

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - Campus de Curitiba (Brasil)
Programa de Pós-graduação em Computação Aplicada (PPGCA/DAINF).
 Curso: **Mestrado Profissional - Disciplina: Programação Avançada.**
1ª Prova – Assunto: Orientação a Objetos e Linguagem C++

Nome do Aluno: _____

Horário de Começo: _____ Horário de Fim: _____

Leia toda a prova antes de começar, pois os enunciados podem estar completados uns nos outros.

Utilize os bons princípios de projeto e programação orientada a objetos.

(Questão 1) Em um programa C++ (para *console*), crie uma classe abstrata chamada de *Personagem*. Esta classe terá um atributo *id*. Isto feito, crie uma classe *Jogador* e uma classe *Inimigo*, ambas derivadas de *Personagem* e cada qual com uma função-membro chamada “*void informe_papel()*”. No caso de *Jogador* tal função-membro informará (em tela) “*bom*”, enquanto que no caso de *Inimigo* informará “*mau*”.

(Questão - 2) Crie uma classe *Lista_Personagens* para ponteiros de objetos de subclasses de *Personagem* baseando-se na classe *template vector* da *STL*. Isto deverá ser feito por meio de agregação ou por meio de herança. Ainda, a classe *Lista_Personagens* deverá ter um método para listar, bem como outro para incluir. Por fim, a listagem apresentará o valor de cada atributo de cada objeto, bem como o seu ‘papel’.

(Questão - 3) Considerando a resolução (correta) das questões anteriores e o código abaixo, responda ou resolva objetivamente:

- Ocorre agregação? Por quê?
- Ocorre polimorfismo? Por quê?
- Explique porque há acoplamento no âmbito da classe Principal e quais as conseqüências disto em termos de reaproveitamento de código.
- Corrija o código eliminando o acoplamento.

```
#include <list>
class Principal {
private:
  Jogador jog1, jog2;
  Inimigo ini1, ini2;
  list <Personagem*> Lps;
public:
  Principal () : jog1(1), jog2(2), ini1(3), ini2(4)
  { Lps.push_back ( &jog1 );
    Lps.push_back ( &jog2 );
    Lps.push_back ( &ini1 );
    Lps.push_back ( &ini2 );
    Executar ();
  }
  ~Principal () { }
  void Executar ()
  {
    list< Personagem*>::iterator it;
    for ( it = Lps.begin(); it != Lps.end(); it++)
    {
      (*it)->informe_papel();
    }
  }
};
```

(Questão - 4) Elabore um diagrama de classes para projetar o solicitado nas questões precedentes.

(Questão - 5) Responda objetivamente as questões:

- O que são padrões de projeto?
- Cite pelo menos cinco padrões de projeto GoF.
- Explique dois destes padrões.
- Houve a utilização de algum padrão de projeto nas soluções das questões precedentes? Qual?

A interpretação faz parte do conteúdo da prova! Inclua comentários (se for o caso) para deixar explicitas suas decisões.