

P2W6 Opdracht Bevolking (EXTRA)

De klasse Persoon

De klasse **Persoon** bevat de volgende attributen: **naam** (een **String**) en **leeftijd** (type **int**)

- Voorzie een constructor om alle attributen een waarde te geven.
- Voorzie een getter voor elk attribuut.
- Voorzie een **toString**-methode → zie gewenste uitvoer.

De klasse Bevolking

Vul in de volgende klasse de gevraagde methoden aan (zie ook de verwachte uitvoer):

```
public class Bevolking {
    private NavigableSet<Persoon> bevolking = new TreeSet<>();

    public Bevolking() {
        populate();
    }

    private void populate() {
        bevolking.add(new Persoon("Jos", 51));
        bevolking.add(new Persoon("Els", 46));
        bevolking.add(new Persoon("Karel", 48));
        bevolking.add(new Persoon("Jos", 55));
        bevolking.add(new Persoon("Marcel", 49));
        bevolking.add(new Persoon("Loes", 50));
    }

    public String toon() {
        // geef de inhoud van de set gebruik makend van de toString methode
        // op 1 regel als een String terug
    }

    public Persoon getEerste() {
        // geef het de eerste person in de set terug
    }

    public Persoon getLaatste() {
        // geef de laatste person in de set terug
    }

    public Persoon getVolgende(Persoon persoon) {
        // geef de eerstvolgende persoon in de set terug (of null)
    }

    public Persoon getVorige(Persoon persoon) {
        // geef de vorige person in de set terug (of null)
    }

    public Persoon getCeiling(Persoon persoon) {
        // geef de person die de parameter het dichtst benadert terug,
        // moet in de rangorde ervoor komen
    }
}
```

```

public Persoon getFloor(Persoon persoon) {
    // geef de person die de parameter het dichtst benadert terug,
    // moet in de rangorde gelijk zijn of erna komen
}

public String toonMetHigher() {
    // doe hetzelfde als in de toon methode maar gebruik nu de higher
    // methode om de set te doorlopen
}

public String toonMetDescendingIterator() {
    // doe hetzelfde als in de toon methode maar gebruik nu een
    // descendingIterator methode om de set te doorlopen (zie Javadoc)
}

public String toonVolgensLeeftijd() {
    // doe hetzelfde als in de toon methode maar geef de personen terug
    // in volgorde van oplopende leeftijd
}
}

```

De klasse RunDemo

Gebruik deze klasse om je programma te testen.

```

public class Demo {
    public static void main(String[] args) {
        Bevolking bevolking = new Bevolking();

        System.out.println("Inhoud set: " + bevolking.toon());

        System.out.println("\nEerste: " + bevolking.getEerste());
        System.out.println("Tweede: " +
            bevolking.getVolgende(bevolking.getEerste()));

        System.out.println("\nVoorlaatste: " +
            bevolking.getVorige(bevolking.getLaatste()));
        System.out.println("Laatste: " + bevolking.getLaatste());

        System.out.println("\nCeiling: " +
            bevolking.getCeiling(new Persoon("Koen", 47)));
        System.out.println("Floor: " +
            bevolking.getFloor(new Persoon("Koen", 47)));

        System.out.println("\nInhoud set: " + bevolking.toonMetHigher());
        System.out.println("Omgekeerd: " +
            bevolking.toonMetDescendingIterator());
        System.out.println("Leeftijd: " + bevolking.toonVolgensLeeftijd());
    }
}

```

Verwachte uitvoer:

Inhoud set: Els(46) Jos(51) Karel(48) Loes(50) Marcel(49)

Eerste: Els(46)

Tweede: Jos(51)

Voorlaatste: Loes(50)

Laatste: Marcel(49)

Ceiling: Loes(50)

Floor: Karel(48)

Inhoud set: Els(46) Jos(51) Karel(48) Loes(50) Marcel(49)

Omgekeerd: Marcel(49) Loes(50) Karel(48) Jos(51) Els(46)

Leeftijd: Els(46) Karel(48) Marcel(49) Loes(50) Jos(51)

