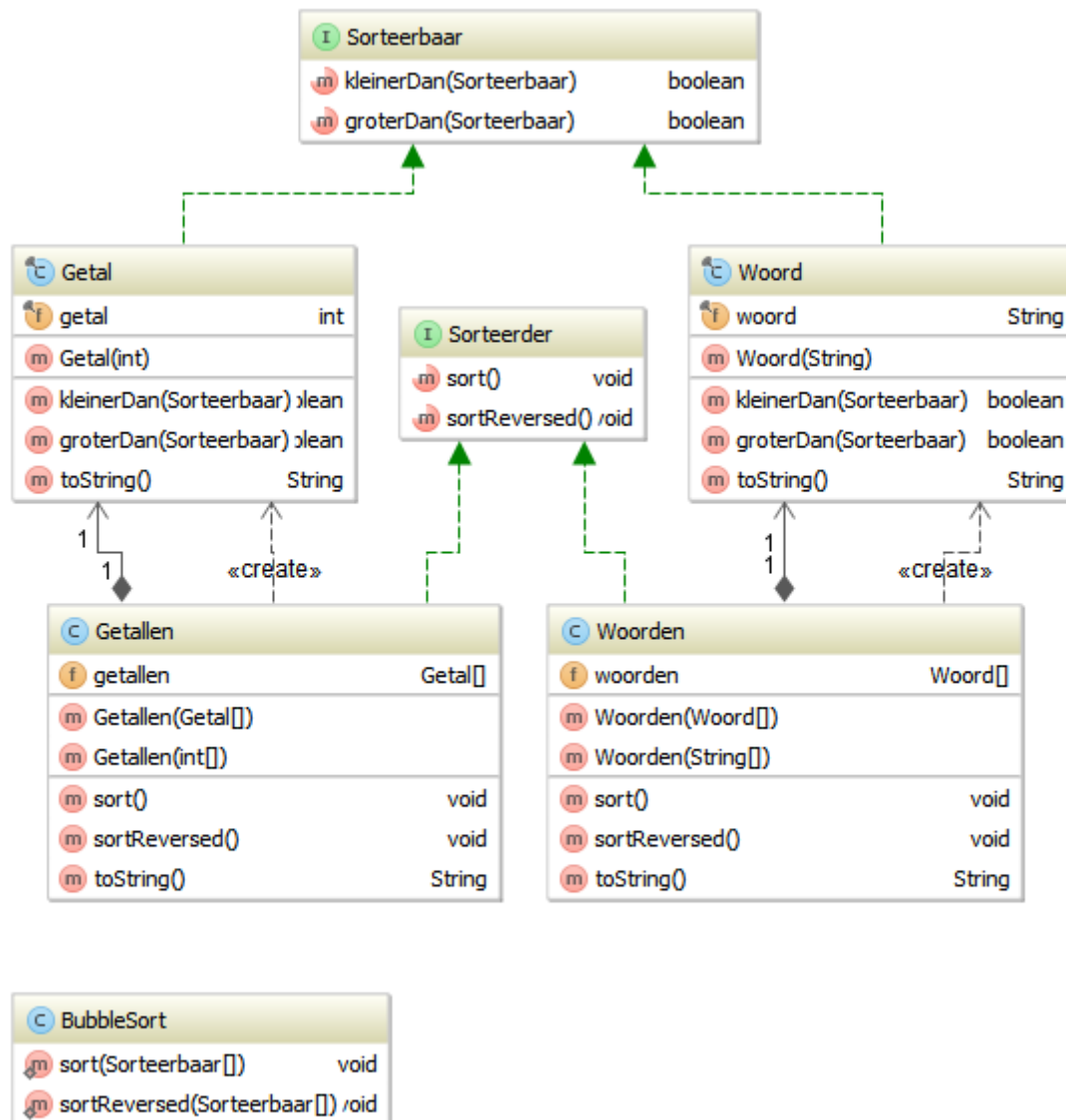


Opdracht Sorteertaar

UML klassendiagram (via IntelliJ)



De interface Sorteertaar

```
public interface Sorteertaar {
    boolean kleinerDan(Sorteertaar s);

    boolean groterDan(Sorteertaar s);
}
```

De interface Sorteerder

```
public interface Sorteerder {
    void sort();

    void sortReversed();
}
```

De klasse BubbleSort

```
/**
 * In deze klasse krijg je de sort methode cadeau.
 * Werk de andere methode zelf uit.
 */
public class BubbleSort {
    public static void sort(Sorteertaar[] tabel) {
        int n = tabel.length;
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (tabel[j + 1].kleinerDan(tabel[j])) {
                    Sorteertaar temp = tabel[j + 1];
                    tabel[j + 1] = tabel[j];
                    tabel[j] = temp;
                }
            }
        }
    }

    public static void sortReversed(Sorteertaar[] tabel) {
    }
}
```

De klasse Getal

```
/**
 * Pas aan waar nodig.
 *
 * Maak gebruik van de Sorteerbaar interface.
 */
public final class Getal {
    private final int getal;

    public Getal(int getal) {
        this.getal = getal;
    }

    /* Deze methode geeft true terug als het eerste getal
     * kleiner is dan het tweede.
     */
    public boolean kleinerdan(Sorteerbaar sorteerbaar) {
        return false;
    }

    /* Deze methode doet net het omgekeerde */
    public boolean groterDan(Sorteerbaar sorteerbaar) {
        return false;
    }

    /* Hier zet je een getalwaarde om naar een String, reeds uitgewerkt. */
    public String toString() {
        return Integer.toString(getal);
    }
}
```

De klasse Woord

```
/**
 * Pas aan waar nodig.
 *
 * Maak gebruik van de Sorteerbaar interface.
 */
public final class Woord {
    private final String woord;

    public Woord(String woord) {
        this.woord = woord;
    }

    /* Deze methode geeft true terug als het eerste woord alfabetisch
     * voor het tweede komt.
     */
    public boolean kleinerdan(Sorteerbaar sorteerbaar) {
        return false;
    }

    /* Deze methode doet het omgekeerde van de voorgaande */
    public boolean groterDan(Sorteerbaar sorteerbaar) {
        return false;
    }

    public String toString() {
        return woord;
    }
}
```

De klasse Getallen

```
/**
 * Vul/pas aan waar nodig. Maak gebruik van de Sorteerder interface.
 */
public class Getallen {
    private Getal getallen[];

    /**
     * Maak hier de nieuwe tabel. Kopieer dan de tabel van int getallen
     * naar de nieuwe tabel van het type Getal. Maak gebruik van
     * new Getal(...) om de int waarden naar type Getal om te zetten.
     */
    public Getallen(int[] getallen) {

    }

    /** Sorteert met de sort methode (zie klasse BubbleSort) */
    public void sort() {

    }

    /** Sorteert met de sortReversed methode (zie klasse BubbleSort) */
    public void sortReversed() {

    }

    /** Zorg ervoor dat de gewenste afdruk op het scherm komt */
    public String toString() {
        return "nonsens";
    }
}
```

De klasse Woorden

```
/**
 * Vul/pas aan waar nodig. Maak gebruik van de Sorteerder interface.
 */
public class Woorden {
    private Woord woorden[];

    /**
     * Maak hier de nieuwe tabel. Kopieer dan de tabel van strings
     * naar de nieuwe tabel van het type Woord. Maak gebruik van
     * new Woord(...) om de strings naar type Woord om te zetten.
     */
    public Woorden(String[] woorden) {

    }

    /** Sorteer met de sort methode (zie klasse BubbleSort) */
    public void sort() {

    }

    /** Sorteer met de sortReversed methode (zie klasse BubbleSort) */
    public void sortReversed() {

    }

    /** Zorg ervoor dat de gewenste afdruk op het scherm komt */
    public String toString() {
        return "nonsens";
    }
}
```

De klasse TestSorteerbaar

```
public class TestSorteerbaar {
    public static void main(String[] args) {
        int[] numbers = {5, 8, 2, 1, 10, 4, 3, 7, 6, 9};
        String[] eurolanden = {
            "Nederland", "België", "Frankrijk", "Duitsland", "Italië",
            "Spanje", "Finland", "Ierland", "Portugal", "Oostenrijk",
            "Luxemburg", "Griekenland", "Slovenië", "Cyprus", "Malta",
            "Slowakije", "Estland", "Letland"
        };

        Sorteerder getallen = new Getallen(numbers);
        System.out.println("getallen = " + getallen);
        getallen.sort();
        System.out.println("getallen = " + getallen);
        getallen.sortReversed();
        System.out.println("getallen = " + getallen);

        Sorteerder woorden = new Woorden(eurolanden);
        System.out.println("woorden = " + woorden);
        woorden.sort();
        System.out.println("woorden = " + woorden);
        woorden.sortReversed();
        System.out.println("woorden = " + woorden);
    }
}
```

Verwachte uitvoer (woorden op 1 regel):

```
getallen = 5 8 2 1 10 4 3 7 6 9
getallen = 1 2 3 4 5 6 7 8 9 10
getallen = 10 9 8 7 6 5 4 3 2 1
woorden = Nederland België Frankrijk Duitsland Italië Spanje Finland
Ierland Portugal Oostenrijk Luxemburg Griekeland Slovenië Cyprus
Malta Slowakije Estland Letland
woorden = België Cyprus Duitsland Estland Finland Frankrijk
Griekeland Ierland Italië Letland Luxemburg Malta Nederland
Oostenrijk Portugal Slowakije Slovenië Spanje
woorden = Spanje Slovenië Slowakije Portugal Oostenrijk Nederland
Malta Luxemburg Letland Italië Ierland Griekeland Frankrijk Finland
Estland Duitsland Cyprus België
```