

## P2W6 Opdracht Personen (Extra)

### De klasse Persoon

De klasse Persoon bevat 3 attributen: id van het type int, naam van het type String en leeftijd van het type int.

- Voorzie een constructor waarmee je beide attributen van een beginwaarde kunt voorzien.
- Voorzie een getter voor het id.
- Voorzie een toString methode om de inhoud van het object netjes in kolommen weer te geven.

### De klasse Personen

De klasse Personen bevat een TreeMap met de naam personen met als key een int (de id van het Persoon object) en als value het Persoon object zelf.

- Voorzie een methode voegtoe om een Persoon object aan de map toe te voegen. Zorg ervoor dat bij een reeds aanwezige key de value niet overschreven wordt. Dit zou op één coderegel moeten kunnen, zoek in de documentatie van de Map interface
- Voorzie een methode getValues om alle in de Map aanwezige values terug te geven.
- Voorzie een methode getAll om alle elementen die in de range van de id waarden vallen (van de minimale tot en met de maximale waarde) terug te geven. Als er geen value aanwezig is moet een Persoon object met de waarden "onbekend" (voor de naam) en 0 voor de leeftijd worden teruggegeven.
- Voorzie een methode vervang om op de eenvoudigste manier (zie javadoc) bij bestaande entry (het id is als key aanwezig) de value door een nieuw Persoon object te vervangen.
- Voorzie een methode vervangAls om op de eenvoudigste manier een bestaande entry (key en value moeten overeenstemmen) de value door een nieuw Persoon object te vervangen.

## De klasse RunPersonen

Gebruik de volgende klasse om je programmeerwerk uit te testen:

```
public class RunPersonen {
    public static void main(String[] args) {
        Personen personen = new Personen();

        personen.voegToe(new Persoon(1000, "Jos", 50));
        personen.voegToe(new Persoon(1010, "Erika", 25));
        personen.voegToe(new Persoon(1005, "Peter", 40));
        personen.voegToe(new Persoon(1000, "Joske", 5));
        personen.voegToe(new Persoon(1003, "Joke", 15));
        personen.voegToe(new Persoon(1004, "Dirk", 45));
        personen.voegToe(new Persoon(1007, "Dorien", 30));

        toonAllePersonen(personen);

        // Toon alle values (al dan niet bestaande) vanaf de
        // id-waarde 1000 tot de maximale id-waarde
        for (Persoon persoon : personen.getAll()) {
            System.out.println(persoon);
        }
        System.out.println();

        // Vervang de persoon met id 1003 door een Persoon object
        // met Karolien als naam en 27 als leeftijd
        personen.vervang(1003, new Persoon(1003, "Karolien", 27));

        // Vervang de persoon met het id 1010 als de value een
        // Persoon object met id 1010, naam Petra en leeftijd 25 is
        // door als nieuwe value een Persoon object met de naam
        // Karel en de leeftijd 70
        personen.vervangAls(1010, new Persoon(1010, "Petra", 25),
            new Persoon(1010, "Karel", 70));

        toonAllePersonen(personen);
    }

    private static void toonAllePersonen(Personen personen) {
        for (Persoon persoon : personen.getValues()) {
            System.out.println(persoon);
        }
        System.out.println();
    }
}
```

## Verwachte uitvoer

1000	Jos	50
1003	Joke	15
1004	Dirk	45
1005	Peter	40
1007	Dorien	30
1010	Erika	25

1000	Jos	50
1001	Onbekend	0
1002	Onbekend	0
1003	Joke	15
1004	Dirk	45
1005	Peter	40
1006	Onbekend	0
1007	Dorien	30
1008	Onbekend	0
1009	Onbekend	0
1010	Erika	25

1000	Jos	50
1003	Karolien	27
1004	Dirk	45
1005	Peter	40
1007	Dorien	30
1010	Erika	25