

Fase 1: Criar Nó - Criar a classe NoVP que tenha os seguintes atributos: int valor, NoVP noDireita, NoVP noEsquerda, NoVP noPai e int cor. Instanciar um nó com valor 10. Obs: nós pretos receberão valor 0 e vermelhos 1.

```
@Test
void Fase1() {
    NoVP n1 = new NoVP(10);
    assertEquals(n1.valor, 10);
}
```



Fase 2: Criando Árvore - Criar a classe ArvoreVP, que tenha como atributo raiz. A raiz deve conter o valor 10.

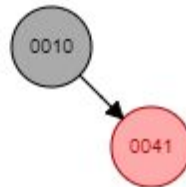
```
@Test
void Fase2() {
    ArvoreVP n1 = new ArvoreVP(10);
    assertEquals(n1.raiz.valor, 10);
}
```



Fase 3: Incluindo Nó - Adicionar a estrutura, um nó com o valor de 41. Para isso, utilize o método com a seguinte assinatura void add( int valor)

```
@Test
void Fase3() {
    ArvoreVP n1 = new ArvoreVP(10);
    assertEquals(n1.raiz.valor, 10);
    assertEquals(n1.raiz.cor, false);

    n1.add(41);
    assertEquals(n1.raiz.noDireita.valor, 41);
    assertEquals(n1.raiz.noDireita.cor, true);
}
```

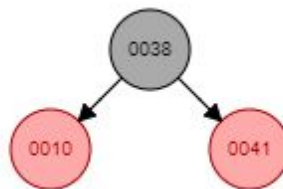


Fase 4: Incluindo Nó - Adicionar a estrutura, um nó com o valor de 38. Para isso, utilize o método com a seguinte assinatura void add( int valor)

```
@Test
void Fase4_10() {
    ArvoreVP n1 = new ArvoreVP(10);
    n1.add(41);
    n1.add(38);

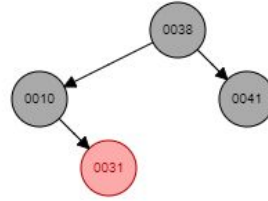
    assertEquals(n1.raiz.valor, 38);
    assertEquals(n1.raiz.noDireita.valor, 41);
    assertEquals(n1.raiz.noEsquerda.valor, 10);
    assertEquals(n1.raiz.noDireita.cor, false);
    assertEquals(n1.raiz.noEsquerda.cor, false);

    //rotação 1 & 2 + Troca de cor
}
```



Fase 5: Incluindo Nó - Adicionar a estrutura, um nó com o valor de 31. Para isso, utilize o método com a seguinte assinatura void add( int valor)

```
n1.add(31);
assertEquals(n1.raiz.noDireita.cor,true);
assertEquals(n1.raiz.noEsquerda.cor,true);
assertEquals(n1.raiz.noEsquerda.noDireita.valor,31);
```



Fase 6: Incluindo Nó - Adicionar a estrutura, um nó com o valor de 12. Para isso, utilize o método com a seguinte assinatura void add( int valor)

Fase 7: Incluindo Nó - Adicionar a estrutura, um nó com o valor de 19. Para isso, utilize o método com a seguinte assinatura void add( int valor)

```
n1.add(19);

assertEquals(n1.raiz.noEsquerda.noDireita.noEsquerda.valor,19);
assertEquals(n1.raiz.noEsquerda.cor,true);
```

Fase 8: Incluindo Nó - Adicionar a estrutura, um nó com o valor de 08. Para isso, utilize o método com a seguinte assinatura void add( int valor)

Fase 9: Incluindo Nó - Adicionar a estrutura, um nó com o valor de 55. Para isso, utilize o método com a seguinte assinatura void add( int valor)

Fase 10: Incluindo Nó - Adicionar a estrutura, um nó com o valor de 56. Para isso, utilize o método com a seguinte assinatura void add( int valor)

Fase 11: Buscando Nó - Busque o nó com o valor 15. Para isso utilize o método com a seguinte assinatura boolean buscaVP(int valor)

Fase 12: Excluindo Nó - Remova o nó com o valor 15. Para isso utilize o método com a seguinte assinatura boolean removeVP(int vaor).

Observação, o método retorna true, se conseguir remover o nó e false, caso contrário.



