***Importante***: Para item abaixo deve ser copiado trechos do código que cumprem o requisito e explicado, se não for aparente, o porquê o requisito é cumprido. Sejam bem explícitos. Deve ser indicado também o arquivo da classe em que está o trecho do código. Eu avaliarei o código do Github a partir desse documento para confirmá-lo e também para detectar possíveis erros. **Quem não seguir o que está indicado aqui, não terá o projeto avaliado e perderá a atividade.**

**Requisitos de implementação**

***Todos os atributos e funções membros devem estar relacionados a classe***

1. Pelo menos 4 atributos

Atributos da classe Computador.java

protected int HD;

protected int RAM;

protected int placaDeVideoRAM;

protected float processadorGHz;

protected String placaDeVideoModelo;

protected String processadorModelo;

protected boolean OnOff;//true para On e false para Off

1. Pelo menos 4 funções membros sem incluir get e set

Funções da classe Computador.java

@Override

public void ligar() {//Implementa da interface Device

if (!isOn()) {

System.out.println("Ligando o computador.");

this.OnOff = true;

Computador.numComputadoresLigador++;

System.out.println("Iniciando SO.");

this.iniciaSO();

} else {

System.out.println("O seu computador ja esta ligado.");

}

}

@Override

public void desligar() {//Implementa da interface Device

if (isOn()) {

System.out.println("Finalizando SO.");

System.out.println("Desligando o computador.");

this.OnOff = false;

Computador.numComputadoresLigador--;

} else {

System.out.println("Voce nao precisa desligar um computador que ja esta desligado.");

}

}

public void reiniciar() {

if (isOn()) {

this.desligar();

this.ligar();

} else {

System.out.println("Nao faz sentido reiniciar um computador que esta desligado.");

}

}

protected abstract void iniciaSO(); //função implementada apenas nas classes base PC e Mac

public abstract boolean executaPrograma(); //função implementada apenas nas classes base PC e Mac

**Requisitos de implementação**

1. Cinco classes: uma superclasse e duas subclasses, e duas classes relacionadas ao projeto

Superclasse: Computador.java

Subclasses: PC.java e Mac.java, ambas herdam de computador.

Classes relacionadas: Monitor e Teclado, ambas são usadas em PC e Mac.

1. Sempre usar o super para o máximo de reaproveitamento de código

Utilizado na sobrecarga em PC.java e Mac.java dos métodos ligar() e desligar()

1. Diagrama de classes (obrigatório salvar também o png do diagrama no gitHub)
2. Todos os atributos devem ser inicializados. Fez validação de dados?

Validação dos dados feita em todos os sets dos atributos da classe Computador.

1. Dois construtores

Construtores da classe Computador.java

public Computador() {

this.HD = 500;

this.RAM = 4;

this.placaDeVideoRAM = 1;

this.processadorGHz = 1.9f;

this.placaDeVideoModelo = "NVidia gtx650";

this.processadorModelo = "Intel core I5-4460T";

}

public Computador(int HD, int RAM, int VRAM, float processadorGHz, String placaDeVideoModelo, String processadorModelo) {

this.setHD(HD);

this.setRAM(RAM);

this.setPlacaDeVideoRAM(VRAM);

this.setProcessadorGHz(processadorGHz);

this.setPlacaDeVideoModelo(placaDeVideoModelo);

this.setProcessadorModelo(processadorModelo);

}

1. Um atributo static.

Atributo static da classe Coputador.java

protected static int numComputadoresLigador = 0;

Este atributo é utilizado para contar o numero de computadores ligados, sendo assim incrementado na função ligar() e decrementado na função desligar() ambas da classe Computador.java

1. Um atributo const static

Atributo static final(const) da classe Mac.java

private static final String SISTEMA\_OPERACIONAL = "MacOsX";

Este atributo guarda uma constante para o sistema operacional de todos os macs pois todos os macs tem o mesmo sistema operacional variando somente na versão, tendo esta uma variável dentro da classe Mac.java que guarda um valor do enum MacOsX, onde estão contidas as versões do SO MacOsX.

*Correta modelagem dos statics?*

1. Um array

O array está na classe Computador.java

protected String []programasPadroesInstalados = new String[3];

1. Método static – deve ser chamado no main
2. Sobrescrita de método: chamar dentro do método da classe. Usar o @override

Métodos sobrescritos na classe PC.java:

@Override

protected void iniciaSO() {

System.out.println("Iniciando o " + this);

System.out.println("Aguarde enquanto o SO e inicializado.");

}

@Override

public boolean executaPrograma() {

if (i < 0 || i >= programasPadroesInstalados.length) {

System.out.println("Index out of range.");

return false;

} else {

System.out.println("Executando " + programasPadroesInstalados[i]);

}

return true;

return true;

}

@Override

protected void instalarProgramasPadroes() {

this.programasPadroesInstalados[0] = "Notepad";

this.programasPadroesInstalados[1] = "Mozilla Firefox";

this.programasPadroesInstalados[2] = "Media Player";

}

Métodos sobrescritos na classe Mac.java:

@Override

protected void iniciaSO() {

System.out.println("Iniciando o " + this);

System.out.println("Aguarde enquanto o SO e inicializado.");

}

@Override

public boolean executaPrograma(int i) {

if (i < 0 || i >= programasPadroesInstalados.length) {

System.out.println("Index out of range.");

return false;

} else {

System.out.println("Executando " + programasPadroesInstalados[i]);

}

return true;

}

@Override

protected void instalarProgramasPadroes() {

this.programasPadroesInstalados[0] = "Editor de Texto";

this.programasPadroesInstalados[1] = "Opera";

this.programasPadroesInstalados[2] = "Media player";

}

1. Usar Protected acessando diretamente os atributos na classe derivada

Todos os atributos da classe Computador.java são protected, como pode ser observado no requisito 1

1. Usar suas classes no main

Opcionais que garantem pontos extras:

Trabalhar com ENUM

Enum em PC.java:

public enum SO {

UBUNTU, FEDORA, MINT, ZENWALK, WINDOWS\_XP, WINDOWS\_VISTA, WINDOWS\_7, WINDOWS\_8, WINDOWS\_8\_1

}

Enum em Mac.java:

public enum MacOsX {

CHEETAH, PUMA, JAGUAR, PANTHER, TIGER, LEOPARD, SNOW\_LEOPARD, LION, MOUNTAIN\_LION, MAVERICKS, YOSEMITE

}

Trabalhar com pacotes