ECOP13A-Lab8

Guia de Laboratório Prof. André Bernardi andrebernardi@unifei.edu.br





8º Laboratório ECOP13A 06 de junho 2025



1^a Questão



Crie um programa principal para testar a hierarquia Politico/Presidente/ Governador/Prefeito, da tarefa 2 do laboratório 7 com as seguintes características.

- Utilize um vetor de ponteiros para a classe base de modo a representar os diversos elementos a serem criados.
- Permita que o usuário escolha qual tipo de objeto será criado através de um menu de opções do tipo: 1-Presidente, 2-Governador, 3-Prefeito.
- Utilize um loop para imprimir todos os objetos criados





Alterar a hierarquia criada no ex2 do laboratório 7 de modo a permitir o uso da propriedade do polimorfismo.

Modifique o tipo da função Imprime() para que ela seja virtual e execute novamente o programa para analisar o resultado.

```
#ifndef POLITICO H
                               1<sup>a</sup> e 2<sup>a</sup> questão
#define POLITICO H
                                Exemplo de Solução
#include <string>
#include <iostream>
using namespace std;
// função de apoio para as classes
int menu();
class Politico{
    protected:
        string nome, partido, numero;
    public:
          Politico(string n="", string p="", string nu=""): nome{n},
                     partido{p}, numero{nu}
            cout << "Construindo Politico!" << endl;</pre>
       virtual void read();
       virtual void imprime();
       friend ostream& operator<<(ostream&, Politico&);</pre>
       friend istream& operator>>(istream&, Politico&);
       virtual ~Politico() { cout << "Destroi Politico!" << endl << endl; }</pre>
};
```







```
class Presidente: public Politico{
   protected:
        string pais;
   public:
        Presidente(string nome="", string partido="", string numero="",
                 string p=""): Politico{nome, partido, numero}, pais{p}
            cout << "Construindo Presidente!" << endl;</pre>
        void read();
        void imprime();
        ~Presidente() { cout << "Destroi Presidente!" << endl; }
};
class Governador: public Presidente{
   protected:
        string estado;
   public:
        Governador(string nome="", string partido="", string numero="",
             string pais="", string e=""): Presidente{nome, partido, numero, pais},
             estado{e}
            cout << "Construindo Governador!" << endl;</pre>
        void read();
        void imprime();
        ~Governador() { cout << "Destroi Governador!" << endl; }
```





```
class Prefeito: public Governador{
    protected:
        string cidade;
    public:
        Prefeito(string nome="", string partido="", string numero="",
                string pais="", string estado="", string c=""):
                Governador{nome, partido, numero, pais, estado}, cidade{c}
            cout << "Construindo Prefeito!" << endl << endl;</pre>
        void read();
        void imprime();
        ~Prefeito() { cout << "Destroi Prefeito!" << endl; }
};
#endif // POLITICO_H
```

```
#include <string>
#include <iostream>
#include "politico.h"
using namespace std;
ostream& operator<<(ostream& out, Politico& p)</pre>
   p.imprime();
   return out;
}
istream& operator>>(istream& in, Politico& p)
   p.read();
   return in;
void Politico::imprime()
    cout << "Numero: " << numero << endl;</pre>
    cout << "Nome: " << nome << endl;</pre>
    cout << "Partido: "</pre>
                             << partido << endl;
```







```
void Politico::read()
   cout << "Digite o nome: ";</pre>
   //getline(cin, nome);
   cin >> nome;
   cout << "Digite o partido: ";</pre>
   //getline(cin, partido);
   cin >> partido;
   cout << "Digite numero do candidato: ";</pre>
   cin >> numero;
void Presidente::imprime()
    Politico::imprime();
    cout << "Pais: " << pais << endl;</pre>
void Presidente::read()
  Politico::read();
   cout << "Digite o pais: ";</pre>
   cin >> pais;
```

UNIFEI - IEST





```
void Governador::imprime()
    Presidente::imprime();
    cout << "Estado: " << estado << endl;</pre>
void Governador::read()
  Presidente::read();
   cout << "Digite o estado: ";</pre>
  cin >> estado;
void Prefeito::imprime()
    Governador::imprime();
    cout << "Cidade: " << cidade << endl << endl;</pre>
void Prefeito::read()
  Governador::read();
  cout << "Digite a cidade: ";</pre>
   cin >> cidade;
```







```
//função de apoio das classes
int menu()
   int op;
   cout << "1 - Presidente" << endl;</pre>
   cout << "2 - Governador" << endl;</pre>
  cout << "3 - Prefeito " << endl;</pre>
   cout << "0 - Sair " << endl;
   cout << "Digite opcao desejada para inserir: ";</pre>
  while(cin >> op) {
      if(op == 0 || op == 1 || op == 2 || op == 3)
         return op;
      cout << "Digite uma das opcoes disponiveis" << endl;</pre>
```

Main



```
#include <string>
#include <iostream>
#include "politico.h"
using namespace std;
#define MAXP 100000
int main(){
    int qtd;
    cout << "Informe quantidade politicos: ";</pre>
    cin >> qtd; cout << endl;</pre>
    if(qtd < 0 \mid | qtd >= MAXP)
        qtd = 5; //limita em 5 caso usuário erre na quantidade
    Politico* vet[ qtd ];
    cout << "Escolha " << qtd</pre>
         << " vezes qual o tipo que deseja criar" << endl;</pre>
```

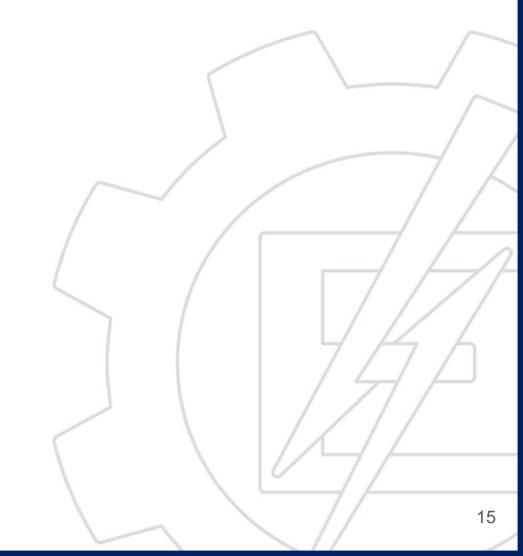


```
for(int j = 0; j < qtd; j++){</pre>
    cout << endl;</pre>
    int op = menu();
     if(op == 0)
        return 0;
    switch( op ) {
        case 1:
             vet[ j ] = new Presidente; cout << endl;</pre>
             cin >> *vet[ j ];
        break;
         case 2:
             vet[ j ] = new Governador; cout << endl;</pre>
             cin >> *vet[ j ];
        break;
         case 3:
             vet[ j ] = new Prefeito; cout << endl;</pre>
             cin >> *vet[ j ];
        break;
```









3^a Questão



Crie um programa principal para testar a hierarquia **Ponto/Circulo/Cilindro**, do ex3 do laboratório 6 com as seguintes características.

- Utilize um vetor de ponteiros para a classe base de modo a representar os diversos elementos a serem criados.
- Permita que o usuário escolha qual tipo de objeto será criado através de um menu de opções do tipo: 1-Ponto, 2-Circulo, 3-Cilindro.
- Utilize um loop para imprimir todos os objetos criados

3^a e 4^a questão Exemplo de Solução – Classe Ponto



```
#include <iostream>
using namespace std;
#define PI 3.1415
#ifndef CIRCULO H
#define CIRCULO H
int menu();
class Ponto
   protected:
      double x, y;
   public:
      Ponto (double x = 0, double y = 0): x\{x\}, y\{y\} { }
       virtual ~Ponto() {}
       virtual void read() { cin >> x >> y; }
       virtual void print() const { cout << "C(" << x << "," << y << ")"; }</pre>
       virtual double area() { return 0; }
        virtual double volume() { return 0; }
       friend istream& operator>>(istream& input, Ponto& in);
        friend ostream& operator<<(ostream& output, const Ponto& out);</pre>
```





3^a e 4^a questão Exemplo de Solução – Classe Circulo

```
class Circulo: public Ponto
protected:
          double raio;
public:
   Circulo(double x = 0, double y = 0, double r = 0): Ponto\{x, y\}, raio \{r\}
   ~Circulo() {}
   void read() { cin >> x >> y >> raio; }
   void print() const { Ponto::print(); cout << " RAIO = " << raio; }</pre>
   double area() { return PI*raio*raio; }
   double volume() { return 0; }
};
```





3^a e 4^a questão Exemplo de Solução – Classe Cilindro

```
class Cilindro: public Circulo
   protected:
             double altura;
   public:
      Cilindro (double x = 0, double y = 0, double raio = 0, double a = 0): Circulo \{x, y, raio\}, altura \{a\}
{ }
      ~Cilindro() {}
      void read() { cin >> x >> y >> raio >> altura; }
      void print() const { Circulo::print(); cout << " ALTURA = " << altura; }</pre>
      double area() { return (2*Circulo::area()) + 2*PI*raio*altura ; }
      double volume() { return (Circulo::area())*altura; }
   };
#endif // CIRCULO H
```





```
#include <iostream>
#include "circulo.h"
using namespace std;
istream& operator>>(istream& input, Ponto& in)
   in.read();
   return input;
ostream& operator<<(ostream& output, const Ponto& out)</pre>
   out.print();
   return output;
```





```
int menu()
{
   int op;
  cout << "1 - Ponto " << endl;</pre>
   cout << "2 - Circulo " << endl;</pre>
   cout << "3 - Cilindro" << endl;</pre>
   cout << "0 - Sair " << endl;</pre>
   cout << "Digite opcao desejada para inserir: ";</pre>
   while(cin >> op) {
      if (op == 0 || op == 1 || op == 2 || op == 3)
         return op;
      cout << "Digite uma das opcoes disponiveis" << endl;</pre>
```

Main





```
#include <iostream>
#include "circulo.h"
using namespace std;
#define PI 3.1415
#define MAXP 100000
int main(){
    int qtd;
    cout << "Informe quantidade formas: ";</pre>
    cin >> qtd; cout << endl;</pre>
    if(qtd < 0 \mid | qtd >= MAXP)
        qtd = 5;
    Ponto* vet[ MAXP ];
    cout << "Escolha " << qtd</pre>
         << " vezes qual o tipo que deseja criar" << endl;</pre>
```





CPP

```
for(int j = 0; j < qtd; j++){</pre>
     cout << endl;</pre>
     int op = menu();
     if(op == 0)
          return 0;
     switch( op ) {
          case 1:
              vet[ j ] = new Ponto; cout << endl;</pre>
              cin >> *vet[ j ];
          break;
          case 2:
              vet[ j ] = new Circulo; cout << endl;</pre>
              cin >> *vet[ j ];
          break;
          case 3:
              vet[ j ] = new Cilindro; cout << endl;</pre>
              cin >> *vet[ j ];
          break;
```

