Explicação da questão 1

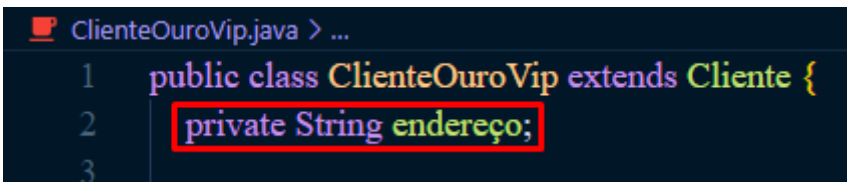
Classe: Representa a estrutura de dados. Na classe estão presentes métodos, atributos, construtores, etc. Nesse exemplo de classe podemos observar que se fazem presentes, construtor, métodos como **get** e **set** e o atributo **private String endereço;**.

ClienteOuroVip.java > ...

```java
public class ClienteOuroVip extends Cliente {
    private String endereço;

    public ClienteOuroVip(String nome, Double valorDaCompra, String endereço) {
        super(nome, valorDaCompra);
        this.endereço = endereço;
    }

    public String getEndereço () {
        return endereço;
    }

    public void setEndereço (String endereço) {
        this.endereço = endereço;
    }

    @Override
    public double CalcularPagamento () {
        double desconto = (this.getValorDaCompra() * 15) / 100;
        return this.getValorDaCompra() - desconto;
    }

    @Override
    public String toString() {
        return "ClienteOuroVip [endereço=" + endereço + "]";
    }
}
```

Exemplo de Classe

Atributo: São os elementos que definem a estrutura de uma classe, são propriedades dos objetos. Nesse exemplo podemos observar o atributo **private String endereço**.



Exemplo de atributo

Construtor: O construtor deve ter o mesmo nome da classe em que está inserido, nele são passados os parâmetros dos atributos. Nesse exemplo podemos observar que o construtor possui o mesmo nome da classe e está sendo passado como parâmetro o atributo **endereço**, **nome** e **valorDaCompra** que estão sendo acessados de uma classe abstrata.

```
public class ClienteOuroVip extends Cliente {
    private String endereço;

    public ClienteOuroVip(String nome, Double valorDaCompra, String endereço) {
        super(nome, valorDaCompra);
        this.endereço = endereço;
    }
```

Exemplo de construtor

Métodos: Métodos podem ser considerados como funções. Nesse exemplo podemos observar os métodos **gets** e **sets**. O método **get** é usado para acessar o valor do atributo, o **set** é usado para modificar o valor do atributo.

```
public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}
```

Exemplo de métodos

Objetos: Representa uma entidade abstrata ou concreta. Nesse exemplo podemos observar objetos sendo utilizados na classe **Main**. O uso do operador new ao construtor da classe que irá retornar uma referência.

```
Main.java > ...
1   public class Main {
        Run | Debug
2       public static void main(String[] args){
3           Mercadinho mercadinho = new Mercadinho();
4
5           ClienteRegular CLRegular = new ClienteRegular ("Mateus", 19.99);
6           ClienteVip CLVip = new ClienteVip ("Joana", 78.00, "400892");
7           ClienteOuroVip CLOuroVip = new ClienteOuroVip ("Lucas", 25.00, "Rua Epitácio Pessoa");
```

Exemplo de objeto

Encapsulamento: Tem a função de organizar os dados relacionados, deixando-os encapsulados em objetos, com o propósito de reduzir as colisões de nomes de variáveis. No meu código não tem esse exemplo, porém, coloquei essas imagens para que facilitem o entendimento. No segundo código também deveria conter o **endereço**.

ClienteOuroVip.java > ...

```java
public class ClienteOuroVip extends Cliente {
    private String endereço;

    public ClienteOuroVip(String nome, Double valorDaCompra, String endereço) {
        super(nome, valorDaCompra);
        this.endereço = endereço;
    }
```

ClienteVip.java > ...

```java
public class ClienteVip extends Cliente {
    private String NumeroDocartão;

    public ClienteVip(String nome, Double valorDaCompra, String NumeroDocartão) {
        super(nome, valorDaCompra);
        this.NumeroDocartão = NumeroDocartão;
    }
```

Exemplo de encapsulamento

Sobrecarga: É uma forma de polimorfismo que basicamente cria variações de um mesmo método, criando dois ou mais métodos com nomes iguais em uma classe diferente. Nesse exemplo podemos observar que um mesmo método foi usado em mais de uma classe.

Desconto.java

```java
public interface Desconto {
    double getValorDaCompra();
}
```

ClienteVip.java > ...

```java
    @Override
    public double CalcularPagamento () {
        double desconto = (this.getValorDaCompra () * 10) / 100;
        return this.getValorDaCompra() - desconto;
    }
```

Exemplo de sobrecarga

Herança: É uma forma que permite criar novas classes tomando como base classes já existentes, tomando proveito das características já existentes. Nesse exemplo podemos observar os atributos sendo herdados.

Cliente.java > ...

```java
public abstract class Cliente {
    private String nome;
    private Double valorDaCompra;

    public Cliente(String nome, Double valorDaCompra) {
        this.nome = nome;
        this.valorDaCompra = valorDaCompra;
    }
```

ClienteOuroVip.java > ...

```java
public class ClienteOuroVip extends Cliente {
    private String endereço;

    public ClienteOuroVip(String nome, Double valorDaCompra, String endereço) {
        super(nome, valorDaCompra);
        this.endereço = endereço;
    }

```

Exemplo de herança

Classe abstrata: É uma classe especial que não pode ser instanciada, podendo apenas ser herdada. Tomando como base o exemplo anterior, podemos contemplar o uso de uma classe abstrata.

Cliente.java > ...

```java
public abstract class Cliente {
    private String nome;
    private Double valorDaCompra;

```

ClienteOuroVip.java > ...

```java
public class ClienteOuroVip extends Cliente {
    private String endereço;
```

Exemplo de classe abstrata

Polimorfismo: Tem como objetivo ser utilizada por duas ou mais classes derivadas de uma superclasse podendo chamar métodos que possuem a mesma identificação, porém, comportamentos distintos. Podemos observar o mesmo método tendo comportamentos diferentes.

Desconto.java > ...

```java
public interface Desconto {
    double getValorDaCompra();
}
```

ClienteOuroVip.java > ClienteOuroVip

```java
@Override
public double CalcularPagamento () {
    double desconto = (this.getValorDaCompra() * 15) / 100;
    return this.getValorDaCompra() - desconto;
}
```

ClienteVip.java > ...

```java
@Override
public double CalcularPagamento () {
    double desconto = (this.getValorDaCompra () * 10) / 100;
    return this.getValorDaCompra() - desconto;
}
```

Exemplo de polimorfismo