



PCS 3111 - Laboratório de Programação Orientada a Objetos para Engenharia Elétrica

Aula 03 – Conceitos Básicos de OO

Atenção

1. As definições das classes usadas nos exercícios encontram-se **disponíveis no e-Disciplinas**.
2. Os nomes, os atributos, os métodos, e as respectivas assinaturas das classes dadas **devem seguir o especificado** em cada exercício para fins de correção automática.
3. A **ordem de declaração** de atributos e métodos fornecidos **não deve ser alterada**. Caso contrário, poderá haver redução automática da nota.
4. A função main **não deve ser submetida**. Caso contrário, a correção automática retornará um *Compilation Error*.

Exercício 01

Usando a definição fornecida e as especificações a seguir, implemente a classe Pessoa:

```
class Pessoa {  
public:  
    string nome;  
    bool disponivel;  
    int horasMensais;  
  
    double getSalarioDiario();  
    void imprimir();  
};
```

- Uma Pessoa representa um trabalhador que possui nome, disponibilidade e horas mensais de serviço;
- O método getSalarioDiario() retorna o salário de uma Pessoa em um dia de trabalho, dado por:

$$\text{salário diário} = 50 + \frac{\text{horas mensais}}{22.0}$$

caso ela esteja disponível e 0, caso contrário.

Observação: O 22 representa a média de dias úteis por mês.

- O método imprimir() deve exibir na tela os dados de cada Pessoa no seguinte formato:

<nome> - <disponibilidade> - <horas>h/mes - R\$<salário> por dia

Por exemplo:

Ana - disponivel - 120h/mes - R\$55.4545 por dia

Pedro - nao disponivel - 0h/mes - R\$0 por dia



Exercício 02

Usando a definição fornecida e as especificações a seguir, implemente a classe Atividade:

```
class Atividade {  
public:  
    string nome;  
    int duracao;  
    Pessoa *p1 = NULL;  
    Pessoa *p2 = NULL;  
  
    Pessoa* getResponsavel();  
    bool adicionar(Pessoa *p);  
    double getCusto();  
    void imprimir();  
};
```

- Uma Atividade é constituída por nome, duração (em dias) e, no máximo, duas pessoas disponíveis para realizá-la. Cada Atividade possui um responsável, obtido pelo método `getResponsavel()`, o qual retorna um ponteiro para a pessoa que apresenta o menor salário. Considere que o salário diário de duas pessoas nunca será igual. Caso não existam pessoas na Atividade, o método `getResponsavel` deve retornar `NULL`.
- O método `adicionar(Pessoa* p)` armazena uma pessoa **disponível** em `p1` ou `p2` (nesta ordem) e retorna `true`. Caso a pessoa não esteja disponível ou não seja possível adicionar mais pessoas (já existem duas pessoas na Atividade) deve-se retornar `false`.
- O método `getCusto()` retorna o custo de uma Atividade e é determinado por:

$$\text{custo} = \text{duração} * \text{salário diário da pessoa responsável}$$

Caso não haja uma pessoa responsável, retornar -1.

- O método `imprimir()` deve exibir na tela os dados de cada Atividade no seguinte formato:

<nome da atividade> - <duração> dias - R\$<custo>

<nome do responsável> - <disponibilidade> - <horas>h/mes - R\$<salário> por dia

Sendo <...> o valor armazenado em cada variável. Por exemplo:

Teste - 2 dias - R\$110.909

Ana - disponível - 120h/mes - R\$55.4545 por dia

Suponha que ao chamar esse método sempre há um responsável.

Considere que os atributos são sempre atribuídos antes de se chamar os métodos que dependem deles.



Testes do Judge

Exercício 1

- Pessoa: Teste getSalarioDiario com pessoa disponível
- Pessoa: Teste getSalarioDiario com pessoa não disponível

Exercício 2

- Atividade: Teste adicionar com nenhuma pessoa adicionada
- Atividade: Teste adicionar com uma pessoa adicionada
- Atividade: Teste adicionar com duas pessoas adicionadas
- Atividade: Teste adicionar com pessoa não disponível
- Atividade: Teste getResponsavel com uma pessoa na Atividade
- Atividade: Teste getResponsavel com duas pessoas na Atividade
- Atividade: Teste getResponsavel sem pessoas na Atividade
- Atividade: Teste getCusto com responsável
- Atividade: Teste getCusto sem responsável