



PCS 3111 - Laboratório de Programação Orientada a Objetos para Engenharia Elétrica 2021

Aula 08 – Classes Abstratas, Herança Múltipla e Métodos e Atributos Estáticos

Atenção

- Código inicial a ser usado na resolução dos exercícios encontra-se **disponível no Discord**.
- 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.
- Submeta um arquivo comprimido (faça um “.zip” – **não pode ser “.rar”**) colocando apenas os arquivos “.cpp” e “.h”. Não crie pastas no “zip”.
- Comente a função *main* ao submeter.

AVISO: para evitar problemas de compilação no Judge, envie os exercícios à medida que implementá-los!

Exercício 1

Altere a classe **Artista** para que ela seja abstrata. Essa classe segue a mesma ideia das aulas anteriores e já está implementada.

```
Artista(int quantidadeMaxima, string nome);  
virtual ~Artista();  
  
Musica** getMusicas();  
int getQuantidadeDeMusicas();  
int getQuantidadeMaxima();  
  
string getNome();  
  
virtual double getNota();  
  
bool adicionarMusica(Musica* musica);  
  
virtual void imprimir();
```

Faça o método `imprimir` ser abstrato.

Implemente também uma classe concreta **Pessoa**, filha de **Artista**, que recebe em seu construtor os parâmetros `quantidadeMaxima` e `nome`. A classe **Pessoa** é útil para permitir que uma banda seja composta apenas por Pessoas e não por Artistas, tendo em vista que isso permitiria que uma Banda seja composta por Bandas. Ela não tem nenhum comportamento adicional. A impressão deve funcionar da mesma forma que na classe **Artista**.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Departamento de Engenharia de Computação e Sistemas Digitais

```
Pessoa(int quantidadeMaxima, string nome);  
virtual ~Pessoa();
```

Modifique a classe **Banda**, filha de **Artista**, para que ela seja concreta (faça as adaptações necessárias) e que seus membros sejam **Pessoas** (ou seja, usando um vetor de **Pessoas**). A sua definição deve ficar com os seguintes métodos específicos:

```
Banda(int quantidadeMaxima, string nome);  
virtual ~Banda();  
  
bool adicionar(Pessoa *a);  
  
Pessoa** getParticipantes();  
double getNota();
```

Mantenha o comportamento do método `adicionar` (retorne falso caso não haja espaço no vetor ou a pessoa já tenha sido adicionada anteriormente) e do método `imprimir`.

Complete o código fornecido. Para tanto considere os seguintes passos na função teste:

1. Crie a Banda com nome *Banda do Mar* e com 3 participantes;
2. Adicione à *Banda do Mar* a pessoa *Marcelo Camelo* (com 3 de `quantidadeMaxima`);
3. Adicione à *Banda do Mar* a pessoa *Maria Luiza* (com 3 de `quantidadeMaxima`);
4. Imprima a Banda.

Exercício 2

Adicione ao projeto a classe Usuario, já implementada. Implemente a classe **UsuarioPessoa** (cuja definição é entregue; adicione-a ao projeto) que deve ser derivada das classes **Pessoa** e **Usuario**. A classe **UsuarioPessoa** possui apenas o construtor (e o destrutor, vazio), que deverá ser igual ao seguinte modelo:

```
UsuarioPessoa(string nome, string login, int maximoFavoritos,  
int quantidadeMaxima);  
virtual ~UsuarioPessoa();
```

Refaça a função teste. Para tanto considere os seguintes passos:

1. Crie o **UsuarioPessoa** “*Tom Jobim*”, de login “*tom@usp.br*”, `quantidadeMaxima` e `maximoFavoritos` igual a 1;
2. Imprima “*Tom Jobim*”.

Exercício 3

Na classe **Banda** adicione e implemente os métodos estáticos `setBonus` e `getBonus`, deixando os já existentes. Ou seja, a classe deve possuir os seguintes métodos públicos:



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Departamento de Engenharia de Computação e Sistemas Digitais

```
Banda(int quantidadeMaxima, string nome);  
virtual ~Banda();  
  
bool adicionar(Pessoa *a);  
  
Pessoa** getParticipantes();  
double getNota();  
void imprimir();  
  
static double getBonus();  
  
static void setBonus(double valor);
```

No código fornecido a nota é calculada como a média da nota das **Pessoas** acrescida de um bônus fixo de 1, limitado a 5. Ao invés de usar um valor fixo de bônus, use o valor retornado pelo método estático `getBonus`. O método `setBonus` deve ser implementado de modo a permitir alterar o valor do bônus. Considere 0,1 como valor inicial do bônus e continue limitando o valor do `getNota` a 5. Por exemplo, suponha que o valor do bônus é o seu valor inicial (0,1) e que uma banda apresenta participantes com notas 5 e 3 e, conseqüentemente, média igual a 4; ao se chamar o método `getNota`, ele deve retornar 4,1. Caso o valor do bônus seja atualizado para 0,5, ao se chamar o método `getNota`, ele deve retornar 4,5. Como é um método estático, caso se chame o método `getNota` para outras bandas, o bônus considerado também deve ser de 0,5.

Refaça a função teste. Para tanto considere os seguintes passos:

1. Crie a Pessoa “*Tom Jobim*”, de `quantidadeMaxima` igual a 1;
2. Adicione a música “*Samba de uma nota so*”, de duração 180 e avaliada com nota 5, a *Tom Jobim*;
3. Crie a Pessoa “*Elis Regina*”, de `quantidadeMaxima` igual a 1;
4. Adicione a música “*Aguas de marco*”, de duração 214 e avaliada com nota 4, a “*Elis Regina*”;
5. Crie a Banda “*Elis & Tom*”, de `quantidadeMaxima` igual a 2;
6. Adicione “*Tom Jobim*” e “*Elis Regina*”, nesta ordem, como participantes de “*Elis & Tom*”;
7. Imprima a nota da Banda “*Elis & Tom*” (somente o valor, pulando uma linha após). Não altere o valor do bônus.
8. Modifique o bônus da Banda para 0,2;
9. Imprima a nota da Banda “*Elis & Tom*” (somente o valor, pulando uma linha após).

Testes do Judge

Exercício 1

- Artista é classe abstrata



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Departamento de Engenharia de Computação e Sistemas Digitais

- Banda é classe derivada de Artista
- Banda: vetor de Pessoa
- Pessoa: getters
- Teste da função teste

Exercício 2

- Usuario: getters
- UsuarioPessoa é classe derivada de Pessoa e de Usuario
- Teste da função teste

Exercício 3

- Banda: getBonus é método estático
- Banda: primeira nota com valor inicial
- Banda: valor inicial do bônus
- Teste da função teste