Programação Orientada a Objetos

Prof. Delano M. Beder

Roteiro 14 – Classes Abstratas

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

2. Implementação da abstração Funcionário

Nova classe C++ denominada Funcionario

Dois arquivos: Funcionario.h (Cabeçalho) e Funcionario.cpp (Código-fonte)

2.1 Arquivo Funcionario.h

Note que Funcionario é uma classe abstrata pois contem um método abstrato – getProfissao() - que precisa ser obrigatoriamente implementada pelas subclasses

```
#ifndef FUNCIONARIO_H
#define FUNCIONARIO_H

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

class Funcionario {
public:
  Funcionario(int CPF, string nome);
  virtual ~Funcionario();
  virtual void imprime() const;
  virtual string getProfissao() const = 0; // método abstrato
private:
  int CPF;
  string nome;
};

#endif /* FUNCIONARIO_H */
```

2.2 Arquivo Funcionario.cpp

```
#include "Funcionario.h"

Funcionario::Funcionario(int CPF, string nome) :
CPF(CPF), nome(nome) {
}

Funcionario::~Funcionario() {
}

void Funcionario::imprime() const {
  cout << "Nome : " << nome << endl;
  cout << "CPF : " << CPF << endl;
  cout << "Profissão : " << getProfissao() << endl;
}
```

3. Implementação da abstração Médico

Nova classe C++ denominada Medico

Dois arquivos: Medico.h (Cabeçalho) e Medico.cpp (Código-fonte)

3.1 Arquivo Medico.h

Note que Medico é uma subclasse da classe abstrata Funcionario. Essa classe deve obrigatoriamente implementar o método getProfissao()

```
#ifndef MEDICO_H
#define MEDICO_H

#include "Funcionario.h"

class Medico : public Funcionario {
public:
  Medico(int CPF, string nome, string especialidade);
  virtual ~Medico();
  virtual void imprime() const;
  virtual string getProfissao() const;
private:
  string especialidade;
};

#endif /* MEDICO_H */
```

3.2 Arquivo Medico.cpp

```
#include "Medico.h"

Medico::Medico(int CPF, string nome, string especialidade) :
Funcionario(CPF, nome), especialidade(especialidade) {
}

Medico::~Medico() {
}

void Medico::imprime() const {
  Funcionario::imprime();
  cout << "Especialidade : " << especialidade << endl;
}

string Medico::getProfissao() const {
  return "Médico";
}
```

4. Implementação da abstração Enfermeiro

Nova classe C++ denominada Enfermeiro

Dois arquivos: Enfermeiro.h (Cabeçalho) e Enfermeiro.cpp (Código-fonte)

4.1 Arquivo Enfermeiro.h

Note que Enfermeiro é uma subclasse da classe abstrata Funcionario. Essa classe deve obrigatoriamente implementar o método getProfissao()

```
#ifndef ENFERMEIRO_H
#define ENFERMEIRO_H

#include "Funcionario.h"

class Enfermeiro : public Funcionario{
public:
  Enfermeiro(int CPF, string nome, int CRE);
  virtual ~Enfermeiro();
  virtual void imprime() const;
  virtual string getProfissao() const;
private:
  int CRE; // Número de registro no Conselho Regional de Enfermagem
};

#endif /* ENFERMEIRO_H */
```

4.2 Arquivo Enfermeiro.cpp

```
#include "Enfermeiro.h"

Enfermeiro::Enfermeiro(int CPF, string nome, int CRE) :
Funcionario(CPF, nome), CRE(CRE) {
}

Enfermeiro::~Enfermeiro() {
}

void Enfermeiro::imprime() const {
  Funcionario::imprime();
  cout << "CRE : " << CRE << endl;
}

string Enfermeiro::getProfissao() const {
  return "Enfermeiro";
}
```

5. Arquivo main.cpp

```
#include "Medico.h"
#include "Enfermeiro.h"

int main(int argc, char** argv) {

  Medico m(12345, "Dr. House", "Neurologista");
  m.imprime();

  cout << endl;

  Enfermeiro e(67890, "Nurse Beth", 12345);
  e.imprime();


  return 0;
}
```

6. Compile e execute (verifique a saída impressa)

7. Fim