ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO
Departamento de Engenharia de Computação e Sistemas Digitais

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

Aula 07 – Herança e Polimorfismo II

Atenção

1. Código inicial a ser usado na resolução dos exercícios encontra-se **disponível 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.

Exercício 1

Considere as classes **Pessoa**, **Atividade**, **AtividadeComSupervisor** e **Projeto**, implementadas nas aulas anteriores.

Na classe **Projeto**:

- Implemente o método `bool adicionar(Pessoa* p)`, o qual possibilita a adição de uma **Pessoa** a um **Projeto**. Se já houver `maximoValor` **Pessoas** ou na hipótese de a **Pessoa** ter sido previamente adicionada (ou seja, o mesmo objeto), o método deve retornar false. Caso contrário, deve-se adicionar a **Pessoa** ao **Projeto** e retornar true.
- Implemente os métodos `Pessoa** getPessoas()` e `int getQuantidadePessoas()` que retornam o vetor de **Pessoas** adicionadas e seu tamanho, respectivamente.
- Modifique o método `adicionar(Atividade* a)`, acrescentando a condição de que somente **Atividades** cujos participantes não supervisores também estejam inseridos no **Projeto** podem ser adicionadas a ele.

  Por exemplo, no cenário abaixo, "Leitura" não pode ser adicionada ao **Projeto** "EP2", ao passo que "Testes", pode.

Nome	Classe	Supervisor	Pessoas
EP2	Projeto	---	Ana, Maria, João
Leitura	Atividade	---	Ana, Paula
Testes	AtividadeComSupervisor	Lucas	Maria, João

Note que o método `adicionar` foi sobrecarregado. Os métodos públicos dessa classe são apresentados a seguir.

```
class Projeto {
public:
    Projeto(string nome, int maximoValor);
    ~Projeto();

    int getDuracao();
    int getQuantidadeAtividades();
    int getQuantidadePessoas();
    Atividade** getAtividades();
    Pessoa** getPessoas();

    bool adicionar(Atividade* a);
    bool adicionar(Pessoa* p);
};
```

Implemente apropriadamente os métodos conforme especificado. **NÃO** altere as classes **Pessoa** e **Atividade** fornecidas.

Exercício 2

Considere a classe **AtividadeComSupervisor** apresentada a seguir.

```
class AtividadeComSupervisor: public Atividade {
private:
    Pessoa* supervisor;

public:
    AtividadeComSupervisor(string nome, int horasNecessarias,
                           int maximoPessoas, Pessoa* supervisor);
    ~AtividadeComSupervisor();

    Pessoa* getSupervisor();
};
```

Redefina o método `bool adicionar(Pessoa* p)` nessa classe, de modo que não seja possível adicionar o supervisor ao vetor de pessoas.

Altere o arquivo `Atividade.h` para que o método mais especializado seja sempre utilizado, independente do tipo da variável. Mas não modifique a implementação dos métodos em `Atividade.cpp`.

Exercício 3

Na classe **Projeto**, implemente o seguinte método:

```
Atividade** getAtividadesSemSupervisor(int& quantidade);
```

O método `getAtividadesSemSupervisor` retorna um vetor alocado dinamicamente contendo todas as **Atividades** sem supervisor de um **Projeto** (ou seja, objetos que não sejam da classe **AtividadeComSupervisor**). Além disso, deve-se retornar a quantidade de **Atividades** sem supervisor através do parâmetro `quantidade`, passado por referência. Caso o **Projeto** não possua **Atividades** sem supervisor, deve-se retornar NULL.

Testes do Judge

Exercício 1

- Projeto: adicionar Pessoa com vetor vazio
- Projeto: adicionar Pessoa com vetor parcialmente preenchido
- Projeto: adicionar Pessoa com vetor cheio
- Projeto: adicionar com Pessoa repetida
- Projeto: adicionar Atividade com Pessoas não participantes do Projeto no início do vetor
- Projeto: adicionar Atividade com Pessoas não participantes do Projeto no meio do vetor
- Projeto: adicionar Atividade com Pessoas não participantes do Projeto no fim do vetor
- Projeto: adicionar Atividade com Pessoas participantes do Projeto

Exercício 2

- AtividadeComSupervisor: adicionar supervisor
- AtividadeComSupervisor: adicionar não supervisor com vetor vazio
- AtividadeComSupervisor: adicionar não supervisor com vetor parcialmente preenchido
- AtividadeComSupervisor: adicionar não supervisor com vetor cheio

Exercício 3

- getAtividadesSemSupervisor sem Atividades sem supervisor
- getAtividadesSemSupervisor com uma Atividade sem supervisor
- getAtividadesSemSupervisor com mais de uma Atividade sem supervisor