Lista de exercícios – Programação orientada a objetos em C++ (LPI)

Professor: Carlos Eduardo Batista

Exercícios práticos

1) Faça um programa em C++ que contenha uma classe que representa um funcionário, registrando seu nome, salário e data de admissão. Crie por último uma classe que representa uma empresa, registrando seu nome e CNPJ. Em todas as classes defina os atributos como privados e crie métodos públicos para acessar e modificar os atributos. Finalmente, faça um programa que:
   - Crie uma empresa;
   - Adicione a empresa alguns funcionários (solicitar no início quantos);
   - Dê aumento de 10% a todos os funcionários de um determinado departamento.

2) Modifique o programa anterior para admitir novos tipos de funcionários, os quais serão definidos a partir de novas classes que herdarão da classe original que representa o funcionário. Serão definidas as subclasses para funcionários do tipo Gerente, Analista e Técnico, os quais deverão sobrecarregar os métodos para recuperação de atributos de forma que exibam qual o cargo do usuário consultado (ex. método que recupera nome deve exibir o cargo e depois o nome). O programa anterior deve ser alterado para:
   - Solicitar o cargo do funcionário adicionado;
   - Aumentar o salário dos gerentes em 15%
   - Exibir ao final do cadastro todos os funcionários e suas informações – utilizar técnicas de polimorfismo (funções virtuais).

3) Faça um programa em C++ que defina uma classe que representa números complexos e que use sobrecarga de operadores para que operações matemáticas básicas (soma, subtração, divisão e multiplicação) sejam realizadas entre os números complexos.

Exercícios teóricos

1) O que é uma classe em programação orientada a objetos?

2) Defina em suas palavras o que é um objeto, e qual a analogia pode ser feita com elementos da programação procedural (imperativa)?

3) O que é herança em programação orientada a objetos? Qual a diferença de herança pública para privada em C++?

4) Para que servem os métodos construtores e destrutores em C++?

5) Cite exemplos de uso para atributos e métodos estáticos em C++ (o da questão abaixo não vale ;).

6) O padrão de projeto *singleton* é utilizado para classes que só devem possuir uma única instância, certo? Como realizar a implementação de uma classe com essas características utilizando métodos e atributos estáticos?

7) Qual a diferença entre herança e composição e como estas técnicas podem ser utilizadas para facilitar o reuso de código em C++?

8) Para que serve e quais são os tipos de encapsulamento em C++?

9) Qual a relação do uso de métodos virtuais em super classes em C++ e o conceito de polimorfismo da programação orientada a objetos?

10) Qual é a saída (na tela) da execução do seguinte programa em C++:

```
#include <stdlib.h>
#include <iostream>

class C1 {
public:
   virtual void f1( );
   void f2( );
};
void C1::f1(){std::cout<<"C1-f1"<< std::endl;}
void C1::f2(){std::cout<<"C1-f2"<< std::endl;}

class C2: public C1 {
public:
   void f1( );
   void f2( );
};
void C2::f1(){std::cout<<"C2-f1"<< std::endl;}
void C2::f2(){std::cout<<"C2-f2"<< std::endl;}


void g(C1 &an) {
   an.f1();
}

int main() {
   C2 a1;
   C1 a2;
   g(a1);
   g(a2);
   C1 * pt = &a1;
   pt-> f2();
   pt-> f1();
   return 0;
}
```