

ECOP13 – Laboratório de Programação Orientada a Objetos

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

Atividade 7 – Herança

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 8” cadastrada no SIGAA até o dia 05/11/2020 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 em C++**. Siga as instruções:

1. **(ex01.cpp)** Escreva um programa que defina uma classe "Shape" com os atributos largura e altura e um construtor padrão, que os inicialize com o valor zero. Sobrecarregue o operador >> para que o usuário entre com os valores de largura e altura, e o operador << para imprimir suas características. A seguir, defina duas classes derivadas "Triangle" e "Rectangle", que herdam de "Shape" e possuem funções membro que calculem a área das respectivas formas. Na função principal, defina duas variáveis e suas respectivas classes, um triângulo e um retângulo, entre com seus atributos e, em seguida, chame a função "area()" nesses dois tipos de variáveis. Veja um exemplo de execução:

```
Entre com base e altura do triângulo: 10 5
Entre com largura e altura do retângulo: 10 5
O triângulo possui largura 10 e altura 5.
O retângulo possui largura 10 e altura 5.
Triângulo (área): 25
Retângulo (área): 50
```

2. **(ex02.cpp)** Escreva um programa em C++ com as seguintes classes:
 1. **Basic** - contendo como atributos nome, ID, idade, e uma função membro que recupere esses dados. A classe deve ter um construtor que receba valores a serem colocados em cada um dos atributos.
 2. **Department** - contendo como atributos nome do departamento, cargo, horas a trabalhar e uma função que recupere esses dados. A classe deve ter um construtor que receba valores a serem colocados em cada um dos atributos.
 3. **Employee** – herda das duas classes anteriores (herança múltipla), possui um construtor que recebe todos os dados necessários e invoca implicitamente os construtores das duas classes base para inicializar o objeto. Implemente também uma função para imprimir todos os dados do funcionário, no formato mostrado abaixo.

Na função principal crie um objeto da classe **Employee**, inicialize com os valores que preferir, e imprima seus dados através de uma chamada de função membro "imprimir()" de **Employee**. Veja como deve ser uma execução:

```
Enter employee's basic info
```

```
Enter Name: Teodomiro
Enter Id: 100
Enter Age: 52
```

```
Enter employee's department info
```

Enter Department Name: IESTI
Enter assigned work: Fundador
Enter time in hours to complete work: 460

=====
EMPLOYEE'S INFORMATION
=====

BASIC INFORMATION
Name: Teodomiro
ID: 100
Age: 52 years old

DEPARTMENT INFORMATION

Department: IESTI
Assigned Work: Fundador
Time to complete work: 460 hours.

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).