

HO GENT

OOSDII

Overerving Deel 2 - Oefeningen

Table of Contents

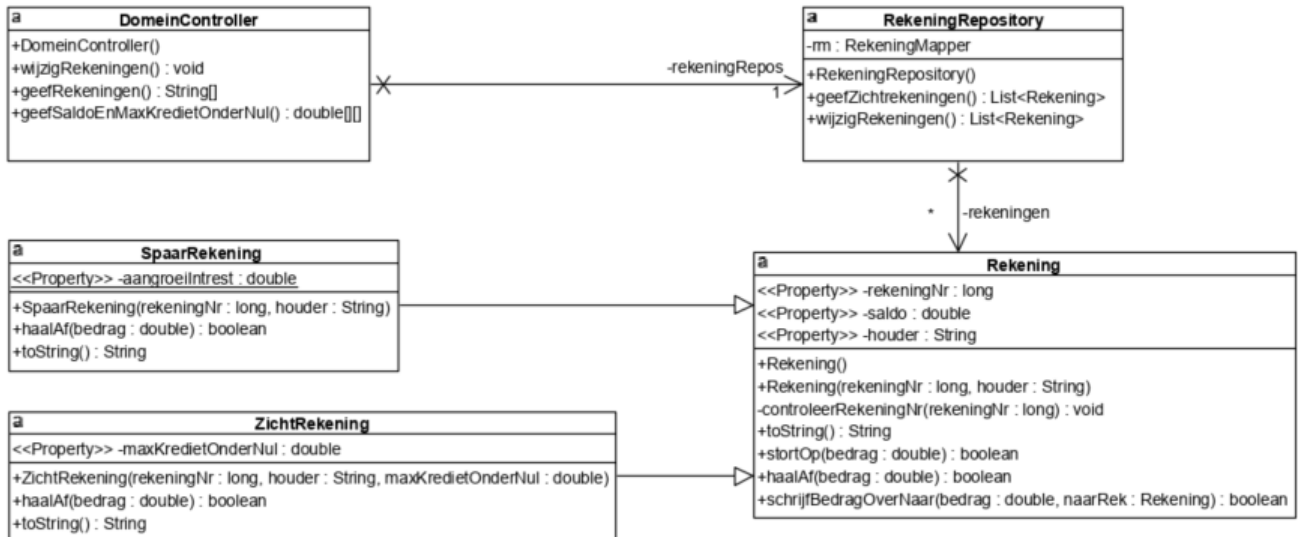
1. Oefening 1 Rekening	1
1.1. Gegeven: Startproject Rekening_start	1
1.2. Gevraagd	1
2. Oefening 2 Film	2
2.1. Gegeven: Startproject Film_start	2
2.2. Gevraagd	3
3. Oefening 3 Tuin	4
3.1. Gegeven: De UML van alle packages	4
3.2. Gevraagd: implementeer de volledige applicatie	5
3.3. Uitvoer van de applicatie	9

1. Oefening 1 Rekening

1.1. Gegeven: Startproject Rekening_start

Gegeven een project met een applicatie netjes uitgewerkt in 3 lagen. We vertrekken van de applicatie die uitgewerkt werd in oefening 4 van H7 in OOSD1 (startproject: Rekening_start).

De eigenschappen, het gedrag en de relaties van en tussen domeinklassen kan je aflezen uit onderstaand klassediagram:



1.2. Gevraagd

1.2.1. Breng volgende wijzigingen aan

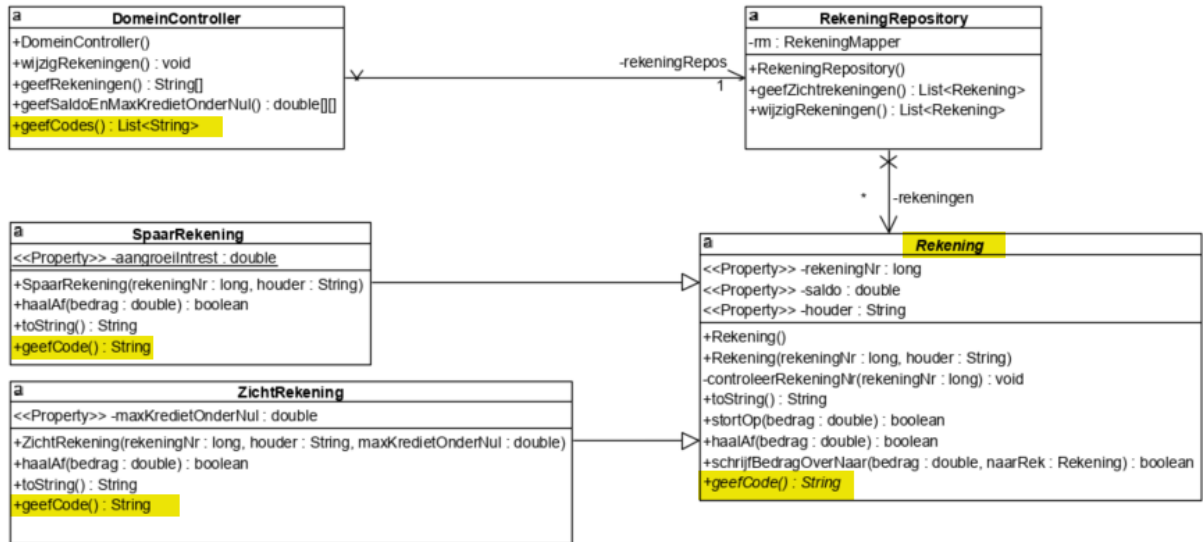
- Een algemene Rekening heeft geen betekenis, zo'n object mag je niet kunnen maken, enkel een zichtrekening of spaarrekening heeft een betekenis. Hoe realiseer je dit?
- Extra functionaliteit toevoegen: Elke rekening heeft een unieke code.

De opbouw van deze code is verschillend per type:

- Zichtrekening: ZR-maxKredietOnderNul-rekeningNr
- Spaarrekening: SR-rekeningNr-aangroeiIntrest.

Voorzie 2 decimalen voor het maxKredietOnderNul (zonder minteken in de code!) en de aangroeiIntrest

We voegen een abstracte methode `geefCode()` toe in de superklasse en maken die concreet in beide subclasses. Wat is het nut van de abstracte methode in de klasse Rekening?



- Pas nu de applicatieklasse en de DomeinController aan zodat een beschrijving van alle rekeningen op het scherm verschijnt.

```

Overzicht rekeningen
SpaarRekening met rekeningnummer 000-0000002-02
staat op naam van Ruben
en bevat 200,00 euro. Aangroeiintrest = 2,00%
ZichtRekening met rekeningnummer 000-0000003-03
staat op naam van Nick
en bevat 300,00 euro. Max krediet onder nul = -50,00
ZichtRekening met rekeningnummer 000-0000004-04
staat op naam van Alexander
en bevat 400,00 euro. Max krediet onder nul = -3286,00
SpaarRekening met rekeningnummer 000-0000001-01
staat op naam van Maxime
en bevat 100,00 euro. Aangroeiintrest = 2,00%

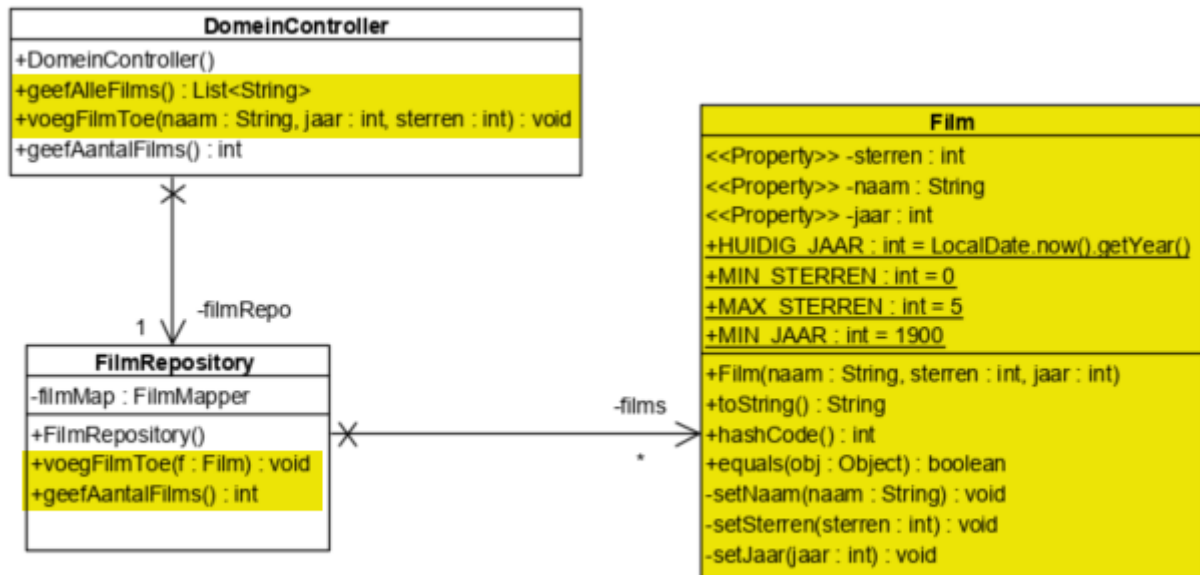
Wijzig rekeningen
Zichtrekening 1 heeft als saldo 300,00 en mag 60,00 onder nul.
Zichtrekening 2 heeft als saldo 400,00 en mag 3296,00 onder nul.

Overzicht codes rekeningen
SR-202-2,00
ZR-60,00-303
ZR-3296,00-404
SR-101-2,00
  
```

2. Oefening 2 Film

2.1. Gegeven: Startproject Film_start

In het startproject vind je terug een uitgewerkte 3lagen applicatie. In het domein vind je enkele TODO's, aangeduid in het fluo in onderstaande UML.



2.2. Gevraagd

- Klasse Film en FilmTest: We werken volledig de domeinklasse Film volgens de principes van TDD. Dit wil zeggen dat we EERST de testen schrijven en daarna pas de domeinklasse implementeren.

- Attributen:

- sterren, naam en jaar hebben een bijhorende getter en private setter.

naam: mag niet leeg zijn of enkel blanco's (spatie, nieuwe lijn, tab, ...) bevatten

sterren: waarde moet in het interval [MIN_STERREN, MAX_STERREN] liggen

jaar: waarde moet in het interval [MIN_JAAR, HUIDIG_JAAR] liggen

Bij een foutieve waarde zorg je telkens dat een IllegalArgumentException met een passende boodschap wordt gegoooid.

- 4 constanten (zie UML)

- Constructor met 3 parameters

- Methodes

- toString methode

Een filmobject wordt als volgt voorgesteld in tekstformaat: *naamKlasse naamFilm - sterren - jaar*

- Twee filmobjecten zijn gelijk als de naam én het jaar identiek zijn.

- De klasse FilmRepository

- Methode voegFilmToe

De film wordt toegevoegd in de lijst films als de film nog niet aanwezig is in de lijst.

Houd rekening met volgende uitleg, gekopieerd uit de API van de klasse ArrayList, methode contains

Package java.util

Class ArrayList<E>

contains

```
public boolean contains(Object o)
```

Returns true if this list contains the specified element. More formally, returns true if and only if this list contains at least one element *e* such that `Objects.equals(o, e)`.

Specified by:

contains in interface Collection<E>

Specified by:

contains in interface List<E>

Overrides:

contains in class AbstractCollection<E>

Parameters:

o - element whose presence in this list is to be tested

Returns:

true if this list contains the specified element

- Methode geefAantalFilms

Deze methode berekent hoeveel films in de lijst zitten.

- De klasse DomeinController

- Methode geefAlleFilms

De methode verzamelt alle tekstweergaves van de filmobjecten die aanwezig zijn in de lijst in de repository.

- Methode geefAantalFilms

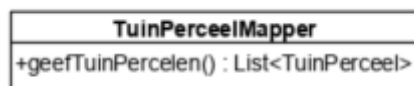
Vraagt aan de repository hoeveel films er zijn

3. Oefening 3 Tuin

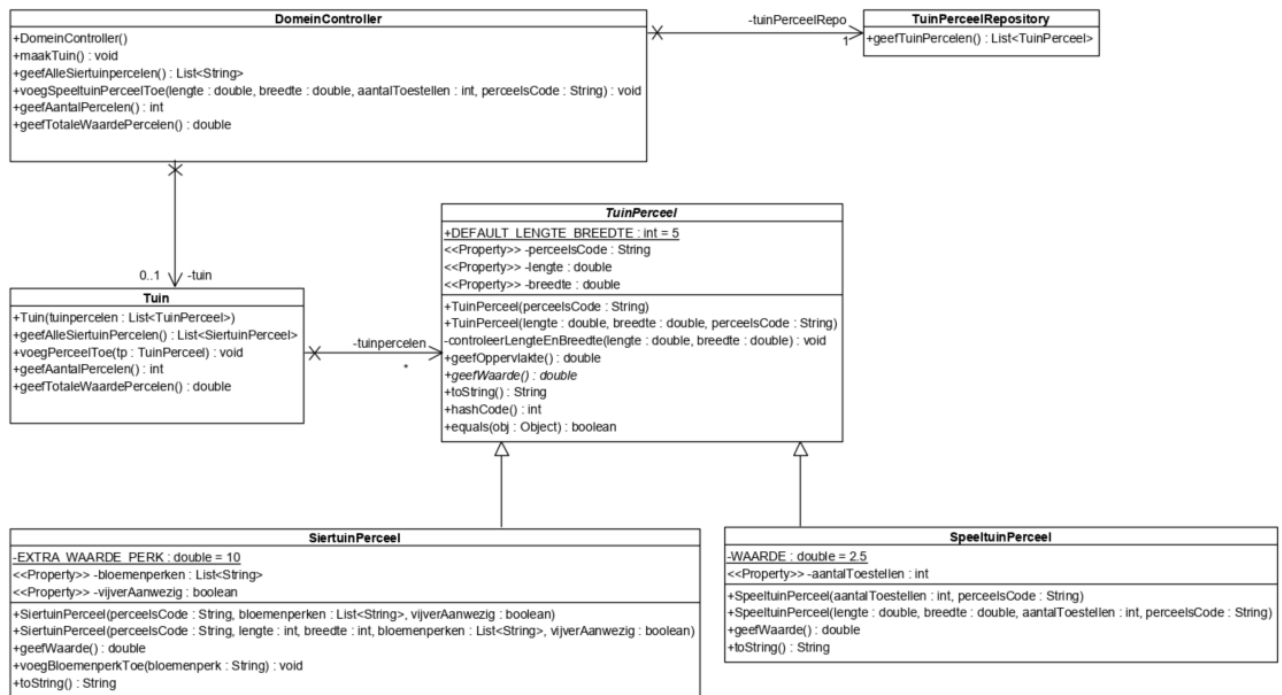
Bouw een volledig werkende drielagenapplicatie. Deze oefening bevat heel wat herhaling van semester 1 én de nieuwe leerstof uit dit hoofdstuk.

3.1. Gegeven: De UML van alle packages

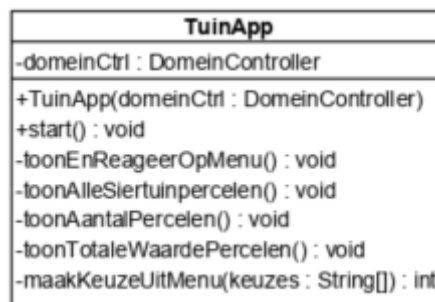
1. Package persistentie



2. Package domein



3. Package ui



4. Package main



3.2. Gevraagd: implementeer de volledige applicatie

Hieronder volgt nog een beschrijving van de functionaliteit.

1. Package persistentie

De klasse TuinPerceelMapper bevat slechts 1 methode. De code hiervan is gegeven:

```

1  public List<TuinPerceel> geefTuinPercelen(){
2      List<TuinPerceel> tuinpercelen = new ArrayList<>();
3      tuinpercelen.add(new SiertuinPerceel("SI001", Arrays.asList("Gladool",
4      "Roos", "Distel"), true));
5      tuinpercelen.add(new SiertuinPerceel("SI002", 20, 20, Arrays.asList
6      ("Gladool", "Roos", "Distel"), true));
7      tuinpercelen.add(new SiertuinPerceel("SI003", 20, 15, Arrays.asList
8      ("Zonnebloem", "Margriet", "Distel"), false));
9      tuinpercelen.add(new SiertuinPerceel("SI004", 3, 3, Arrays.asList
10     ("Sneeuwvlokje", "Roos", "Tulp"), false));
11     tuinpercelen.add(new SpeeltuinPerceel(20, 10, 3, "SP001"));
12     return tuinpercelen;
13 }

```

2. Package domein



Vergeet niet de klassen `Tuin`, `SiertuinPerceel` en `SpeeltuinPerceel` grondig te testen. Schrijf je eerst de testen vooraleer je de domeinklasse implementeert?

a. De abstracte klasse `TuinPerceel`

- De 4 attributen in de klasse krijgen een waarde bij creatie van het object en kunnen nadien niet meer gewijzigd worden. De attributen `lengte` en `breedte` zijn getallen en worden uitgedrukt in meter.
- Het attribuut `perceelsCode` is uniek voor een object van `TuinPerceel`. Dat wil zeggen dat twee objecten met dezelfde `perceelsCode` beschouwd worden als dezelfde objecten.
- In de methode **`controleerLengteEnBreedte`** valideer je volgende domeinregels:
 - `lengte >= 1` én `breedte >= 1`
 - `lengte >= breedte`
 - indien niet aan één van bovenstaande regels is voldaan werp je een `IllegalArgumentException` met een gepaste boodschap:

"Lengte en breedte moeten strikt positief zijn"

"Lengte moet groter dan of gelijk aan breedte zijn"

- De **constructors**:
 - De constructor met 1 parameter maakt een `tuinperceel` met `lengte` en `breedte` ingesteld op de `DEFAULT LENGTE BREEDTE`
- De methode **`geefWaarde`** is een abstracte methode.
- De methode **`geefOppervlakte`** geeft het product van `lengte` en `breedte` terug.
- Voorbeeld uitvoer methode **`toString()`**: `Tuinperceel` van 3,0 op 3,0 met een waarde van 3,00

b. De klasse `SiertuinPerceel`

- De attributen `EXTRA_WAARDE_PERK` en bloemenperken zijn niet meer wijzigbaar na constructie van een siertuinperceel.
- De **constructors**:
 - Indien de parameter bloemenperken null is dan wordt het attribuut bloemenperken geïnitieerd op een lege lijst.
 - Er kan niet meer dan 1 bloemenperk per m2 zijn, werp een `IllegalArgumentException` met een gepaste boodschap indien niet aan de deze regel is voldaan.

"Te veel bloemenperken voor dit perceel"

- De constructor zonder lengte en breedte parameters maakt een siertuinperceel met lengte en breedte ingesteld op `DEFAULT LENGTE BREEDTE`
- De methode **geefWaarde**:

Indien er een vijver aanwezig is wordt de waarde van het tuinperceel het aantal bloemenperken verhoogd met de constante `EXTRA_WAARDE_PERK`. Is er geen vijver, dan beperkt de waarde zich tot het aantal bloemenperken.

- Voorbeelden uitvoer methode **toString()**:

SiertuinPerceel van 5,0 op 5,0 met een waarde van 13,00 met vijver

SiertuinPerceel van 20,0 op 15,0 met een waarde van 3,00 zonder vijver

c. De klasse `SpeeltuinPerceel`

- Het attribuut `aantalToestellen` is niet meer wijzigbaar na constructie van een speeltuinperceel.
- De **constructors**:
 - Er kan niet meer dan 1 toestel per 3m2 zijn, werp een `IllegalArgumentException` met de volgende boodschap indien niet aan de deze regel is voldaan: "Er past niet meer dan 1 toestel per 3 vierkante meter"
 - De constructor zonder lengte en breedte parameters maakt een speeltuinperceel met lengte en breedte ingesteld op `DEFAULT LENGTE BREEDTE`
- De methode **geefWaarde**:
 - De waarde van het tuinperceel is het aantal toestellen vermenigvuldigd met de constante `WAARDE`
- Voorbeeld uitvoer methode **toString()**:

Speeltuinperceel van 5,0 op 5,0 met een waarde van 13,00 en met 3 toestellen om te spelen.

d. De klasse Tuin

- Het attribuut `tuinpercelen` bevat een List van `Tuinpercelen`, er is geen getter of setter voor dit attribuut.
- Bij creatie van een `Tuin`-object wordt de lijst van `tuinpercelen` meegegeven.
- De methode **`geefAlleSiertuinpercelen`** retourneert alle `siertuinpercelen`, en dus bijvoorbeeld NIET de `SpeeltuinPercelen`.
- De methode **`voegPerceelToe`** kan een `TuinPerceel`-object toevoegen aan de List `tuinpercelen`, op voorwaarde dat dit object nog niet in de lijst zit. Als het object er al in zit wordt een `IllegalArgumentException` met passende boodschap gegoooid.
- De methode **`geefAantalPercelen`** geeft het aantal objecten in de List `tuinpercelen` terug.
- De methode **`geefTotaleWaardePercelen`** maakt de som van alle waarden van de `tuinperceel` objecten en geeft die som terug.

e. De klasse `TuinPerceelRepository`

- In deze klasse vinden we slechts 1 methode terug, **`geefTuinPercelen`**. De methode vraagt de lijst van `TuinPerceel`-objecten op aan de mapperklasse en geeft de lijst terug.

f. De klasse `DomeinController`

- Deze klasse heeft 2 attributen (zonder getter/setter).
- In de **constructor** wordt het attribuut `tuinPerceelRepo` geïnitieerd.
- De methode **`maakTuin`** initialiseert het attribuut `tuin`. Hiervoor wordt een nieuw `Tuin`-object gemaakt met de lijst van `tuinpercelen` die je terugkrijgt van het object `tuinPerceelRepo`.
- De methode **`geefAlleSiertuinpercelen`** vraagt de `siertuin` objecten aan de `tuin` en verzamelt dan alle tekstweergaves in een List.
- De methode **`voegSpeeltuinPerceelToe`** maakt een `SpeeltuinPerceel`-object aan en voegt het toe aan de `tuin`.
- De methode **`geefAantalPercelen`** geeft het aantal percelen van de `tuin` terug.
- De methode **`geefTotaleWaardePercelen`** geeft de totale waarde van de percelen van de `tuin` terug.

3. Package `ui`

a. De klasse `TuinApp`

- De **constructor** krijgt een object van de klasse `DomeinController` door en initialiseert daarmee zijn attribuut.
- De methode **`start`** maakt een `tuin` aan (methode `maakTuin` uit de klasse `DomeinController` aanroepen) en roept daarna de methode `toonEnReageerOpMenu` aan.
- De methode **`toonEnReageerOpMenu`** heeft een lokale array

```
String[] keuzes = { "Toon alle siertuinpercelen", "Voeg speeltuinperceel toe", "Geef aantal percelen", "Geef totale waarde percelen", "Stop" };
```

Deze array wordt doorgegeven aan de methode `maakKeuzeUitMenu` en die methode geeft dan het nummer dat gekozen werd uit het menu terug. Op basis van dat nummer wordt nu de juiste hulpmethode aangeroepen, wordt de applicatie gestopt of wordt direct een methode uit de `DomeinController` aangeroepen (`domeinCtrl.voegSpeeltuinPerceelToe(10, 5, 3, "SP007");`)

- De methode **`maakKeuzeUitMenu`** drukt alle opties uit de array af en nummert die opties, te beginnen met 1. Nu kiest de gebruiker een cijfer, dat wordt gecontroleerd of dat cijfer inderdaad een optie is. Indien niet vraag je een nieuwe keuze en anders geef je het cijfer terug als gekozen waarde.
- De methode **`toonAlleSiertuinpercelen`** vraagt alle siertuinpercelen op en drukt ze netjes af.
- De methodes **`toonAantalPercelen`** en **`toonTotaleWaardePercelen`** drukken de opgevraagde waarden netjes af.

4. Package main

a. De klasse `StartUp`

Roep op correcte wijze de methode start van de klasse `TuinApp` aan.

3.3. Uitvoer van de applicatie

Uitvoer1:

MENU

=====

1. Toon alle siertuinpercelen
2. Voeg speeltuinperceel toe
3. Geef aantal percelen
4. Geef totale waarde percelen
5. Stop

Je keuze (1-5): 7

Geef een getal in tussen 1 en 5

MENU

=====

1. Toon alle siertuinpercelen
2. Voeg speeltuinperceel toe
3. Geef aantal percelen
4. Geef totale waarde percelen
5. Stop

Je keuze (1-5): -1

Geef een getal in tussen 1 en 5

MENU

=====

1. Toon alle siertuinpercelen
2. Voeg speeltuinperceel toe
3. Geef aantal percelen
4. Geef totale waarde percelen
5. Stop

Je keuze (1-5): 1

Alle siertuinpercelen:

SiertuinPerceel van 5,0 op 5,0 met een waarde van 13,00 met vijver
SiertuinPerceel van 20,0 op 20,0 met een waarde van 13,00 met vijver
SiertuinPerceel van 20,0 op 15,0 met een waarde van 3,00 zonder vijver
SiertuinPerceel van 3,0 op 3,0 met een waarde van 3,00 zonder vijver

MENU

=====

1. Toon alle siertuinpercelen
2. Voeg speeltuinperceel toe
3. Geef aantal percelen
4. Geef totale waarde percelen
5. Stop

Je keuze (1-5): 5

Uitvoer2:

MENU

=====

1. Toon alle siertuinpercelen
2. Voeg speeltuinperceel toe
3. Geef aantal percelen
4. Geef totale waarde percelen
5. Stop

Je keuze (1-5): 3

Aantal percelen in de tuin: 5

MENU

=====

1. Toon alle siertuinpercelen
2. Voeg speeltuinperceel toe
3. Geef aantal percelen
4. Geef totale waarde percelen
5. Stop

Je keuze (1-5): 2

MENU

=====

1. Toon alle siertuinpercelen
2. Voeg speeltuinperceel toe
3. Geef aantal percelen
4. Geef totale waarde percelen
5. Stop

Je keuze (1-5): 3

Aantal percelen in de tuin: 6

MENU

=====

1. Toon alle siertuinpercelen
2. Voeg speeltuinperceel toe
3. Geef aantal percelen
4. Geef totale waarde percelen
5. Stop

Je keuze (1-5): 4

Totale waarde percelen in de tuin: 47,00

MENU

=====

1. Toon alle siertuinpercelen
2. Voeg speeltuinperceel toe
3. Geef aantal percelen
4. Geef totale waarde percelen
5. Stop

Je keuze (1-5): 5