

1)

Na classe Oficina ele importa a biblioteca Random para gerar números aleatórios, cria uma variável veículo do tipo Veículo, que possui duas subclasses, a Automóvel e a Bicicleta. Criando uma variável para a biblioteca Random, ele gera números de 0 a 100, e utiliza um if que, se esse número for par, ele fala para o código que é um automóvel, e se for ímpar, ele fala que é uma bicicleta. Logo após, ele cria um método revisar, que dentro dele, chama os métodos verificar, ajustar e limpar.

Já na classe UsaOficina, ele cria um objeto para a oficina, e um para o veículo, logo após cria um for para executar o método do número aleatório e definir se vai ser uma bicicleta ou um automóvel. Em seguida chama o método revisar, para verificar, ajustar e limpar o automóvel ou bicicleta.

2)

a)

```
App.java X Animal.java Cachorro.java Gato.java Preguiça.java
src > Principal > App.java > App
1 package Principal;
2
3 import Animais.*;
4
5 public class App {
6     public static void main(String[] args) throws Exception {
7         Animal animal = new Animal();
8         Animal cachorro = new Cachorro();
9         Animal gato = new Gato();
10        Animal preguica = new Preguiça();
11
12        animal.emitirSom();
13        cachorro.emitirSom();
14        gato.emitirSom();
15        preguica.emitirSom();
16    }
17 }
```

```
App.java Animal.java X Cachorro.java Gato.java Preguiça.java
src > Animais > Animal.java > Animal
1 package Animais;
2
3 public class Animal {
4     public void emitirSom(){
5         System.out.println(x:"Nha Nha Nha");
6     }
7 }
```

```
App.java Animal.java Cachorro.java X Gato.java Preguiça.java
src > Animais > Cachorro.java > Cachorro
1 package Animais;
2
3 public class Cachorro extends Animal {
4     public void emitirSom(){
5         System.out.println(x:"Au Au Au");
6     }
7 }
```

```
App.java Animal.java Cachorro.java Gato.java X Preguiça.java
src > Animais > Gato.java > Gato
1 package Animais;
2
3 public class Gato extends Animal{
4     public void emitirSom(){
5         System.out.println(x:"Miau Miau Miau");
6     }
7 }
```

```
App.java Animal.java Cachorro.java Gato.java Preguiça.java X
src > Animais > Preguiça.java > Preguiça
1 package Animais;
2
3 public class Preguiça extends Animal{
4     public void emitirSom(){
5         System.out.println(x:"AAAAAAHHHHZZZZ");
6     }
7 }
```

```
Nha Nha Nha
Au Au Au
Miau Miau Miau
AAAAAAHHHHZZZZ
```

b)

```
TesteFigura.java X Figura.java Quadrado.java Circulo.java
src > TesteFigura.java
1 public class TesteFigura {
2     public static void main(String[] args) throws Exception {
3         Quadrado quadrado1 = new Quadrado();
4         Circulo circulo1 = new Circulo();
5         Figura figura1 = new Figura();
6         System.out.println("Lado do quadrado: " + quadrado1.area(lado1:25) + "\nRaio do círculo: " + circulo1.area(raio1:15) + "\nÁrea da figura: " + figura1.area(areal:55));
7     }
8 }
```

```
J TesteFigura.java J Figura.java X J Quadrado.java J Circulo.java
src > J Figura.java > Figura
1 public class Figura extends TesteFigura{
2     public double area;
3
4     public double area(double area1){
5         return area1;
6     }
7 }
```

```
J TesteFigura.java J Figura.java J Quadrado.java X J Circulo.java
src > J Quadrado.java > Quadrado
1 public class Quadrado {
2     public double area(double lado1){
3         return lado1;
4     }
5 }
```

```
J TesteFigura.java J Figura.java J Quadrado.java J Circulo.java X
src > J Circulo.java > Circulo
1 public class Circulo {
2     public double area(double raio1){
3         return raio1;
4     }
5 }
```

```
Lado do quadrado: 25.0
Raio do círculo: 15.0
Area da figura: 55.0
```

O polimorfismo ocorre quando as classes têm o mesmo método, no entanto cada método tem funções diferentes (área do Quadrado: lado, área do Círculo: raio, área da Figura: área).

---

c)

```
J App.java X J Cliente.java J ClienteEspecial.java
src > J App.java > App
1 public class App {
    Run | Debug
2     public static void main(String[] args) throws Exception {
3         Cliente c1 = new Cliente();
4         Cliente ce1 = new ClienteEspecial();
5         Cliente c2 = new Cliente();
6         Cliente ce2 = new ClienteEspecial();
7
8         c1.desconto(valor:2000);
9         ce1.desconto(valor:2000);
10        c2.desconto(valor:6000);
11        ce2.desconto(valor:6000);
12    }
13 }
```

```
J App.java J Cliente.java X J ClienteEspecial.java
src > J Cliente.java > Cliente
1 public class Cliente {
2     public void desconto(double valor){
3         valor *= 0.9;
4         System.out.println("Valor já com o desconto para clientes: " + valor);
5     }
6 }
```

```
J App.java J Cliente.java J ClienteEspecial.java X
src > J ClienteEspecial.java > ClienteEspecial
1 public class ClienteEspecial extends Cliente{
2     public void desconto(double valor){
3         valor *= 0.8;
4         System.out.println("Valor já com o desconto para clientes especiais: " + valor);
5     }
6 }
```

```
Valor já com o desconto para clientes: 1800.0
Valor já com o desconto para clientes especiais: 1600.0
Valor já com o desconto para clientes: 5400.0
Valor já com o desconto para clientes especiais: 4800.0
```