**DESCRIÇÃO**

A atividade consiste em resolver os exercícios abaixo utilizando o paradigma orientado a objetos, na linguagem Java.

1. Pesquise sobre interfaces e elabore um exemplo que utilize este recurso.
2. Implemente a classe Senha, que deve receber uma senha, simulando um sistema de autenticação.
3. Um método entraSenha deve ser implementado para que o usuário tente efetuar a autenticação. Neste método deve ser exibida uma mensagem informando se a senha estava correta ou incorreta. Caso o usuário erre três vezes seguidas a senha, esta deve ser bloqueada, não sendo mais possível efetuar a autenticação. Caso acerte, o usuário volta a ter três tentativas.
4. Implemente também o método trocarSenha. Este método deve receber a senha antiga e a senha nova, e não poderá ser acessado caso a senha já esteja bloqueada. Caso o usuário erre a senha neste método, o erro deve ser contabilizado para fins de bloqueio da senha. Caso o usuário acerte a senha, seu número de tentativas para autenticação deve voltar a ser três.

**Observação**: se achar necessário, crie outros métodos.

1. Implemente a classe descrita abaixo:

Classe: Porta

Atributos: aberta, cor, dimensaoX, dimensaoY, dimensaoZ

Métodos: public void abre()

public void fecha()

public void pinta(String cor)

public boolean estaAberta()

1. Aproveitando a classe Porta, crie a classe Casa.

Atributos:

* três portas (porta1, porta2 e porta3);
* cor.

Métodos:

* public void pinta(String cor)
* public int quantasPortasEstaoAbertas()
* public String getCor()
* public Porta getPorta1()
* public Porta getPorta2()
* public Porta getPorta3()

Além disso, crie também a classe TestaCasa, onde você deve criar um objeto do tipo Casa e efetuar chamadas a seus métodos.

1. Implemente a classe Calculadora, que deve implementar os métodos somar, subtrair, dividir e multiplicar. Implemente também a classe CalculadoraCientifica, que herda de Calculadora. Esta classe deve conter o método potencia. Crie a classe TestaCalculadoras, onde você deve criar instância(s) de Calculadora e CalculadoraCientifica e realizar chamadas a todos os métodos.
2. Implemente a classe Data. Esta classe deve receber três atributos: dia, mês e ano. Você deve implementar alguma forma de verificar se a data informada é válida. Crie um método para imprimir a data (por exemplo: se dia = 1, mes = 4 e ano = 1900, deve imprimir algo como 1/4/1900). Crie também um método proximoDia(), que deve atualizar a data para o dia seguinte. Os atributos devem ser inicializados através do construtor. Crie objetos do tipo Data em uma classe de teste. **(EXTRA)** Crie um construtor sem parâmetros que, quando invocado, deve criar um objeto do tipo Data com a data atual do sistema operacional.
3. Considere a existência da classe Lampada e os códigos da tabela abaixo. O código da esquerda (Código 1) foi inserido na função main em uma classe chamada TesteDesafio1, e o código da direita (Código 2) foi inserido na função main em uma classe chamada TestaDesafio2. Com base nas informações e nos códigos abaixo, faça o que se pede:

| **Código 1** | **Código 2** |
| --- | --- |
| Lampada lamp1 = new Lampada();  Lampada lamp2 = lamp1;  int i = 10;  if (lamp1 == lamp2) {  System.out.println(i++);  } | Lampada lamp1 = new Lampada();  lamp1.ligar();  lamp1.trocarTipo(“LED”);  Lampada lamp2 = new Lampada();  lamp2.ligar();  lamp2.trocarTipo(“LED”);  int i = 10;  if (lamp1 == lamp2) {  System.out.println(i++);  } |

1. O que é exibido na tela quando o código da esquerda é executado? Por quê?
2. O que é exibido na tela quando o código da direita é executado? Por quê?
3. Qual é o valor de i ao final da execução do código da esquerda?
4. Qual é o valor de i ao final da execução do código da direita?
5. O que acontece se acrescentarmos as duas linhas abaixo no código da esquerda? E no da direita? Por quê?

| lamp1.trocarTipo(“Halógena”);  lamp2.mostrarInformacoesGerais(); |
| --- |

**PESO DA AVALIAÇÃO**

Notas de aula.

**OBSERVAÇÕES**

* Plágio = **ZERO (inclui cópia ou simples alteração de trabalho de colegas)**