### Uitwerking opgave 4: Sequence Diagrams and Java Code - Basic

**Opgave 4.1: Sequence diagram vertalen naar Java-code**

**public class RegistreerLuchthaven** {

private Land land;

private Luchthaven lh;

private TreeMap<String,Land> alleLanden;

public start() {  
 alleLanden = Land.*geefAlleLanden()*;

lh = new Luchthaven(); }

public int landNaam(String naam) {

land = alleLanden.get(naam); // Add check on null

lh.zetLand(land);

int landCode = land.geefCode();

return landCode; }

public void naam(String naam) {

lh.zetNaam(naam); }

public void code(String code); {

lh.zetCode(code)}

public void werkPlaats(boolean wp) {

lh.zetWerkplaats(wp); }

public void ok() {

lh.bewaar(); }

public void cancel() { }

}

**public class Luchthaven** {

private String naam;

private String code;

private boolean werkPlaats;

private Land land;

public Luchthaven() { }

public void zetLand(Land land) { }

public void zetNaam(String naam) {

if (land.geefLuchthaven(naam) == null)

this.naam = naam; } // Else 🡪 Throw exception

public String geefNaam(String naam) { }

public void zetCode(String code) { }

public void zetWerkPlaats(boolean wp) { }

public void bewaar() {

land.voegLuchthavenToe(this); }

}

**public class Land** {

private static ArrayList<Land> alleLanden;

private String naam;

private int code;

private HashSet<Luchthaven> luchthavens;

public static TreeMap<String,Land> geefAlleLanden() {

TreeMap< String,Land > landenMap = new TreeMap<String,Land>();

for(Land land : alleLanden) {

landenMap.put(land.geefNaam(), land); }

return landenMap; }

public String geefNaam() { }

public int geefCode() { }

public Luchthaven geefLuchthaven(String naam) { // Incomplete

for(Luchthaven luchthaven : luchthavens) {

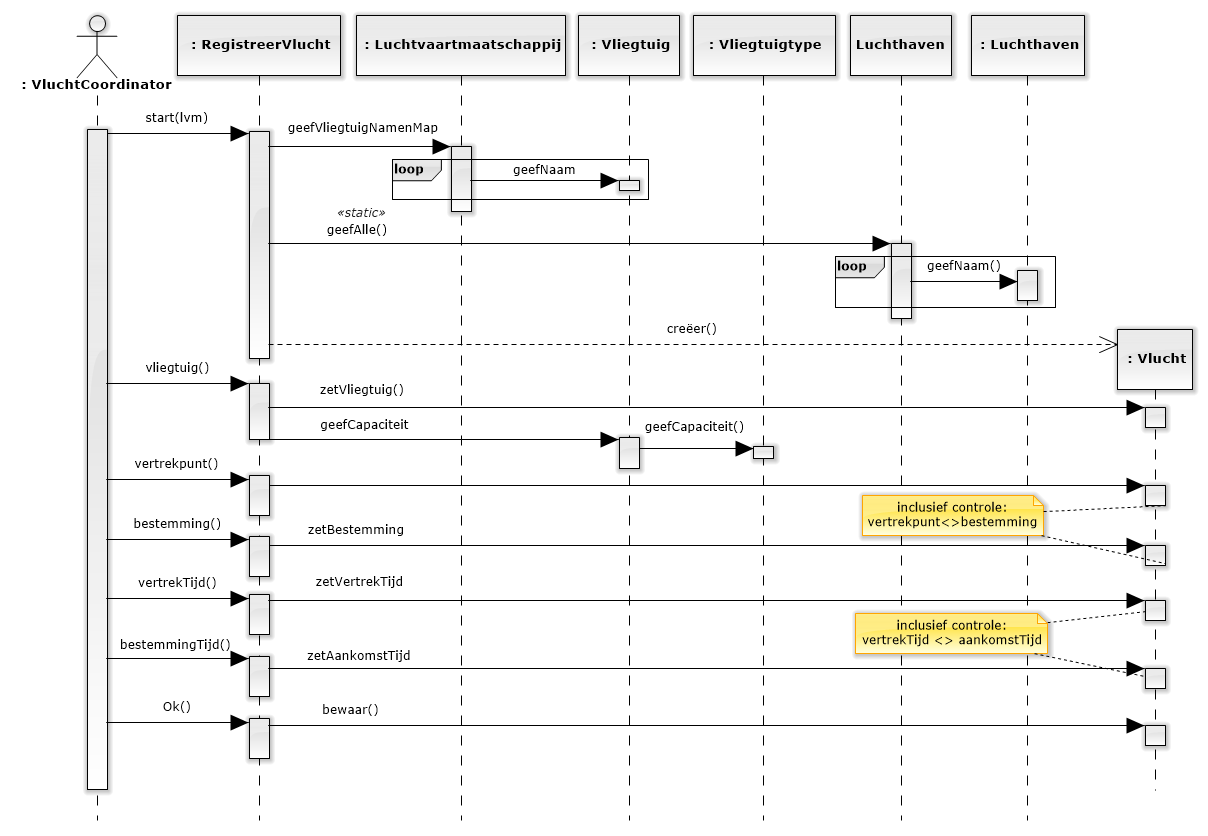
if (luchthaven.geefNaam().equals(naam)) return luchthaven; } }

public void voegLuchthavenToe(Luchthaven lh) { }

}

**Opgave 4.2 Opstellen van een sequence diagram voor een use case**

**Sequence Diagram**



**Klassediagram met aanvullingen**

