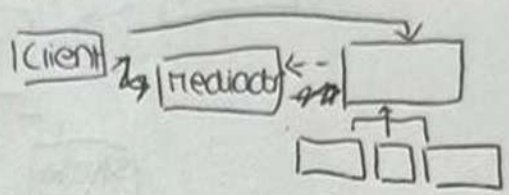
Nome: [redacted]

5

1. [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*).

O padrão Mediator, é um padrão comportamental que vai descrever relações entre classes este vai ser um "mediador" como o nome indica, guardando todos os estados e quando este pretende ir para o estado anterior, acede ao mediator.

Um exemplo da utilização deste seria num leilão, onde teríamos um mediador, que ira guardar o estado de cada Artigo, consoante se está em leilão, foi vendido ou ainda em stock, no entanto este não vai avisar as pessoas que estão na licitação, nesse caso ter-se-ia de emplementar um padrão Observer.



2. [4] Considere o código abaixo e o resultado da sua execução. Identifique o padrão utilizado e o descreva o seu funcionamento. Apresente a estrutura UML para este problema em concreto.

```
public class TestTranslator {
    public static void main(String[] args) {
        String s = "Hoje está um lindo dia de sol.";

        Translator t = new TranslatorEN(new TranslatorES(new TranslatorPT( new TranslatorUK())));

        t.translate(s, "UK");
        t.translate(s, "PT");
        t.translate(s, "ES");
        t.translate(s, "EN");
        t.translate(s, "FR");
    }
}
```

```
[Сьогодні прекрасний сонячний день.]
[Hoje está um lindo dia de sol.]
[Hoy es un hermoso día soleado.]
[Today is a beautiful sunny day.]
Translation not available for the language FR
```

1

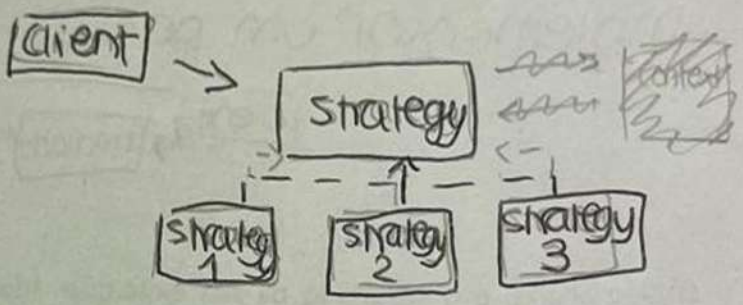
O padrão utilizado neste exemplo é o Strategy. Neste padrão vão existir várias classes (línguas possíveis), cada uma vai ter uma representação diferente, mas em termos de código igual, a única coisa que muda é a linguagem em que vai ser traduzido.

3. [4] Uma revista eletrónica quer fornecer aos seus clientes um conjunto de propriedades sobre diferentes telemóveis, como por exemplo, processador, preço, memória, câmara, etc. Os resultados devem ser apresentados numa lista, ordenada por qualquer um dos atributos. Por outro lado, existem vários algoritmos de ordenação com diferentes desempenhos, relativamente ao tempo de processamento e ao espaço ocupado. Assim, é necessário podermos selecionar facilmente o melhor algoritmo (por exemplo, em tempo de execução).

a. Que padrão pode ser aplicado para cumprir estes requisitos?

3 O padrão que pode ser utilizado é o strategy, este permite utilizarmos diferentes métodos (estratégias) para chegarmos a resultados similares.

b. Desenhe um diagrama de classes para responder a este problema.



4. [4] Explique a organização da arquitetura de software *microkernel*, descrevendo, nomeadamente, o modelo, principais vantagens e desvantagens, e um exemplo em que o padrão é usado.

A arquitetura microkernel tem como modelo, uma arquitetura centralizada, que pode 3 tipos de implementação API REST based, application REST base ou microservice, em que em todos criam uma "camada" para comunicar com os outros serviços. As vantagens desta são em termos de performance e agilidade e as desvantagens em termos de facilidade de desenvolvimento (devido às camadas).

5. [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> e = listIterator(size());
    if (o==null) {
        while (e.hasPrevious())
            if (e.previous()==null) return e.nextIndex();
    } else {
        while (e.hasPrevious())
            if (o.equals(e.previous())) return e.nextIndex();
    }
    return -1;
}
```

1

a. **Que padrões de software identifica nesta classe?**

Um padrão que se identifica aqui é o Iterator, um padrão comportamental.

b. **Explique o seu funcionamento.**

Este padrão funciona utilizando um list Iterator, que guarda todos os ~~[illegible]~~ estão conseguindo devolver sempre o próximo na lista