

Domain Design

Inhoud

Tekstueel	1
Domeinklassen	1
Overzicht variabelen en types	2
Relaties	3
Methoden	4
UML klassendiagram	5

Tekstueel

“De applicatie moet voor Nederland overzicht geven van de faciliteiten en huizen van een park, en per huis of het gehuurd is, zo ja, voor hoeveel dagen, door wie, en wat de prijs dan is. Ook moet het geslacht en de leeftijdsgroep per klant getoond worden.”

Domeinklassen

- Land
- Park
- Faciliteit
- Huis
- (super)Klant
 - o (sub)Man
 - o (sub)Vrouw
 - o (sub)Adolescent
 - o (sub)Volwassene
 - o (sub)Senior
- HuisHuur

Overzicht variabelen en types

- Land
 - o naam : String
 - o park : Park
- Park
 - o naam : String
 - o plaats : String
 - o provincie : String
 - o beoordeling : double
 - o faciliteiten : List<Faciliteit>
 - o huizen : List<Huis>
- Faciliteit
 - o type : String
 - o naam : String
- Huis
 - o type : String
 - o prijsPerNacht : int
 - o oppervlakte : int
 - o aantalSlaapkamers : int
 - o aantalVerdiepingen : int
 - o verwarming : boolean
 - o wifi : boolean
 - o vaatwasser: boolean
 - o koffiemachine: boolean
 - o aantalGaspitten : int
 - o aantalBadkamers : int
 - o föhn : boolean
 - o sauna : boolean
 - o zonnebank : boolean
- (super)Klant
 - o naam : String
 - o straat : String
 - o huisnummer : int
 - o postcode : String
 - o woonplaats : String
 - o leeftijdsgroep : String
 - o telefoonnummer : int
 - o e-mailadres: String
 - o manOfVrouw : String
 - o kortingsPercentage : double
- HuisHuur
 - o aantalNachten : int
 - o huurder : Klant
 - o gehuurdeHuis : Huis
- (sub)Man
 - o man : Klant
- (sub)Vrouw
 - o vrouw : Klant
- (sub)Adolescent
 - o adolescent : Klant
- (sub)Volwassene
 - o volwassene : Klant
- (sub)Senior
 - o volwassene : Klant

Relaties

- **Land <-> Park**
 1. Een **land** heeft één of meer **parken**
 2. Een **park** hoort altijd bij één **land**
- **Park <-> Faciliteit**
 1. Een **park** heeft nul of meer **faciliteiten**
 2. Een **faciliteit** hoort altijd bij één **park**
- **Park <-> Huis**
 1. Een **park** heeft één of meer **huizen**
 2. Een **huis** hoort altijd bij één **park**
- **Park <-> Klant**
 1. Een **park** heeft nul of meer **klanten**
 2. Een **klant** hoort altijd bij één of meer **parken**
- **Huis <-> Klant**
 1. Een **huis** heeft nul of één **klant**
 2. Een **klant** hoort altijd bij nul of één **huis**
- **Huis <-> HuisHuur**
 1. Een **huis** hoort nul of één keer bij **huishuur**
 2. **Huishuur** hoort bij nul of meer **huizen**
- **Klant <-> HuisHuur**
 1. Een **klant** hoort nul of één keer bij **huishuur**
 2. **Huishuur** hoort bij nul of meer **klanten**

Methoden

- Land
 - o Land(n : String)
 - o setPark(p : Park) : void
 - o toString() : String
- Park
 - o Park(n : String, pl : String, pr : String, b : double)
 - o setFaciliteit(f : Faciliteit) : void
 - o setHuis(h : Huis) : void
 - o toString() : String
- Faciliteit
 - o Faciliteit(t : String, n : String)
 - o toString() : String
- Huis
 - o Huis(t : String, pPN : int, o : int, aS : int, aV : int, ve : boolean, w : boolean, va : boolean, k : boolean, aG : int, aB : int, f : boolean, s : boolean, z : boolean)
 - o getType() : String
 - o getPrijsPerNacht() : int
 - o booleanCheck(wa : boolean, wo : String) : String
 - o toString() : String
- (super)Klant
 - o Klant(n : String, s : String, h : int, p : String, w : String, gb : String, t : String, e : String, gs : char, kP : double)
 - o getKorting() : double
 - o toString() : String
 - (sub)Man
 - o setMan(m : Klant) : void
 - o toString() : String
 - (sub)Vrouw
 - o setVrouw(v : Klant) : void
 - o toString() : String
 - (sub)Adolescent
 - o setAdolescent(a : Klant) : void
 - o toString() : String
 - (sub)Volwassene
 - o setVolwassene(v : Klant) : void
 - o toString() : String
 - (sub)Senior
 - o setSenior(s : Klant) : void
 - o toString() : String
- HuisHuur
 - o HuisHuur()
 - o setAantalNachten(aN : int) : void
 - o setHuurder(h : Klant) : void
 - o setGehuurdeHuis(gH : Huis) : void
 - o berekenPrijs() : double
 - o huisCheck() : String
 - o klantCheck() : String
 - o toString() : String

UML klassendiagram

Visual Paradigm Standard(Tobias Kramer(Utrecht University of Applied Sciences))

