

Aluno: Brendon Hudson Cardoso
SCC0504 - Programação Orientada a Objetos

nºUSP: 10284879
Profº.: Luiz Eduardo V. Silva

Exercício 1

- a) Adotaria o desenvolvimento de um contrato para todos os desenvolvedores de dispositivos os desenvolverem de forma que sejam compatíveis com o Sistema Operacional e sem a necessidade de abrir o código fonte de cada um, contrato esse que em Java pode ser codificado como “Interfaces”, nela, explicitada todas as assinaturas dos métodos que serão utilizados pelo Sistema Operacional para manusear cada dispositivo, com os retornos específicos e parâmetros necessários. Assim sendo, cada desenvolvedor implementará a interface “Device” para seu dispositivo da maneira que queira, desde que, implemente todos os métodos assinados na Interface.

Um exemplo para a interface:

```
public interface Device {  
    void turnOn();  
    void turnOff();  
    int statusCheck();  
    void calibrate();  
}
```

```
public class Video implements Device {  
    void turnOn() {  
        // implementação...  
    }  
  
    void turnOff() {  
        // implementação...  
    }  
  
    int statusCheck() {  
        // implementação...  
    }  
  
    void calibrate() {  
        // implementação...  
    }  
}
```

// O mesmo para a classe dos outros dispositivos... Mouse, Keyboard, Printer.

- b) No caso que seja necessário desenvolver um driver para toda uma família de dispositivos semelhantes, como uma família de impressoras, onde todas as impressoras possuem a mesma “arquitetura” (atributos e métodos) para manuseá-las, porém, com diferentes “organizações”, como tipo de cartucho, consumo de energia, quantidade de memória, entre outros, adotaria a seguinte implementação em java para que o Sistema Operacional ainda possa se comunicar com qualquer um dos dispositivos da família da mesma forma, implementaria uma classe abstrata para a família de impressoras ‘X’, por exemplo, classe que deverá implementar a interface “Device” da alternativa anterior, assim, cada impressora da família necessariamente deve herdar dessa classe abstrata ao ser implementada para poder ser instanciada como um objeto Device do Sistema Operacional.

Códigos de exemplo:

```
public abstract class PrinterFamilyX implements Device {
    boolean on;
    double InkLevel;
    int numberOfPrints;
    // demais campos que sejam comuns para as impressoras da família...

    void turnOn() {
        // implementação pode ser única para todas
    }

    void turnOff() {
        // implementação pode ser única para todas
    }

    /* Obrigatoriamente, os seguintes métodos (comportamentos) serão implementados
    na subclasse não abstrata desta classe, pois, são implementações que dependem
    da organização interna de cada impressora da família 'X':

    int statusCheck();

    void calibrate();
    */
}
```