

Exercícios:

1. Faça um programa que calcule a área de uma figura geométrica. Aceite quatro tipos de figura geométrica: quadrado, retângulo, triângulo e círculo. Use herança e polimorfismo.
2. Desenvolver uma classe Agenda que pode armazenar 10 pessoas e que seja capaz de realizar as seguintes operações:

```
void armazenaPessoa(String nome, int idade, float altura);  
void removePessoa(String nome);  
int buscaPessoa(String nome); // informa em que posição da agenda está a  
pessoa  
void imprimeAgenda(); // imprime os dados de todas as pessoas da agenda  
void imprimePessoa(int index); // imprime os dados da pessoa que está na  
posição "i" da agenda.
```

3. Desenvolver uma classe denominada Elevador para armazenar as informações de um elevador dentro de um prédio. A classe deve armazenar o andar atual (térreo = 0), total de andares no prédio (desconsiderando o térreo), capacidade do elevador e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:
 - inicializa: deve receber como parâmetros do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazio).
 - entra: acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço).
 - sai: remover uma pessoa do elevador (só deve remover se houver alguém dentro dele).
 - sobe: para subir um andar (não deve subir se já estiver no último andar).
 - desce: para descer um andar (não deve descer se já estiver no térreo).

Encapsular todos os atributos da classe.

4. Criar uma classe Televisão e uma classe ControleRemoto que pode controlar o volume e trocar os canais da televisão. O controle de volume permite:
 - Aumentar ou diminuir a potência do volume do som em uma unidade de cada vez.
 - Aumentar ou diminuir o número do canal em uma unidade.

- Trocar para um canal indicado.
- Consultar o valor do volume de som e o canal selecionado.

Utilizar interface para as operações na Televisão.

5. Qual a diferença entre uma exceção (subclasses de Exception) e um erro (subclasses de Error)?
6. Criar uma lista de 100 inteiros gerados aleatoriamente. Gerar uma segunda lista mostrando os inteiros reordenados.
7. A classe Avaliação recebe como atributos o nome do aluno, a disciplina e a lista de notas. Desenvolver os métodos: registrar nota e calcular média. Proteger esses métodos de qualquer tipo de exceção que pode ocorrer durante a interação com outras classes.
8. Implementar uma classe que gerencia uma pilha de caracteres preparada para atender duas threads: uma que coloca caracteres na pilha (Produtor) e outra que retira caracteres da pilha (Consumidor). O processo deve funcionar de tal maneira que o Produto só insere caracteres na pilha se ela não estiver cheia. Se ela estiver cheia, ele aguarda que o Consumidor retire alguma coisa da pilha para que vague uma posição. Em contrapartida, o Consumidor verifica se há elementos na pilha. Se ela estiver vazia, ele aguarda que algo seja colocado para que ele possa consumir.

Escreva um programa de teste no qual o Produtor gere uma sequência de 1000 números aleatórios para colocar na pilha. O consumidor deve imprimir o que ele retira na tela.

9. Escreva um programa para simular o funcionamento de um restaurante. Esse programa deve definir duas classes, cujos objetos funcionarão em threads independentes, conforme descrito a seguir:

Caixa – solicita pelo teclado uma String que contém o nome do prato comprado e coloca na lista de pedidos pendentes. Esse processo se repete indefinidamente. Cada item da lista possui o nome de um único prato. Como só existe espaço para 20 pedido na lista de pedidos pendentes, ao alcançar este número, o caixa deve dar uma pausa nas vendas e aguardar que seja retirado pedidos da lista para poder prosseguir.

Cozinha – retira pedidos da lista de pedidos pendentes e aciona o seu preparo. Por uma questão de eficiência, todos os pedidos para o mesmo prato devem ser feitos em conjunto. Por este motivo, a cada rodada, o objeto Cozinha deve retirar da lista de pedidos pendentes todos os pedidos com o mesmo nome do prato e acionar o método `fazPrazo` que simula a produção de pratos. Este processo se repete indefinidamente.

`fazPrato`: recebe como parâmetros o nome do prato e a quantidade de pratos a ser produzida, e providencia a sua produção. Esse método produz um único tipo de prato, cada vez que é chamado.

Considere que este é um método estático da classe `Cozinheiro`. As classes `Caixa` e `Cozinha` devem ser construídas de forma que sejam executadas em paralelo. Escreva um programa principal que instancie e execute as duas classes (`Caixa` e `Cozinha`) em threads independentes.

Podem ser implementadas classes auxiliares que sejam necessárias no processo.

10. Implementar um programa que leia um número “n” informado pelo usuário e diga quantos números primos há entre 0 e “n”. Esse programa deve rodar em 2 threads, de forma que o esforço computacional seja uniformemente dividido entre as threads.
11. Para que serve o modificador `synchronized`? Em que situações ele deve ser usado? Por que não usar em todos os métodos do programa?
12. Qual a diferença entre o método `sleep()` e o método `join()` da classe `Thread`?