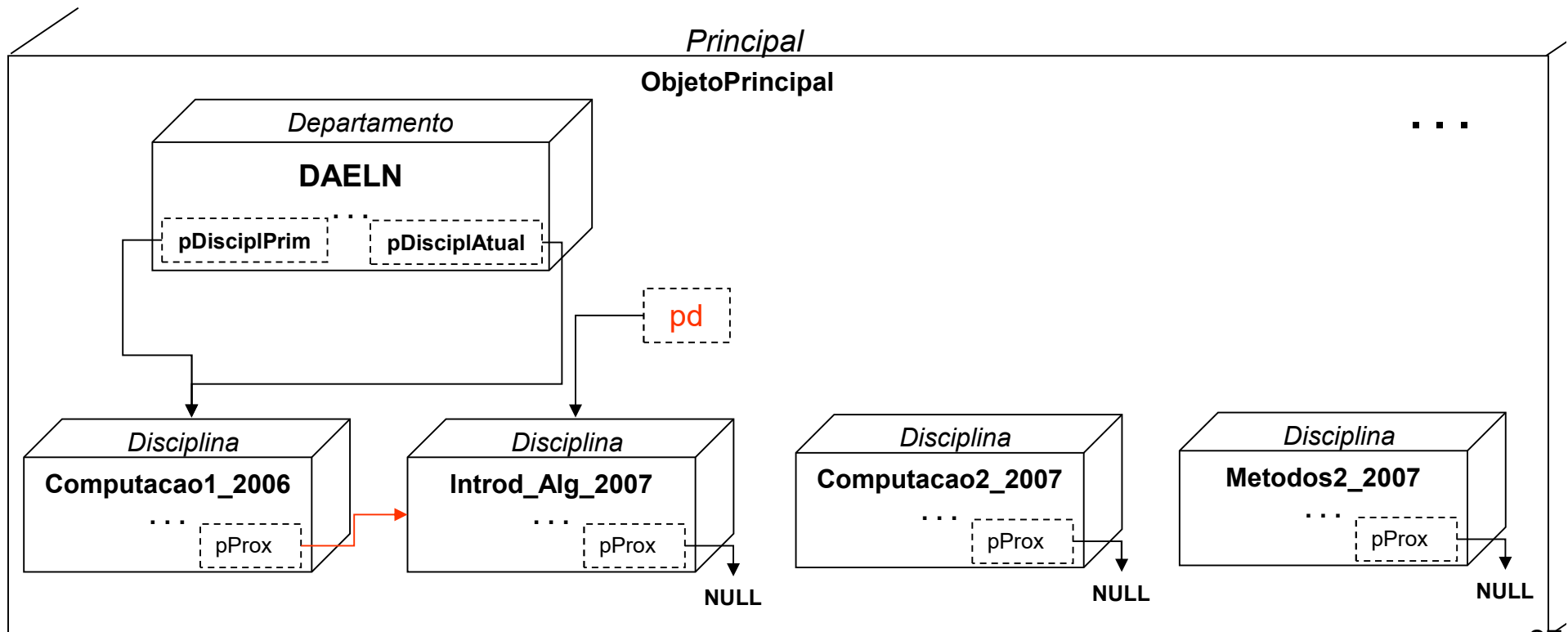
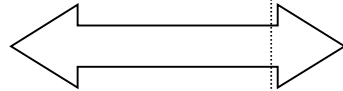
  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  DAELN.incluaDisciplina ( &Computacao1_2006);
  DAELN.incluaDisciplina ( &Introd_Alg_2007);
  DAELN.incluaDisciplina ( &Computacao2_2007);
  DAELN.incluaDisciplina ( &Metodos2_2007);

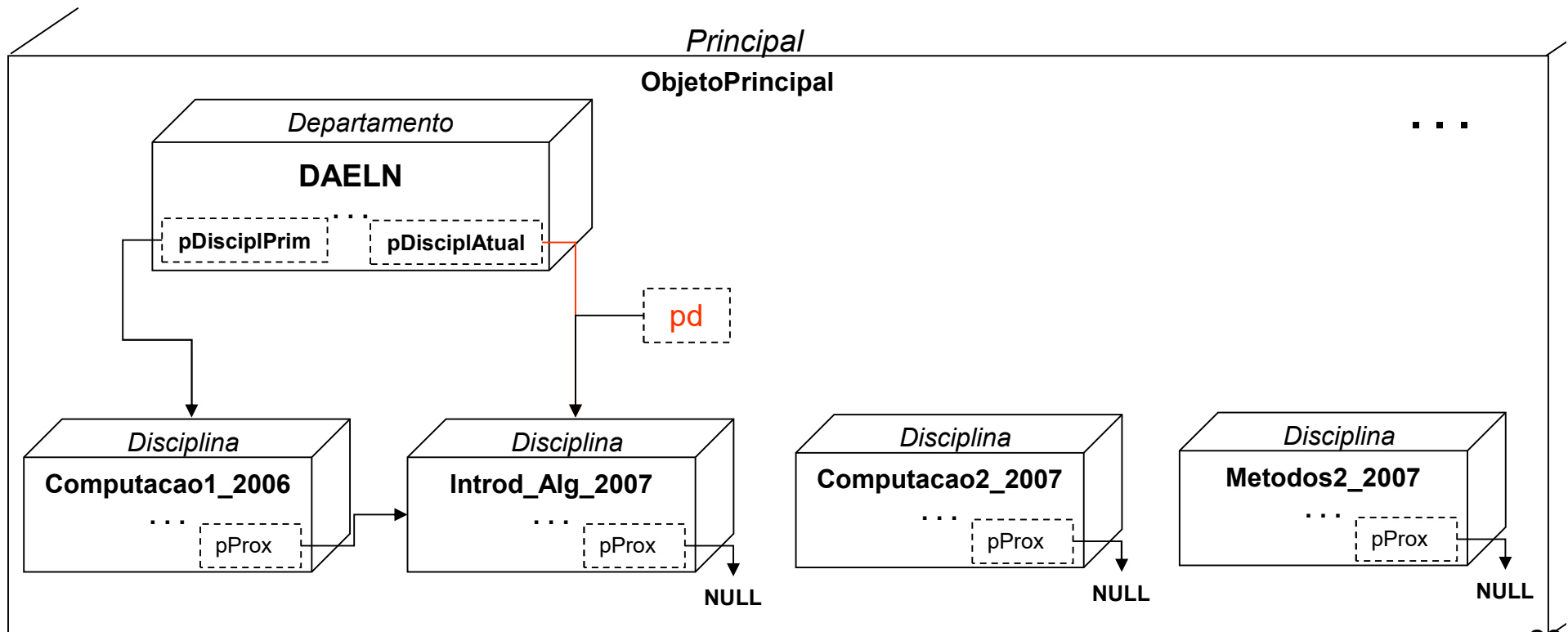
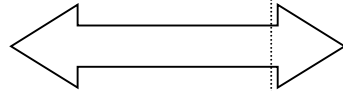
  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  DAELN.incluaDisciplina ( &Computacao1_2006);
  DAELN.incluaDisciplina ( &Introd_Alg_2007);
  DAELN.incluaDisciplina ( &Computacao2_2007);
  DAELN.incluaDisciplina ( &Metodos2_2007);

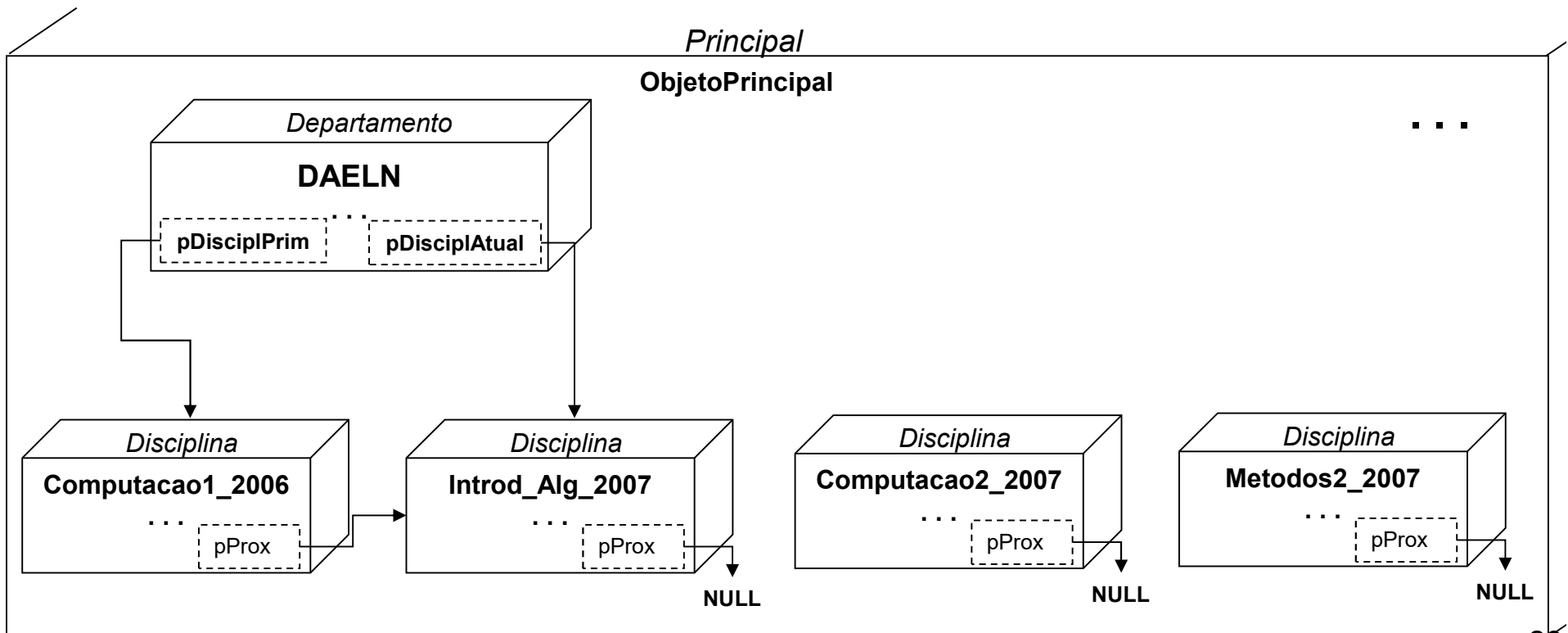
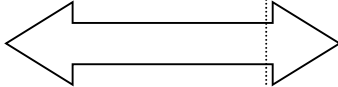
  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



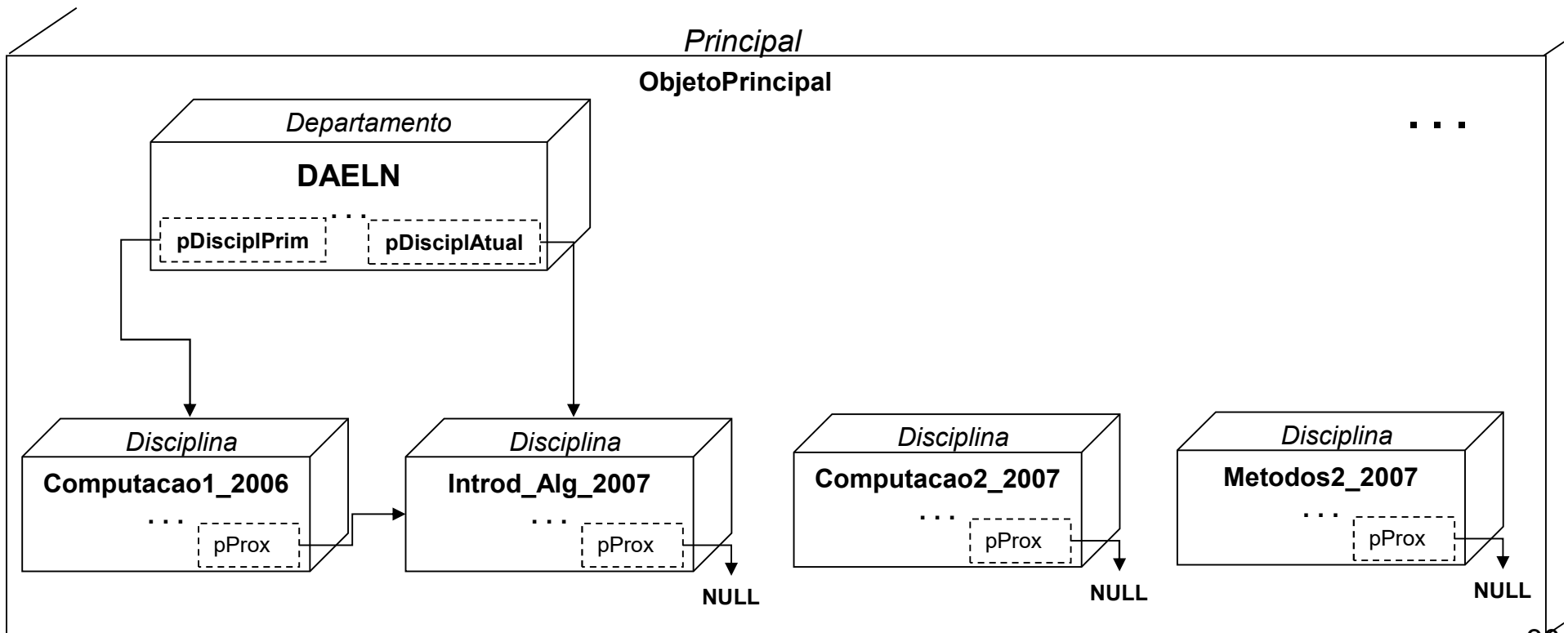
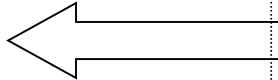
```

void Principal::InicializaDisciplinas ( )
{
    Computacao1_2006.setNome ( "Computacao I 2006" );
    Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
    Computacao2_2007.setNome ( "Computao II" );
    Metodos2_2007.setNome ( "Métodos II" );

    DAELN.incluaDisciplina ( &Computacao1_2006);
    DAELN.incluaDisciplina ( & Introd_Alg_2007);
    DAELN.incluaDisciplina ( &Computacao2_2007);
    DAELN.incluaDisciplina ( &Metodos2_2007);

    Computacao1_2006.setDepartamento ( &DAELN );
    Introd_Alg_2007.setDepartamento ( &DAELN );
    Computacao2_2007.setDepartamento ( &DAELN );
    Metodos2_2007.setDepartamento ( &DAELN );
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  DAELN.incluaDisciplina ( &Computacao1_2006);
  DAELN.incluaDisciplina ( & Introd_Alg_2007);
  DAELN.incluaDisciplina ( &Computacao2_2007);
  DAELN.incluaDisciplina ( &Metodos2_2007);

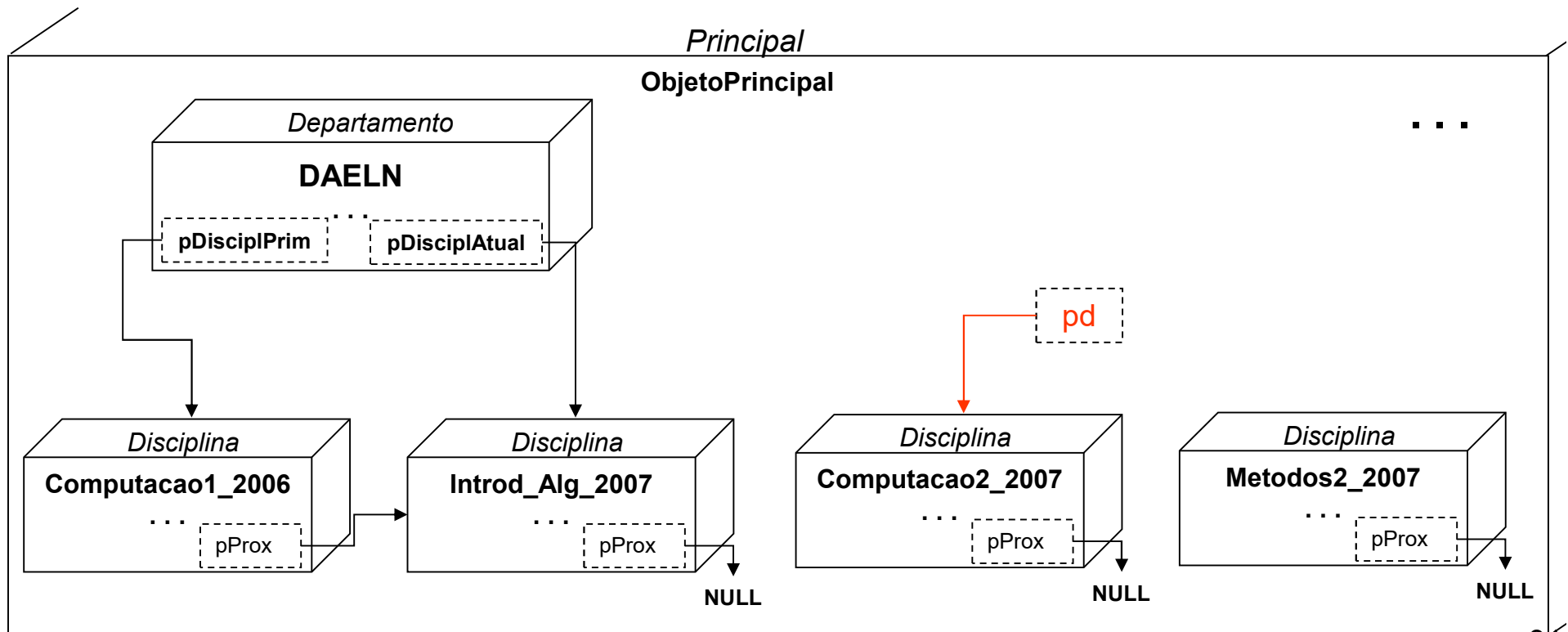
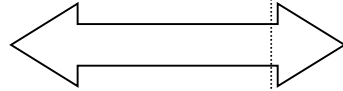
  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
    Computacao1_2006.setNome ( "Computacao I 2006" );
    Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
    Computacao2_2007.setNome ( "Computao II" );
    Metodos2_2007.setNome ( "Métodos II" );

    DAELN.incluaDisciplina ( &Computacao1_2006);
    DAELN.incluaDisciplina ( & Introd_Alg_2007);
    DAELN.incluaDisciplina ( &Computacao2_2007);
    DAELN.incluaDisciplina ( &Metodos2_2007);

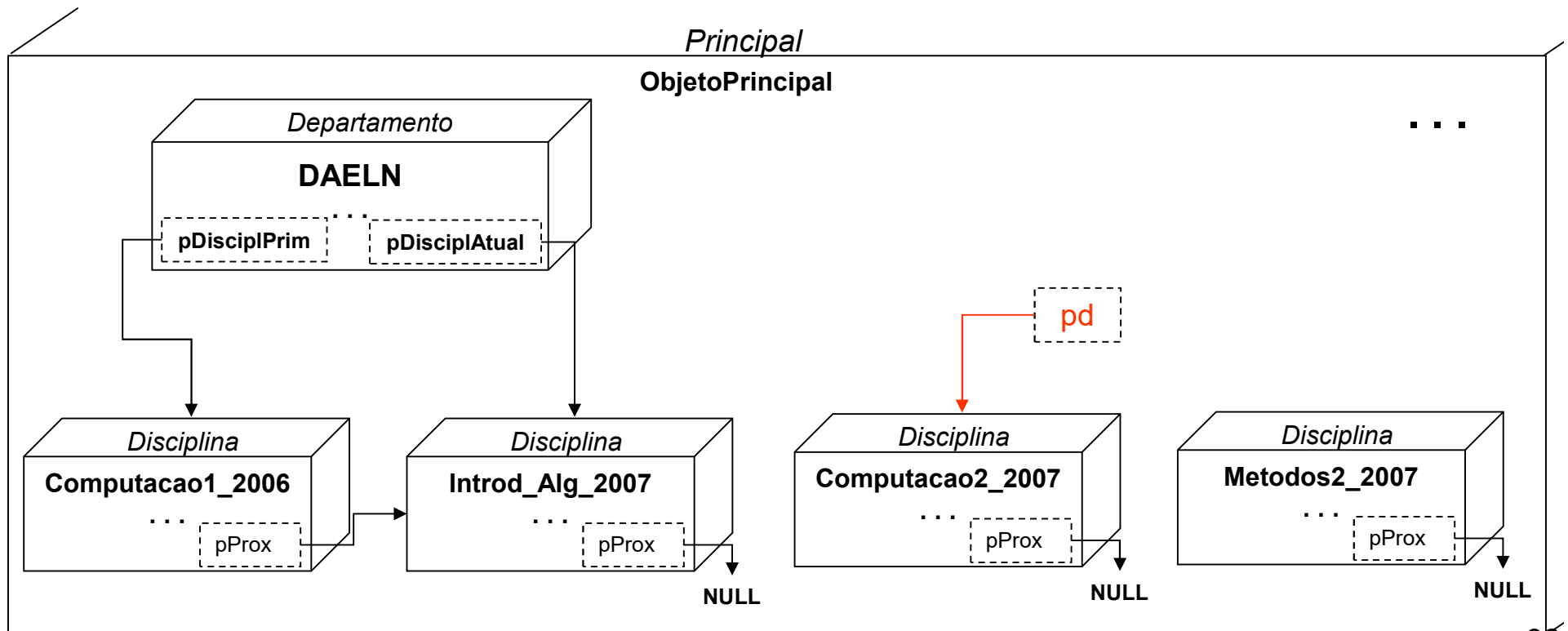
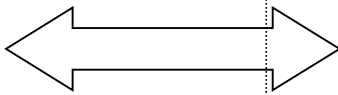
    Computacao1_2006.setDepartamento ( &DAELN );
    Introd_Alg_2007.setDepartamento ( &DAELN );
    Computacao2_2007.setDepartamento ( &DAELN );
    Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
    if ( pDisciplPrim == NULL )
    {
        pDisciplPrim = pd;
        pDisciplAtual = pd;
    }
    else
    {
        pDisciplAtual->pProx = pd;
        pDisciplAtual = pd;
    }
}

```




```

void Principal::InicializaDisciplinas ( )
{
    Computacao1_2006.setNome ( "Computacao I 2006" );
    Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
    Computacao2_2007.setNome ( "Computao II" );
    Metodos2_2007.setNome ( "Métodos II" );

    DAELN.incluaDisciplina ( &Computacao1_2006);
    DAELN.incluaDisciplina ( & Introd_Alg_2007);
    DAELN.incluaDisciplina ( &Computacao2_2007);
    DAELN.incluaDisciplina ( &Metodos2_2007);

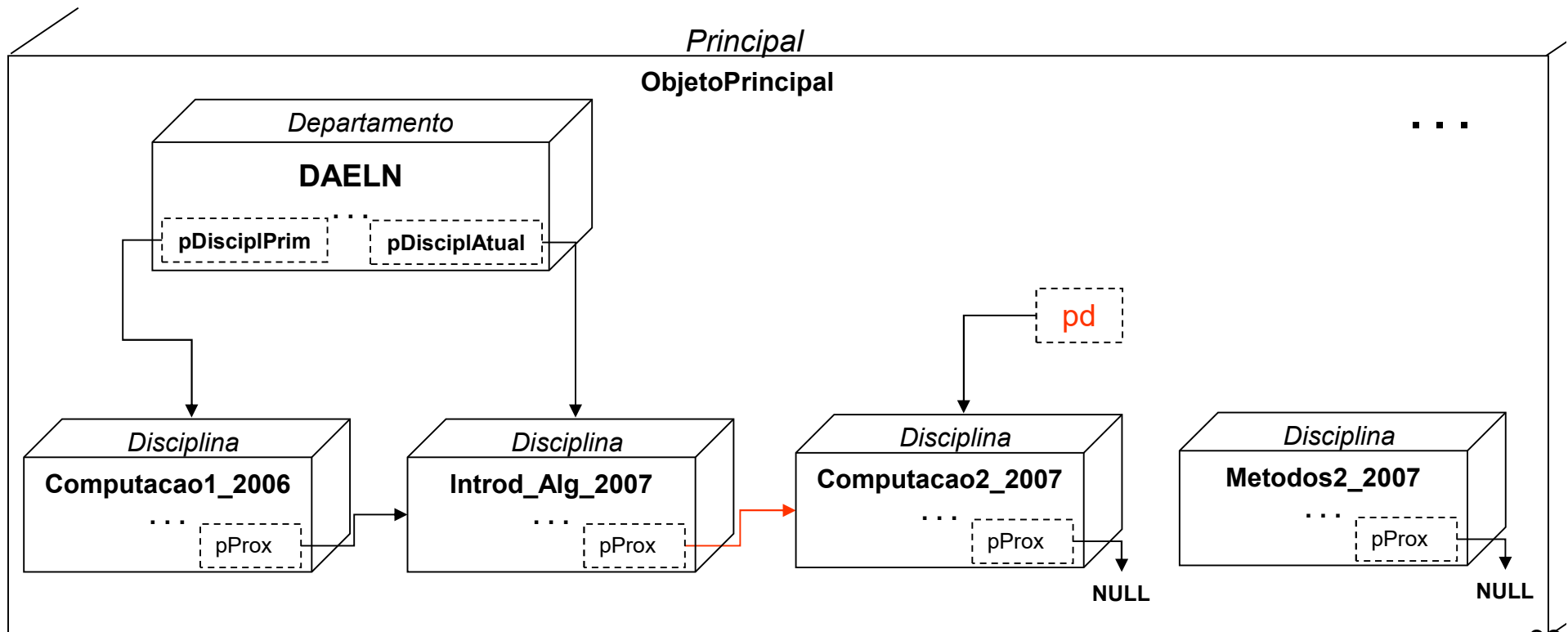
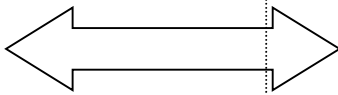
    Computacao1_2006.setDepartamento ( &DAELN );
    Introd_Alg_2007.setDepartamento ( &DAELN );
    Computacao2_2007.setDepartamento ( &DAELN );
    Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
    if ( pDisciplPrim == NULL )
    {
        pDisciplPrim = pd;
        pDisciplAtual = pd;
    }
    else
    {
        pDisciplAtual->pProx = pd;
        pDisciplAtual = pd;
    }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  DAELN.incluaDisciplina ( &Computacao1_2006);
  DAELN.incluaDisciplina ( & Introd_Alg_2007);
  DAELN.incluaDisciplina ( &Computacao2_2007);
  DAELN.incluaDisciplina ( &Metodos2_2007);

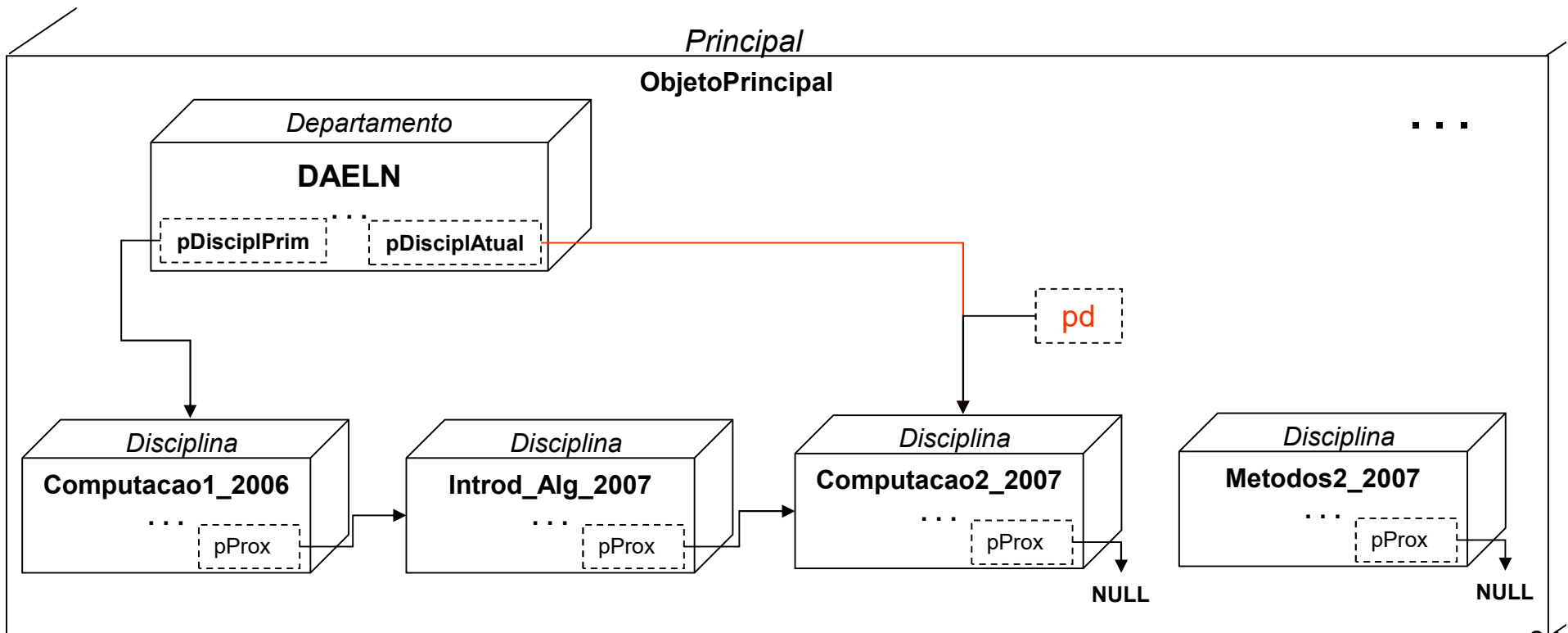
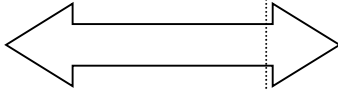
  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  DAELN.incluaDisciplina ( &Computacao1_2006);
  DAELN.incluaDisciplina ( & Introd_Alg_2007);
  DAELN.incluaDisciplina ( &Computacao2_2007);
  DAELN.incluaDisciplina ( &Metodos2_2007);

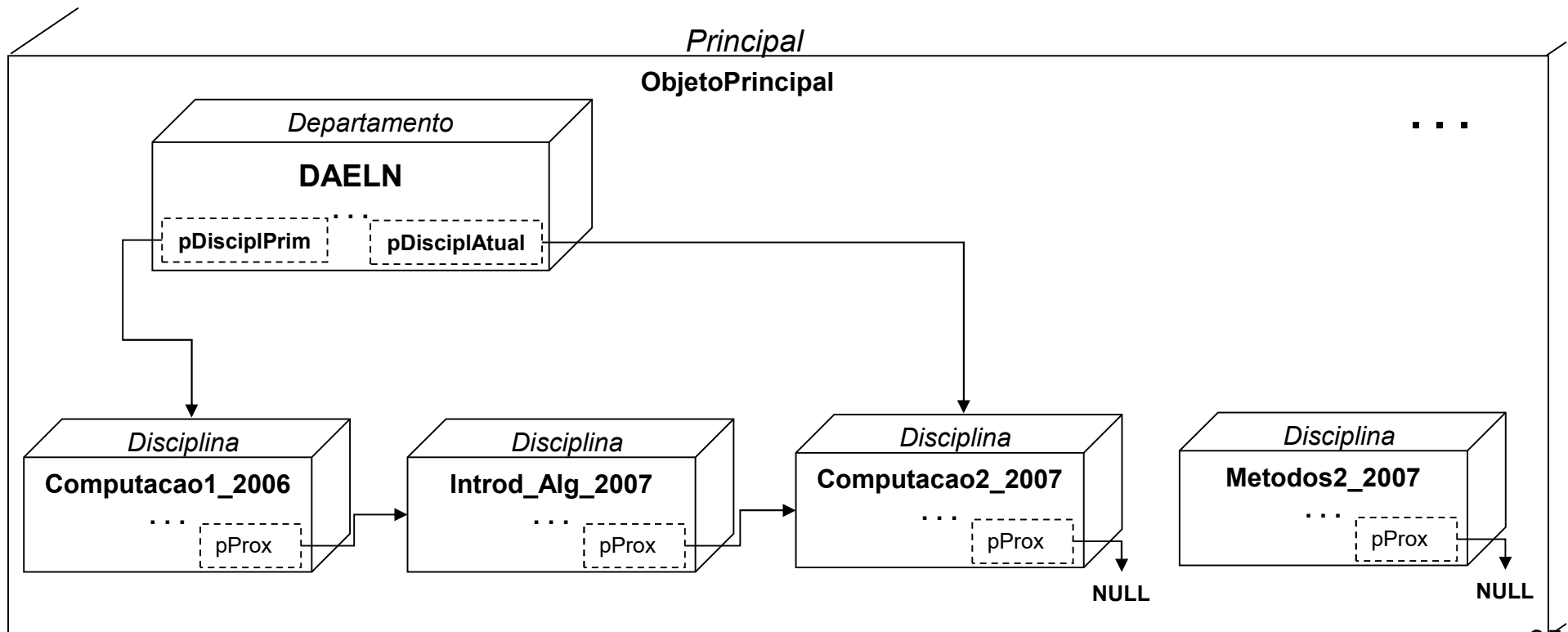
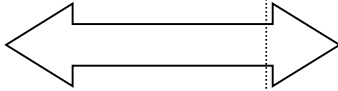
  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



Continue a “simulação”...

por meio do “diagrama de cubos”

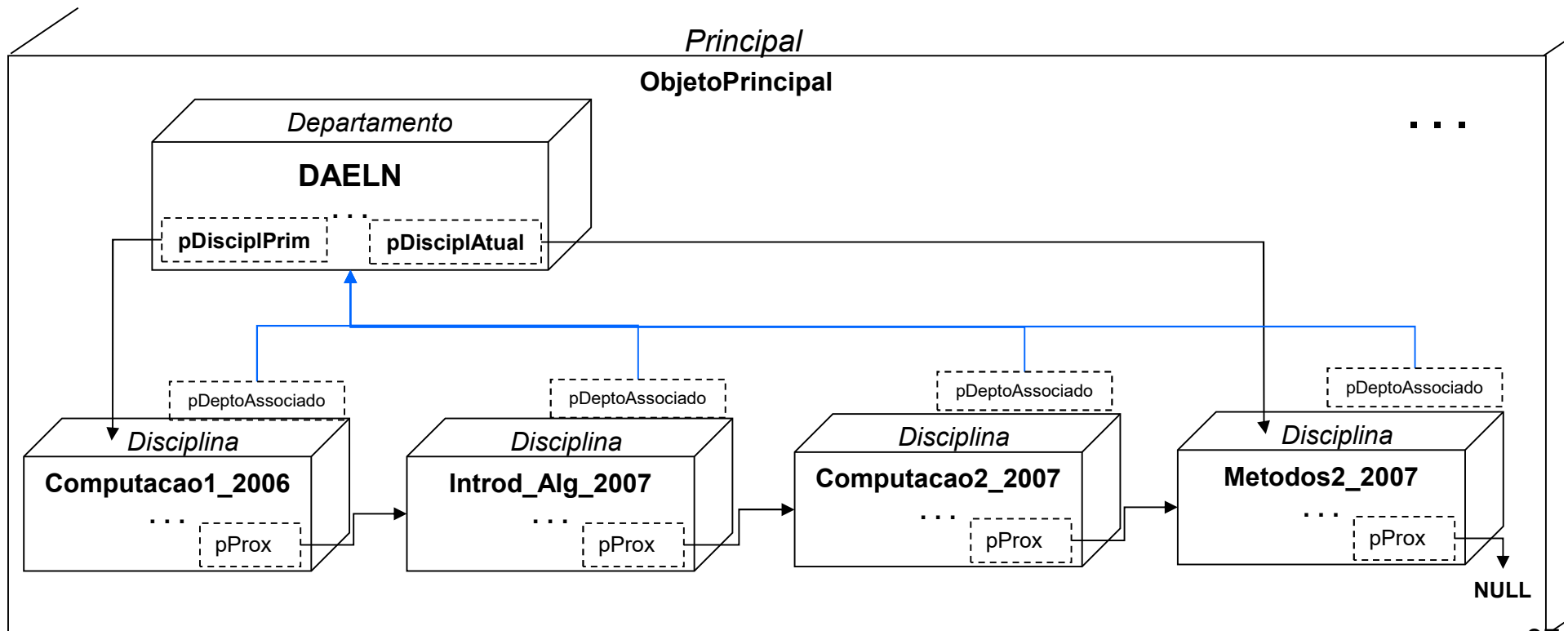
```

void Principal::InicializaDisciplinas ( )
{
    Computacao1_2006.setNome ( "Computacao I 2006" );
    Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
    Computacao2_2007.setNome ( "Computao II" );
    Metodos2_2007.setNome ( "Métodos II" );

    DAELN.incluaDisciplina ( &Computacao1_2006);
    DAELN.incluaDisciplina ( & Introd_Alg_2007);
    DAELN.incluaDisciplina ( &Computacao2_2007);
    DAELN.incluaDisciplina ( &Metodos2_2007);

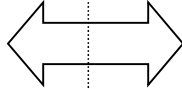
    Computacao1_2006.setDepartamento ( &DAELN );
    Introd_Alg_2007.setDepartamento ( &DAELN );
    Computacao2_2007.setDepartamento ( &DAELN );
    Metodos2_2007.setDepartamento ( &DAELN );
}

```



Neste cenário simule o código considerado

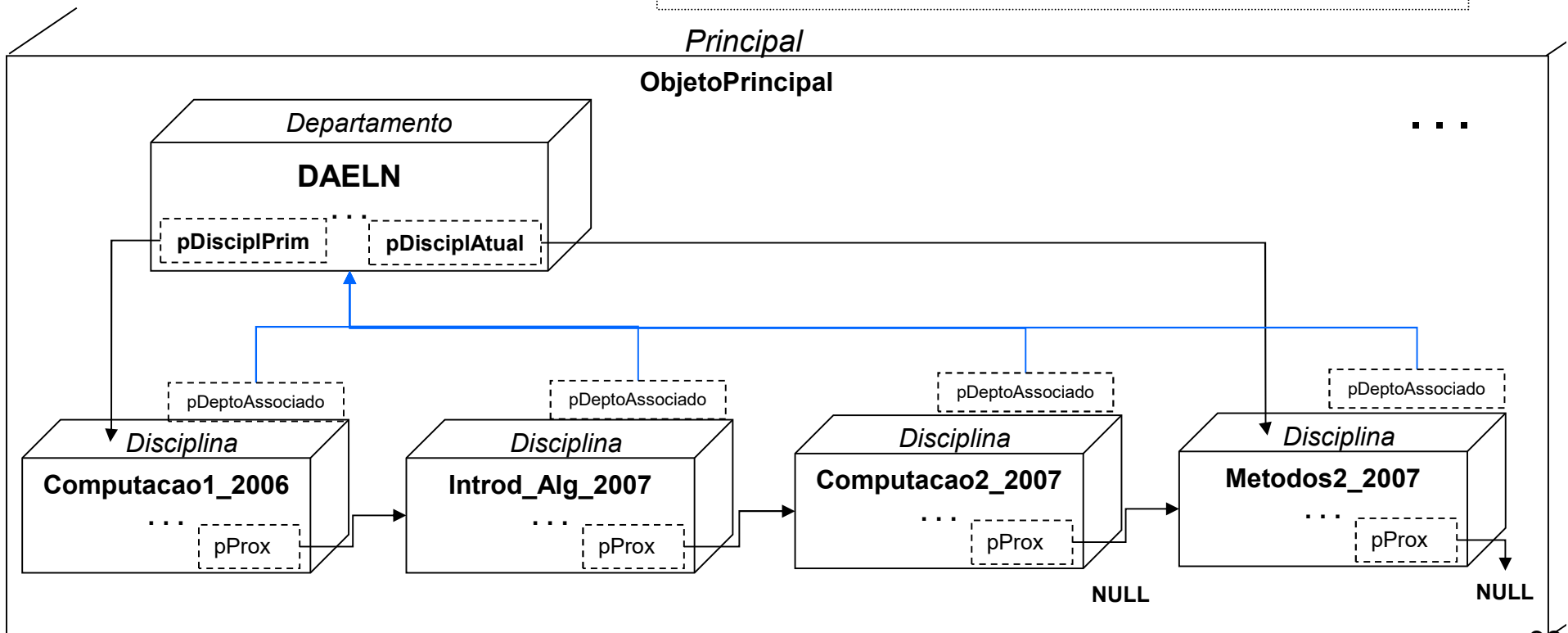
```
void Principal::ListeDiscDeptos ( )  
{  
    DAELN.listeDisciplinas ( );  
    printf ("\n");  
}
```



```
void Departamento::listeDisciplinas ( )  
{  
    Disciplina* pAux;  
    pAux = pDisciplPrim;  
  
    while ( pAux != NULL )  
    {  
        printf (" A disciplina %s pertence ao Departamento %s \n",  
                pAux->getNome(), nome );  
  
        pAux = pAux->pProx;  
    }  
}
```

Principal

ObjetoPrincipal



Voltando ao código...

para melhorá-lo

Usando o ponteiro implícito *this*.

```
#include "Disciplina.h"
#include <stdio.h>
#include <string.h>

Disciplina::Disciplina ( char* ac )
{
    pDeptoAssociado = NULL;
    pProx           = NULL;
    strcpy (area_conhecimento, ac);
}

Disciplina::~Disciplina ( )
{
    DeptoAssociado = NULL;
    pProx          = NULL;
}

void Disciplina::setId ( int n )
{
    id = n;
}

int Disciplina::getId ( )
{
    return id;
}

void Disciplina::setNome (char* n)
{
    strcpy ( nome, n );
}

char* Disciplina::getNome ( )
{
    return nome;
}
```

```
void Disciplina::setDepartamento ( Departamento* pdpto )
{
    pDeptoAssociado = pdpto;

    // Cada vez que um Departamento é associado a uma Disciplina,
    // esta Disciplina passa a fazer parte da lista de disciplina
    // do Departamento, por meio do comando abaixo.

    pdpto->incluaDisciplina ( this );
}

Departamento* Disciplina::getDepartamento ( )
{
    return pDeptoAssociado;
}
```

2a versão.


```

void Principal::InicializaDisciplinas ( )
{
    Computacao1_2006.setNome ( "Computacao I 2006" );
    Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
    Computacao2_2007.setNome ( "Computao II" );
    Metodos2_2007.setNome ( "Métodos II" );

    Computacao1_2006.setDepartamento ( &DAELN );
    Introd_Alg_2007.setDepartamento ( & DAELN );
    Computacao2_2007.setDepartamento ( & DAELN );
    Metodos2_2007.setDepartamento ( & DAELN );
}

void Principal::CalcIdadeProfs ( )
{
    // Cálculo da idade.
    Simao.Calc_Idade ( diaAtual, mesAtual, anoAtual );
    Einstein.Calc_Idade ( diaAtual, mesAtual, anoAtual );
    Newton.Calc_Idade ( diaAtual, mesAtual, anoAtual );
    printf ("\n");
}

void Principal::UnivOndeProfsTrabalham ( )
{
    // Universidade que a Pessoa trabalha.
    Simao.OndeTrabalho ( );
    Einstein.OndeTrabalho ( );
    Newton.OndeTrabalho ( );
    printf ("\n");
}

void Principal::DepOndeProfsTrabalham ( )
{
    // Departamento que a Pessoa trabalha.
    Simao.QualDepartamentoTrabalho ( );
    Einstein.QualDepartamentoTrabalho ( );
    Newton.QualDepartamentoTrabalho ( );
    printf ( "\n" );
}

```

```

void Principal::ListeDiscDeptos ( )
{
    EletronicaUTFPR.listeDisciplinas ( );
    printf ("\n");
}

void Principal::Executar ( )
{
    CalcIdadeProfs ( );
    UnivOndeProfsTrabalham ( );
    DepOndeProfsTrabalham ( );
    ListeDepPorUniv ( );
    // Metodos2.listeAlunos();
    ListeDiscDeptos ( );
}

```

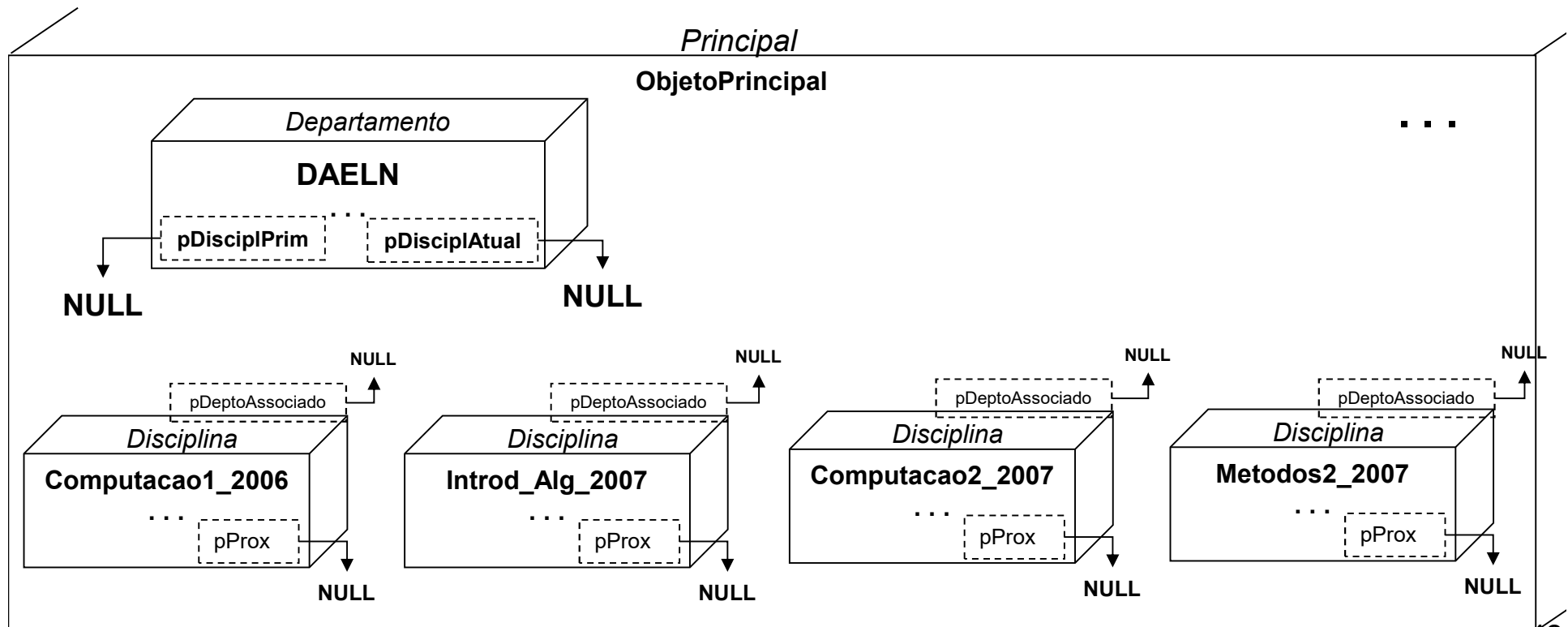
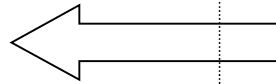
2a versão.

```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

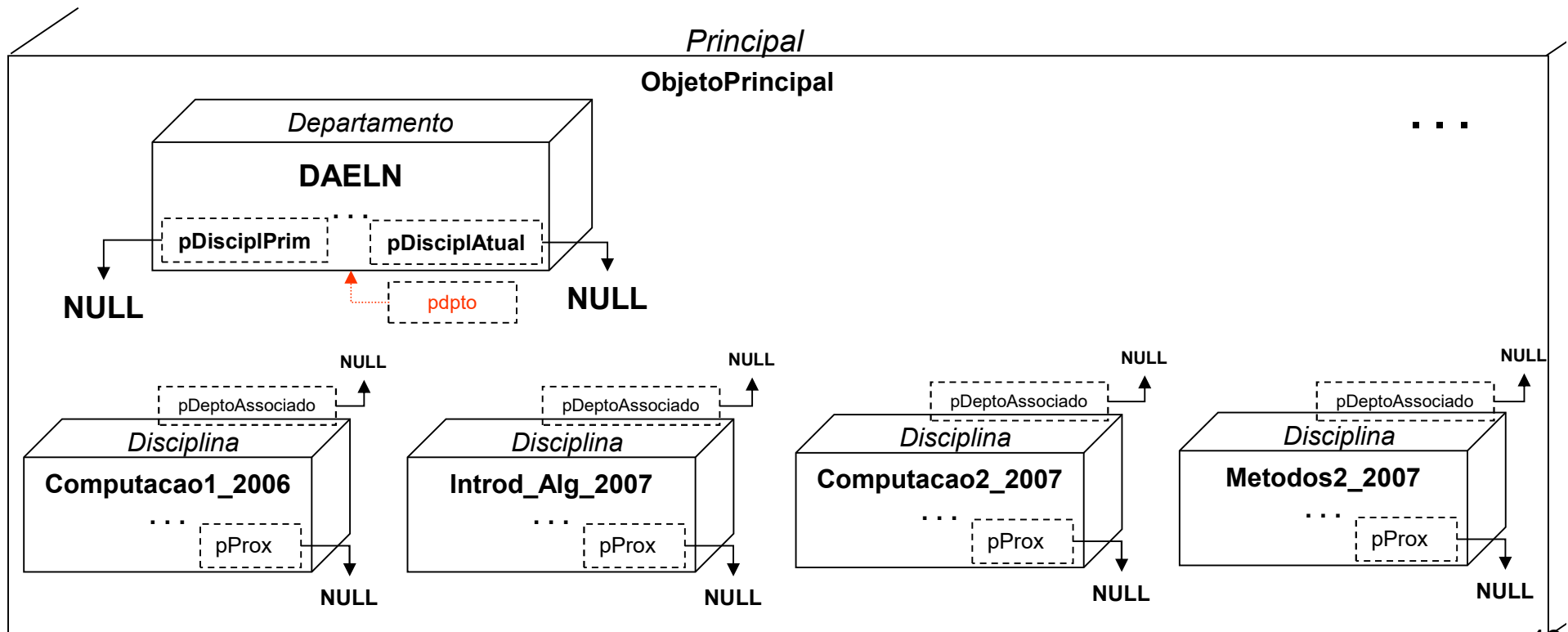
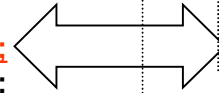
```

```

void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;

  pdpto->incluaDisciplina ( this );
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

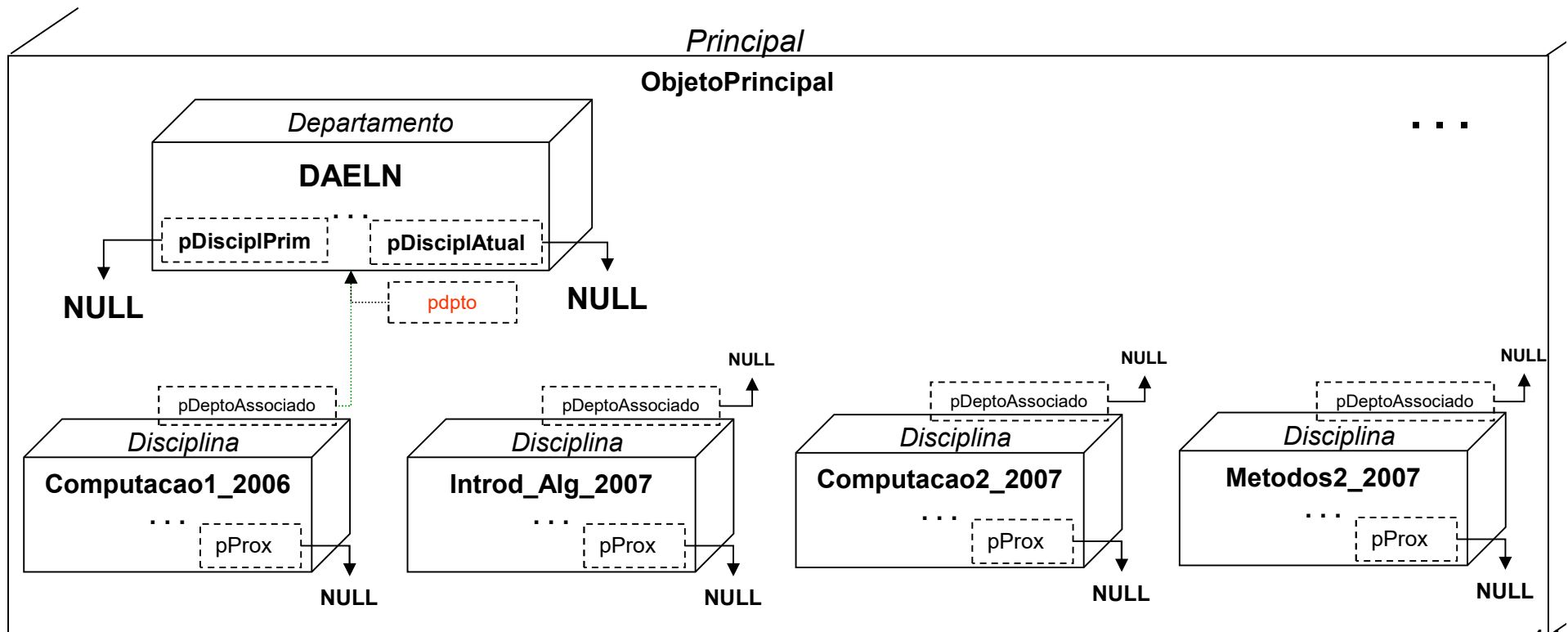
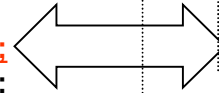
```

```

void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;

  pdpto->incluaDisciplina ( this );
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

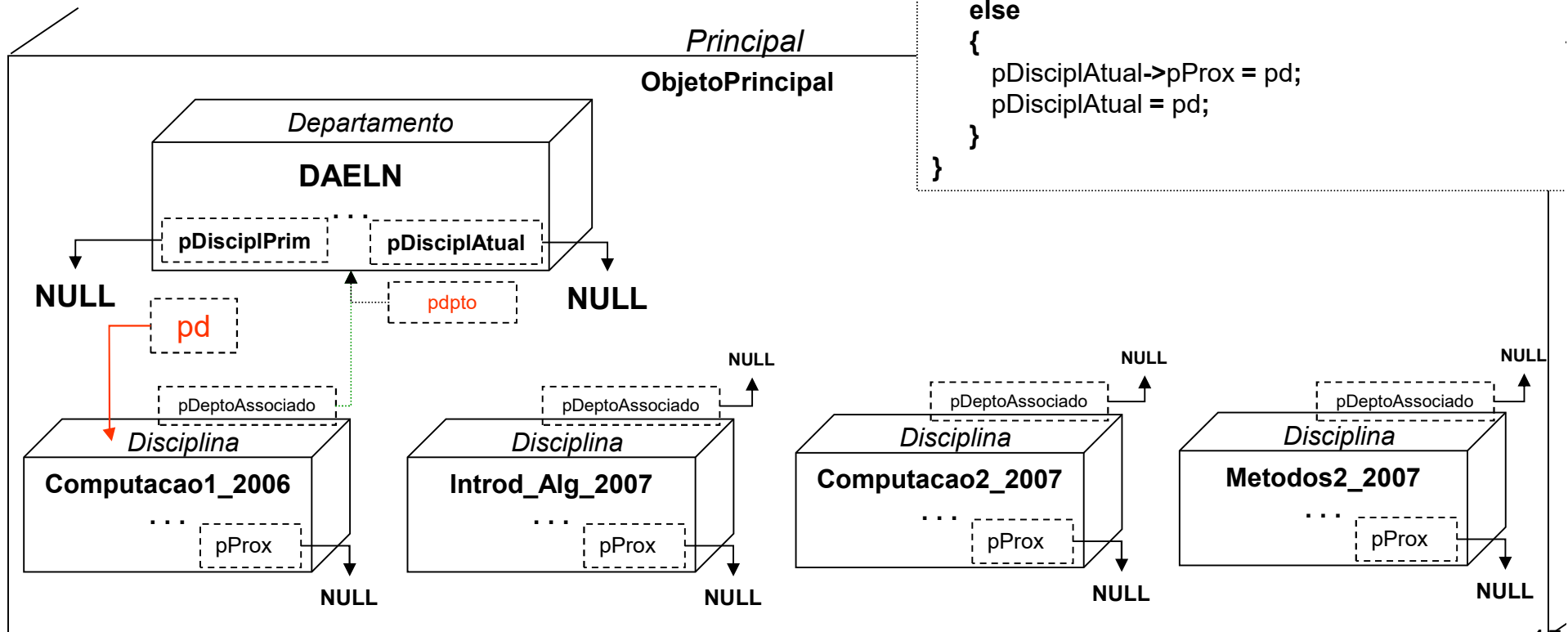
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

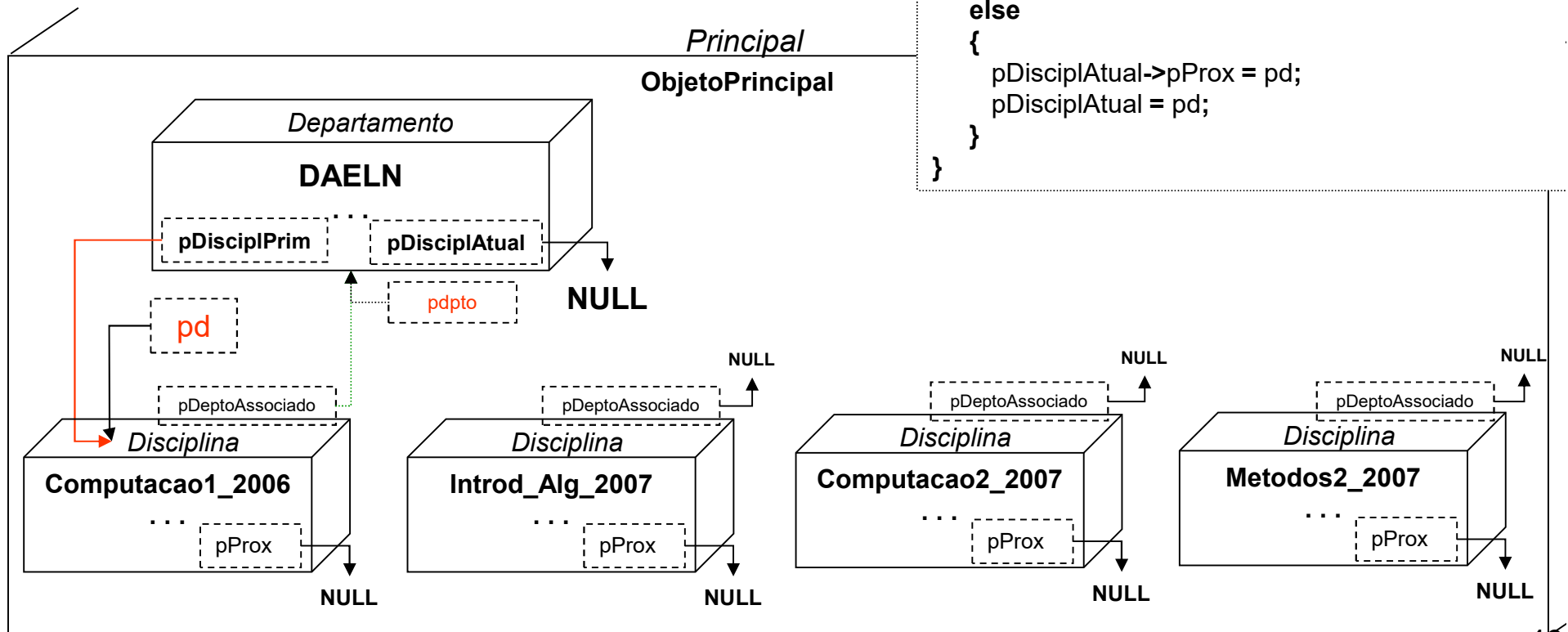
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

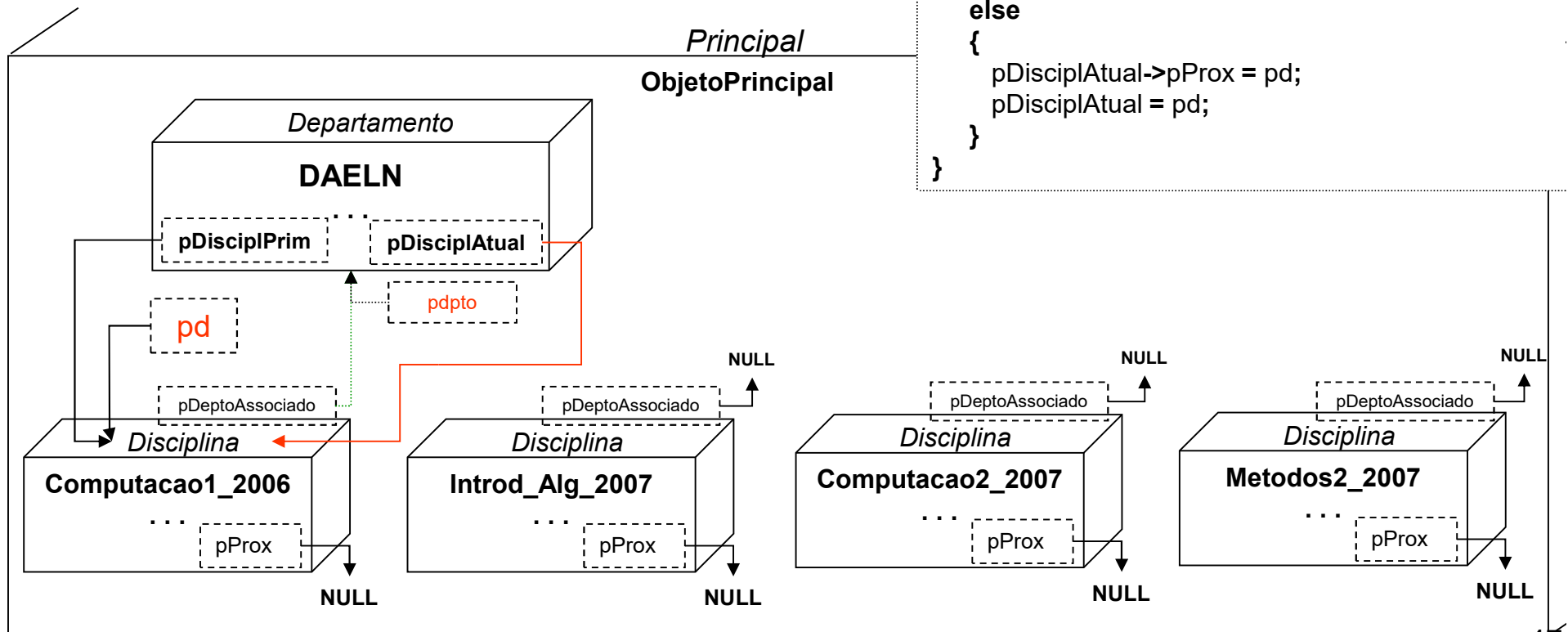
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

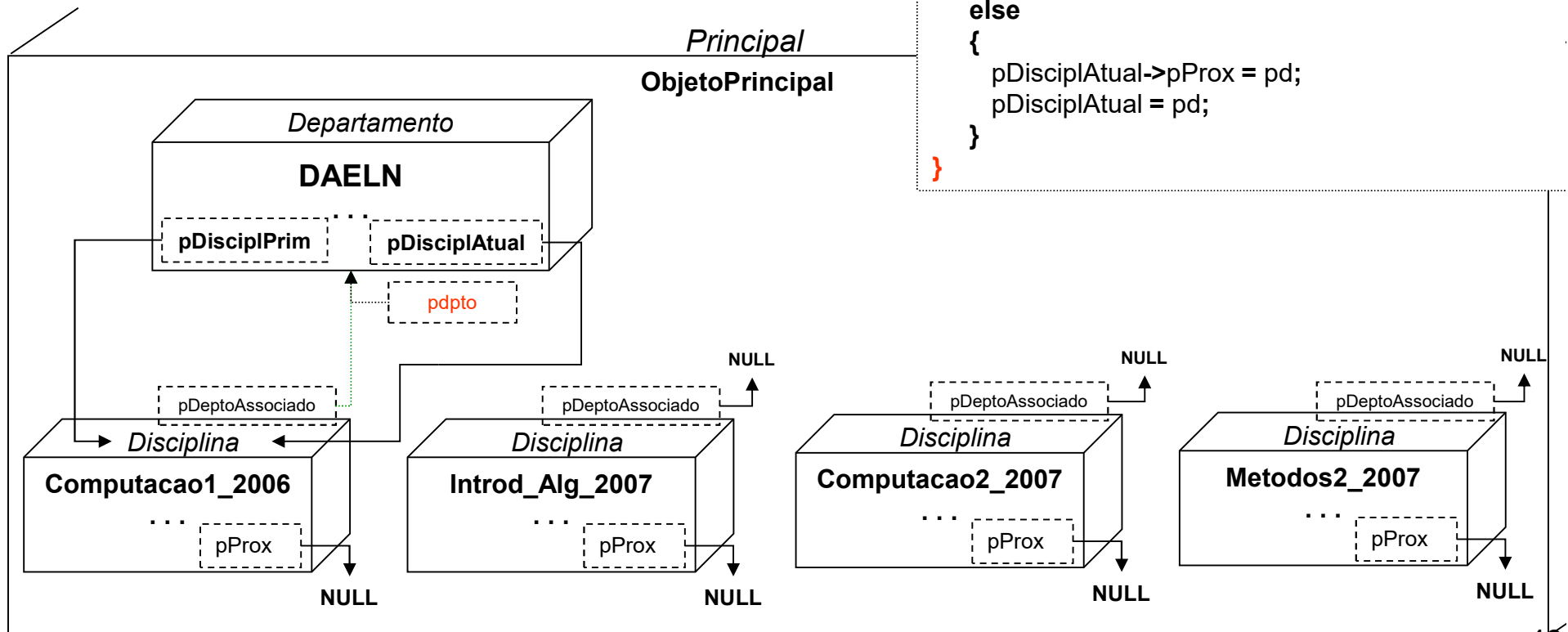
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```




```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

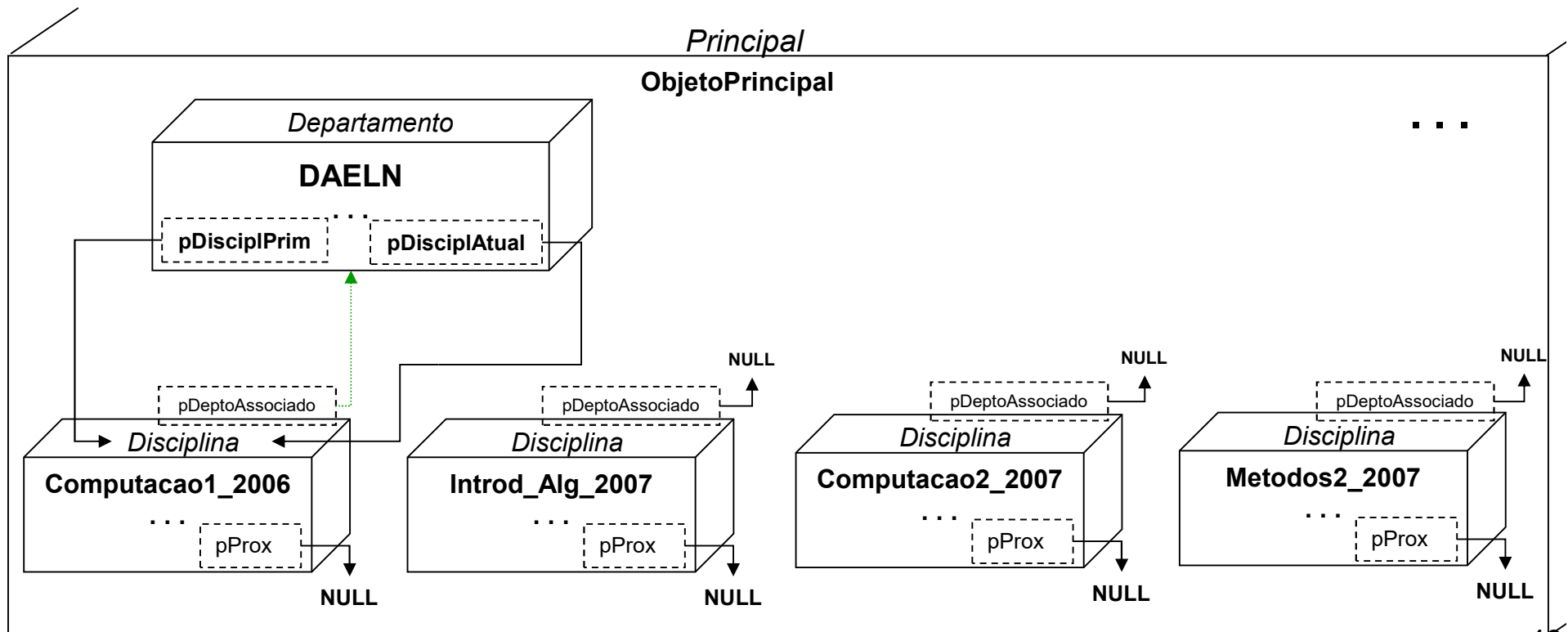
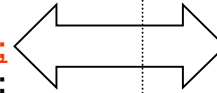
```

```

void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;

  pdpto->setDisciplina ( this );
}

```

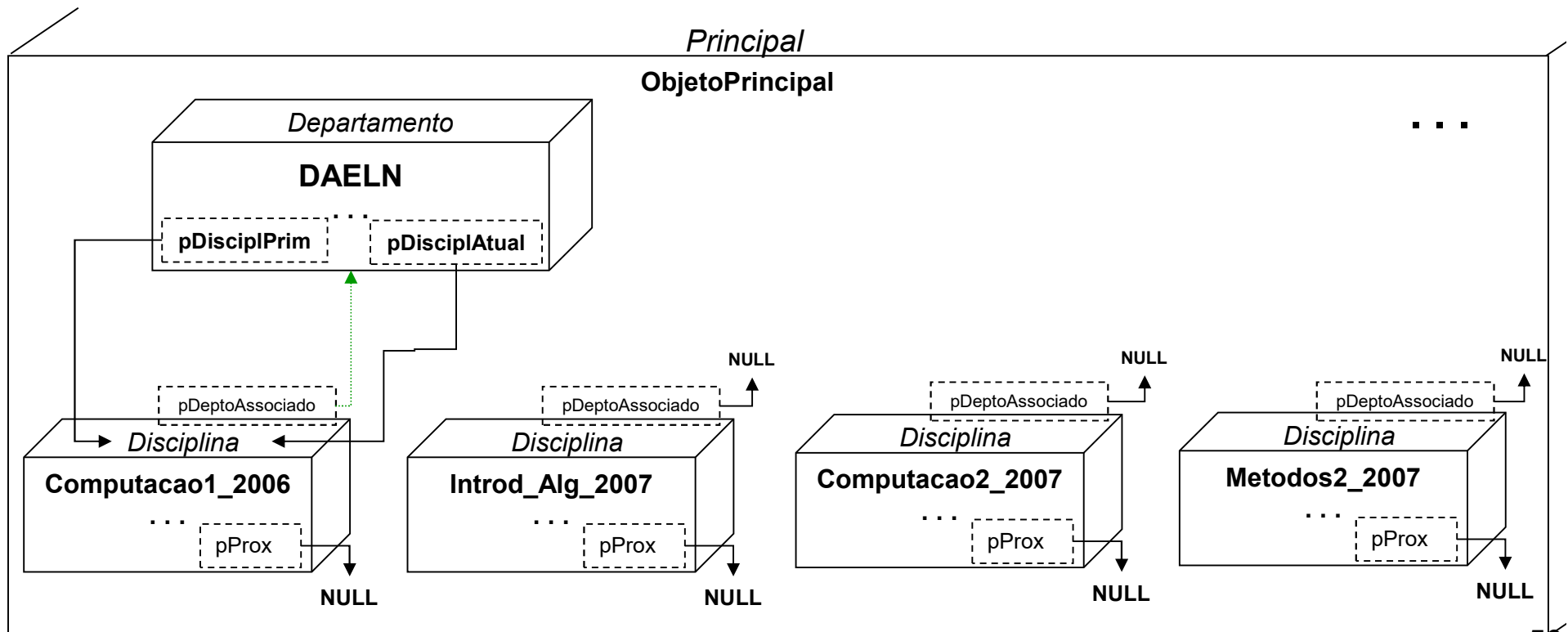
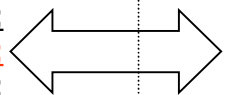


```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

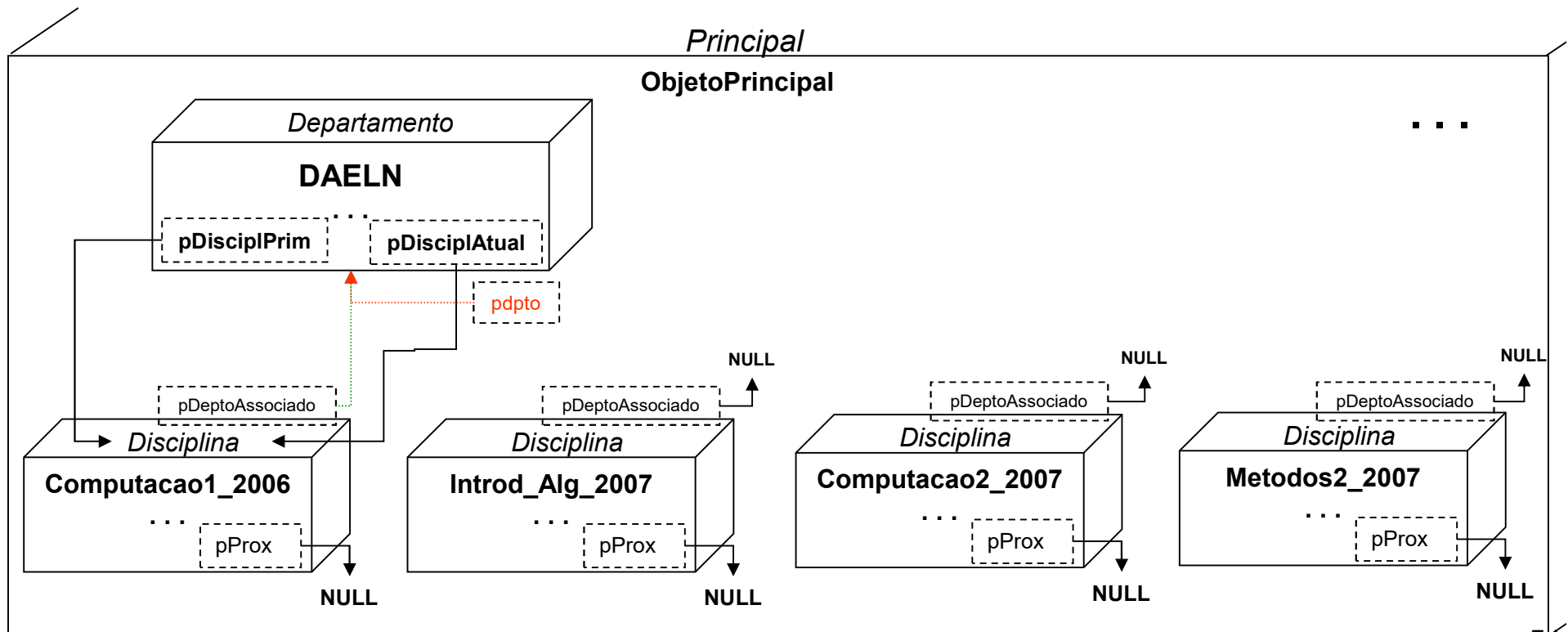
  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

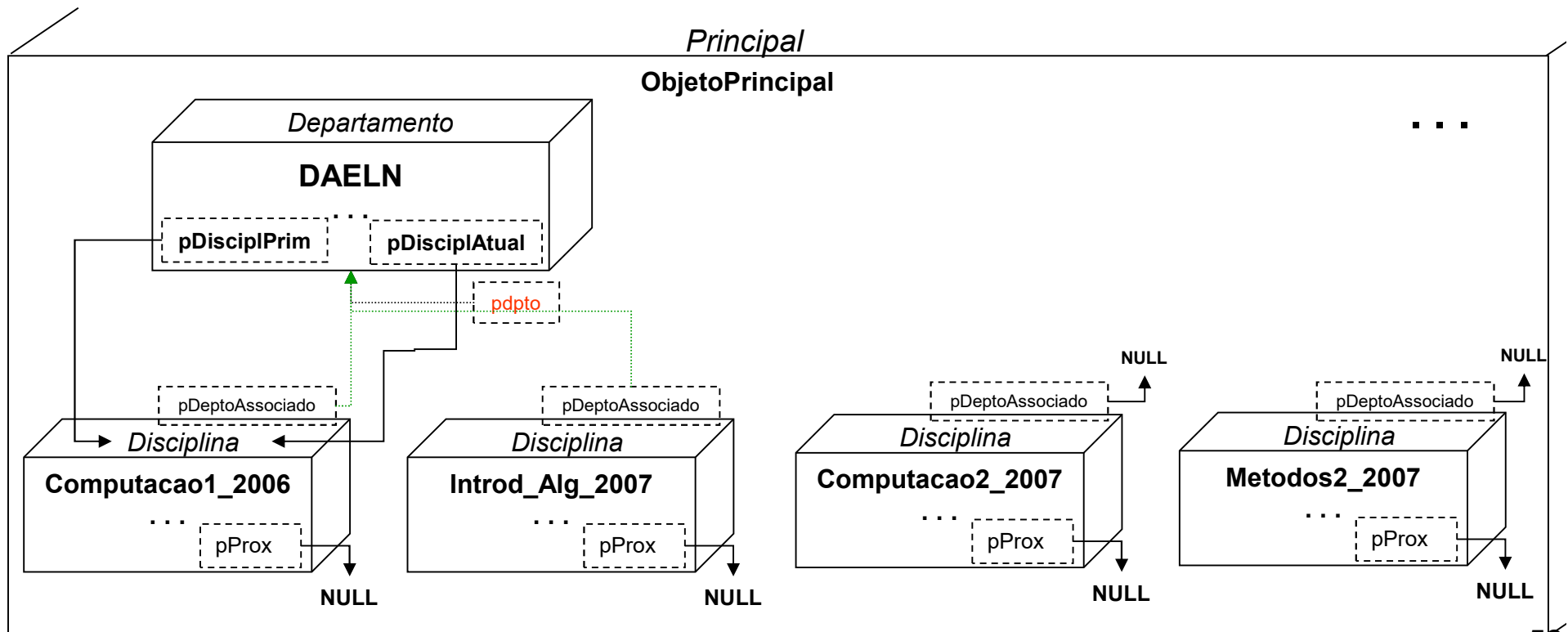
```

```

void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;

  pdpto->incluaDisciplina ( this );
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

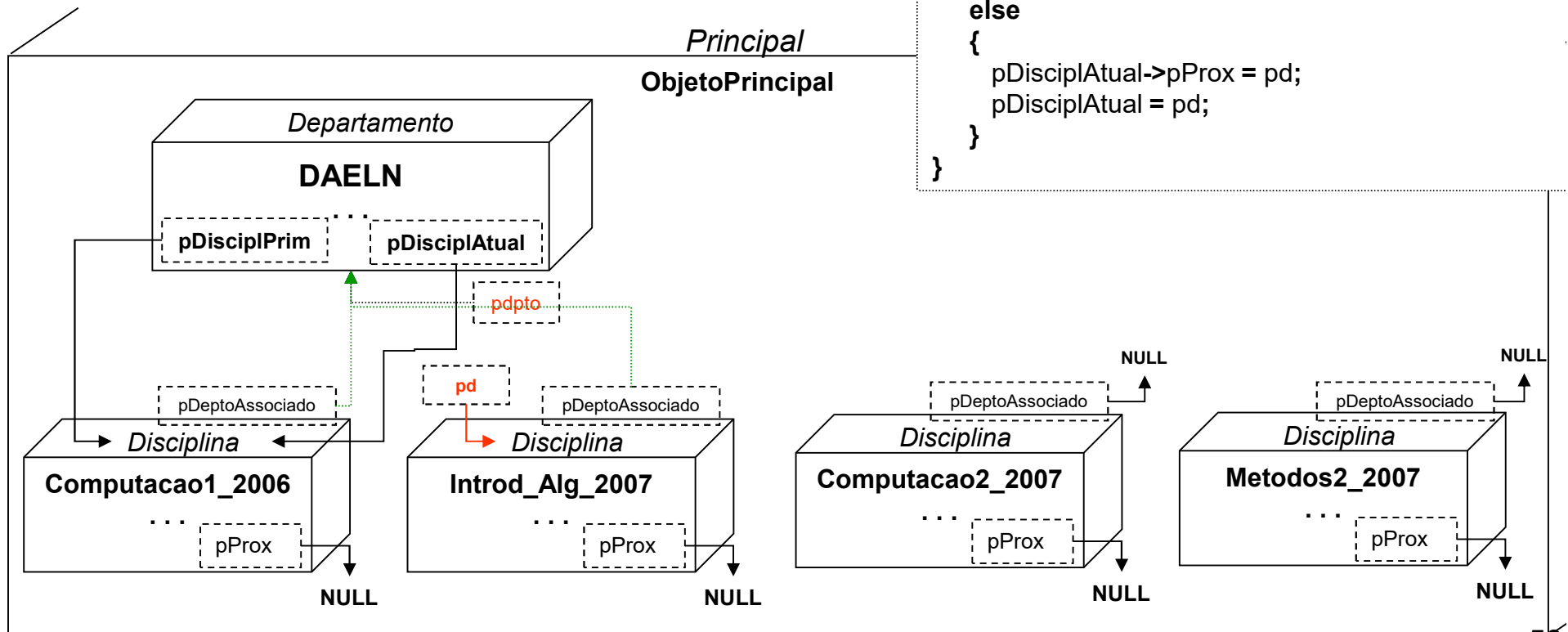
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

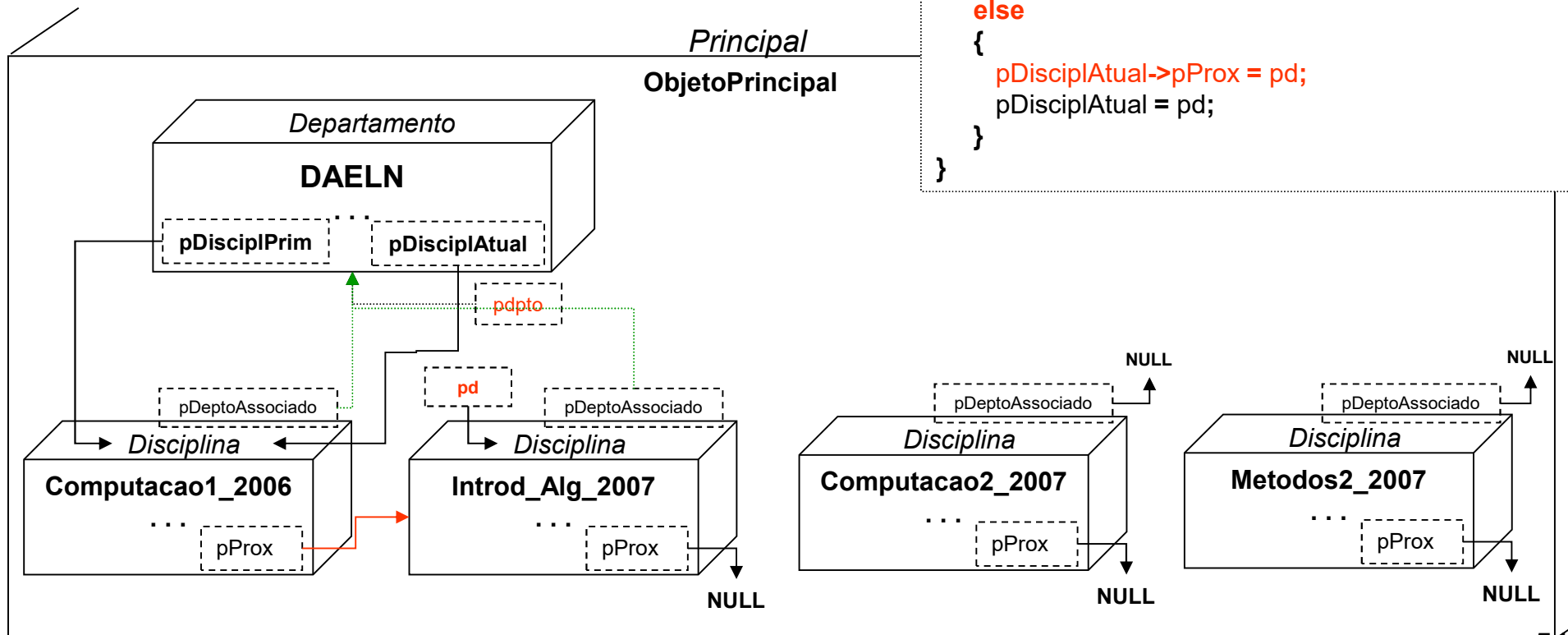
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

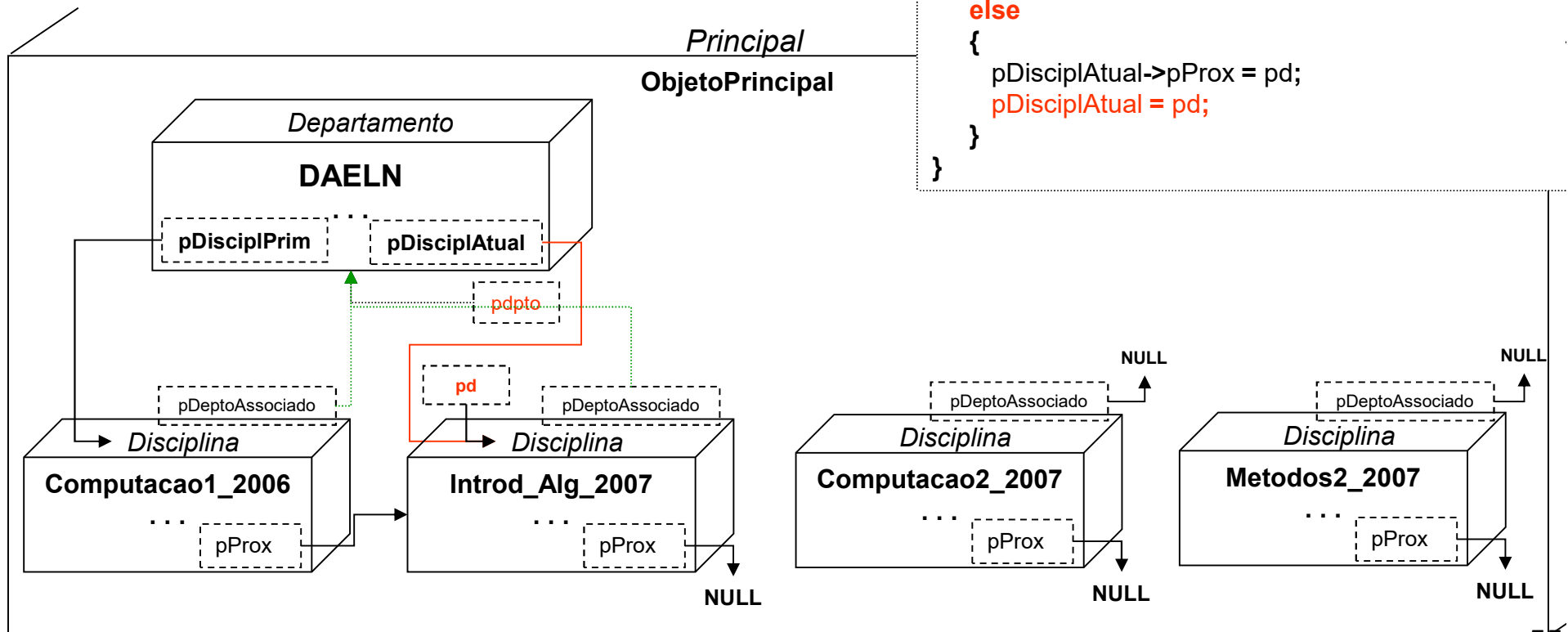
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

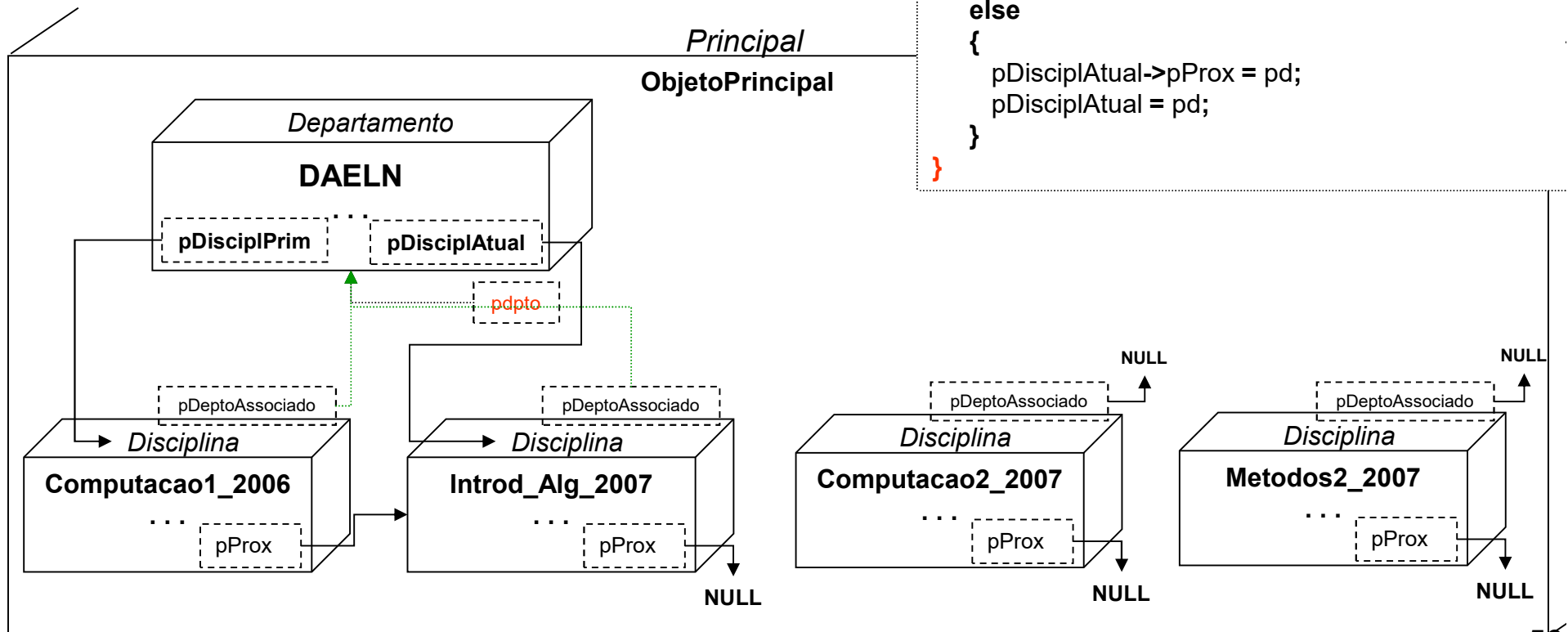
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pDisciplAtual = pd;
  }
}

```




```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

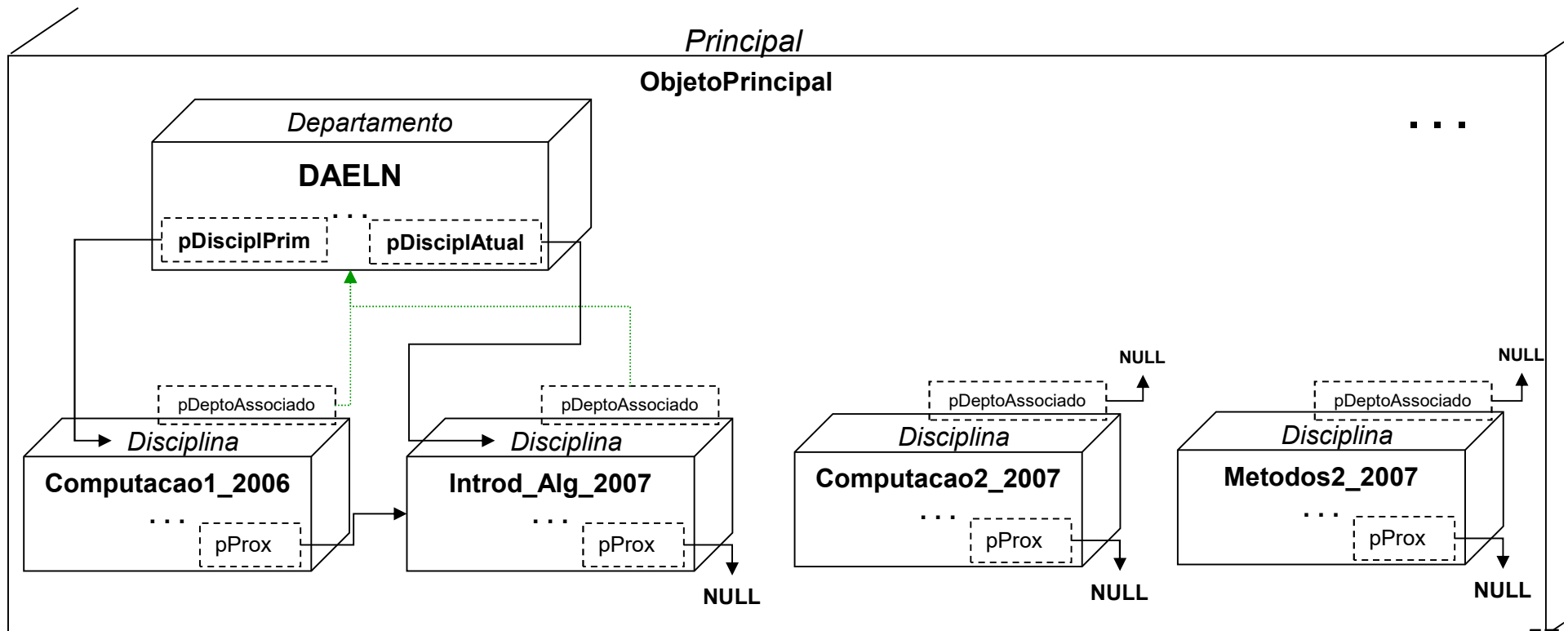
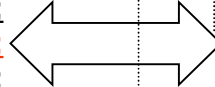
  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->setDisciplina ( this );
}

```

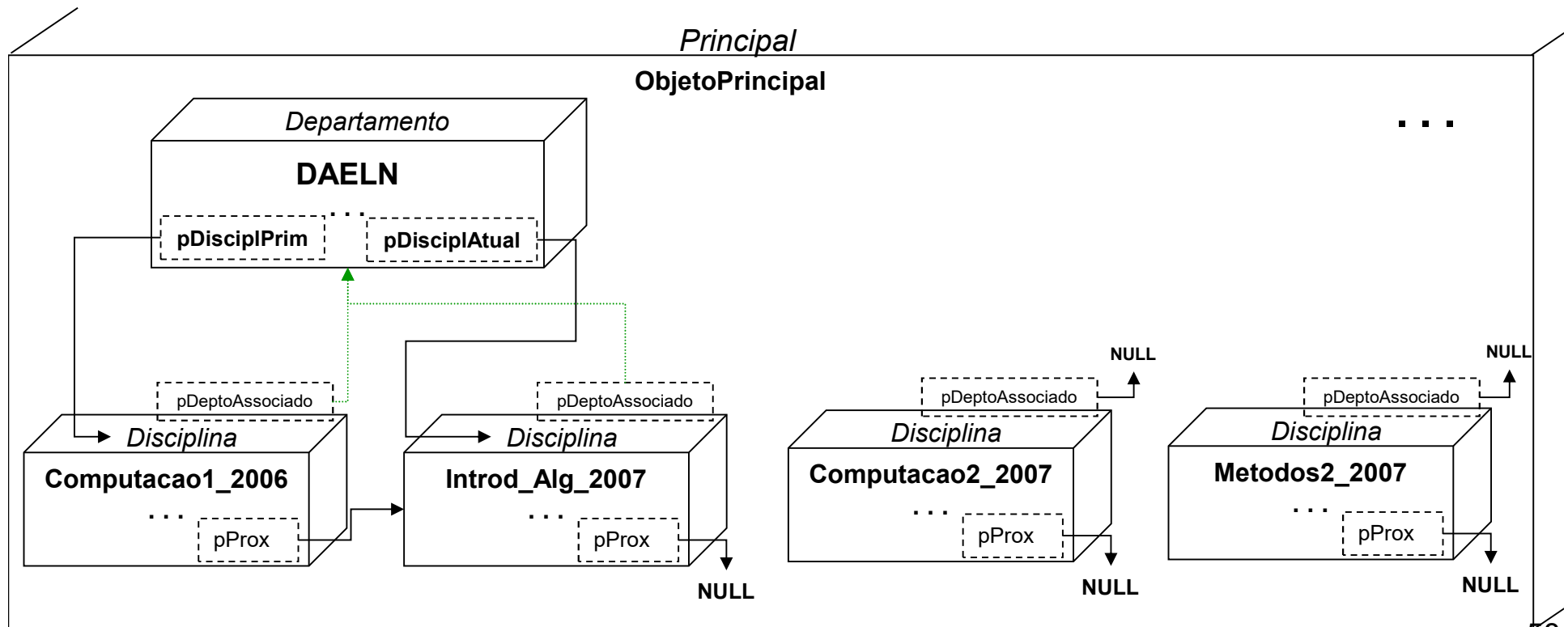
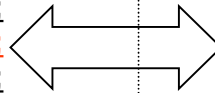


```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```



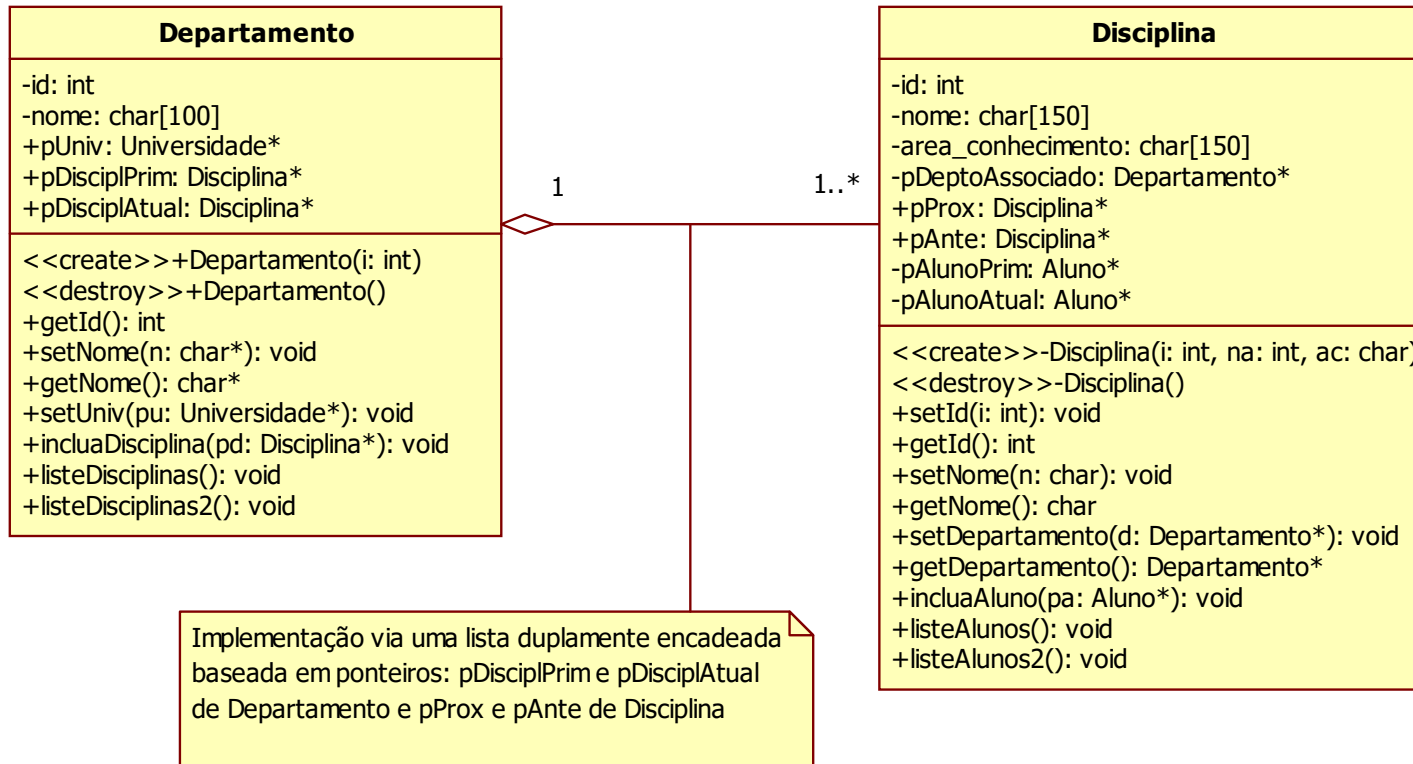
Continue a “simulação”...

por meio do “diagrama de cubos”

Exercício 1B

No exercício anterior, faça com que a lista seja duplamente encadeada.

Digrama de Classes - Projeto



'Correção de Exercício'

```
#ifndef _DEPARTAMENTO_H_
#define _DEPARTAMENTO_H_
class Universidade;
class Disciplina;

class Departamento
{
private:
    char nome[130];
    // Associação para com uma Universidade.
    Universidade* pUniv;
    // Associação para com várias Disciplinas.
    Disciplina *pDisciplPrim;
    Disciplina *pDisciplAtual;
public:
    Departamento ( );
    ~Departamento ( );

    void setNome ( char* n );
    char* getNome ( );
    void setUniversidade(Universidade* u);
    Universidade* getUniversidade ( );

    void incluaDisciplina ( Disciplina* d );
    void listeDisciplinas ( );
    void listeDisciplinas2 ( );
};
#endif
```

Cada Departamento deve ser capaz de armazenar uma lista de disciplinas

Esta lista deverá ser duplamente encadeada.

```
#ifndef _DISCIPLINA_H_
#define _DISCIPLINA_H_
#include "Aluno.h"
#include "Departamento.h"
class Disciplina
{
private:
    int id;
    char nome[150];
    char area_conhecimento[150];
    Departamento* pDeptoAssociado;
public:
    Disciplina ( char* ac = "" );
    ~Disciplina ( );

    Disciplina* pProx; // Deveria ser privado:
                       // re-fazer...
    Disciplina* pAnte; // Idem

    void setId ( int n );
    int getId ( );
    void setNome ( char* n );
    char* getNome ( );

    void setDepartamento (Departamento* pd);
    Departamento* getDepartamento ( );
};
#endif
```

```

#include "Departamento.h"
#include "Universidade.h"
#include "Disciplina.h"
#include <stdio.h>

Departamento::Departamento ( )
{
    pDisciplPrim = NULL;
    pDisciplAtual = NULL;
}

Departamento::~Departamento ( )
{
    pDisciplPrim = NULL;
    pDisciplAtual = NULL;
}

void Departamento::setNome ( char* n )
{
    strcpy ( nome, n );
}

char* Departamento::getNome ( )
{
    return nome;
}

void Departamento::setUniversidade ( Universidade* pu )
{
    pUniv = pu;
}

Universidade* Departamento::getUniversidade ( )
{
    return pUniv;
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
    if ( DisciplPrim == NULL )
    {
        pDisciplPrim      = pd;
        pDisciplAtual     = pd;
    }
    else
    {
        pDisciplAtual->prox = pd;
        pd->pAnte           = pDisciplAtual;
        pDisciplAtual     = pd;
    }
}

void Departamento::listeDisciplinas ( )
{
    Disciplina* pAux;
    pAux = pDisciplPrim;
    while ( pAux != NULL )
    {
        printf(" A disciplina %s pertence ao Departamento %s \n",
            pAux->getNome(), nome );
        pAux = pAux->pProx;
    }
}

void Departamento::listeDisciplinas2 ( )
{
    Disciplina* pAux;
    pAux = pDisciplAtual;
    while ( pAux != NULL )
    {
        printf(" A disciplina %s pertence ao Departamento %s \n",
            pAux->getNome(), nome);
        pAux = pAux->pAnte;
    }
}

```

```

#include "Disciplina.h"
#include <stdio.h>
#include <string.h>

Disciplina::Disciplina ( char* ac )
{
    pDeptoAssociado = NULL;
    pProx           = NULL;
    pAnte           = NULL;
    strcpy (area_conhecimento, ac);
}

Disciplina::~Disciplina ( )
{
    pDeptoAssociado = NULL;
    pProx           = NULL;
    pAnte           = NULL;
}

void Disciplina::setId ( int n )
{
    id = n;
}

int Disciplina::getId ( )
{
    return id;
}

void Disciplina::setNome (char* n)
{
    strcpy ( nome, n );
}

char* Disciplina::getNome ( )
{
    return nome;
}

```

```

void Disciplina::setDepartamento ( Departamento* pd )
{
    DeptoAssociado = pd;

    // Cada vez que um Departamento é associado a uma Disciplina,
    // esta Disciplina passa a fazer parte da lista de disciplina
    // do Departamento, por meio do comando abaixo.

    pd->incluaDisciplina ( this );
}

Departamento* Disciplina::getDepartamento ( )
{
    return DeptoAssociado;
}

```


Faça uma “simulação”...

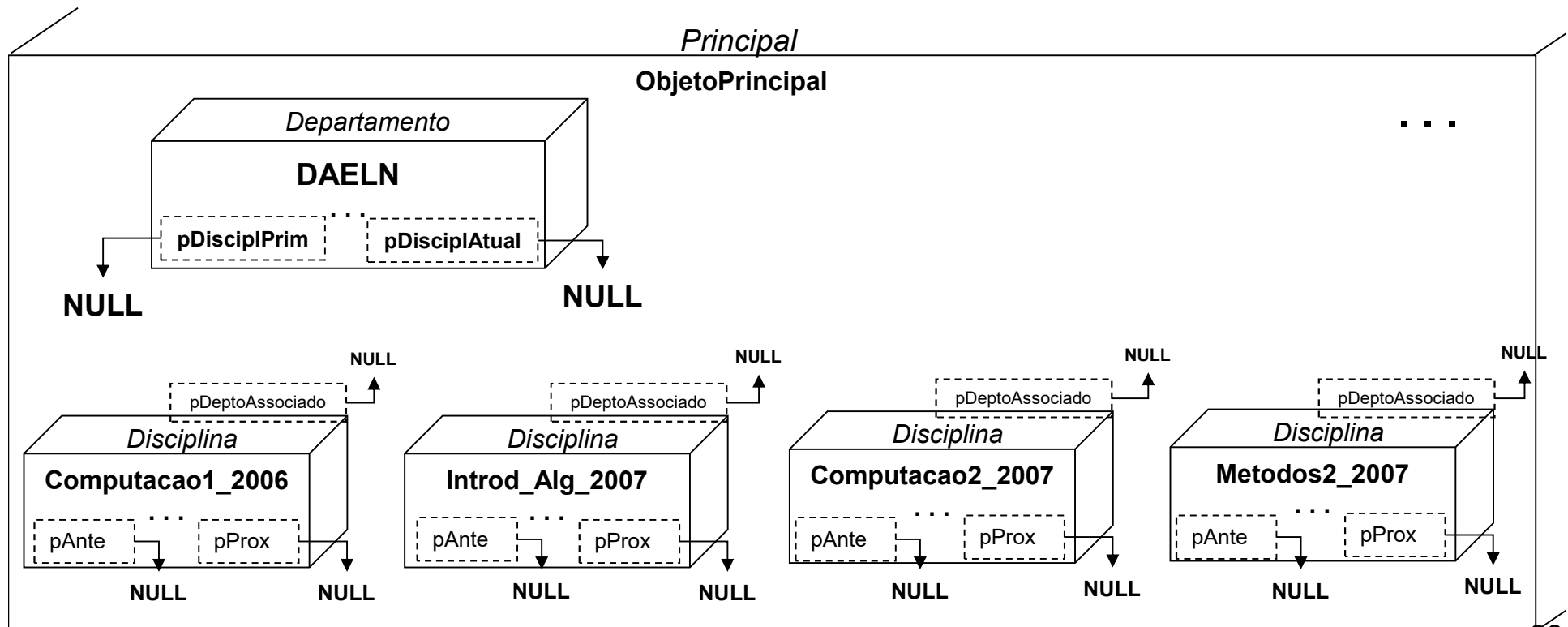
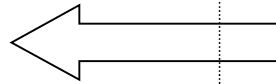
por meio do “diagrama de cubos” da
constituição da lista duplamente encadeada.

```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

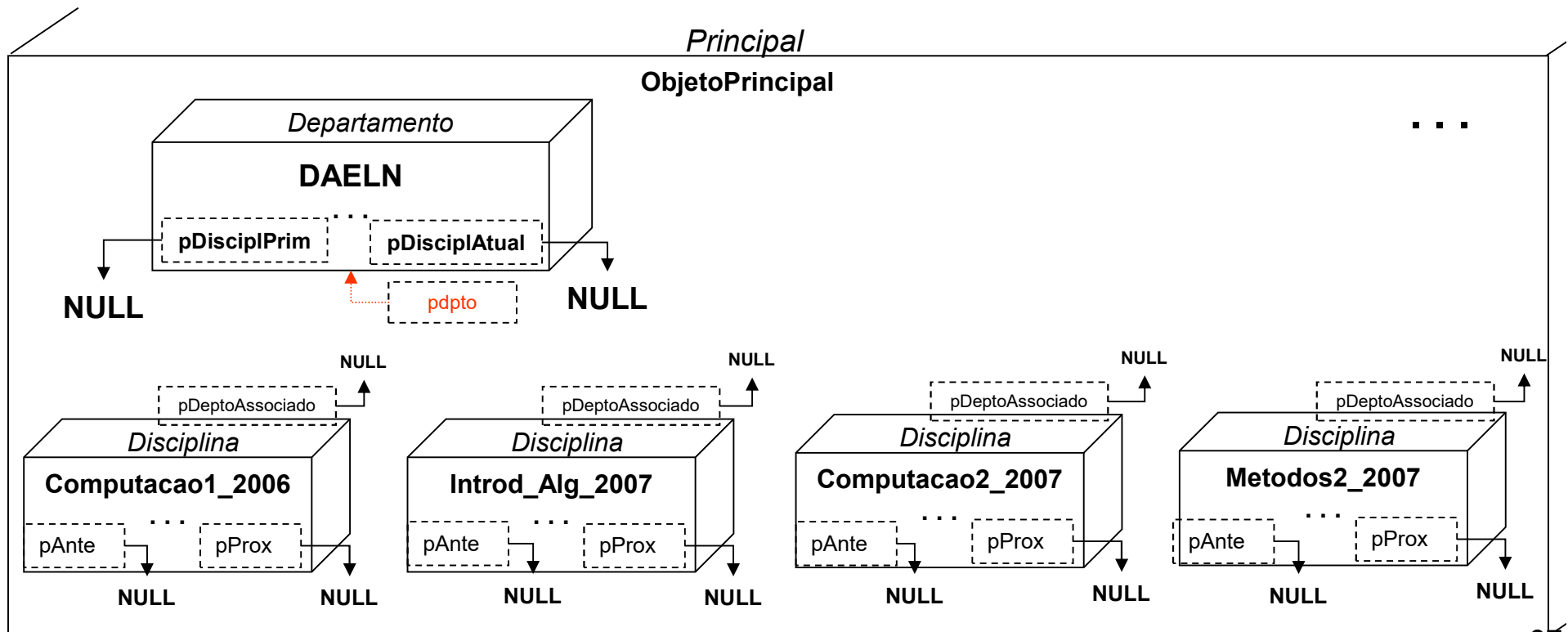
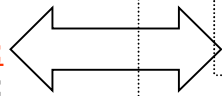
```

```

void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;

  pdpto->incluaDisciplina ( this );
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

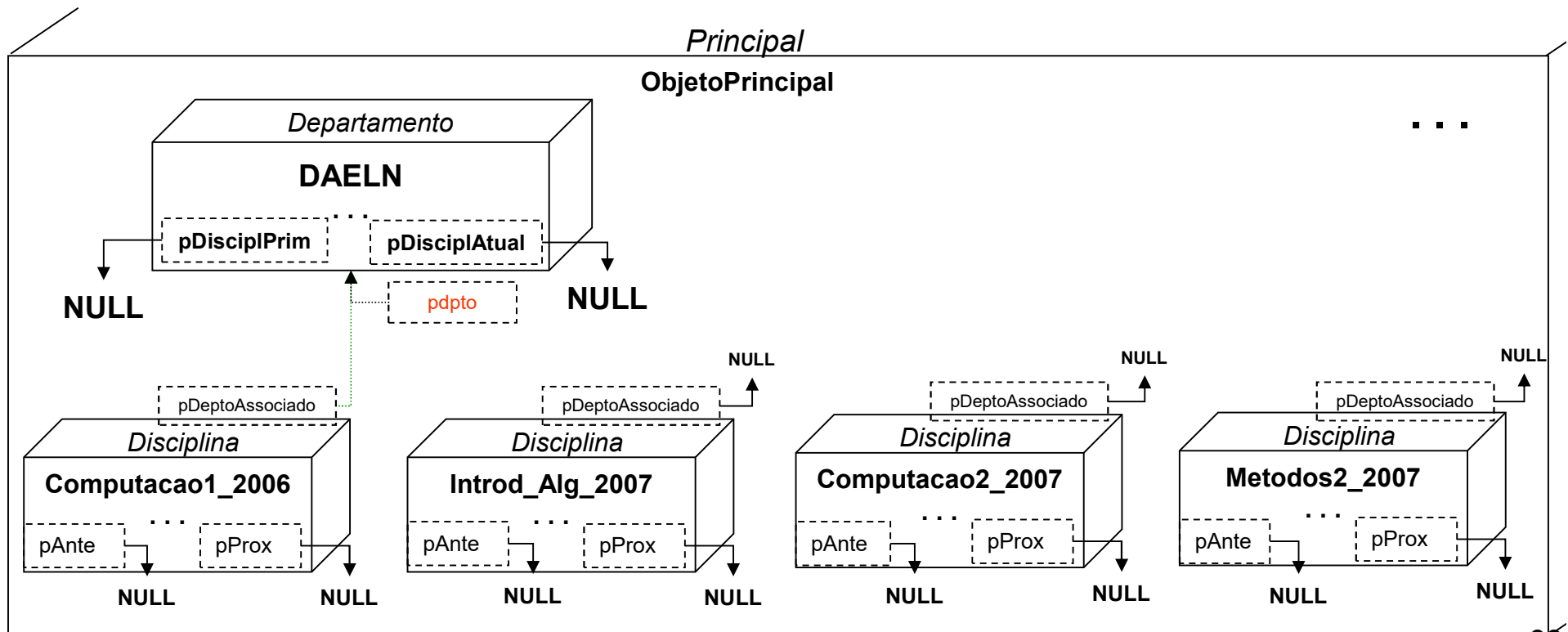
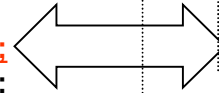
```

```

void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;

  pdpto->incluaDisciplina ( this );
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

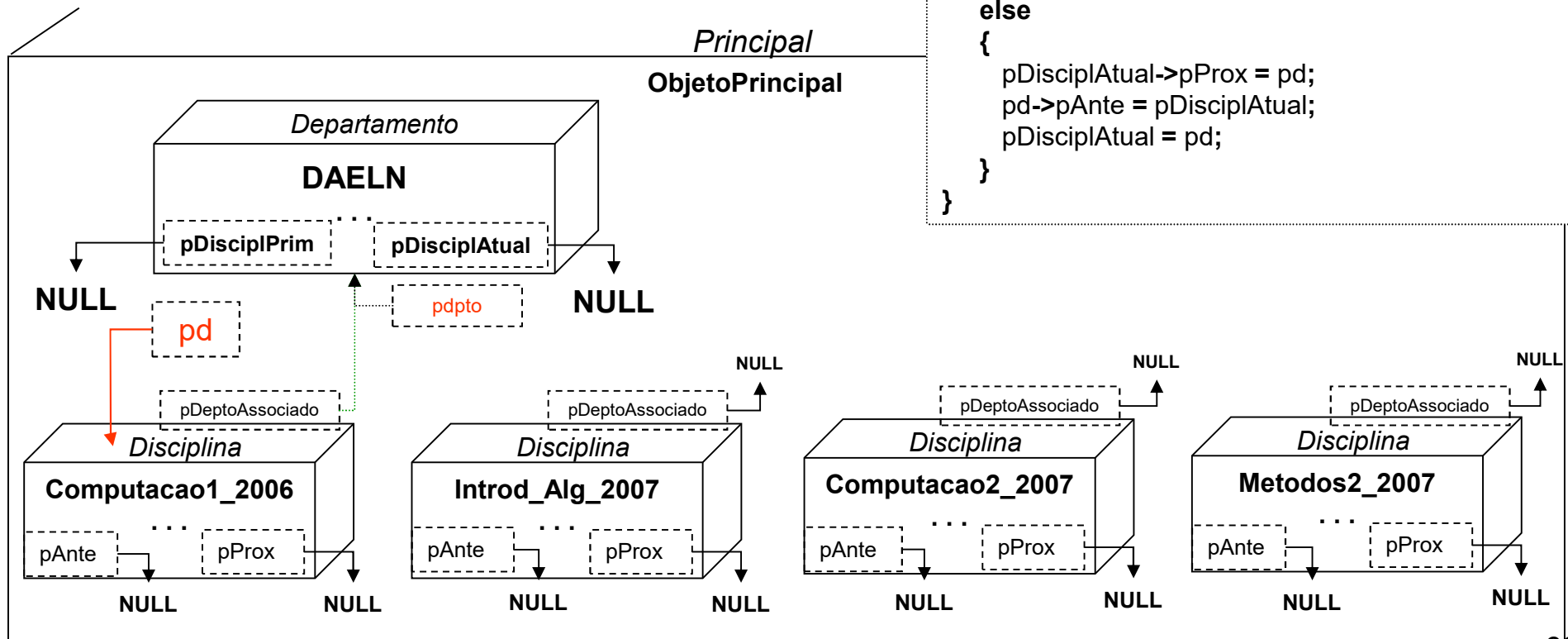
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pd->pAnte = pDisciplAtual;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

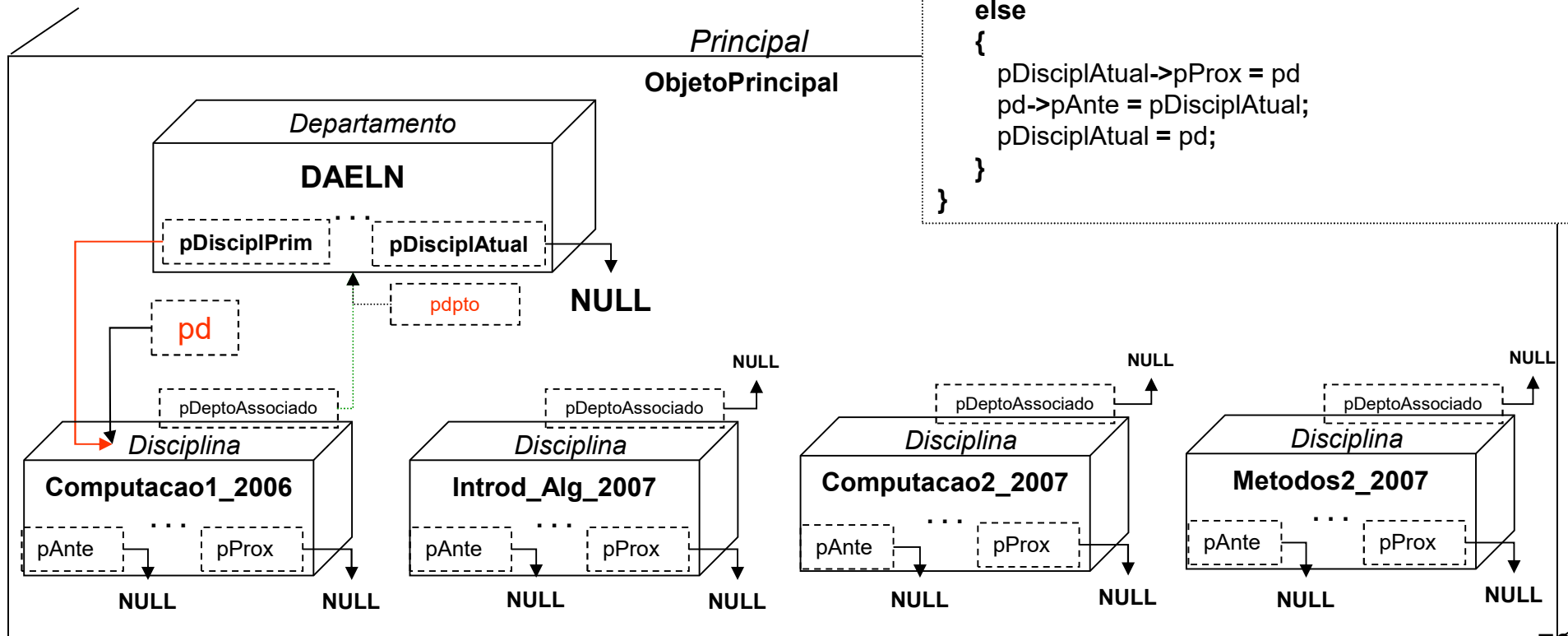
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pd->pAnte = pDisciplAtual;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

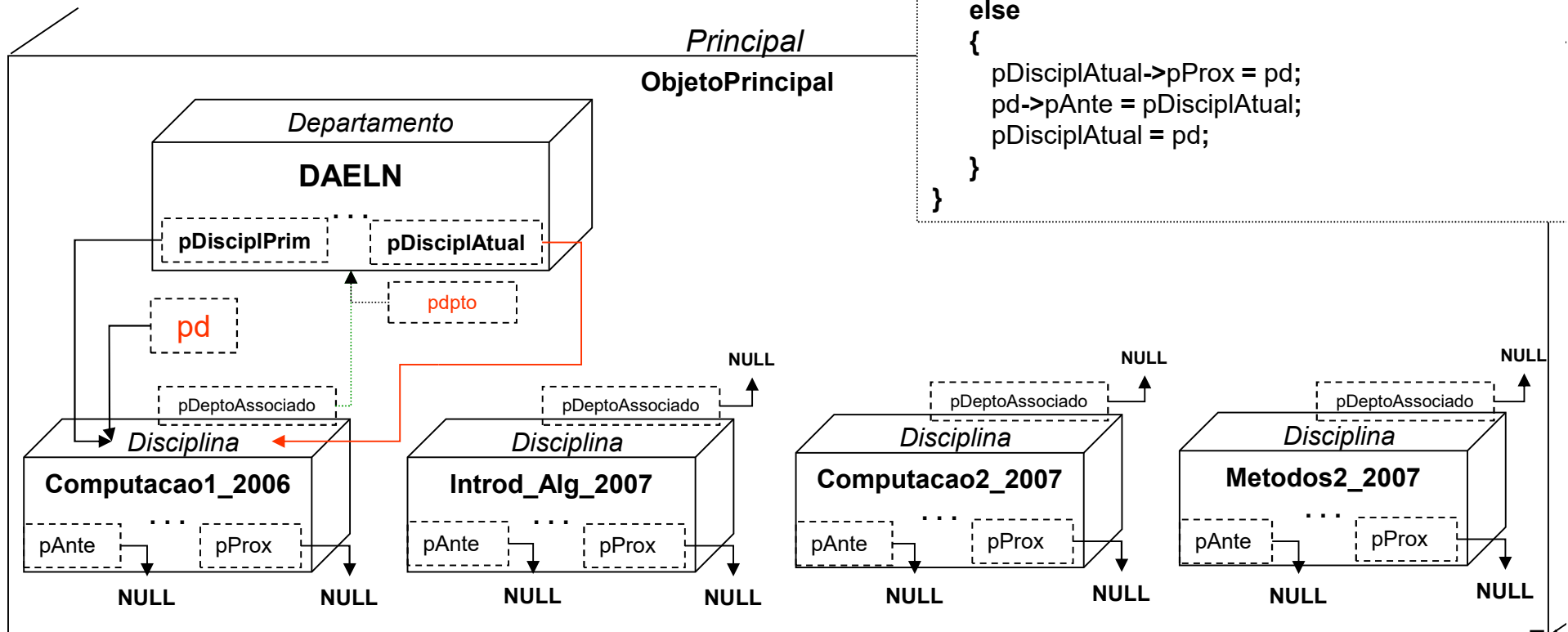
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pd->pAnte = pDisciplAtual;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

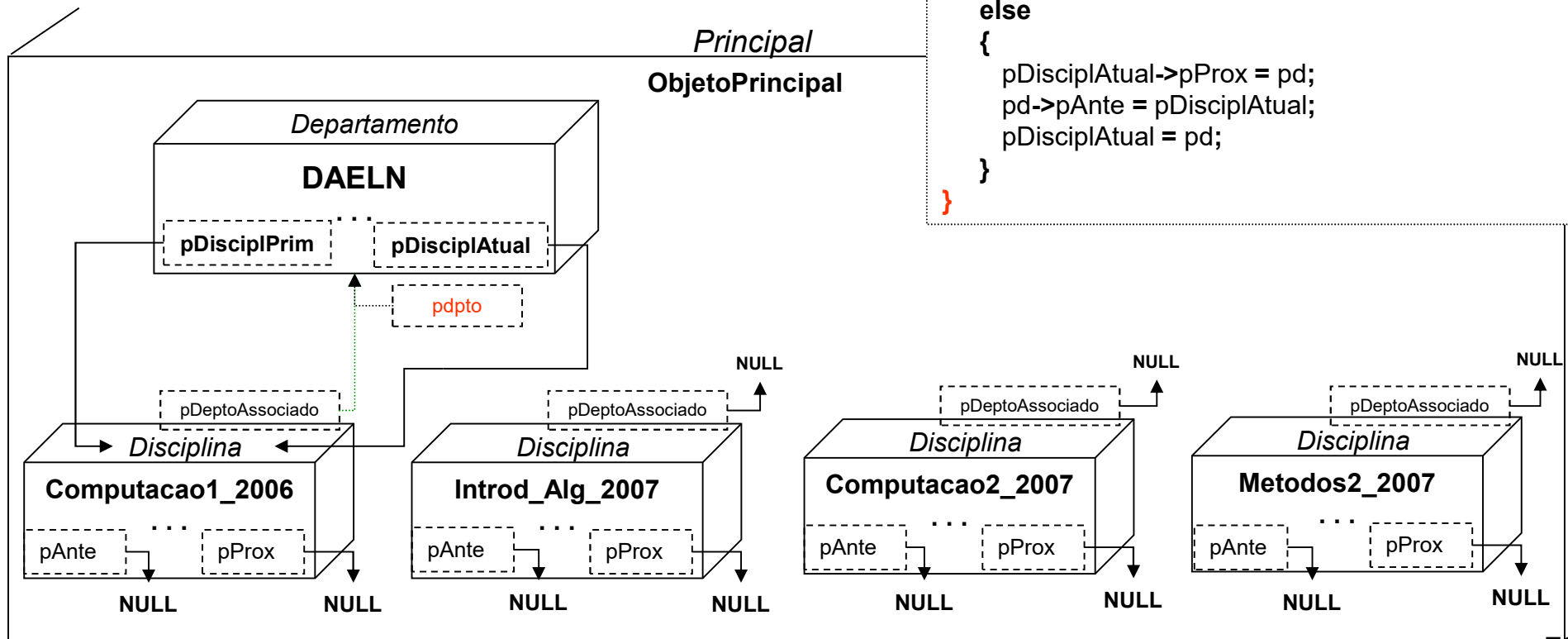
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pd->pAnte = pDisciplAtual;
    pDisciplAtual = pd;
  }
}

```




```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

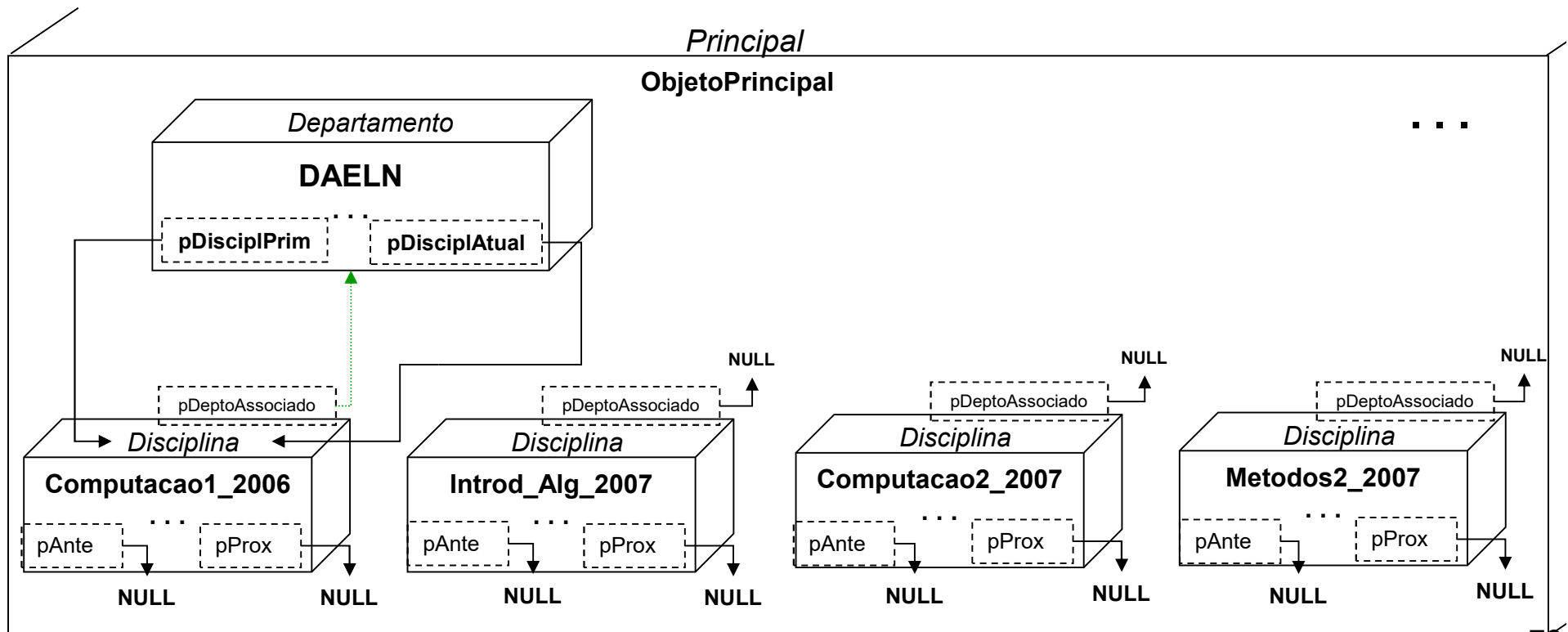
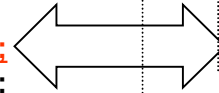
```

```

void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;

  pdpto->incluaDisciplina ( this );
}

```

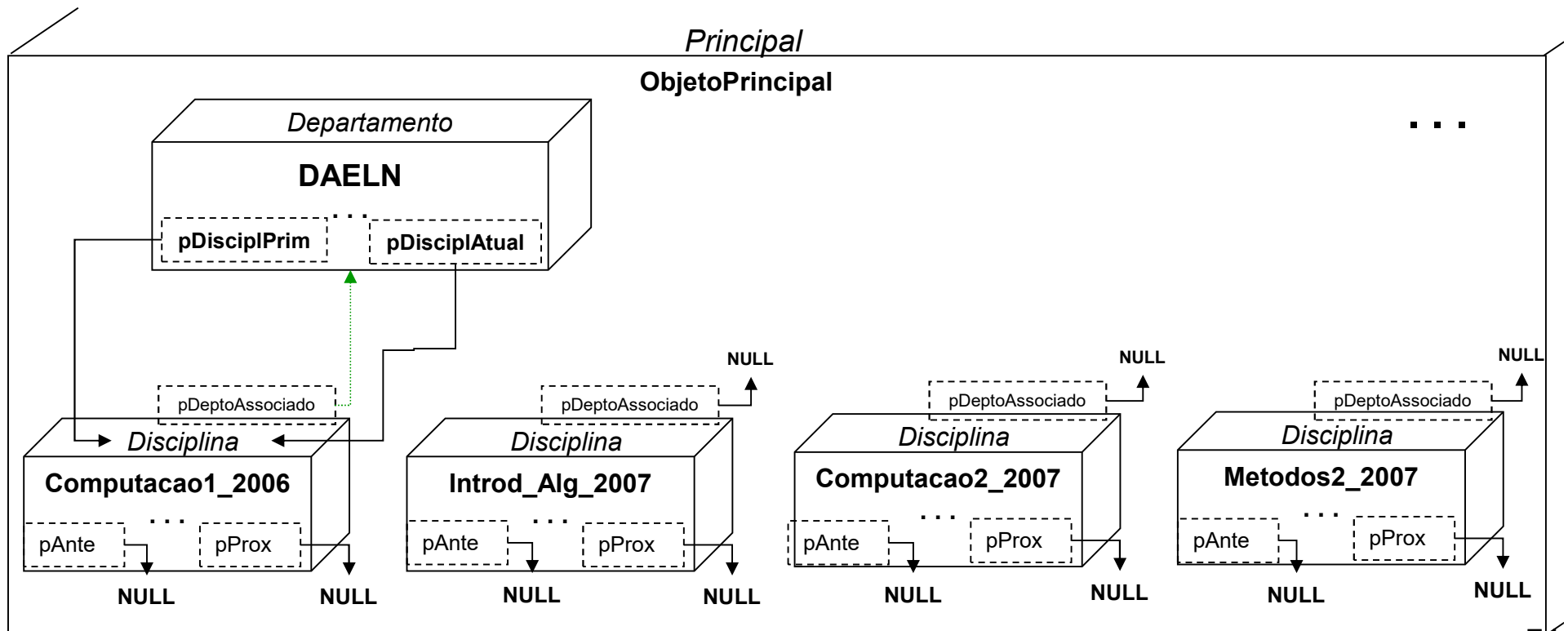
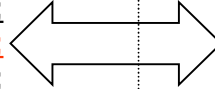


```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

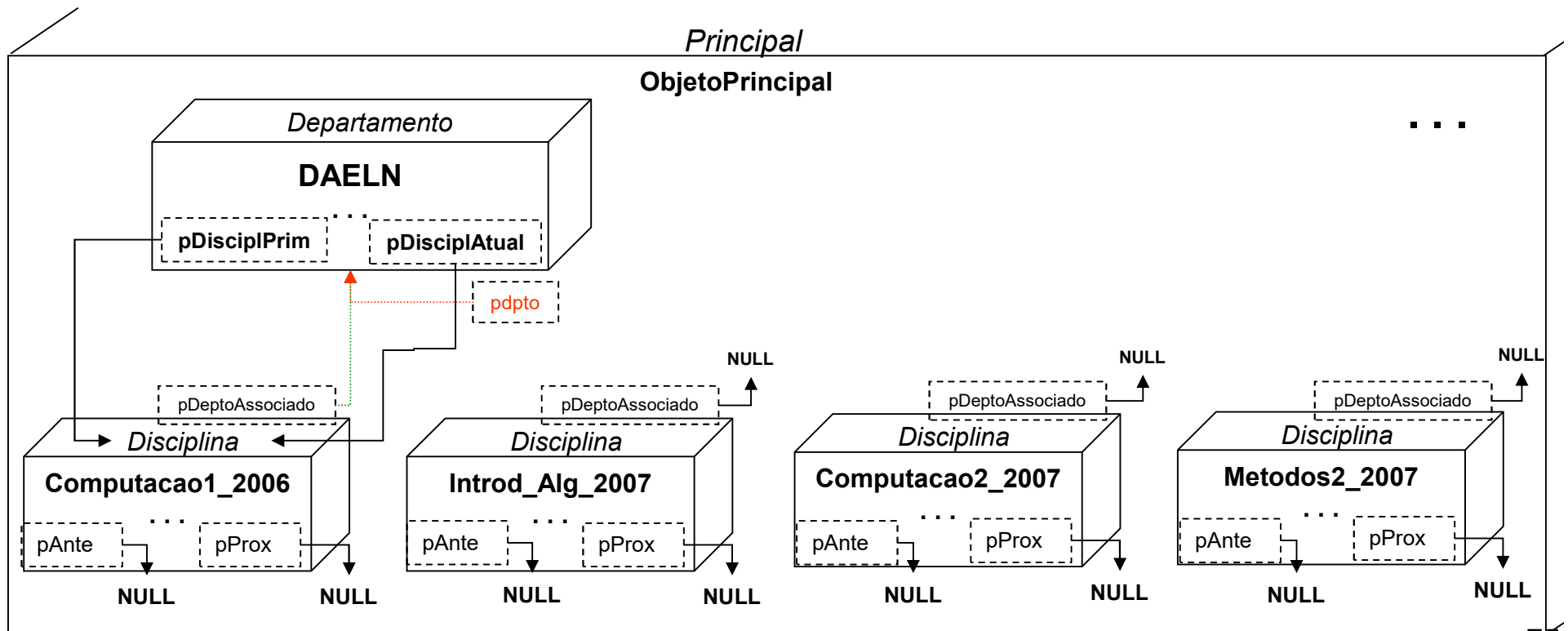
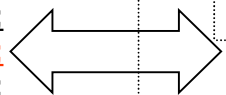
```

```

void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;

  pdpto->incluaDisciplina ( this );
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

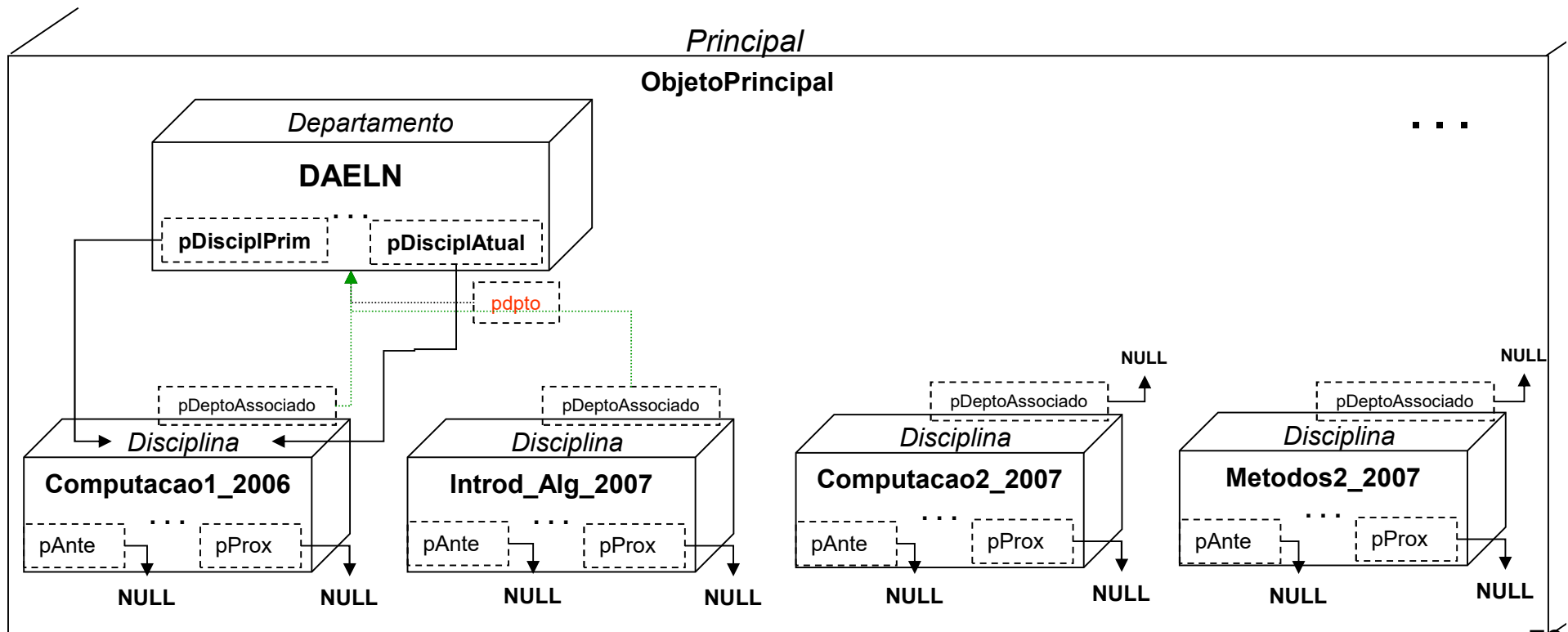
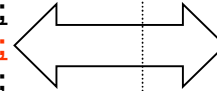
```

```

void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;

  pdpto->incluaDisciplina ( this );
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

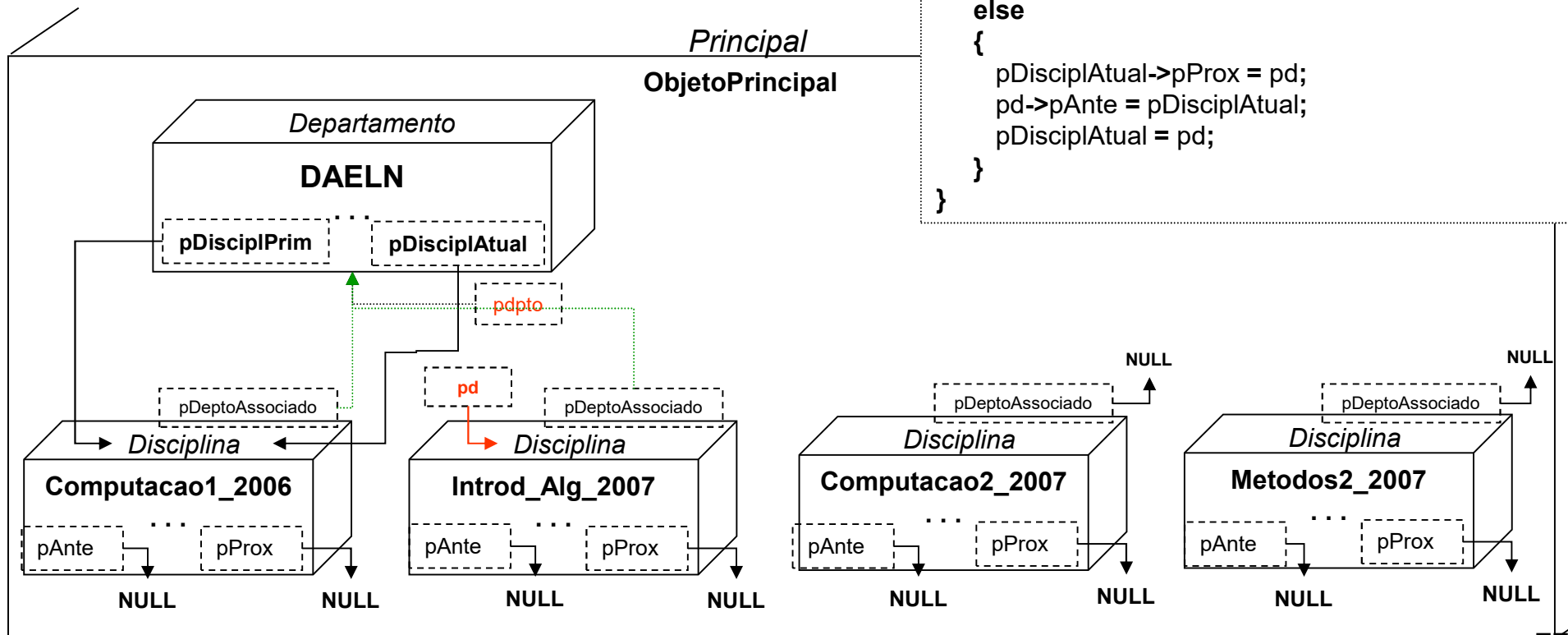
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pd->pAnte = pDisciplAtual;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

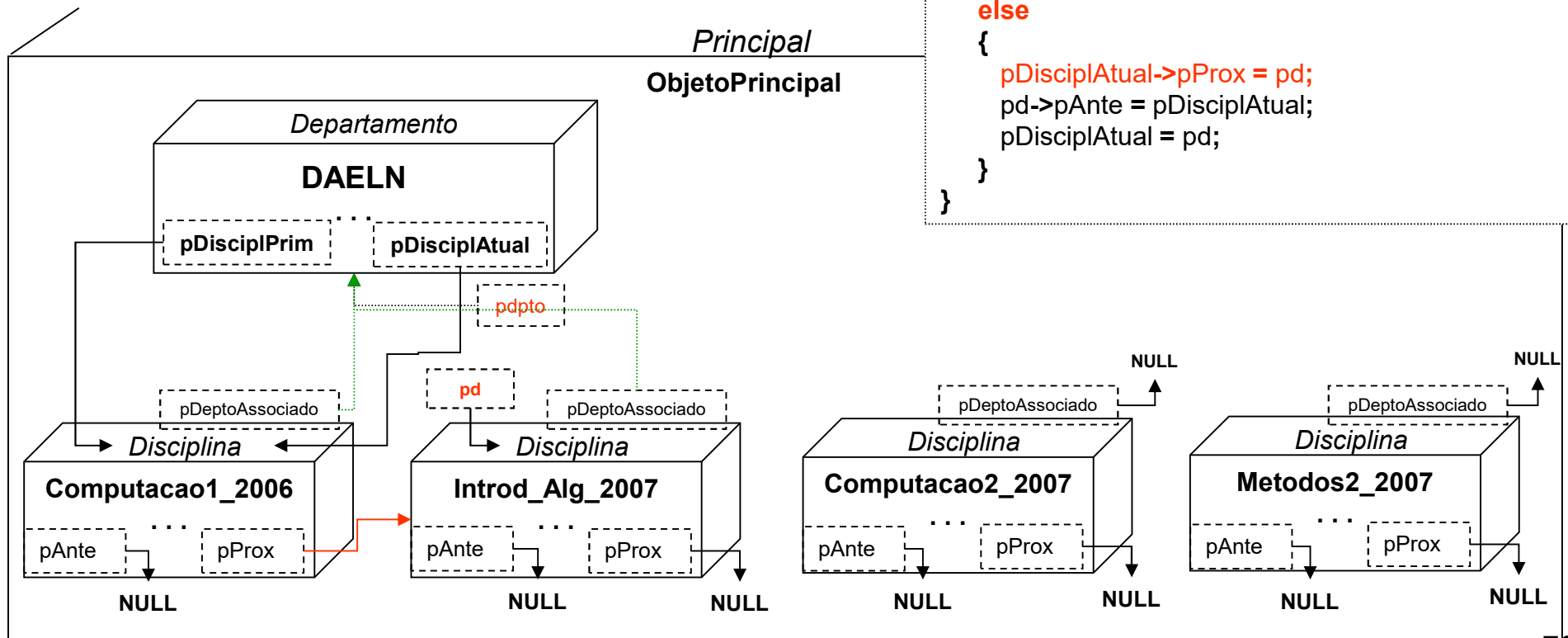
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pd->pAnte = pDisciplAtual;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

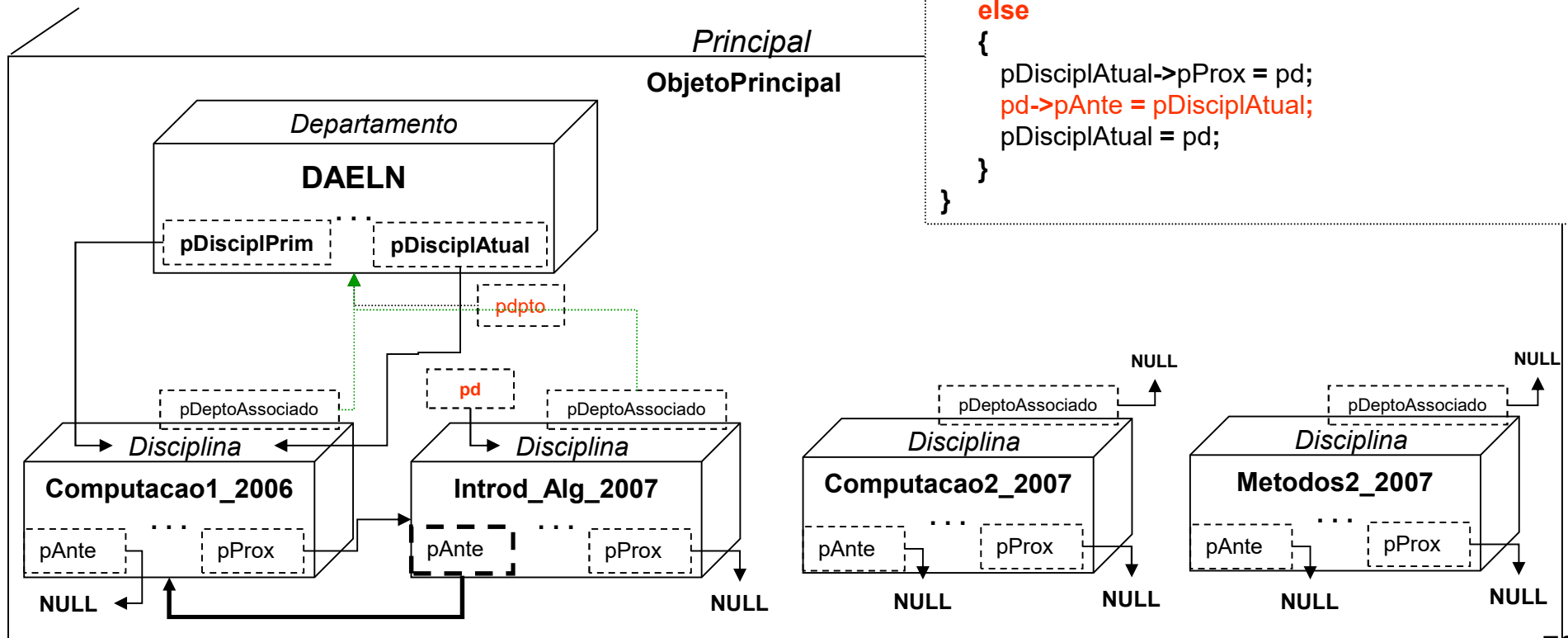
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pd->pAnte = pDisciplAtual;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

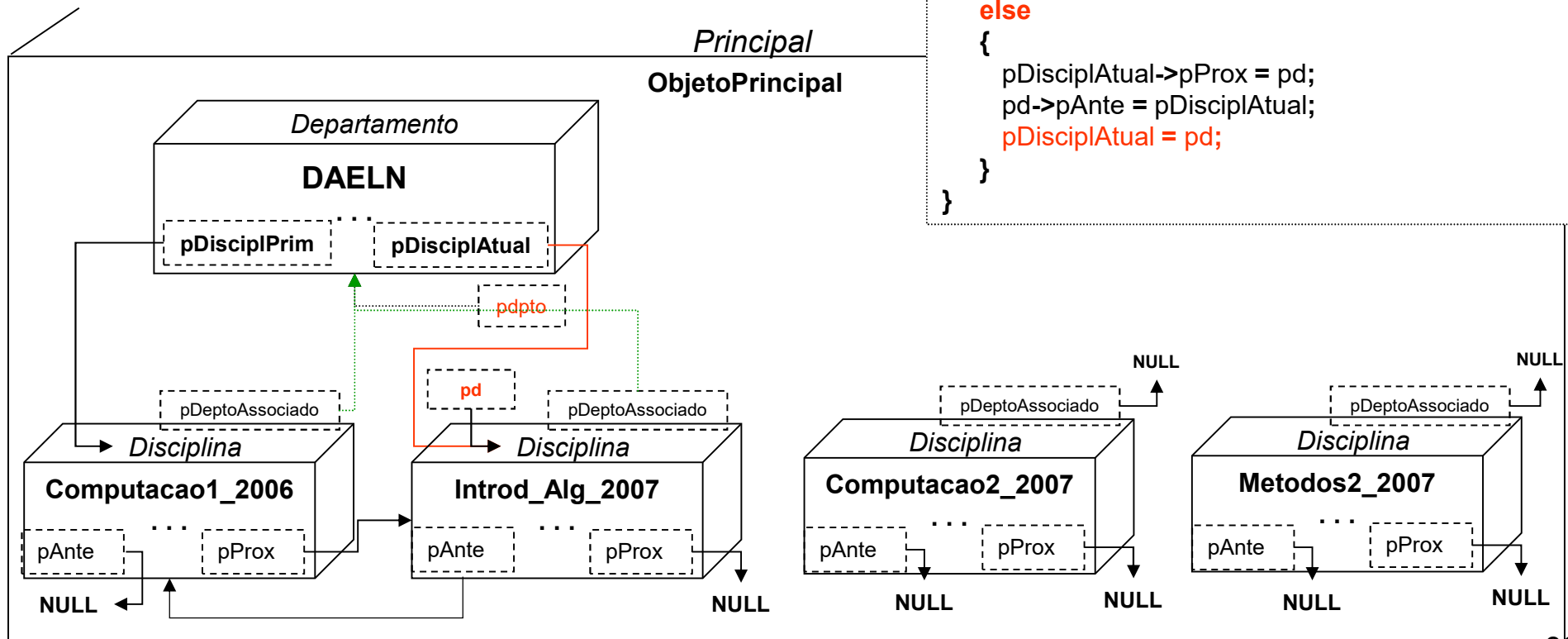
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pd->pAnte = pDisciplAtual;
    pDisciplAtual = pd;
  }
}

```




```

void Principal::InicializaDisciplinas ()
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```

```

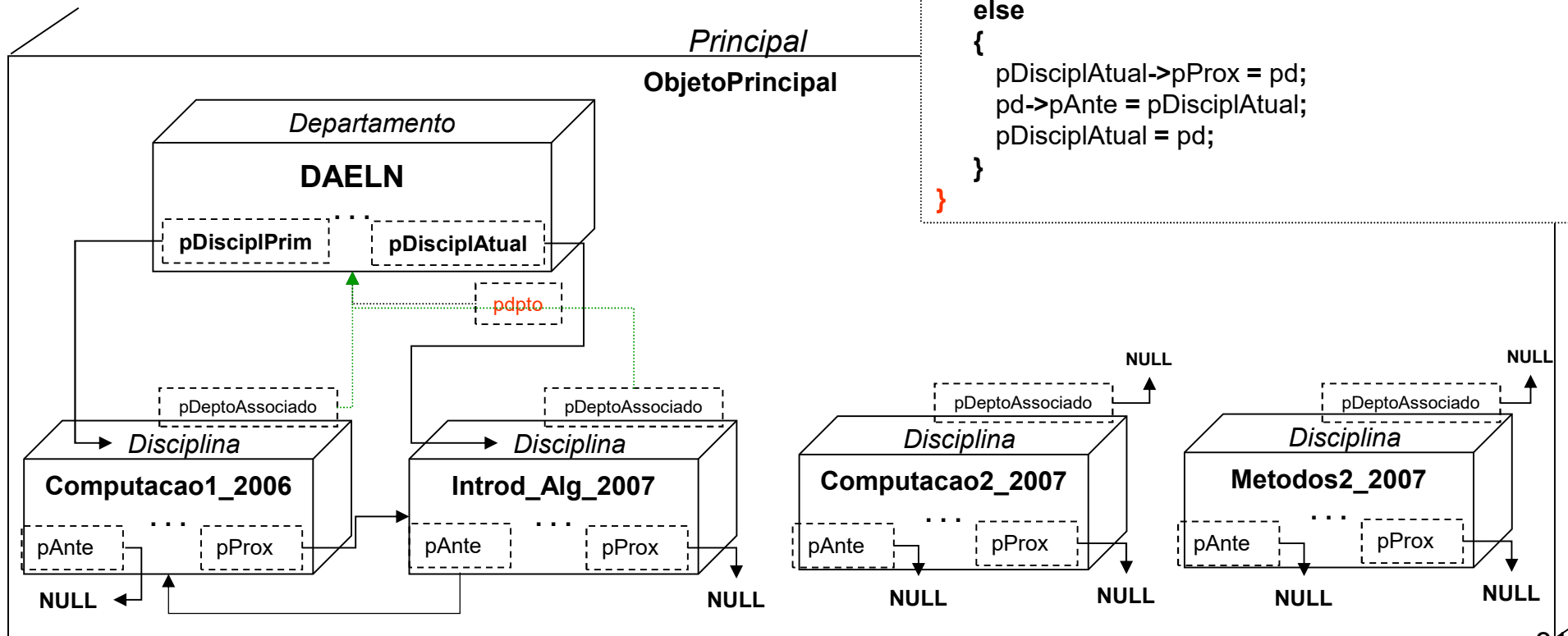
void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;
  pdpto->incluaDisciplina ( this );
}

```

```

void Departamento::incluaDisciplina ( Disciplina* pd )
{
  if ( pDisciplPrim == NULL )
  {
    pDisciplPrim = pd;
    pDisciplAtual = pd;
  }
  else
  {
    pDisciplAtual->pProx = pd;
    pd->pAnte = pDisciplAtual;
    pDisciplAtual = pd;
  }
}

```



```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

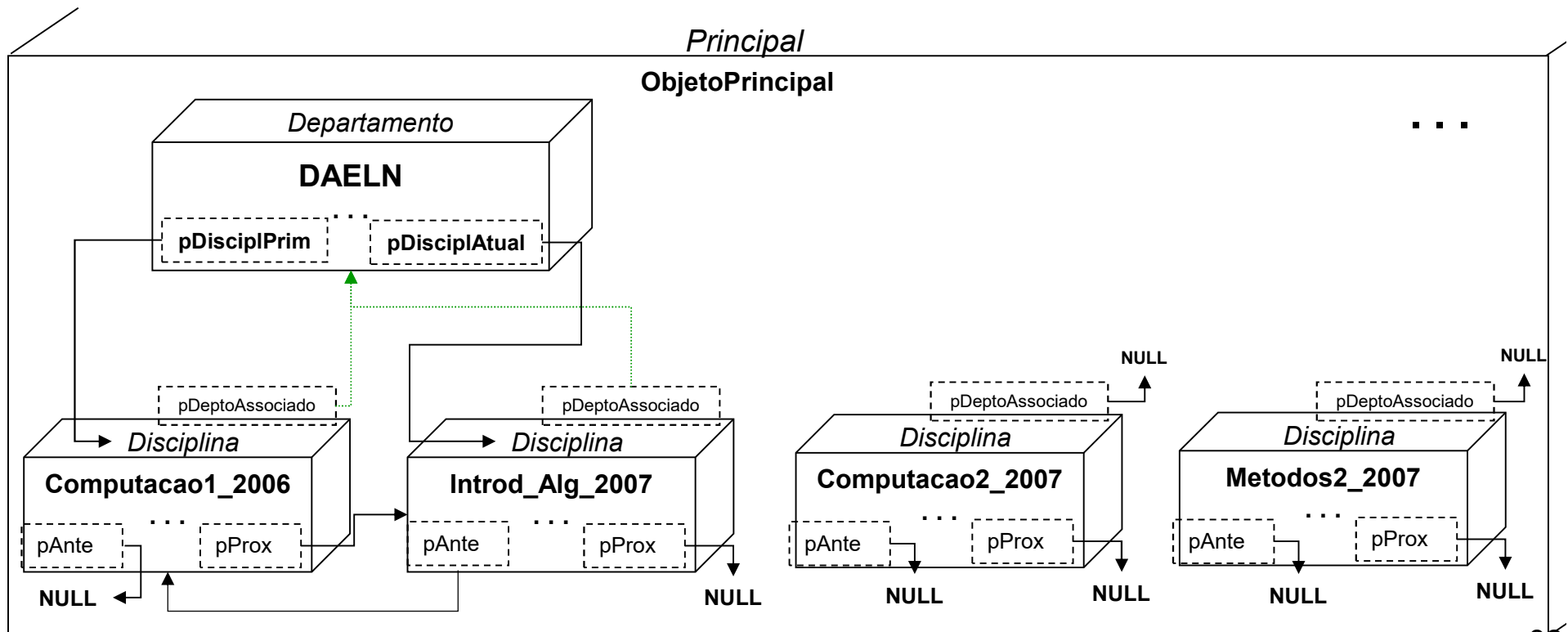
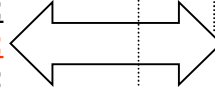
```

```

void Disciplina::setDepartamento ( Departamento* pdpto )
{
  pDeptoAssociado = pdpto;

  pdpto->incluaDisciplina ( this );
}

```

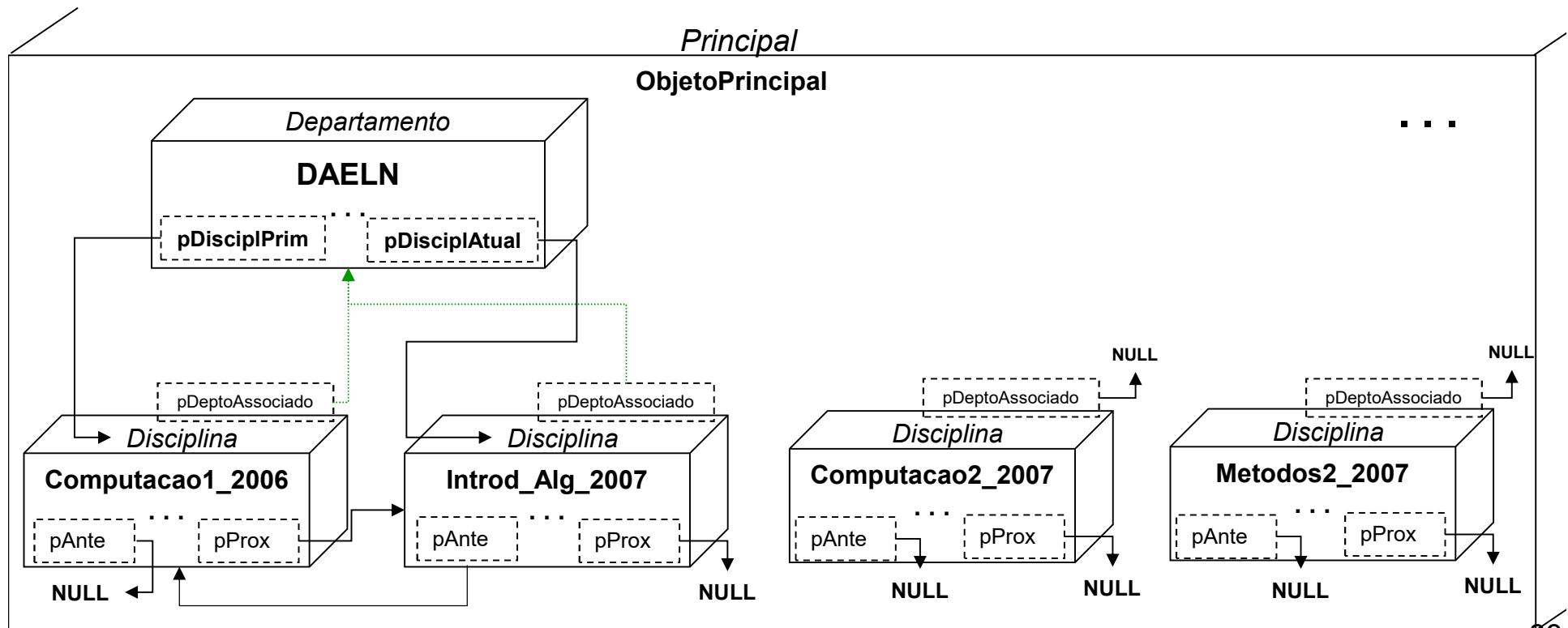
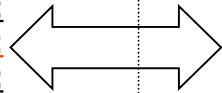


```

void Principal::InicializaDisciplinas ( )
{
  Computacao1_2006.setNome ( "Computacao I 2006" );
  Introd_Alg_2007.setNome ( "Intro de Algde Programacao 2007" );
  Computacao2_2007.setNome ( "Computao II" );
  Metodos2_2007.setNome ( "Métodos II" );

  Computacao1_2006.setDepartamento ( &DAELN );
  Introd_Alg_2007.setDepartamento ( &DAELN );
  Computacao2_2007.setDepartamento ( &DAELN );
  Metodos2_2007.setDepartamento ( &DAELN );
}

```



Continue a “simulação”...

por meio do “diagrama de cubos”

Exercício 2

- Cada Objeto oriundo da Classe *Disciplina* poderá ter um número determinado de objetos *Alunos* relacionados (de uma Classe *Aluno*).
- Este número será determinado no construtor da classe, cujo valor padrão (*default*) será 45.
- As referências (*endereços*) dos objetos *Aluno* serão armazenados em uma lista duplamente encadeada em cada objeto *Disciplina*.
- ~~- Os objetos *Aluno* serão “registrados” em ordem alfabética nos objetos *Disciplina* relacionados.~~
- *Alunos* poderão ser incluídos e excluídos das listas das *Disciplinas*.

Diagrama de Classes – Análise - Associação

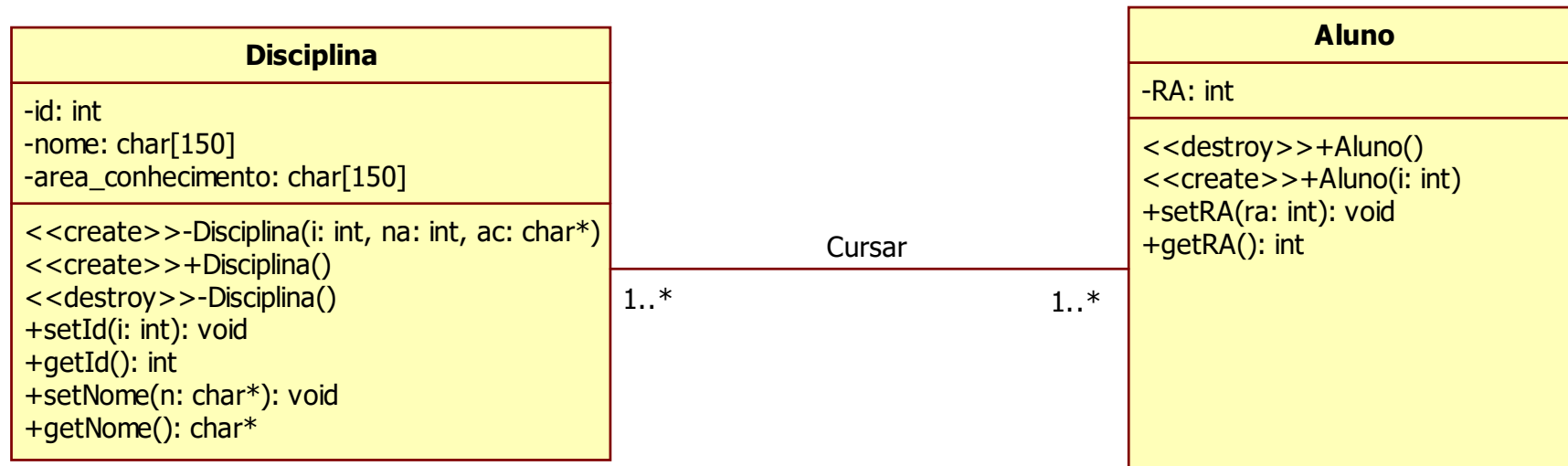


Diagrama de Classes – Análise – Agregação-Simples

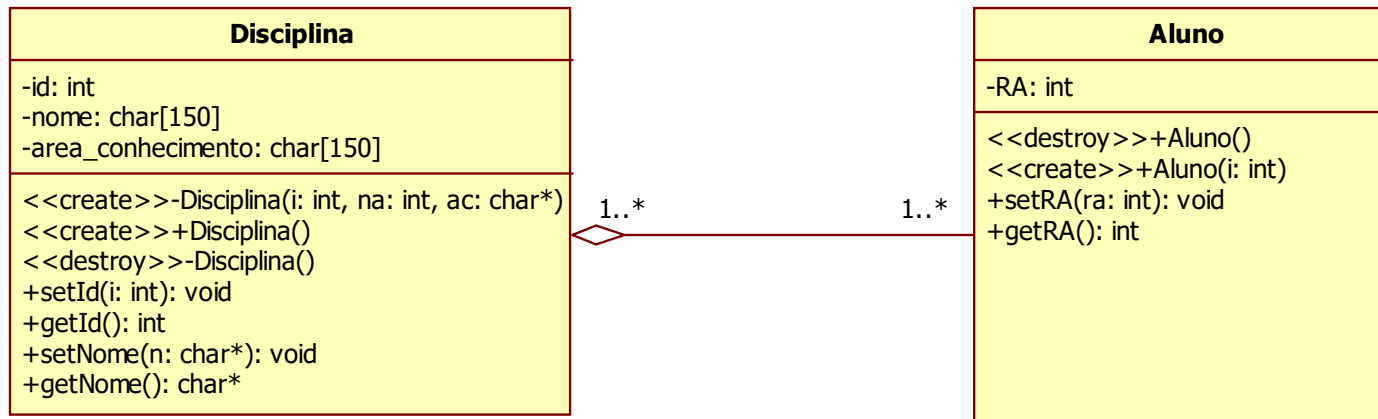
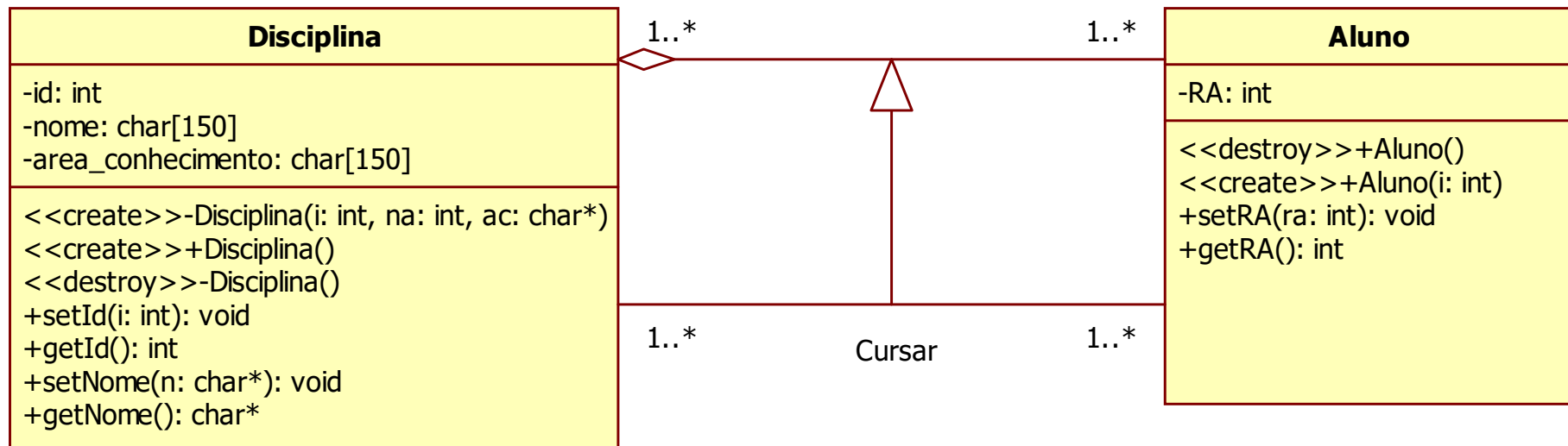
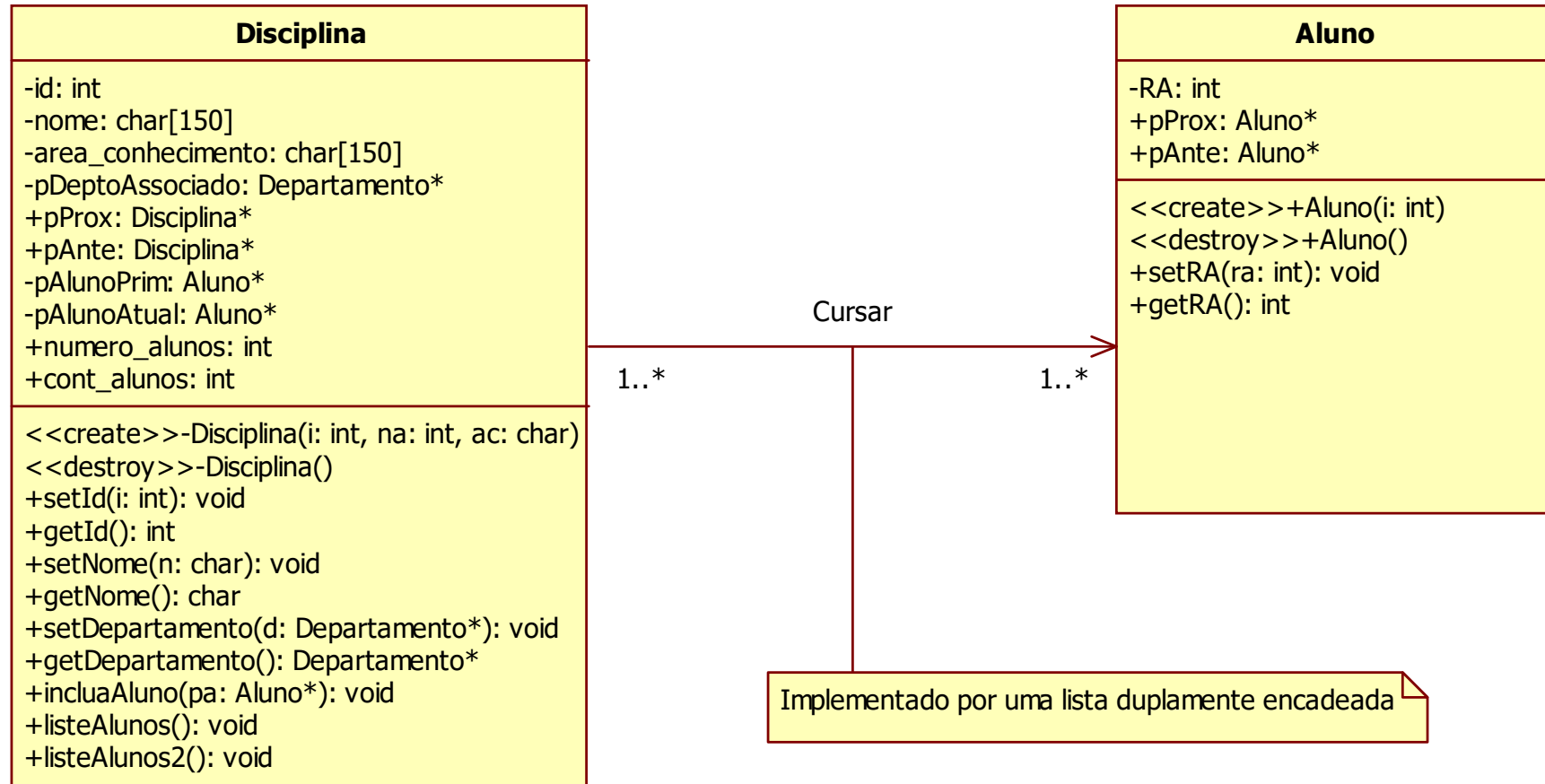


Diagrama de Classes – Análise



A agregação simples se REALIZA por meio de uma associação

Diagrama de Classes – Projeto



'Correção de Exercício'

```
#ifndef _DISCIPLINA_H_
#define _DISCIPLINA_H_
#include "Aluno.h"
#include "Departamento.h"
class Disciplina
{ private:
  int id;
  char nome[150];
  char area_conhecimento[150];
  int numero_alunos;
  int cont_alunos;
  Departamento* pDeptoAssociado;

  Aluno *pAlunoPrim;
  Aluno *pAlunoAtual;
public:
  Disciplina (int na = 45, char* ac = "");
  ~Disciplina ();
  ...
  Disciplina* pProx;
  Disciplina* pAnte;

  void setId ( int n );
  int getId ();

  void setNome ( char* n );
  char* getNome ();

  void setDepartamento ( Departamento* pdpto );
  Departamento* getDepartamento ();

  void incluuaAluno (Aluno* pa );
  void listeAlunos ();
  void listeAlunos2 ();
};
#endif
```

Cada Objeto oriundo da Classe *Disciplina* poderá ter um número determinado de objetos *Alunos* relacionados.

As referências (*endereços*) dos objetos *Aluno* serão armazenados em uma lista duplamente encadeada em cada objeto *Disciplina*.

Este número será determinado no construtor da classe, cujo valor padrão (*default*) será 45.

```
#ifndef _ALUNO_H_
#define _ALUNO_H_
#include "Pessoa.h"

class Departamento;

class Aluno : public Pessoa
{ private:
  int RA;

public:
  Aluno *pProx;
  Aluno *pAnte;

  Aluno ();
  ~Aluno ();

  void setRA ( int ra );
  int getRA ();

};
#endif
```

```

#include "Disciplina.h"
#include <stdio.h>
#include <string.h>

Disciplina::Disciplina ( int na, char* ac )
{
    pDeptoAssociado = NULL;
    pAlunoPrim      = NULL;
    pAlunoAtual     = NULL;
    pProx           = NULL;
    pAnte           = NULL;
    cont_alunos     = 0;
    numero_alunos   = na;
    strcpy (area_conhecimento, ac );
}

Disciplina::~Disciplina ( )
{
    pDeptoAssociado = NULL;
    pAlunoPrim      = NULL;
    pAlunoAtual     = NULL;
    pProx           = NULL;
    pAnte           = NULL;
}

void Disciplina::setId ( int n )           { id = n; }
int Disciplina::getId ( )                 { return id; }
void Disciplina::setNome ( char* n )      { strcpy (nome, n); }
char* Disciplina::getNome ( )             { return nome; }

void Disciplina::setDepartamento ( Departamento* pdpto )
{
    // Cada vez que um Departamento é associado a uma Disciplina,
    // esta Disciplina passa a fazer parte da lista de disciplina
    // do Departamento, por meio do comando abaixo.
    pDeptoAssociado = pdpto;
    pd->setDisciplina ( this );
}

Departamento* Disciplina::getDepartamento ( )
{
    return pDeptoAssociado;
}

```

```

void Disciplina::incluaAluno ( Aluno* pa )
{
    if ( ( cont_alunos < numero_alunos ) && ( pa != NULL ) )
    {
        if ( pAlunoPrim == NULL )
        {
            pAlunoPrim      = pa;
            pAlunoAtual     = pa;
        }
        else
        {
            pAlunoAtual->pProx = pa;
            pa->pAnte          = pAlunoAtual;
            pAlunoAtual     = pa;
        }
        cont_alunos++;
    }
    else
    {
        printf ( "Aluno não incluído. Turma já lotada \n" );
    }
}

void Disciplina::listeAlunos ( )
{
    Aluno* paux;
    paux = pAlunoPrim;
    while ( paux != NULL )
    {
        printf ( "Aluno %s matriculado em %s \n", paux->getNome(), nome );
        paux = paux->pProx;
    }
}

void Disciplina::listeAlunos2 ( )
{
    Aluno* paux;
    paux = pAlunoAtual;
    while ( paux != NULL )
    {
        printf ( "Aluno %s matriculado em %s \n", paux->getNome( ), nome );
        paux = paux->pAnte;
    }
}

```

```

void Disciplina::incluaAluno( Aluno* pa )
{
    if ( ( cont_alunos < numero_alunos ) && ( pa != NULL ) )
    {
        if ( pAlunoPrim == NULL )
        {
            pAlunoPrim      = pa;
            pAlunoAtual     = pa;
        }
        else
        {
            pAlunoAtual->pProx = pa;
            pa->pAnte          = pAlunoAtual;
            pAlunoAtual       = pa;
        }
        cont_alunos++;
    }
    else
    {
        printf ( "Aluno não incluído. Turma já lotada \n" );
    }
}

```

Este slide apenas destaca/salienta os métodos *setAluno* e *listeAlunos* da classe *Disciplina*.

```

void Disciplina::listeAlunos ( )
{
    Aluno* paux;
    paux = pAlunoPrim;
    while ( paux != NULL )
    {
        printf ( "Aluno %s matriculado em %s \n", paux->getNome ( ), nome);
        paux = paux->pProx;
    }
}

void Disciplina::listeAlunos2 ( )
{
    Aluno* paux;
    paux = pAlunoAtual;
    while ( paux != NULL )
    {
        printf ("Aluno %s matriculado em %s \n", paux->getNome(), nome);
        paux = paux->pAnte;
    }
}

```

```

#ifndef _PRINCIPAL_H_
#define _PRINCIPAL_H_
#include "Professor.h"
#include "Universidade.h"
#include "Aluno.h"
#include "Disciplina.h"

class Principal
{
private:
    // Universidades
    Universidade UTFPR;
    Universidade Princeton;
    Universidade Cambridge;

    // Departamentos
    Departamento DAELN;
    Departamento MatematicaUTFPR;
    Departamento FisicaUTFPR;

    Departamento MatematicaPrinceton;
    Departamento FisicaPrinceton;

    Departamento MatematicaCambridge;
    Departamento FisicaCambridge;
    ...
    // Professores
    Professor Simao;
    Professor Einstein;
    Professor Newton;

    // Disciplinas
    Disciplina Computacao1_2006;
    Disciplina Introd_Alg_2007;
    Disciplina Computacao2_2007;
    Disciplina Metodos2_2007;

```

```

// Alunos
Aluno AAA;
Aluno BBB;
Aluno CCC;
Aluno DDD;
Aluno EEE;

int diaAtual;
int mesAtual;
int anoAtual;

public:

    Principal ( );

    // Inicializações...
    void Inicializa ( );
    void InicializaUnivesidades ( );
    void InicializaDepartamentos ( );
    void InicializaProfessores ( );
    void InicializaAlunos ( );
    void InicializaDisciplinas ( );

    void Executar ( );

    void CalcldadeProfs ( );
    void UnivOndeProfsTrabalham ( );
    void DepOndeProfsTrabalham ( );
    void ConhecPessoa ( );
    void ListeDiscDeptos ( );
    void ListaAlunoDisc ( );

};

#endif

```

'Correção de Exercício'

```
void Principal::InicializaAlunos ( )  
{  
    ...  
}
```

```
void Principal::InicializaDisciplinas ( )  
{  
    Computacao1_2006.setNome ( "Computacao I 2006" );  
    Introd_Alg_2007.setNome ( "Introducao de Algoritmos de Programacao 2007" );  
    Computacao2_2007.setNome ( "Computao II" );  
    Metodos2_2007.setNome ( "Métodos II" );  
  
    Computacao1_2006.setDepartamento ( &DAELN );  
    Introd_Alg_2007.setDepartamento ( &DAELN );  
    Computacao2_2007.setDepartamento ( &DAELN );  
    Metodos2_2007.setDepartamento ( &DAELN );  
  
    Metodos2_2007.incluaAluno ( &AAA );  
    Metodos2_2007.incluaAluno ( &BBB );  
    Metodos2_2007.incluaAluno ( &CCC );  
    Metodos2_2007.incluaAluno ( &DDD );  
    Metodos2_2007.incluaAluno ( &EEE );  
}
```

```
void Principal::ListeAlunosDisc ( )  
{  
    Metodos2_2007.listeAlunos ( );  
    printf ( "\n" );  
  
    Metodos2_2007.listeAlunos2 ( );  
    printf ( "\n" );  
}
```

```
void Principal::Executar ( )  
{  
    CalcIdadeProfs ( );  
    UnivOndeProfsTrabalham ( );  
    DepOndeProfsTrabalham ( );  
    ConhecPessoa ( );  
    ListeDiscDeptos ( );  
    ListeAlunosDisc ( );  
}
```

```

void Principal::InicializaDisciplinas ( )
{
    Computacao1_2006.setNome ( "Computacao I 2006" );
    Introd_Alg_2007.setNome ( "Introducao de Algoritmos de Programacao 2007" );
    Computacao2_2007.setNome ( "Computao II" );
    Metodos2_2007.setNome ( "Métodos II" );

    Computacao1_2006.setDepartamento ( & DAELN);
    Introd_Alg_2007.setDepartamento ( & DAELN);

    Computacao2_2007.setDepartamento ( & DAELN);
    Metodos2_2007.setDepartamento ( & DAELN);

    Metodos2_2007.incluaAluno ( &AAA );
    Metodos2_2007.incluaAluno ( &BBB );
    Metodos2_2007.incluaAluno ( &CCC );
    Metodos2_2007.incluaAluno ( &DDD );
    Metodos2_2007.incluaAluno ( &EEE );

    Computacao2_2007.incluaAluno ( &AAA );
    Computacao2_2007.incluaAluno ( &EEE );
    Computacao2_2007.incluaAluno ( &FFF );
}

```

PROBLEMA!

Qual é o problema ?
Como resolver?

Faça uma “simulação”...

por meio do “diagrama de cubos”
para descobrir o problema