Programação Orientada a Objeto (ICP239)

Prof. Ronald Chiesse de Souza – IC/UFRJ

Segunda Prova – 18/06/2024

Questão 1)

Uma empresa de tecnologia possui um servidor físico com Configuração de grande porte, e oferece a seus clientes máquinas virtuais (VMs) Linux e Windows, alocadas com os recursos disponíveis deste servidor. Considere as classes abaixo (que modelam o gerenciamento do sistema) e em seguida responda às perguntas:

```
public class Configuracao {
    //Freq. de cada núcleo, em GHz:
    public final double clock;
    public final int n;  //Total de núcleos
    public final int r;  // Total de RAM, Em GB
    public Configuracao(double ck, int n, int r){
        clock = ck; this.n = n; this.r = r;
    }
}
```

```
public class Alocacao {
    public int n; //NÚCLEOS
    public int r; // RAM, em GB
    public Alocacao(int n, int r){
        this.n = n;
        this.r = r;
    }
}
```

```
public abstract class VM {
    public final String so;
    public final Alocacao a;
    //Construtor da Máquina Virtual (VM):
    public VM(Alocacao meusRecursos, String s){
        a = meusRecursos;
        so = s;
    }
}
```

```
public class Servidor {
    private final Configuracao c;
    private Alocacao a;
    private ArrayList<VM> vms;
    public Servidor(Configuracao cfg){
        c = cfg;
        a = new Alocacao(0, 0);
        vms = new ArrayList<>();
    }
    public void alocar(VM vm) {
        a.n += vm.a.n;
        a.r += vm.a.r;
        vms.add(vm);
    }
}
```

```
public class WindowsVM extends VM {
    public WindowsVM(Alocacao meusRecursos){
        super(meusRecursos, "Windows");
    }
}
public class LinuxVM extends VM {
    public LinuxVM(Alocacao meusRecursos){
        super(meusRecursos, "Linux");
    }
}
```

```
public class Driver {
    public static void main(String[] args) {
        //Servidor com 3.0GHz, 128 núcleos e 512GB de RAM:
        Servidor s = new Servidor(new Configuracao(3.0, 128, 512));
        Random rand = new Random();
        int demanda = rand.nextInt(100);

        for (int i = 0; i <= demanda; i++)
            s.alocar(new LinuxVM(new Alocacao(8, 16)));
        System.out.println("Servidores alocados!");
    }
}
```

Veja que na Driver Class é criado um novo servidor da empresa. Daí, é suprida uma demanda (i.e. uma quantidade) aleatória de pedidos de máquinas virtuais (VMs), sendo criados e alocados exclusivamente servidores Linux. Note que cada nova alocação consome 8 núcleos e 16GB de memória do total previsto na Configuração do Servidor. **Implemente as melhorias a seguir**:

a) (2.0 pontos) **Crie um Singleton da classe Servidor**. Lembre-se de modificar também a driver class apropriadamente. *Na sua resposta, basta escrever as linhas que devem ser incluídas/modificadas nas duas classes.*

b) (1.0 ponto) Crie a exceção personalizada SemRecursoException (que será usada nos próximos itens). Ela deverá ser do tipo *checked*.

c) (1.5 pontos) Modifique o método alocar() da classe Servidor como segue: tanto para núcleo quanto para RAM, ao invés de cegamente incrementar-se seus totais alocados no Servidor (como está sendo feito), primeiro verifique, para os 2 campos, se o total já alocado + a nova alocação ultrapassam a configuração do servidor. Se sim, o método deve lançar a exceção do item b. *Na sua resposta, basta escrever as linhas que devem ser incluídas/modificadas.*

d) (1.5 pontos) Trate na driver class o possível lançamento de exceção do método alocar(): imprima a mensagem "Não há recursos" se for o caso. *Na sua resposta, bastam as linhas relativas ao tratamento.*

Questão 2) (2.0 pontos) A mesma empresa do item anterior deseja, num futuro próximo, automatizar a alocação de VMs para usuários leigos. Estes não sabem ao certo a alocação de núcleos e RAM que precisam. Por outro lado, sabe-se que tal valor é muito baixo. Considerando as classes do item anterior, tal como enunciadas, **crie uma Factory** que seja capaz de criar VMs a partir apenas do nome: "Linux" ou "Windows". Em ambos os casos, a VM deverá alocar apenas 1 núcleo e 4GB de RAM.

Questão 3) (1.0 ponto) Resumidamente, em que consiste o padrão Observer mais básico?

Questão 4) (0.5 ponto) Descreva pelo menos 2 vantagens de se implementar uma interface Emissor no padrão Observer.

Questão 5) (0.5 ponto) Em que tipo de contexto o Observer tradicional faz mais sentido que o Lazy Observer?