

Vamos criar um exemplo prático em Java no NetBeans para ilustrar o conceito de polimorfismo.

Imagine que temos uma superclasse chamada Veiculo e duas subclasses Moto e Carro. Ambas as subclasses podem ter um método acelerar(), mas a forma como cada uma acelera pode ser diferente. Através do polimorfismo, podemos tratar objetos de ambas as subclasses como objetos da superclasse Veiculo e chamar o método acelerar() de forma polimórfica.

1. Crie um novo projeto Java no NetBeans.

2. Crie a superclasse Veiculo:

```
public class Veiculo {  
  
    private int velocidade;  
  
    public int getVelocidade() {  
        return velocidade;  
    }  
  
    public void acelerar() {  
        velocidade += 10;  
        System.out.println("Acelerando o veículo. Velocidade atual: " + velocidade + " km/h");  
    }  
}
```

3. Crie a subclasse Moto:

```
public class Moto extends Veiculo {  
  
    @Override  
    public void acelerar() {  
        int velocidade = getVelocidade();  
        velocidade += 20; // A moto acelera mais rapidamente.  
        System.out.println("Acelerando a moto. Velocidade atual: " + velocidade + " km/h");  
    }  
}
```

4. Crie a subclasse Carro:

```
public class Carro extends Veiculo {  
  
    @Override  
    public void acelerar() {  
        int velocidade = getVelocidade();  
        velocidade += 5; // O carro acelera mais lentamente.  
        System.out.println("Acelerando o carro. Velocidade atual: " + velocidade + " km/h");  
    }  
}
```

```
}
```

5. No método main, use o polimorfismo para tratar ambos, Moto e Carro, como Veiculo e chamar seus métodos acelerar() polimorficamente:

```
public class TestePolimorfismo {  
  
    public static void main(String[] args) {  
        Veiculo minhaMoto = new Moto();  
        Veiculo meuCarro = new Carro();  
  
        System.out.println("Para a Moto:");  
        minhaMoto.acelerar();  
  
        System.out.println("\nPara o Carro:");  
        meuCarro.acelerar();  
    }  
}
```

Quando você executar o programa TestePolimorfismo, verá que o método acelerar() de cada subclasse é chamado corretamente, ilustrando o conceito de polimorfismo. Mesmo que estejamos tratando ambos os objetos como Veiculo, o método correto é chamado para cada tipo real de objeto, seja ele Moto ou Carro.