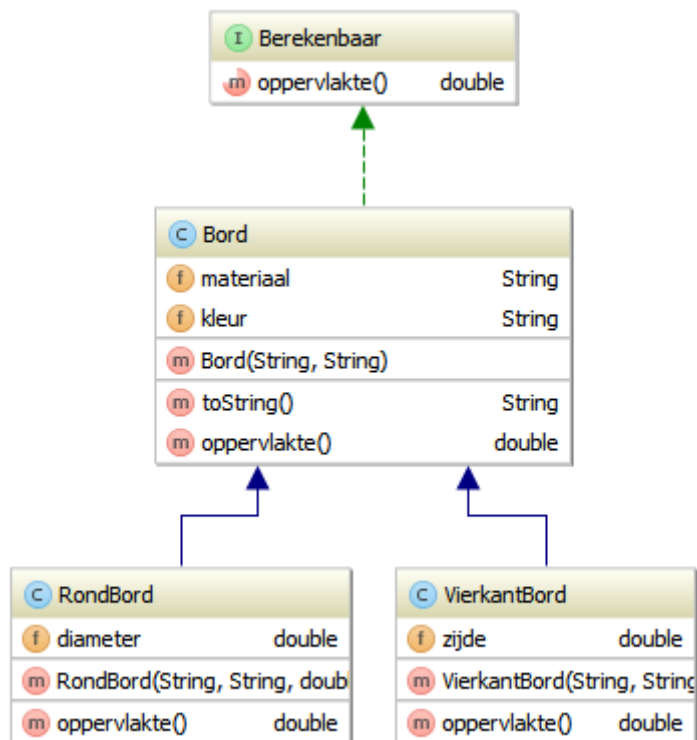


P2W3 Opdracht Bord Interface

UML klassendiagram (via IntelliJ)



De interface Berekenbaar

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

De klasse Bord

De klasse `Bord` implementeert 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.
- **Implementeer** een methode `oppervlakte` die 0.0 teruggeeft. Dit is nodig om *polymorfisme* toe te laten.

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 TestBordInterface

Gebruik deze klasse om de andere klassen te testen.

```
public class TestBordInterface {
    public static void main(String[] args) {
        Berekenbaar rond = new RondBord("karton", "groen", 30);
        Berekenbaar vierkant = new VierkantBord("porcelijn", "wit", 20);
        Berekenbaar bord = new Bord("karton", "blauw");

        System.out.printf("%-8s = %s %.0fcm³%n",
                           "rond", rond, rond.oppervlakte());
        System.out.printf("%-8s = %s %.0fcm²%n",
                           "vierkant", vierkant, vierkant.oppervlakte());
        System.out.printf("%-8s = %s%n",
                           "bord", bord);
    }
}
```

Gewenste uitvoer:

```
rond      = karton    groen  707cm³
vierkant  = porcelijn wit   400cm²
bord      = karton    blauw
```