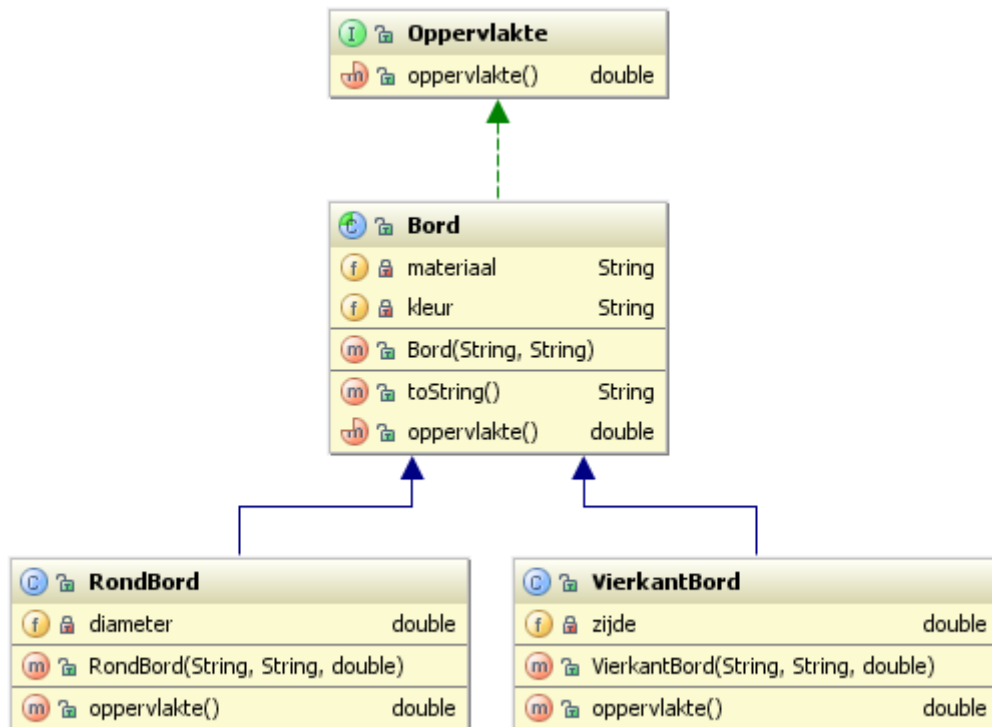


Bord (Interface en Abstract)

UML klassendiagram (via IntelliJ)



De interface *Oppervlakte*

```
public interface Oppervlakte {
    public abstract double oppervlakte();
}
```

De klasse *Bord*

De klasse `Bord` is een *abstract* klasse. Dit wil zeggen dat je er rechtstreeks geen objecten kunt van maken. Ze implementeert ook de interface `Oppervlakte`. De attributen zijn `materiaal` en `kleur`, beide van het type `String`.

- Voorzie een *constructor* die beide attributen een beginwaarde geeft.
- Voorzie een `toString` methode om `materiaal` en `kleur` als één string terug te geven.
- Voorzie een *abstract* methode `oppervlakte` (om te voldoen aan de implementatie van de interface).

De klasse RondBord

De klasse `RondBord` is een uitbreiding van de klasse `Bord`.

- Er is slechts 1 attribuut, met name de `double` `diameter`.
- Voorzie een *constructor* om alle drie de attributen een beginwaarde te geven.
- **Implementeer** de methode `oppervlakte`.

De klasse VierkantBord

De klasse `VierkantBord` is een uitbreiding van de klasse `Bord`.

- Er is slechts 1 attribuut, met name de `double` `zijde`.
- Voorzie een *constructor* om alle drie de attributen een beginwaarde te geven.
- **Implementeer** de methode `oppervlakte`.

De klasse TestBord

Gebruik deze klasse om de andere klassen te testen.

```
public static void main(String[] args) {  
    Bord een = new RondBord("karton", "groen", 30);  
    Bord twee = new VierkantBord("porcelijn", "wit", 20);  
    // Bord bord = new Bord("karton", "blauw"); <-- Verboden!  
  
    System.out.printf("een = %-15s %5.2f cm²\n", een, een.oppervlakte());  
    System.out.printf("twee = %-15s %5.2f cm²\n", twee, twee.oppervlakte());  
}
```

Gewenste uitvoer:

```
een = karton  groen    706,86 cm²  
twee = porcelijn wit   400,00 cm²
```