

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

Nome do(a) Discente: _____ Turma: S71
Curso: _____ 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++ (para *console*), crie uma classe chamada de *Inseto* com um atributo protegido inteiro chamado *energia* e um método “*virtual void executar()= 0*”. (B) Esta classe terá também um método puro “*virtual void locomover () const = 0*”, sendo que nas classes derivadas o método redefinido apenas deve informar em tela se “voa” ou “anda”, bem como o valor do atributo *energia*. (C) Crie uma classe *Abelha* derivada de *Inseto* que terá um atributo booleano protegido chamado *ferrão*, bem como um conjunto apropriado de métodos, dentre eles um método *executar* que chama *locomover*. (D) Crie uma classe *Formiga* derivada de *Inseto*, que terá um atributo protegido ponteiro de *Abelha*, bem como um método void *atacar()* que diminui a *energia* do objeto de *Abelha* apontado caso o seu atributo *ferrão* seja *false*. Ainda haverá nela um método virtual *executar* que chama *atacar* e então *locomover*. Por fim, cada objeto de *Formiga* deve apontar para um objeto de *Abelha*.

(Questão - 2) (A) Crie uma classe *ForAmiga* (acrônimo de formiga amiga) derivada de *Formiga*. Esta classe terá um atributo privado real *amigável*. (B) Neste sentido, a classe em questão terá *ForAmiga* terá um método chamado “*void ajudar()*”, que permitirá incrementar energia do objeto de *Abelha* supostamente apontado se *amigável* for maior que a constante real estática *MIN* de valor 3,1. (C) Ainda, a classe terá um método *executar* que chama *ajudar* e então *locomover*. (D) Por fim, esta classe terá o operador de maior sobrecarregado para que cada objeto de *ForAmiga* possa verificar se é mais amigável que outro objeto de mesmo tipo recebido por meio de referência escondida.

(Questão - 3) (A) Crie uma classe *Colonia* para ponteiros de objetos de *Inseto* por assim dizer. Na prática, a classe *Colonia* servirá para apontar para objetos de subclasses de *Inseto*, apontados como *Inseto* bem entendido. (B) A classe *Colonia* deve ser baseada no componente *map* da STL. (C) Ainda, a classe *Colonia* deverá ter um método *void incluir (Inseto* p)* que incluirá duplas de inteiros e apontamentos insetos, sendo este inteiro gerenciado por um atributo estático inicializado em zero e incrementado após cada inclusão. (D) Por fim, a classe *Colonia* terá outro método *void percorrer()*, o qual fará cada objeto apontado chamar seu método *executar ()*. **Obs.:** Poderia usar outro componente da STL para o item B, sem pontuá-lo, mas habilitando demais questões.

(Questão - 4) (A) Crie uma classe *Habitat* que será instanciada na função *main ()* do programa. (B) Na classe *Habitat* devem ser criados os objetos de *Abelha*, *Formiga* e *ForAmiga* que exercitem as possibilidades de interações entre objetos destes tipos. (C) Ainda, a classe *Habitat* terá método para verificar qual objeto de *ForAmiga* é mais amigável usando operador ternário. (D) Outrossim, a classe *Habitat* terá um objeto de *Colonia*, no qual os endereços objetos pertinentes serão registrados, bem como terá um método *void executar()* chamará o método *percorrer* tal objeto.

(Questão - 5) (A) Primeiramente, antes das demais questões, elabore um diagrama em *UML* das classes solicitadas nelas e de seus relacionamentos. (B) Defina os principais atributos e métodos das classes no diagrama, sendo desnecessário construtoras e destrutoras no diagrama, mas sendo necessário no código. Ainda, no diagrama e no código, *set/get* basta fazer em *Inseto* e indicar se há nas demais classes. (C) Explique por quê há caso evidente de polimorfismo previsto nas questões. (D) Explique se há ou não classe abstrata prevista e o porquê.