

INSTITUTO FEDERAL DA PARAÍBA
CAMPUS CAMPINA GRANDE
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO
DISCIPLINA DE POO e LAB. POO
PROF. VICTOR ANDRÉ PINHO DE OLIVEIRA

Atividade Avaliativa de POO e Lab. POO - Unidade III

Instruções

DIVERTIDA MENTE 3



Responda às questões abaixo.

Atenção: **Escreva suas respostas em cor de raiva (vermelho).**

Questões

1. Devia ser um dia como qualquer outro, mas o clima chuvoso acabou nublando as emoções de Riley. Não havia caído ainda uma gota, mas, ainda assim, Riley preferia ficar em seu quarto, dominada pelo **tédio**.

Encorujada em sua cama, sentia **inveja** dos transeuntes apressados, os quais eram vistos do outro lado da janela. Alguns exibiam um largo sorriso, outros exalavam um ar de satisfação. Cada um seguindo a plenos pulmões os rumos de suas próprias vidas. Por um breve instante, a **raiva** se fixou em seu encalço.

Em certo momento, algo lá fora chamou sua atenção, dissonante da multidão apressada. Suas vestes eram incomuns naquela parte da cidade. Em qualquer parte do mundo, a bem da verdade. O estranho vestia uma capa preta, cujas bordas estampavam um vermelho sangue. Porém, o mais estranho era o chapéu de abas com um cume pontudo, levemente arqueado. E sua mão direita empunhava uma... varinha. Riley teve **medo**.

Ainda absorta diante daquela visão, a qual ninguém parecia notar, o estranho parou bem diante de sua janela, o que permitia uma visão ainda mais nítida do estranho. O chapéu começou a girar, e um par de óculos bem redondos miraram em sua direção. Um raio na testa transformou seu **medo** em terror, mas o sorriso no rosto foi capaz de dissipar qualquer emoção negativa.

Sim, ela conhecia aquele rapaz. Era Harry Potter.

Seus olhos fizeram contato visual e Riley sentiu seu coração acelerar, com certa **ansiedade**. Sem ser capaz de estar no controle de suas emoções, Riley correu escada abaixo, e saiu porta afora o mais rápido que pôde. Parecia saltitante de **alegria**.

Trocaram rápidos cumprimentos, como uma dupla que já se conhecia há tempos. Riley era fã da saga Harry Potter, e Harry adorava Divertida Mente. Depois de alguns minutos de conversa, Harry precisou se despedir, mas, notando que Riley poderia mudar seu estado de ânimo, propôs uma brincadeira: iria conjurar algumas memórias de Riley. Cada memória seria capaz de evocar ou dissipar uma ou mais emoções.

Apesar de ser uma brincadeira inocente, Harry não sabia exatamente como a sala de controle ficaria após a brincadeira, isto é, quais emoções estariam no controle.

Ajude Harry a descobrir.

Considerando o acima exposto, crie um conjunto de classes para representar a brincadeira que Harry propôs a Riley. Dê uma solução que atenda minimamente aos seguintes requisitos, bem como permita rodar a função main abaixo. Tudo baseado

no que foi estudado em sala (**sobrecarga de operadores, herança, polimorfismo etc.**):

a. Hierarquia Emocao:

1. Escreva uma classe básica abstrata chamada **Emocao** (1,0 ponto). Essa classe tem como atributos um nome e um booleano para indicar se está ativa ou não.
O construtor deve receber o nome e o estado da emoção; esta última deve possuir o argumento default falso.
A classe deve possuir um método virtual puro chamado processar, que recebe uma string contendo a descrição da emoção.
Também deve possuir um método estaAtiva que retorna o estado da emoção.
2. Escreva uma classe **Alegria** que herda de **Emocao** (1,0 ponto). Essa classe deve sobrescrever o método virtual processar, de modo a se comportar da seguinte forma: ativa a emoção toda vez que a quantidade de caracteres for par.
3. Escreva uma classe **Raiva** que herda de **Emocao** (1,0 ponto). Essa classe deve sobrescrever o método virtual processar, de modo a se comportar da seguinte forma: inverte o estado da emoção toda vez que a quantidade de vogais for ímpar.
4. Escreva uma classe **Tristeza** que herda de **Emocao** (1,0 ponto). Essa classe deve sobrescrever o método virtual processar, de modo a se comportar da seguinte forma: ativa a emoção toda vez que a quantidade de caracteres for ímpar.

b. Classe Riley: (1,5 ponto)

1. Escreva uma classe chamada **Riley** com a capacidade de suportar todas as emoções descritas anteriormente.
2. Sua classe deve sobrecarregar o operador << para permitir a entrada de uma emoção na sala de controle de Riley. (Lembre-se que Riley suporta várias emoções).
3. A classe deve implementar um método chamado processar, que recebe uma string contendo a descrição da memória e chama o método processar de cada emoção.
4. Por fim, sua classe também deve sobrecarregar o operador << para operar junto do cout, a fim de mostrar na tela as emoções ativas de Riley.

Para demonstrar a sua solução, copie e cole no seu projeto a main abaixo.

```
int main() {  
    Riley riley;  
  
    riley << Alegria("Alegria");  
    riley << Raiva("Raiva");  
}
```

```

riley << Tristeza("Tristeza");

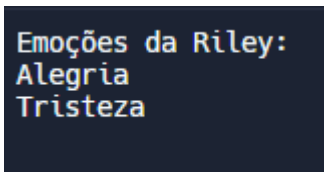
riley.processar("Brincando no parquinho.");
riley.processar("Viagem para Disney.");
riley.processar("Primeiro encontro.");
riley.processar("Uma noite no museu.");
riley.processar("O dia em que quebrou um osso.");

std::cout << riley << std::endl;

return 0;
}

```

Cole o print da saída: (0,5 ponto)



```

Emoções da Riley:
Alegria
Tristeza

```

2. Ainda baseado na questão anterior, responda:

- a. Quais os pilares do paradigma Orientado a Objetos foram aplicados? E o que eles significam? (1 ponto)

R: Foram usados os pilares da Herança e do Polimorfismo dentro do código, além disso foi usado a sobrecarga de operadores para poder realizar a função de entregar as funções a Riley. A herança é um pilar que baseia o código em classes bases que vão possuir outras que vão se derivar dela, recebendo os atributos e métodos públicos e protegidos pela classe pai. O polimorfismo é uma forma de sobrescrever métodos através da herança das classes, dessa forma, sempre o método da classe derivada acaba sendo a responsável por implementar o método, exceto nos casos que a classe pai tem uma forma padrão de executar, fora este, o método é implementado pela classe filha baseada no que ela precisa fazer, ou seja, é o que permite vários objetos responderem o mesmo método de maneiras diferentes.

- b. O que é um método virtual, e qual a diferença entre um método virtual de um método virtual puro? (1 ponto)

R: Um método virtual é um método que permite que ele seja sobrescrito por uma outra classe filha, ou seja, o virtual está na classe pai, para que a classe filha a herde e possa sobrescrever o método. A diferença entre um método virtual e método virtual puro é que o método virtual puro é completamente vazio, não recebe argumentos, nem parâmetros, é completamente vazio.

- c. Quais os mecanismos básicos para se obter o polimorfismo? (1 ponto)

R: Ter uma classe pai que possua um método virtual e ter um classe filha que herde essa classe pai para poder sobrescrever esse método, naturalmente, é necessário ter mais de uma classe que se derive da pai para ter mais de uma função relacionada à um método específico, dessa forma, quando os objetos forem criados e executarem o método, cada um vai executar de uma forma diferente uma da outra.

d. Para que serve a sobrecarga de operadores? (1 ponto)

R: A sobrecarga de operadores serve para que um mesmo operador que tenha sido reservado pela linguagem possa realizar mais de uma função, além da função padrão definida pela linguagem C++.

“Aconteceram muitas coisas ultimamente, mas continuamos a gostar da nossa menina. Ela tem novos amigos, uma casa bonita... As coisas não podiam estar melhores. Afinal de contas, a Riley tem 12 anos. O que é que pode acontecer?”