

Estes exercícios envolvem: *criação de classes, definição de atributos, tipos de dados, criação de métodos, manipulação dos tipos de retorno, expressões aritméticas simples, lógica de programação, associação entre classes, sobrecarga de métodos, classe de teste, método main.*

Orientação inicial: Abra o BlueJ, crie um novo projeto e faça o que pede.

Exercício 1

Crie a classe Time que possui um nome, uma categoria (que receberá valores como vôlei, futebol, basquete, etc.), um ano de criação, uma quantidade de sócios e um histórico.

a) quando um time é criado (construtor), devem ser informados, obrigatoriamente, o nome do time e a categoria

b) crie os métodos de acesso SET dos atributos: ano de criação, quantidade de sócios e históricos

c) crie os métodos de acesso GET para todos os atributos

d) crie um método chamado “calculalddade”, que recebe um atributo inteiro referente a um determinado ano e retorna a idade do time (em relação ao ano recebido por parâmetro)

e) crie um método chamado “adicionaSocios” que recebe como parâmetro de entrada uma quantidade de novos sócios, e atualiza a quantidade de sócios (ou seja, o número de sócios agora deve ser o número de sócios existentes mais o número de sócios novos). Este método também imprime na tela a seguinte mensagem: “Foram adicionados mais X sócios.”

f) crie um método chamado “imprimeInfo” que imprime na tela (de maneira legível e organizada) todas as informações do time, inclusive a idade do time

g) crie um método chamado “contagemRegressiva”, que imprime na tela a seguinte mensagem: “Faltam X dias para o centenário do time.”. O valor X deve ser calculado utilizando a idade do time.

Exercício 2

Agora, altere a classe Time e crie mais um método construtor. Este novo construtor não recebe parâmetros.

- a) crie os métodos de acesso SET para os atributos nome e categoria
- b) crie o método chamado “imprimeMensagem” que recebe uma mensagem X e imprime esta mensagem na tela da seguinte forma: “Mensagem recebida: X”
- c) crie um novo método chamado “imprimeMensagem” (sobrecarga), que não recebe parâmetros e imprime na tela a mensagem: “Nenhuma mensagem recebida. Utilize o outro método”
- d) crie um novo método chamado “imprimeMensagem” que recebe duas mensagens X e Y como parâmetro de entrada e imprime na tela da seguinte forma:

Mensagem 1 recebida: X

Mensagem 2 recebida: Y

Exercício 3

No mesmo projeto, crie a classe Torcedor. Um torcedor possui um nome, um sexo e uma idade.

- a) quando um torcedor é criado, devem ser informados todos os atributos (nome, sexo e idade)
- b) crie os métodos GET para todos os atributos
- c) crie um método “imprimeInfo” que imprime todas as informações do torcedor

Exercício 4

Agora, altere a classe Torcedor. Crie os seguintes métodos:

- a) crie mais um construtor que recebe apenas o nome e a idade do torcedor
- b) crie o método de acesso SET para todos os atributos
- c) crie um construtor que não recebe parâmetros de entrada

Exercício 5

Agora, altere a classe Torcedor da seguinte maneira:

- a) um torcedor tem um time

b) crie um novo construtor que recebe um nome, um sexo, uma idade e um time (um objeto do tipo Time)

c) crie os métodos de acesso SET e GET para o atributo time

d) crie um método “imprimeTime” que imprime na tela a mensagem: “Torcedor X torce para o time Y” (onde X é o nome do torcedor e Y é o nome do time)

e) crie o método “enviaMensagem”, que recebe uma String X, e acessa o método “imprimeMensagem” do seu time, repassando este parâmetro

f) crie o método “enviaMensagem”, que não recebe parâmetros e acessa o método “imprimeMensagem” do seu time

g) crie o método “enviaMensagem” que recebe duas Strings (X e Y) e acessa o método “imprimeMensagem” do seu time

Exercício 6

Crie uma classe chamada Teste. Nesta classe, você vai colocar o método principal de um programa Java, o método main. Crie esta classe exatamente como segue:

```
public class Teste {  
  
    public static void main(String args[]){  
  
        //aqui você vai colocar as instruções  
  
    }  
  
}
```

Exercício 7

Agora, complete a classe Teste com as seguintes informações (na parte reservada para as instruções, onde está comentado no código acima):

a) crie um novo Time passando um nome e uma categoria. Uma possível linha de código para isto seria: Time time1 = new Time(“Grêmio”, “Futebol”);

b) crie agora um novo time chamado time2. Este time será criado com a utilização do construtor sem parâmetros de entrada

c) dê o nome Ulbra para este time, com a linha de comando time.setNome(“Ulbra”);

d) depois disto, defina a categoria Vôlei para o time2

- e) agora, defina um número de sócios para o time1, por exemplo, 300
- f) defina o histórico do time1, por exemplo, “Sempre foi o melhor time do mundo”
- g) defina um ano de criação para o time1, por exemplo, 1987
- h) crie agora um torcedor chamado Vandressa, do sexo feminino, com 27 anos
- i) crie um torcedor que não recebe parâmetro de entrada (veja qual construtor deve ser utilizado para tal)
- j) defina o nome deste torcedor como Ribelton Cruz
- k) defina que Ribelton é do sexo masculino
- l) agora, diga que Wandressa torce para o Grêmio
- m) o time de Ribelton, por sua vez, é a Ulbra
- n) imprima quanto tempo falta para o time1 chegar ao seu centenário
- o) imprima a idade do time2
- p) faça com que Ribelton envie 2 mensagens para o seu time
- q) crie mais 3 torcedores: André com 32 anos, Richeli com 22 anos e Petrúcio com 54 anos
- r) crie mais um time agora. O time é o São José, de Basquete, e possui como torcedores o André, o Richeli e o Petrúcio
- s) altere o nome do torcedor Richeli para Rochele

Exercício 8

Faça o diagrama de classes UML referente às classes criadas