

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - Campus de Curitiba Central –
Dep. Acadêmico de Informática (DAINF). Disciplina: **Técnicas de Programação – CSE20.**
Prova sobre linguagem C++ / Diagrama de Classes (*UML*) / Orientação a Objetos - 1ª Parcial

Nome do(a) Discente: _____ Turma: S71

Curso: Eng. Comp. _____ Horário de Começo: _____ Horário de Fim: _____

Leia toda a prova, pois os enunciados estão completados uns nos outros.

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

Em tempo, questões de mesmo peso, idem para os tópicos dentro delas (quando houver).

(Questão - 1) (a) Em um programa C++ (*console*) para representação de animais, crie uma classe abstrata *Canino* com um método virtual puro chamado *reproduzir*. (b) Ainda, esta classe terá pelo menos como atributos protegidos um *booleano* chamado *hibrido*, um *booleano descendente_hibrido*, um apontamento para outro *Canino* chamado *par* e um atributo estático do tipo *ListaCaninos* chamado *Matilha*. (c) Crie um método estático *incluir(Canino* pc)* relativo a *Matilha*. (d) Crie3 um método estático *listar()* relativo a *Matilha*.

(Questão - 2) (a) A classe *Canino* também permitirá duas classes derivadas dela: *Cão* e *Lobo*. Essas duas classes terão cada qual um atributo privado, respectivamente *booleano domestico* e estático inteiro *quantidade*. (b) Em cada uma dessas classes crie um conjunto de métodos que permita o recebimento de apontamento para *par* de objeto de mesma classe (fazendo o atributo *descendente_hibrido* ser *false* em ‘descendentes’) ou o recebimento de apontamento para *par* de objeto de outra classe (fazendo o atributo *descendente_hibrido* ser *true* em ‘descendentes’). (c) redefina método *reproduzir* permitindo, caso o objeto tenha recebido um *par*, criar dinamicamente novo objeto do mesmo tipo da sua classe, definindo apropriadamente se este ‘descendente’ é *hibrido* ou não. Certamente, cada novo objeto será registrado em *Matilha*. (d) no caso de *lobo*, serão criados dois filhotes enquanto o número total de lobos for menor que 10 (dez).

(Questão - 3) (a) No contexto do programa C++ em questão, crie uma classe chamada *ListaCaninos*, seja a implementando via encadeamentos, seja a implementando via algum componente da STL. (b) Essa classe *ListaCaninos* deve ter método de inclusão, além de construtora e destrutora. (c) Essa classe *ListaCaninos* deve ter um método de percorrimto que começa pelo 1º elemento e vai até último, o qual imprime a posição do canino na lista e se ele é *hibrido* ou não. (d) Por fim, tal classe deve ter um método *limpar* (a ser chamado na destrutora) que desaloca cada objeto de *cão* ou *lobo* apontado enquanto *canino*.

(Questão - 4) (a) Crie uma classe *Principal* a ser instanciada na função *main*. Em tal classe será agregado três objetos de *Cão*, os quais serão alocados dinamicamente e apontados por *pc1*, *pc2* e *pc3*. Em tal classe será ainda agregado um objeto de *Lobo*, o qual será alocado dinamicamente e apontado por *pl1*. (b) Na construtora, crie par de *pc1* para com *pc2*, bem como par de *pc3* para com *pl1*. (c) Cada uma dessas instâncias nominadas de *Cão* e de *Lobo* será incluída em *Matilha* via chamada estática do método *Canino::incluir(Canino* pc)*. (d) Crie um método *executar* em *Principal* que chame o método *reproduzir()* de cada um dos objetos nominados, bem como o método estático *Canino::listar()*.

(Questão - 5) (a) No tocante as questões anteriores, crie gets e sets desde que eles sejam efetivamente utilizados. (b) Tal qual, crie construtoras e destrutoras apropriadas (c) Utilize constantes para parâmetros, retorno de métodos e mesmo para métodos em si tanto quanto possível. (d) Primeiramente, antes das demais questões, elabore um diagrama em *UML* das classes solicitadas nelas e de seus relacionamentos, sem necessariamente descrever os atributos e métodos, mas podendo fazê-lo.

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