

Universidade Federal de Ouro Preto – UFOP Instituto de Ciências Exatas e Biológicas – ICEB Departamento de Computação – DECOM Disciplina: Programação Orientada a Objetos 1s2016



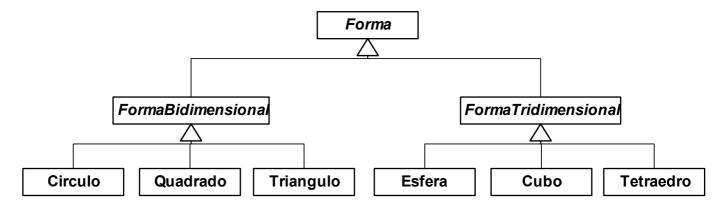


Adaptado da lista elaborada pelo professor: Marco Antonio M. Carvalho em 2s2014

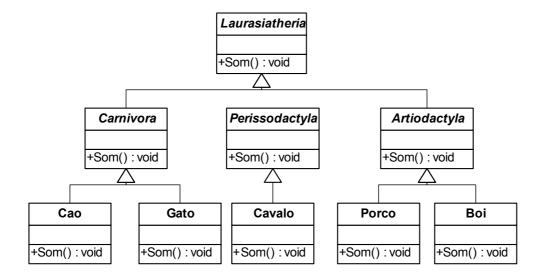
Lista de Exercícios 07 – Polimorfismo e Classes Abstratas

Instruções

- Todos os exercícios que envolvem programas devem ser resolvidos por programas em linguagem C++;
- Na solução dos exercícios, devem ser utilizados os conceitos listados no cabeçalho desta lista;
- Data de entrega, via Moodle, 11 de julho até às 23h55.
- 1. Implemente a hierarquia Formato como ilustrado pelo diagrama abaixo e descrito a seguir:
 - a. Cada FormaBidimensional deve conter um método getArea() que calcula sua área;
 - b. Cada *FormaTridimensional* deve conter os métodos *getArea()* e *getVolume()*, para calcular área e volume, respectivamente;
 - c. Cada classe deve possuir um destrutor específico, que imprima o nome da própria classe;
 - d. Crie um *driver* com um vetor de ponteiros para *Forma*, cada ponteiro deve apontar para um dos objetos de cada classe concreta da hierarquia;
 - e. O *driver* deve imprimir a classe de origem do objeto apontado por cada ponteiro, o que deve ser determinado em tempo de execução;
 - f. O driver deve imprimir, para cada objeto apontado, se trata-se de uma FormaBidimensional ou FormaTridimensional, o que deve ser determinado em tempo de execução
 - i. Se a forma for bidimensional, imprima sua área;
 - ii. Se a forma for tridimensional, imprima sua área e volume.
 - g. Processando os objetos polimorficamente, o destrutor deve ser invocado através do método delete().



- 2. Implemente as classes e o relacionamento entre classes descritos no diagrama UML abaixo. Note que apenas as classes do último nível da hierarquia são concretas. No *driver*, crie um vetor de ponteiros para classe *Laurasiatheria*, em cada posição, adicione ponteiros para cada uma das classes concretas, em ordem alfabética.
 - a. Utilizando o comportamento polimórfico, utilize os ponteiros criados para invocar o método Som() adequado para cada classe apontada;
 - b. Determine, em tempo de execução, a classe de cada objeto apontado pelo vetor;



- 3. Adicione um método *desenhar()* à classe *FormaBidimensional* e à hierarquia que a tem como classe base.
 - a. O construtor de cada classe concreta deverá especificar o tamanho (em quantidade de caracteres) e o caractere de preenchimento para desenhar cada forma, gerando figuras como as apresentadas abaixo;
 - b. Crie um *driver* com um vetor de ponteiros para *FormaBidimensional*, cada ponteiro deve apontar para um dos objetos de cada classe concreta da hierarquia
 - i. Invoque o método desenhar() de cada objeto apontado.

XXX	A	
XXX	A A	000
XXX	A A A	00000
	A A A A	000

4. Descreva, num arquivo .doc, as principais diferenças entre Java e C++ no que tange a implementação de interfaces, classes abstratas, polimorfismo e acoplamento dinâmico, ilustrando com códigos eventuais diferenças de sintaxe e conceitos.