

## Lista de exercícios 2

### Exercício 1

Crie um diretório para armazenar os arquivos referentes ao exercício 1.

Crie as seguintes classes. Para cada classe, crie o arquivo de cabeçalho e implementação. Depois, crie a função main que integrará todas as classes.

#### A. Classe Animal:

Crie uma classe Animal com os seguintes atributos e métodos:

- nome: String que deverá conter o nome do animal.
- idade: Inteiro que representa a idade do animal.
- nivelEnergia: Inteiro que varia de 0 a 100 e representa o nível de energia do animal.
- Construtor Animal(string nome, int idade, int nivelEnergia).
- setNome(string nome): Define o nome do animal.
- setIdade(int idade): Define a idade do animal.
- setNivelEnergia(int energia): Define o nível de energia do animal, verificando se o valor está entre 0 e 100.
- emitirSom(): Exibe a seguinte mensagem: *"Som genérico do animal"*.
- mostrarInfo(): Exibe o nome, a idade e a energia do animal.
- dormir(int energia) que recebe um inteiro como parâmetro e aumenta o valor de nivelEnergia. Se o nível de energia for maior do que 100, nenhum valor deverá ser somado.

#### Observação!

Para que as subclasses possam herdar atributos ou métodos de uma superclasse, eles devem ser do tipo `protected` ou `public`. Se forem do tipo `private`, eles não serão acessíveis pela subclasse.

#### B. Classe Cachorro:

Crie uma classe Cachorro que herda da classe Animal.

- Os atributos de nome, idade e nivelEnergia devem ser herdados de Animal.

- Os métodos *setters* devem ser herdados de animal.
- O método dormir deve ser herdado de Animal.
- Adicione o atributo raça: String que representa a raça do cachorro.
- Construtor Cachorro(string nome, int idade, int nivelEnergia, string raca).
- Adicione o método setRaça(string raca) que define a raça do animal.
- Sobrescreva o método emitirSom( ) para exibir a mensagem: “Au Au”.
- Sobrescreva o método mostrarInfo( ) para que também exiba a raça do cachorro.
- Crie o método brincar() que deverá verificar se o nível de energia do cachorro é maior que 10, se for, diminuir o valor de nivelEnergia em 10 unidades e exibir a seguinte mensagem: “[Nome do cachorro] está brincando!”. Caso contrário, exibir a mensagem: “[Nome do cachorro] está muito cansado para brincar.”.

### C. Classe Gato:

Crie uma classe Gato que herda da classe Animal.

- Os atributos de nome, idade e nivelEnergia devem ser herdados de Animal.
- Os métodos *setters* devem ser herdados de Animal.
- O método dormir deve ser herdado de Animal.
- Adicione o atributo corPelagem: String que representa a cor da pelagem do gato.
- Construtor Gato(string nome, int idade, int nivelEnergia, string corPelagem).
- Adicione o método setCorPelagem(*string cor*) que define a cor de pelagem do gato.
- Sobrescreva o método emitirSom( ) para exibir a mensagem: “Miau”.
- Sobrescreva o método mostrarInfo() para que também exiba a cor de pelagem do gato.
- Crie o método ronronar( ) que deverá verificar se o nível de energia do gato é maior do que 5 unidades, se for, diminuir o valor de nivelEnergia em 5 unidades e exibir a seguinte mensagem: “[Nome do gato] está

ronronando: purr purr.”. Caso contrário, exibir a mensagem: “[Nome do gato] está muito cansado para ronronar”.

#### **D. Classe Passaro:**

Crie uma classe Pássaro que herda da classe Animal.

- Os atributos de nome, idade e nivelEnergia devem ser herdados de Animal.
- Os métodos setters devem ser herdados de Animal.
- O método dormir deve ser herdado de Animal.
- Adicione o atributo corPenas: String que representa a cor das penas do pássaro.
- Construtor Passaro(string nome, int idade, int nivelEnergia, string corPenas).
- Adicione o método setCorPenas(string cor) que define a cor das penas do pássaro.
- Sobrescreva o método emitirSom() para exibir a mensagem: “Piu piu”.
- Sobrescreva o método mostrarInfo() para que também exiba a cor das penas do pássaro.
- Crie o método voar() que deverá verificar se o nível de energia do pássaro é maior do que 15 unidades, se for, diminuir o valor de nivelEnergia em 15 unidades e exibir a seguinte mensagem: “[Nome do pássaro] está voando alto!”. Caso contrário, exibir a mensagem: “[Nome do pássaro] está muito cansado para voar”.

#### **E. Classe Peixe:**

Crie uma classe Peixe que herda da classe Animal.

- Os atributos de nome, idade e nivelEnergia devem ser herdados de Animal.
- Os métodos setters devem ser herdados de Animal.
- O método dormir deve ser herdado de Animal.
- Adicione o atributo especie: String que representa a espécie do peixe de estimação.
- Construtor Peixe(string nome, int idade, int nivelEnergia, string especie).

- Adicione o método `setEspecie(string especie)` que define a espécie do peixe.
- Sobrescreva o método `emitirSom()` para exibir a mensagem: “Glub glub”.
- Sobrescreva o método `mostrarInfo()` para que também exiba a espécie do peixe.
- Crie o método `nadar()` que deverá verificar se o nível de energia do peixe é maior do que 5 unidades, se for, diminuir o valor de `nivelEnergia` em 5 unidades e exibir a seguinte

## F. Main

Na função main:

- Crie uma instância da classe `Cachorro` chamada `c1` usando o construtor e defina o nome, a idade, o nível de energia e a raça como quiser. Chame os métodos `emitirSom()`, `brincar()` e `mostrarInfo()` para essa instância.
- Crie uma instância da classe `Gato` chamada `g1`, usando o construtor e defina o nome, a idade, o nível de energia e a cor da pelagem como quiser. Chame os métodos `emitirSom()`, `brincar()` e `mostrarInfo()` para essa instância.
- Crie uma instância da classe `Pássaro` chamada `p1`, usando o construtor e defina o nome, a idade, o nível de energia e a cor das penas como quiser. Chame os métodos `emitirSom()`, `brincar()` e `mostrarInfo()` para essa instância.
- Crie uma instância da classe `Peixe` chamada `pe1`. Defina o nome, a idade, o nível de energia e a espécie como quiser. Chame os métodos `setNome()`, `setIdade()`, `setNivelEnergia()`, `setEspecie()`, `emitirSom()`, `nadar()` e `mostrarInfo()` para essa instância.

## Exercício 2

Crie um novo diretório para armazenar os arquivos referentes ao exercício 2.

Crie as seguintes classes. Para cada classe, crie o arquivo de cabeçalho e implementação. Depois, crie a função main que integrará todas as classes.

#### **A. Classe Terrestre:**

Crie uma classe Terrestre com os seguintes atributos e métodos:

- velocidadeMaxima: Inteiro que representa a velocidade máxima do animal terrestre em km/h.
- Construtor Terrestre(int velocidade).
- setVelocidadeMaxima(int velocidade): Define a velocidade máxima do animal terrestre.

#### **B. Classe Aquatico:**

Crie uma classe Aquatico com os seguintes atributos e métodos:

- profundidadeMaxima: Inteiro que representa a profundidade máxima que o animal aquático pode atingir em metros.
- Construtor Aquatico(int profundidadeMaxima).
- setProfundidadeMaxima(int profundidade): Define a profundidade máxima do animal aquático.

#### **C. Classe Anfíbio:**

Crie uma classe Anfíbio que herda das classes Terrestre e Aquatico.

- Os atributos e métodos de Terrestre e Aquatico devem ser herdados ao mesmo tempo.
- Adicione o atributo tipoPele: String que representa o tipo de pele do anfíbio (por exemplo, "Lisa" ou "Rugosa").
- Construtor Anfíbio(int velocidadeMaxima, int profundidadeMaxima, tipoPele).
- Adicione o método setTipoPele(string tipo) que define o tipo de pele do anfíbio.
- Adicione o método mostrarInfo( ) para que também exiba todas as informações de Anfíbio (velocidadeMaxima, profundidadeMaxima e tipoPele).

#### **D. Main**

Na função main:

Crie uma instância da classe Anfíbio chamada a1 usando o construtor e defina a velocidade máxima, a profundidade máxima e o tipo de pele.. Defina a velocidade máxima, a profundidade máxima e o tipo de pele como quiser. Chame o método mostrarInfo( ) para essa instância.

### **Exercício 3**

Crie um novo diretório para armazenar os arquivos referentes ao exercício 3.

Crie as seguintes classes. Para cada classe, crie o arquivo de cabeçalho e implementação. Depois, crie a função main que integrará todas as classes.

#### **A. Classe Transacao:**

Crie uma classe Transacao com os seguintes atributos e métodos:

- id: Inteiro que representa o identificador único da transação.
- valor: Double que representa o valor da transação.
- data: String que representa a data da transação no formato ("XX/XX/XXXX").
- hora: String que representa a hora da transação no formato ("XX:XX:XX").
- ConstrutorTransacao(Int id, double valor, string data, string hora).
- setTransacao(int id, double valor, string data, string hora): Define os detalhes da transação.
- processar( ): Método virtual puro que será implementado nas subclasses para processar diferentes tipos de transações.
- mostrarDetalhes( ): Exibe a mensagem: "Transacao - ID: [id], Valor: [valor], Data: [data], Hora: [hora]".

#### **Observação!**

Este exercício aborda o conceito de herança híbrida/diamante, que é uma situação um pouco mais delicada na programação orientada a objetos. Pesquisa sobre o problema e como a herança virtual pode ser usada para resolvê-lo.

### **B. Classe PagamentoOnline:**

Crie uma classe PagamentoOnline que herda virtualmente da classe Transacao:

- Os atributos id, valor, data e hora devem ser herdados de Transacao.
- Os métodos setters devem ser herdados de Transacao.
- Construtor PagamentoOnline(int id, double valor, string data, string hora).
- Sobrescreva o método processar para exibir a mensagem "Processando pagamento online".
- Sobrescreva o método mostrarDetalhes para exibir a mensagem: "Pagamento Online - ID: [id], Valor: [valor], Data: [data], Hora: [hora]".

### **C. Classe PagamentoCartao:**

Crie uma classe PagamentoCartao que herda virtualmente da classe Transacao.

- Os atributos id, valor, data e hora devem ser herdados de Transacao.
- Os métodos setters devem ser herdados de Transacao.
- Construtor PagamentoCartao(int id, double valor, string data, string hora).
- Sobrescreva o método processar() para exibir a mensagem: "Processando pagamento com cartão".
- Sobrescreva o método mostrarDetalhes() para exibir detalhes específicos do pagamento com cartão, como: "Pagamento com cartão - ID: [id], Valor: [valor], Data: [data], Hora: [hora]".

### **D. Classe PagamentoCartaoOnline:**

Crie uma classe PagamentoCartaoOnline que herda de PagamentoOnline e PagamentoCartao:

- Os atributos id, valor, data e hora devem ser herdados de PagamentoOnline e PagamentoCartao.
- Construtor PagamentoCartaoOnline(int id, double valor, string data, string hora).

- Sobrescreva o método `processar()` para exibir a mensagem "Processando pagamento com cartão online".
- Sobrescreva o método `mostrarDetalhes()` para exibir detalhes específicos do pagamento com cartão online, como: "Pagamento com cartão online - ID: [id], Valor: [valor], Data: [data], Hora: [hora]".

#### **E. Main:**

Na função `main`:

Crie uma instância da classe `PagamentoCartaoOnline` chamada `pco` usando o construtor e defina o `id`, o `valor`, `data` e `hora`). Chame os métodos `processar()` e `mostrarDetalhes` para a instância `pco`.