

ECOP13 – Laboratório de Programação Orientada a Objetos

Prof. João Paulo R. R. Leite (joaopaulo@unifei.edu.br)

Atividade 8 – Herança e Polimorfismo

As atividades devem ser feitas em arquivos “.CPP” e “.H”. Os nomes dos arquivos “.CPP” principais de cada exercício estão indicados (ex01.cpp, etc.). Ao final, coloque todos em um mesmo **arquivo com extensão “.ZIP”**, que você deverá enviar na tarefa “Laboratório 9” cadastrada no SIGAA até o dia 12/11/2021 às 10:10h. **Não serão aceitas entregas via e-mail ou atrasadas.** A entrega valerá nota e presença. Os exercícios da semana têm como objetivo principal fazer com que o aluno pratique o conceito de **Herança e Polimorfismo em C++**. Siga as instruções:

1. **(ex01.cpp)** Escreva uma classe em C++ chamada Animal e três subclasses: Cavalo, Baleia e Girafa. A classe base deve ter uma **função virtual** chamada carrega_detalhes(), responsável por imprimir os detalhes dela e das subclasses. Cada classe possui uma versão dessa função, que imprime uma das mensagens abaixo:

Animal: "Carregando detalhes do animal."

Cavalo: "Carregando detalhes do cavalo."

Baleia: "Carregando detalhes da baleia."

Girafa: "Carregando detalhes da girafa."

Você também pode colocar outros atributos e funções membro de sua escolha em cada uma das classes, se quiser. Na **main**, faça um vetor de ponteiros para a classe base Animal que tenha 6 posições. Para cada posição do vetor, instancie um objeto de uma das diferentes classes (base ou derivadas). A seguir, faça um laço de repetição em que é chamada, a partir de cada ponteiro do vetor, a função (virtual) carrega_detalhes.

A execução deve ser semelhante ao exemplo abaixo:

```
Carregando detalhes do animal.  
Carregando detalhes do cavalo.  
Carregando detalhes do animal.  
Carregando detalhes da baleia.  
Carregando detalhes do animal.  
Carregando detalhes da girafa.
```

2. **(ex02.cpp)** Crie um programa em C++ que tenha uma hierarquia de classes, com a classe Investimento como base. A classe Investimento tem as seguintes propriedades (membros de dados): valor (float, valor inicial) e taxa (float, taxa mensal de juros). A seguir, a partir da classe base, crie as subclasses PapelBolsa, RendaFixa e Imobiliario. A classe Investimento deve ter a **função virtual** retorno() que receba como parâmetro o inteiro "meses", e cada classe derivada deve sobrescrevê-la e calcular o retorno de maneira diferente, como descrito a seguir:

- **PapelBolsa**: somatório do valor multiplicado pela taxa positiva para meses pares e taxa negativa para meses ímpares.
- **RendaFixa**: somatório do valor multiplicado pela taxa mensal menos 15% de rendimento mensal $Taxa\ Real = (Taxa - Taxa * 0.15)$.
- **Imobiliario**: somatório do valor multiplicado por três quartos (3/4) da taxa mensal

Veja abaixo um exemplo de execução. É importante que você utilize de **polimorfismo** na chamada da função retorno (ponteiros da classe base apontando para objetos de qualquer uma das classes da hierarquia). Não declare simplesmente um objeto de cada classe.

=== Programa de Rendimentos ===

Entre com os seguintes valores:

Valor inicial da aplicação (R\$): 3500 (cin)
Taxa do papel da bolsa: 0.08 (cin)
Taxa do fundo renda fixa: 0.02 (cin)
Taxa do fundo imobiliário: 0.05 (cin)
Número de meses: 24 (cin)

Expectativa de rendimentos:

Rendimento do papel da bolsa R\$ 3420.46
Rendimento do fundo renda fixa R\$ 5245.31
Rendimento do fundo imobiliário R\$ 8468.03

--Fim do programa--

Se tiver dúvidas entre em contato por e-mail! Não deixe para a última hora...

Este guia foi baseado em material desenvolvido pelo Prof. Carlos Henrique Valério de Moraes (UNIFEI).