UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIAS DE COMPUTAÇÃO
SCC0204/SCC0504 – Programação Orientada a Objetos
Prof. José Fernando Rodrigues Júnior

Lista 5 – Classe Abstrata e Interface

1. Imagine um sistema operacional. Este sistema usa vários dispositivos de diferentes naturezas, como vídeo, impressora, mouse, e teclado. São todos dispositivos distintos, mas que para trabalhar com um dado sistema operacional devem ter as seguintes funcionalidades em seus drivers:

- ligar/desligar;
- checar status, o que retorna um número indicado a condição do dispositivo;
- calibrar.

a) Qual solução de projeto você adotaria para que os desenvolvedores de dispositivos pudessem desenvolver dispositivos para este sistema sem que o código do sistema fosse revelado?
Escreva o correspondente código, incluindo um método main que demonstra o uso polimórfico da sua solução.

b) Considere o caso em que o driver de um determinado tipo de dispositivo, além de satisfazer aos requisitos do sistema operacional considerado, também será usado para a criação de toda ***uma família*** de drivers que tem funcionalidades comuns e que possuem especificidades em cada uma de sua variações. Escolha um dos tipos de dispositivo e escreva o correspondente código (apenas *println*), incluindo um método main que demonstra o uso polimórfico da sua solução.

Obs.: os métodos não precisam ter funcionalidades de fato, apenas println.

2. Considere o seguinte código:

```
class System{
      public void execute(Executable e){
            int iStatus = -1;
            if(e.initialize( )){
                  this.setLoad(e.LOAD *  this.timeUnits);
                  iStatus = e.execute( );
            }
            e.close( );
      }
}
```

Escreva o código correspondente à definição do contrato **Executable** usando:
a) Classe abstrata
b) Interface

3. Implemente um sistema de automação residencial com base nas seguintes especificações.
Crie uma interface chamada `ControlavelPorVoz` que possui o método `void executarComandoVoz(String comando)`.

Crie uma superclasse abstrata chamada `Dispositivo`, que contém os atributos `String nome` e `boolean ligado` e os métodos:

```
void ligar() → liga o dispositivo
void desligar() → desliga o dispositivo
boolean estaLigado() → retorna se o dispositivo está ligado
void status() → imprime o nome do dispositivo e seu estado atual (ligado/desligado)
abstract void descricao() → descreve o que o dispositivo faz
```

Crie as seguintes subclasses de `Dispositivo`:

- `Lampada` -- Não implementa a interface `ControlavelPorVoz`.
- `CaixaSom` -- Implementa `ControlavelPorVoz`.
  Ao receber o comando "tocar música" e estando ligada, deve imprimir "Música tocando...".
  Caso contrário, imprime "Caixa desligada ou comando inválido."

- `CortinaAutomatica` -- Implementa `ControlavelPorVoz`.
  Ao receber o comando "abrir" e estando ligada, imprime "Cortina abrindo...".
  Ao receber o comando "fechar" e estando ligada, imprime "Cortina fechando...".
  Caso contrário, imprime "Cortina desligada ou comando inválido."

Na função `main`:
Crie uma lista com objetos de cada tipo (`Lampada`, `CaixaSom`, `CortinaAutomatica`), todos como instâncias de `Dispositivo`. Ligue todos os dispositivos.

Para cada item da lista que seja controlável por voz (pesquise e use o recurso `instanceof`), execute um comando de voz. Ao final, chame o método `status()` para todos os dispositivos da lista.

4. Considere uma classe `TimeFutebol` a qual guarda informações de times de futebol que participam de um dado campeonato, as informações são: número de vitórias, derrotas, e empates, número de gols marcados e de gols sofridos, número de cartões amarelos e vermelhos. Crie um método que retorna a quantidade de pontos de um dado time = vitórias*3 + empates*1.

Em seguida, usando a interface `Comparable`, ordene os times de acordo com os critérios de desempate da federação paulista de futebol (http://pt.wikipedia.org/wiki/Campeonato_Paulista_de_Futebol_de_2014_-_S%C3%A9rie_A1#Crit.C3.A9rios_de_desempate), inclusive o item 6 (Sorteio).

Escreva um programa principal (main) com vários times cujos atributos irão acionar cada um dos critérios de desempate para ordenação.