

PROGRAMAÇÃO ORIENTADA A OBJETOS I – SISTEMAS DE INFORMAÇÃO

Lista de exercícios 5 – Classes, objetos e encapsulamento

Prof. Jean Eduardo Glazar

EXERCÍCIO 1:

Etapa 1: Crie uma classe chamada **Ponto**, que represente uma coordenada com três atributos: **X** (tipo float), **Y** (tipo float) e **cor** (tipo Color do java). Sua classe deve ter um **construtor** que inicializa os três atributos, passados como parâmetro. Crie um método para alterar o valor da coordenada **X** e **Y**. Descreva um método **set** e um **get** para o atributo cor. Ao alterar os atributos com o método **set**, verifique se a nova informação é correta. Faça uma função para **mostrar** os dados da classe Ponto. Faça uma classe (Principal) com o método **main**, para testar todas as funções da classe Ponto. Crie um objeto do tipo Ponto, inicialize com os valores e imprima. Depois altere as informações e imprima novamente.

Etapa 2: Crie uma classe para representar uma **Reta**. Uma reta é composta por dois pontos, ou seja, dois objetos da classe **Ponto** acima. Uma Reta também possui uma **cor**, no mesmo esquema da cor do ponto. Sua classe deve ter um **construtor** que inicializa os dois pontos mais a cor da reta, passados como parâmetro. Crie uma função para alterar o valor da coordenada **X** e **Y**, de cada ponto. Descreva um método **set** e um **get** para o atributo cor. Ao alterar os atributos com o método **set**, verifique se a nova informação é correta. Faça um método para **calcular** e retornar o comprimento da reta (distância entre dois pontos). Faça outro método para **mostrar** os dados da classe Reta, ou seja, imprimir os dois pontos, seu comprimento e a cor da reta.

Altere a classe (Principal) com o método **main**, para testar todas as funções da classe Reta. Crie um objeto do tipo Reta, inicialize com os valores e imprima. Depois altere as informações e imprima novamente.

Etapa 3: Crie uma classe para representar um **Círculo**. Um círculo é composto por um ponto que define o **centro**, ou seja, um objeto da classe **Ponto** acima, e um **raio** (tipo double). Um círculo também possui uma **cor**, no mesmo esquema da cor do ponto. Sua classe deve ter um **construtor** que inicializa o objeto ponto, o raio e a cor do círculo, passados como parâmetro. Crie uma função para alterar (**set**) o valor da coordenada **X** e **Y**, do ponto do centro. Descreva métodos **set** e um **get** para o atributo cor e para o raio (lembre-se que não existe raio negativo). Ao alterar os atributos com o método **set**, verifique se a nova informação é correta. Faça um método para **calcular** e retornar a área do círculo (a área do círculo é dada pela fórmula: $\text{área} = \pi \cdot \text{raio}^2$). Faça outro método para **mostrar** os dados da classe Círculo, ou seja, imprimir o ponto do centro, o raio, a área e a cor do círculo.

Altere a classe (Principal) com o método **main**, para testar todas as funções da classe Círculo. Crie um objeto do tipo Círculo, inicialize com os valores e imprima. Depois altere as informações e imprima novamente.

Etapa 4: Crie uma classe para representar um **Retângulo**. É necessário armazenar apenas os dois pontos que estão nas quinas em diagonal. Armazene também a **cor**, no mesmo esquema da cor do ponto. Sua classe deve ter um **construtor** que inicializa os dois pontos mais a cor do retângulo, passados como parâmetro. Crie uma função para alterar o valor da coordenada **X** e **Y**, de cada ponto. Descreva um método **set** e um **get** para o atributo cor. Ao alterar os atributos com o método **set**, verifique se a nova informação é correta. Faça um método para **calcular** e retornar a área do retângulo (comprimento x largura). Faça outro método para **mostrar** os dados da classe Retângulo, ou seja, imprimir os dois pontos, sua área e a cor.

Altere a classe (Principal) com o método **main**, para testar todas as funções da classe Reta. Crie um objeto do tipo Reta, inicialize com os valores e imprima. Depois altere as informações e imprima novamente.

EXERCÍCIO 2:

Etapa 1: Crie uma classe chamada **Data** com três atributos: **dia** (tipo int), **mês** (tipo int) e **ano** (tipo int). Sua classe deve ter um **construtor** que inicializa os três atributos, passados como parâmetro, e valida se a data está correta. Descreva um método **set** e um **get** para cada atributo (encapsulamento). Ao alterar os atributos com o método **set**, verifique se a nova informação é correta. Faça três funções para **imprimir** a data: uma que imprima no formato **dd/mm/aaaa** (por exemplo: 30/08/2018); outra que imprima no formato americano **aaaa/mm/dd** (por exemplo: 2018/08/30); e outra que imprima no formato **dd de nome do mês de ano** (por exemplo: 07 de setembro de 2018). Para pegar o nome do mês, utilize a estrutura **switch** ou um **array** com os nomes dos meses. Faça outra classe (Principal) com o método **main**, para testar todas as funções da classe Data.

Etapa 2: Crie outra classe para representar um **Produto**, com os atributos **código**, **descrição**, **preço**, **quantidade em estoque** e **data de validade** (que deverá ser um objeto da classe **Data** criada acima). Sua classe deve ter um **construtor** que inicializa os atributos (não esqueça de inicializar o objeto data). O código e a quantidade não podem ser negativos nem iguais a zero. Descreva um método **set** e um **get** para cada atributo. Ao alterar os atributos com o método **set**, verifique se a nova informação é correta. Crie outro método para representar a venda de um produto, onde é passado a quantidade vendida como parâmetro e diminuído do estoque (verifique se existe quantidade em estoque para vender). Crie outro método para verificar se a quantidade do produto está abaixo de um limite mínimo, que deve ser passado como parâmetro. Faça uma função para imprimir os dados do produto. Altere a classe (Principal) com o método **main**, para testar todas as funções da classe Produto.

Etapa 3: Na classe (Principal) no método **main**, declare uma lista (classe **List**) para armazenar vários objetos do tipo **Produto**. Implemente o processo para **ler** os dados de vários produtos e inseri-lo na lista (use o método **add** da lista). Ao final, **listar** os dados de todos os produtos que estão abaixo do limite mínimo (peça ao usuário para informar esse limite e use a função criada na classe Produto).