

1. [4] Considere o código abaixo e o resultado da sua execução.

```
public class Ex2Main {  
    public static void main(String[] args) {  
        Handler h0 = new SushiChef();  
        Handler h1 = new PastaChef();  
        Handler h2 = new BurgerChef();  
        Handler h3 = new PizzaChef();  
  
        order ("I'll definitely take the pasta", h0);  
        order("I'd like a pizza", h0);  
        order("I'll pick the donut", h0);  
        order("Today I think I'll have a burger", h0);  
        order("I want to eat sushi", h0);  
  
        public void order(String request, Handler h) {  
            h.handle(request.substring(request.length() - 7));  
        }  
    }  
}
```

“Cooking a pasta.”
“Cooking a pizza.”
“No chef can prepare this food.”
“Cooking a burger.”
“Preparing sushi.”

- a) Identifique o padrão utilizado e descreva o seu funcionamento.

- b) Apresente a estrutura UML para este problema em concreto.

2. [4] Tendo por base o padrão *Mediator*, descreva um problema e apresente uma implementação onde aplique este padrão (apenas o diagrama de classe e o exemplo de utilização na *main*).

3. [4] Considere a seguinte implementação de *Collections.sort*:

```
public class Collections {  
    public static <T extends Comparable<? super T>> void sort(List<T> list) {  
        list.sort(null);  
    }  
}
```

a. Indique de que padrão se trata e indique as suas vantagens.

b. Apresente o diagrama UML deste padrão.

4. [4] A classe `java.util.AbstractList` é uma classe abstrata que apresenta alguns métodos não abstratos, que usam exclusivamente métodos abstratos. Um exemplo é `lastIndexOf`:

```
public int lastIndexOf(Object o) {  
    ListIterator<E> it = listIterator(size());  
    if (o==null) {  
        while (it.hasPrevious())  
            if (it.previous()==null) return it.nextInt();  
    } else {  
        while (it.hasPrevious())  
            if (o.equals(it.previous())) return it.nextInt();  
    }  
    return -1;  
}
```

- a) Que padrões de software identifica nesta classe?
- b) Explique o seu funcionamento.
5. [4] Explique a organização da arquitetura de software microkernel, descrevendo, nomeadamento, o modelo, principais vantagens e desvantagens, e um exemplo em que o padrão é usado.