

















<b>Titel</b>	<b>De Gokkers</b>
--------------	-------------------

Tijd totaal project/projectdag	3 dagdelen per week
Kerntaak/kerntaken	B1-K1_Levert een bijdrage aan het ontwikkeltraject B1-K2_Realiseert en test (onderdelen van) een product B1-K3_Levert een product op
Werkproces(sen)	B1-K1-W1_Stelt de opdracht vast B1-K1-W2_Levert een bijdrage aan het projectplan B1-K1-W3_Levert een bijdrage aan het ontwerp B1-K1-W4_Bereidt de realisatie voor B1-K2-W1_Realiseert (onderdelen van) een product B1-K2-W2_Test het ontwikkelde product B1-K3-W1_Optimaliseert het product B1-K3-W2_Levert het product op B1-K3-W3_Evalueert het opgeleverde product
Tijdsverdeling PR per lesdag	Aansturing (AS): 40 minuten Zelfstandig Trainen (ZT): 140 minuten Terugkoppeling (TK): 40 minuten
Extra bijlagen bij dit project	1. Bijlage 2, Protocol presentatie
Leermiddelen en materialen	1. Laptop 2. Bestand 'Dog' en 'PictureBox1' 3. Project- en Teamrapport 4. Sjablonen bereikbaarheidslijst en samenwerkingscontract 5. Visual Studio
Literatuur en bronnen	Boeken • - Internet • - Overig • -







									
Veiligheid	Nieuwsgierigheid	Leeromgeving	Herhalen	Samenwerking	Focus	Zintuigen	Voortbouwen	Feedback	Structuur

## Aansturing: 40 minuten

TIJD	VORM	BESCHRIJVING OEFENING OF ONDERDEEL	
5 min	Opstart  	<ul style="list-style-type: none"><li>• Welkom heten</li><li>• Eventueel terug komen op zaken van vorige projectdag</li></ul>	
1 min	CASUS VOOR DE PROJECTGROEP		
	  	DIDACTISCHE VORM Maak hier een keuze	
	<p>Lidy, een fervent gokster, probeert steeds maar weer om Sietse en Fer over te halen om mee naar de hondenrenbaan te gaan. Maar Sietse en Fer vinden het niet leuk om al hun geld, dat ze met hard werken hebben verdiend, te verliezen. Daarom aan jullie de opdracht om een simulatiespel voor hen te maken.</p> <p>Ze willen dus wedden bij een hondenrenwedstrijd. Sietse start met 50 euro, Lidy met 75 euro (Zij heeft per slot van rekening het meeste geld.) en Fer met slechts 45 euro. Vóór elke race besluiten ze óf ze wedden en voor welk bedrag, of niet.</p> <p>Als de race eenmaal begonnen is kan er niets meer veranderd worden!!</p>		

TIJD	VORM	BESCHRIJVING OEFENING OF ONDERDEEL
xx min	<b>PROJECTOPDRACHT (DEZE WERK JE UIT TIJDENS DE ZELFSTANDIGE TRAINING)</b> <b>DE PRODUCTEN EEN-VOOR-EEN UITGEWERKT:</b> <ul style="list-style-type: none"> <li>• Zie bijlage</li> </ul>	
5 min	Doorstart 	<ul style="list-style-type: none"> <li>• Voorbereiding op Zelfstandig Trainen</li> </ul>

### Terugkoppeling: 40 minuten

TIJD	VORM	BESCHRIJVING OEFENING OF ONDERDEEL	
5 min	Opstart  	<ul style="list-style-type: none"><li>• Welkom heten</li><li>• Eventueel aandachtspunten / overdracht instructeur uit ZT-tijd</li><li>•</li></ul>	
30min	<b>ACTIVERENDE DIDACTIEK</b>	<b>ASSESSMENT OVER ALLE OF EEN DEEL VAN DE VAARDIGHEIDSDOELEN</b>	
	  	<b>DIDACTISCHE VORM</b> Maak hier een keuze	
	<b>NABESPREKEN TERUGKOPPELING:</b> <ul style="list-style-type: none"><li>• Feedback geven en ontvangen</li><li>• Laat studenten eerst zichzelf beoordelen</li><li>• Daarna wat de instructeur/docent zag</li><li>• En tot slot (indien nodig) nog aanvullen</li></ul>		
5 min	Afsluiting 	<ul style="list-style-type: none"><li>• Feedback op het project inhoudelijk</li><li>• Feedback voor de trainer: wat kan volgende keer beter, van welk moment of welke aanpak hebben de studenten het meest geleerd?</li><li>• Gebruik je gemaakte producten eventueel als bewijskaart.</li></ul>	

## Projectbeoordeling

Groep

Naam student	Naam student	
Projectgroep	1. Naam 1 <sup>e</sup> student	3. Naam 3 <sup>e</sup> student
	2. Naam 2 <sup>e</sup> student	4. Naam 4 <sup>e</sup> student
Datum beoordeling	11 april 2017	
Projectnaam /-nummer	De gokkers	
Periode en leerjaar	3	1
Procesbegeleider		
Afdeling	ICO	
Opleiding en crebo	Applicatie- en mediaontwikkeling	25187

#	WP	Niveau	Productcriteria	0	1
1.	B1-K1-W1-M	B	Opdracht wordt eenduidig beschreven		
2.					
3.					
4.	B1-K1-W2-E	B	Samenwerking krijgt professionele vormgeving		
5.	B1-K1-W2-Q	B	De taakverdeling is in een planning opgenomen		
6.	B1-K1-W3-M	B	Een aangeleverd functioneel ontwerp wordt gebruikt om een applicatie te maken		
7.	B1-K1-W3-U	B	Gegeven opdracht wordt aangepast		
8.	B1-K2-W1-L	B	Ontwikkelomgeving wordt ingericht		
9.	B1-K2-W1-M	B	Gewenste programma wordt gecodeerd		
10.	B1-K2-W1-S	B	Er is gecodeerd volgens gangbare conventies		
11.	B1-K2-W2-K	B	Er zijn een technische en functionele test opgesteld		
12.	B1-K2-W2-M	B	Technische en functionele test zijn uitgevoerd.		
13.	B1-K3-W1-D	B	Wensen van opdrachtgever bij acceptatietest worden aanvaard		
14.	B1-K3-W1-E	B	Afspraken m.b.t. op te leveren product zijn vastgelegd		
15.	B1-K3-W1-J	B	Er is een contract opgesteld		
16.	B1-K3-W1-K	B	Programma doorloopt de acceptatietest		
17.	B1-K3-W1-R	B	Specifieke latere wensen van de klant zijn in beeld gebracht		
18.	B1-K3-W2-I	B	Het product wordt gepresenteerd		
19.	B1-K3-W3-S	B	Het programma is technisch getest		
Productresultaat over dit project:					
Handtekening procesbegeleider/ opdrachtgever/ klant:			Handtekening voorzitter/ projectleider		

## OPERATIONALISATIE BEOORDELING PROJECT (product/proces)

<b>1</b> <b>B1-K1-W1-M</b> <b>B</b>	Opdracht wordt eenduidig beschreven
	Eisen:
	<ul style="list-style-type: none"> <li>Form design is vastgelegd</li> <li>Eisen aan het programma zijn vastgelegd</li> </ul>
	1 = 2 behaald
	• —
	• —
<b>4</b> <b>B1-K1-W2-E</b> <b>B</b>	Samenwerking krijgt professionele vormgeving
	Eisen:
	<ul style="list-style-type: none"> <li>samenwerkingscontract</li> <li>bereikbaarheidslijst</li> </ul>
	1 = 2 behaald
<b>5</b> <b>B1-K1-W2-Q</b> <b>B</b>	De taakverdeling is in een planning opgenomen
	Eis:
	<ul style="list-style-type: none"> <li>planning</li> </ul>
	1 = 1 behaald
<b>6</b> <b>B1-K1-W3-M</b> <b>B</b>	Een aangeleverd functioneel ontwerp wordt gebruikt om een applicatie te maken
	Eis:
	<ul style="list-style-type: none"> <li>aangepast functioneel ontwerp</li> </ul>
	1 = 1 behaald
<b>7</b> <b>B1-K1-W3-U</b> <b>B</b>	Gegeven opdracht wordt aangepast
	Eis:
	<ul style="list-style-type: none"> <li>Oorspronkelijke opdrachtbeschrijving wordt herschreven</li> </ul>
	1 = 1 behaald
<b>8</b> <b>B1-K2-W1-L</b> <b>B</b>	Ontwikkelomgeving wordt ingericht
	Eisen:
	<ul style="list-style-type: none"> <li>Visual Studio is geïnstalleerd</li> <li>Een tekstverwerker en een spreadsheetprogramma zijn geïnstalleerd</li> </ul>
	1 = 2 behaald
<b>9</b> <b>B1-K2-W1-M</b> <b>B</b>	Gewenste programma wordt gecodeerd
	Eisen:
	<ul style="list-style-type: none"> <li>Programmacode</li> <li>Door Visual Studio gegenereerd klassediagram</li> </ul>
	1 = 2 behaald
<b>10A</b> <b>B1-K2-W1-S</b> <b>B</b>	Er is gecodeerd volgens gangbare conventies
	Eisen:
	<ul style="list-style-type: none"> <li>Voldoende commentaarregels</li> <li>Inspringingen en witregels</li> <li>Logische engelstalige namen voor knoppen, klassen en methodes</li> </ul>
	1 = minimaal 2 behaald
<b>10B</b> <b>B1-K2-W1-S</b> <b>B</b>	Er is gecodeerd volgens gangbare conventies
	Eis:

	<ul style="list-style-type: none"> <li>• Er is gebruik gemaakt van klassen</li> </ul>
	1 = 1 behaald
<b>11</b> <b>B1-K2-W2-K</b> <b>B</b>	Er zijn een technische en functionele test opgesteld
	Eisen: <ul style="list-style-type: none"> <li>• Technische test</li> <li>• Functionele test</li> </ul>
	1 = 2 behaald
<b>12</b> <b>B1-K2-W2-M</b> <b>B</b>	Technische en functionele test zijn uitgevoerd.
	Eisen: <ul style="list-style-type: none"> <li>• Beide testen zijn volledig ingevuld</li> <li>• Er is aangegeven of het product de testen heeft doorstaan</li> <li>• Er is aangegeven door wie en wanneer de test is uitgevoerd (voor beide testen)</li> </ul>
	1 = 2 behaald
<b>13</b> <b>B1-K3-W1-D</b> <b>B</b>	Wensen van opdrachtgever bij acceptatietest worden aanvaard
	Eis: <ul style="list-style-type: none"> <li>• Verslag met daarin de gewenste aanvullingen/aanpassingen van de opdrachtgever die bij acceptatietest naar voren zijn gekomen.</li> </ul>
	1 = 1 behaald
<b>14</b> <b>B1-K3-W1-E</b> <b>B</b>	Afspraken m.b.t. op te leveren product zijn vastgelegd
	Eis: <ul style="list-style-type: none"> <li>• Contract waarin afspraken m.b.t. betaling en nazorg vermeld staan.</li> </ul>
	1 = 1 behaald
<b>15</b> <b>B1-K3-W1-J</b> <b>B</b>	Er is een contract opgesteld
	Eis: <ul style="list-style-type: none"> <li>• Contract dat voldoet aan algemeen gangbare richtlijnen v.w.b. opmaak en samenstelling documenten.</li> </ul>
	1 = 1 behaald
<b>16</b> <b>B1-K3-W1-K</b> <b>B</b>	Programma doorloopt de acceptatietest
	Eis: <ul style="list-style-type: none"> <li>• Acceptatietest is opgesteld</li> </ul>
	1 = 1 behaald
<b>17</b> <b>B1-K3-W1-R</b> <b>B</b>	Specifieke latere wensen van de klant zijn in beeld gebracht
	Eis: <ul style="list-style-type: none"> <li>• De opdrachtoomschrijving is uitgebreid met de toegevoegde wensen</li> </ul>
	1 = 1 behaald
<b>18</b> <b>B1-K3-W2-I</b> <b>B</b>	Het product wordt gepresenteerd
	Eis: <ul style="list-style-type: none"> <li>• Voldoende beoordeeld verslag van de presentatie</li> </ul>
	1 = 1 behaald
<b>19</b> <b>B1-K3-W3-S</b> <b>B</b>	Het programma is technisch getest
	Eis: <ul style="list-style-type: none"> <li>• Technische test</li> </ul>
	1 = 1 behaald

## Bijlage 1

### 1 Inleiding

In dit project ga je in groepjes van drie een nieuw programma maken. Alle benodigde kennis heb je al opgedaan tijdens de C#-lessen.

Dit project is groter en complexer dan het vorige ("Torens van Hanoi"). Lees dadelijk eerst t/m hoofdstuk 6 door voordat je begint. Neem hier echt de tijd voor!!

Het functioneel ontwerp hebben we al voor je gemaakt.

Aan jullie (omdat je in groepjes werkt) de taak om er een mooi, werkend product van te maken.

Heel veel succes ermee en plezier.

## 2 Het programma

Uit een al gehouden interview is de volgende uitgewerkte informatie naar voren gekomen.

### 2.1 Het wedbureau

Door het wedbureau wordt bijgehouden hoeveel saldo iedereen heeft en hoeveel hij/zij wedt. Het minimum bedrag waarmee gewed kan worden is 5 euro en het maximum bedrag is 15 euro. Er kan slechts met één weddenschap op één race meegedaan worden.

Uiteraard wordt ook bijgehouden of iemand het ingezette bedrag wel kan betalen.

### 2.2 Het wedden

Voor elke weddenschap geldt: de winnaar verdubbelt zijn inzet of hij verliest wat hij heeft ingezet.

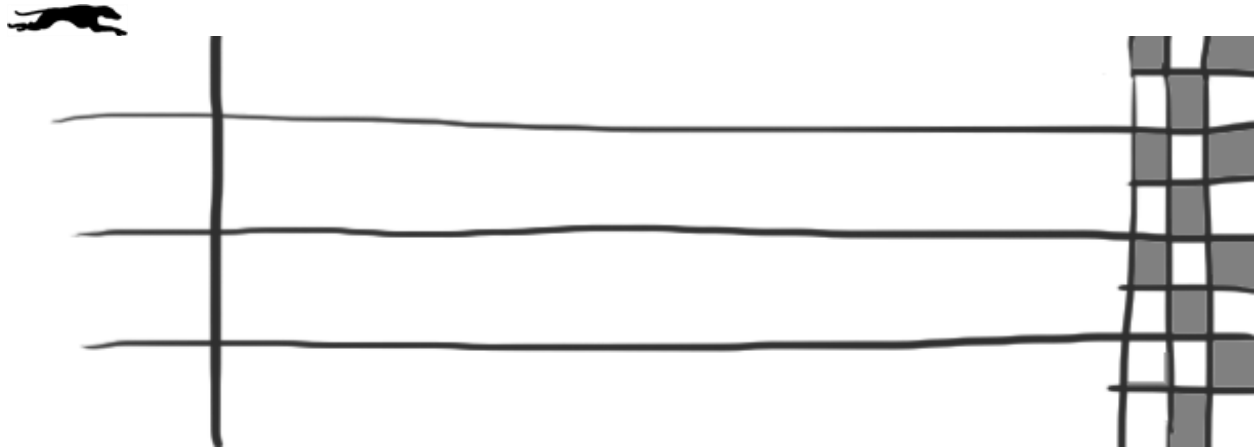
### 2.3 De race

Er zijn vier honden (Greyhounds) die een recht stuk racebaan af moeten rennen. De hond die als eerste over de finishlijn komt wint.

De race is volkomen willekeurig (Engels: random). Een hond die gewonnen heeft, heeft geen grotere kans om de volgende race te winnen dan de andere drie honden.

### 3. De lay-out van het formulier

De interface zal uit twee delen komen te bestaan; bovenin zien we de honden rennen en onderin kunnen we gegevens invoeren. Het moet er ongeveer uit komen te zien zoals hiernaast weergegeven. Het bovenste gedeelte bestaat uit de PictureBox *PictureBox1* en vier PictureBoxes *Dog* voor de honden. De benodigde afbeeldingen kun je downloaden van MySite (lesmateriaal [ICO](#) > [Leerjaar 1](#) > amo > p03 > pr > De gokkers ).



Mooier is natuurlijk als je zelf iets maakt zoals op de volgende pagina gedaan is. Het onderste gedeelte betreft het wedbureau. Hier kunnen de drie gokkers Lidy, Sietse en Fer hun weddenschap invoeren.



Elk van de vier honden heeft zijn eigen PictureBox control.

Alle drie personen kunnen wedden, alleen slechts één keer per race. Daarom wordt gebruik gemaakt van de radiobuttons.



Tip: Denk er bij het instellen van de grootte (SizeMode) van elke PictureBox aan om deze op *StretchImage* te zetten.

Wanneer je elk van de vier Greyhound objecten initialiseert, krijgt elk MyPictureBox veld een referentie naar een van deze objecten.

De referentie betreft de referentie van de lengte van de renbaan, de startpositie van elke hond naar de Greyhound's objects initializer.

Gebruik de property *Width* van de achtergrond om de racetrack-lengte in het Greyhound object te bepalen. Deze is daar weer nodig om te bepalen of hij de race heeft gewonnen.

Wanneer iemand een weddenschap plaatst, wordt de vorige overschreven. De huidige weddenschap wordt zichtbaar in de labels.

**Tip1:** Bij elk label staat de *AutoSize* op *False* en *BorderStyle* op *FixedSingle*.

#### 4 Het plaatsen van de weddenschappen

Gebruik de controls in het Wedbureau-vak voor het plaatsen van de weddenschappen. Er zijn drie verschillende mogelijkheden:

1<sup>e</sup>: Er zijn geen weddenschappen geplaatst

Wanneer het programma opstart, of als een race beëindigd is, zijn er geen weddenschappen geplaatst bij het wedbureau.

2<sup>e</sup> Iedereen plaatst zijn weddenschap

Om een weddenschap te plaatsen moet je de radiobutton van de betreffende persoon selecteren, een hond en een bedrag kiezen en op de 'wedt'-knop klikken. De gokker betaalt zijn inzet.

Een UpdateLabels() methode zal het label en de radioknop updaten:

The screenshot shows a betting interface with two radio buttons: 'Lidy heeft 75 euro.' (selected) and 'Fer heeft 45 euro.'. To the right, there are two text boxes: 'Lidy wedt 15 euro op hond 2' and 'Fer heeft geen weddenschap geplaatst.'. Below the radio buttons, there is a label 'Lidy' followed by a 'wedt' button, a numeric input field with '15', a label 'euro op hond nummer:', a numeric input field with '2', and a green 'Go!' button.

3<sup>e</sup> Na afloop van de race, ontvangt iedere gokker die gewonnen heeft zijn winst.

Als de race ~~gelopen is~~ teneinde is en er is een winnaar, wordt door elk Guy-object (gokker) zijn Collect() methode aangeroepen zodat zijn winst toegevoegd wordt of zijn inzet van zijn saldo afgehaald wordt.

## 5 Klassen en arrays

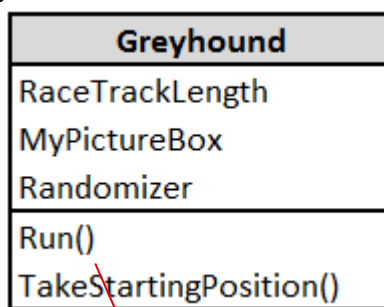
Naast de grafische interface voor de simulator heb je nog drie klassen nodig. Ook heb je een array van drie Guy objecten nodig om bij te kunnen houden hoe het met hun gokken gaat en een array van vier Greyhound objecten die de race gaan lopen. Iedere instantie van Guy heeft zijn eigen weddenschap-object die zijn weddenschap bijhoudt en zijn saldo.

### 5.1 De klasse Greyhound

We geven je hier de klasse Greyhound en een gedeelte van de benodigde code. Aan jullie de taak om het af te maken.

De Greyhound-klasse houdt het afgelegde stuk van de race bij. Het houdt dus de positie bij van de PictureBox die de hond vertegenwoordigt. Elke instantie van *Greyhound* gebruikt een veld genaamd *MyPictureBox* om naar PictureBox op het form te refereren. Deze PictureBox toont de afbeelding van de hond.

De commentaarregels in de methodes zijn bedoeld om je een beeld te geven wat er van je verlangd wordt. We geven je het raamwerk van de klasse die je moet gaan maken. Aan jullie om de genoemde methodes te definiëren.



```

Public class Greyhound
{
    public int RaceTrackLength; //De lengte van de renbaan
    public PictureBox MyPictureBox = null;
    public Random Randomizer; //Een instantie van Random (= Willekeurig)

    public bool Run()
    {
        //Ga willekeurig 1, 2, 3 of 4 posities naar voren.
        //Werk de positie van PictureBox bij op het formulier.
        //Geef de waarde 'true' terug als ik de race win.
    }

    public void TakeStartingPosition()
    {
        //Wijzig mijn locatie naar de startlijn.
    }
}
  
```

Zie je hoe het klassendiagram correspondeert met de code?

Je hebt maar één instantie van Random nodig: Elke Greyhound Randomizer referentie verwijst naar hetzelfde Random object.

De Greyhound object initializer is behoorlijk recht-toe, recht-aan. Zorg voor een referentie naar de juiste PictureBox op het formulier naar elk Greyhound object.

Dit is hoogstwaarschijnlijk makkelijker dan je denkt... Je hoeft maar één variabele te definiëren en je bent klaar.

## 5.1.1 De greyhound gaat rennen

De klasse Greyhound houdt de positie op de renbaan bij tijdens de race. Het werkt ook steeds de locatie van de PictureBox (die de hond vertegenwoordigt) bij. Elke instantie van Greyhound gebruikt een veld MyPictureBox om te refereren naar de PictureBox controller op het formulier die de afbeelding van de hond toont.

Maak een variabele 'distance' aan die de afstand bevat die de hond naar voren moet rennen. Je code zal dan de locatie van MyPictureBox updaten door de variabele distance aan de X-waarde toe te voegen:

```
MyPictureBox.Location = new Point(  
MyPictureBox.Location.X + distance, MyPictureBox.Location.Y);
```

Vraag de juiste locatie op, voeg waarde aan X-coördinaat toe en verplaats de afbeelding op het formulier.

Je moet er voor zorgen dat het formulier de juiste afbeelding aan de juiste hond toekent.

## 5.2 De klasse Guy

Guy
Name
MyBet
Cash
MyRadioButton
MyLabel
UpdateLabels()
PlaceBet()
ClearBet()
Collect()

Public class Guy

```

{
    public string Name; // De naam van de gokker
    public Bet MyBet;   // Een instantie van Bet()
    public int Cash;    // Het saldo van de gokker

    //Deze twee velden zijn de gokkers GUI controls op het formulier
    public RadioButton MyRadiobutton;
    public Label MyLabel;

    public void UpdateLabels()
    {
        //Verander mijn label in de omschrijving van mijn weddenschap.
        //Verander de label op mijn radioknop zodat deze mijn saldo laat zien.
        //(Bijv. "Lidy heeft 43 euro.")
    }

    public bool PlaceBet(int amount, int dog)
    {
        //Plaats een nieuwe weddenschap en sla het op in de variabele MyBet.
        //Retourneer een true als de gokker genoeg geld heeft om te wedden.

        //Onderstaande regel staat er tijdelijk om foutmeldingen te voorkomen.
        //Haal deze regel later weg.
        return true;
    }

    public void ClearBet()
    {
        //Maak de weddenschap leeg.
    }

    public void Collect(int Winner)
    {
        //Betaal mijn weddenschap uit.
        //Maak mijn weddenschap leeg.
        //Werk mijn labels bij.
    }
}

```

Als je MyLabel aan één van de labels op het formulier hebt toegekend, ben je in staat om de tekst van het label aan te passen d.m.v. MyLabel.Text. Hetzelfde geldt voor MyRadioButton!

Wanneer je het Guy-object initialiseert, zorg er dan voor dat het veld MyBet leeg is en roep zijn UpdateLabels() methode aan.

Denk eraan dat weddenschappen gerepresenteerd worden door instanties van Bet.

Tip: Gebruik het Bet-object...Laat deze het werk doen.

### 5.3 De klasse Bet

Bet
Amount
Dog
Bettor
GetDescription
PayOut

Tip: Instantieer Bet in de Guy klasse. Guy zal het this-keyword gebruiken voor een referentie naar zijn weddenschap.

Public class Bet

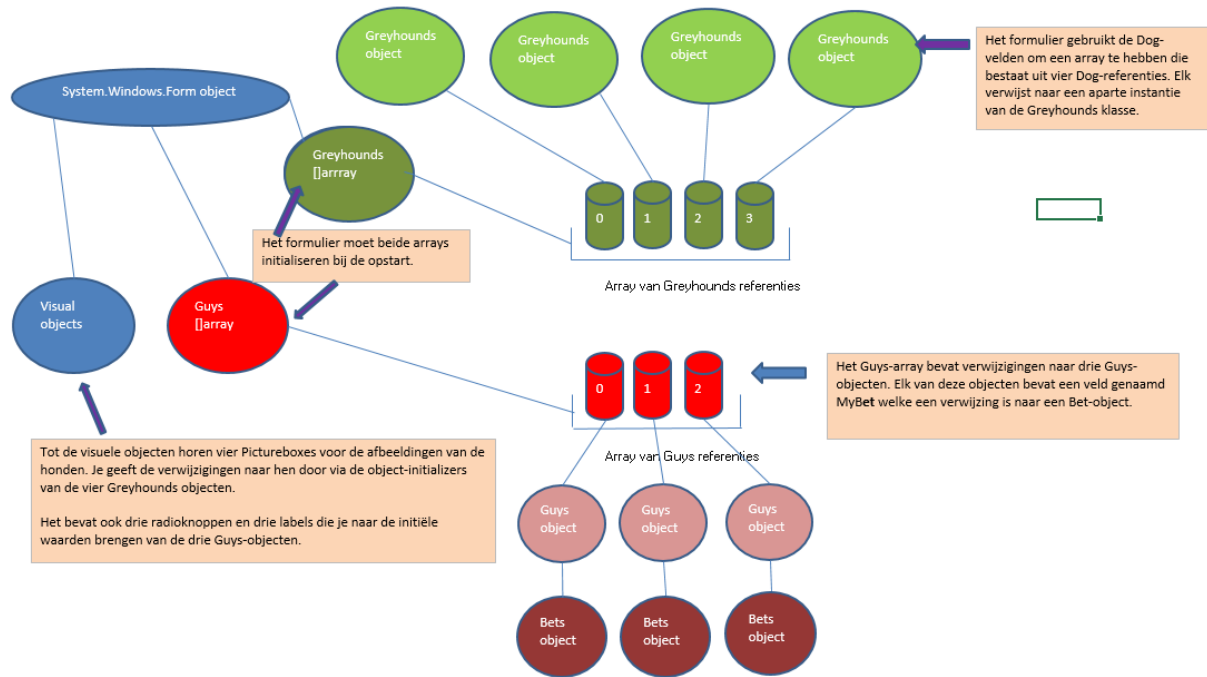
```
{
    public int Amount; //Het bedrag van de weddenschap.
    public int Dog;    //Het nummer van de hond waarop weddenschap is afgesloten.
    public Guy Bettor; //De gokker die gewed heeft.

    public string GetDescription()
    {
        //Retourneer een string die aangeeft wie de weddenschap heeft gedaan,
        //voor welk bedrag er is gewed en op welke hond er is gewed.
        //Bijv. "Sietse wedt 8 euro op hond 4".
        //Als het bedrag 0 is, is er geen weddenschap geplaatst.
        //De string die dan geretourneerd wordt is bijv.
        //"Sietse heeft geen weddenschap geplaatst.")
    }

    public int PayOut(int Winner)
    {
        //De parameter van deze methode is de winnaar van de race.
        //Als de hond gewonnen heeft, retourneer dan het bedrag dat gewed is
        //Anders, retourneer het tegengestelde van het gewedde bedrag.
    }
}
```

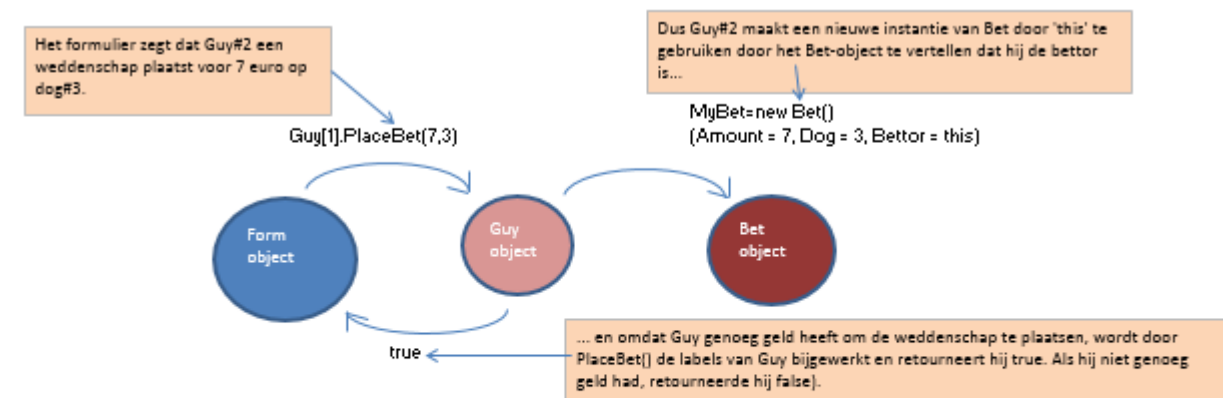
## 6 Het logisch ontwerp

Hieronder vind je een afbeelding van een gedeelte van het logisch ontwerp. Dit ziet er in eerste instantie ingewikkeld uit. Toch staat er niets in wat je niet kent. Neem er je tijd voor om het te bestuderen.



Vertaal dan dit ontwerp in code. Begin met de Greyhound- en Guy arrays op je hoofdformulier.

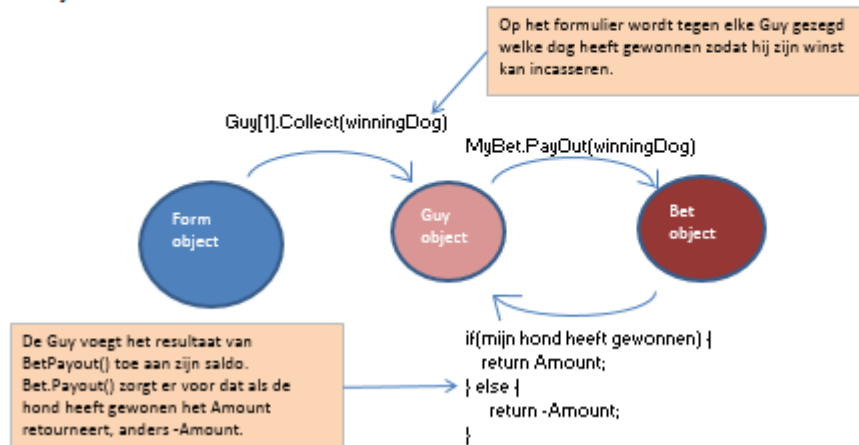
Bij het plaatsen van een weddenschap wordt een nieuw Bet-object gemaakt



Het formulier zegt de honden door te rennen totdat er een winnaar is.



Het Bet-object zoekt uit of het moet uitbetalen.





## 7 En nu aan de slag

Om dit programma te maken kun je kiezen om eerst te beginnen met het formulier-ontwerp of eerst voor het klassenontwerp. Wij stellen voor dat je voor het eerste kiest. Dit stemt dan nl. overeen met je projectbeoordeling (zie verderop).

We raden je aan om je programma van veel gepaste commentaarregels te voorzien. Nogmaals, veel succes.

Je zult diverse documenten moeten maken. Deze dienen per project een uniforme lay-out te hebben. Maak dus een 'sjabloon' wat je telkens kunt gebruiken. Het dient in ieder geval aan de volgende eisen te voldoen:

- Het bevat een kaft, met daarop de naam van het project, de datum, de namen van degenen die het gemaakt hebben, eventueel een plaatje of logo, de plaatsnaam waar het gemaakt is.
- Inhoudsopgave die automatisch is gegenereerd.
- Bladzijdenummering waarbij geldt dat de kaft geen nummer heeft en de pagina daarna nr. 1
- Als het document meerdere hoofdstukken bevat, begint elk hoofdstuk op een nieuwe bladzijde. De hoofdstukken zijn genummerd en hebben een grote letter, bijv. puntgrootte 18. Paragrafen hebben ook een nummer en een iets kleinere letter, bijv. puntgrootte 14.
- Er staan geen taal- of *spelvauten* in. Maak dus in ieder geval gebruik van de spellingscontrole. Een opmerking als 'ik heb dyslexie' gaat niet op. Het document dient er gewoon goed uit te zien. Zorg ook voor goed lopende zinnen.
- Een koptekst met daarin per hoofdstuk de naam van het hoofdstuk (indien je hoofdstukken hebt uiteraard).
- Een voettekst met daarin in kleine letters de naam van het project en de namen van de groepsleden.

Uiteraard heb je verder nog wat mogelijkheden om het document te verfreaaien. Denk hierbij aan een rand, een lijn boven de voettekst, opnemen van illustraties, gebruik van pijlen, een index etc. Wellicht is het zinnig om versiebeheer toe te passen....

Bekijk de bladzijde Operationalisatie beoordeling project (product/proces). Hier staan de eisen waar je aan dient te voldoen. In het derde jaar moet je dit zelfstandig kunnen. Nu zullen we je er nog enigszins mee helpen. Bewaar alles wat je produceert. Dit zal op deze lijst afgetekend worden als beoordeling van dit project

## 1 Opdracht wordt eenduidig beschreven

- Maak met potlood een prototype van de interface van je te maken programma.
- Maak een document Opdrachtbeschrijving waarin je alle eisen van je programma zet (De zogenaamde functional requirements). Gebruik bijvoorbeeld de MoSCoW methode: <https://nl.wikipedia.org/wiki/MoSCoW-methode>.

## 2 Samenwerking krijgt professionele vormgeving

Jullie gaan afspraken maken hoe jullie gaan samenwerken (Dit is een gedeelte van het Plan van aanpak, wat je bij het volgende project zult gaan maken.) Zorg er ook voor dat je weet hoe je elkaar kunt benaderen. Gebruik hiervoor de twee sjablonen die je van Mysite kunt downloaden (Samenwerkingscontract en bereikbaarheidslijst.) Stel het samenwerkingscontract zo concreet mogelijk op, zodat er geen speld tussen te krijgen is. Gebruik van woorden als 'op tijd', 'mooi', 'goed je best doen' zijn niet duidelijk genoeg. Wat is 'op tijd'? Wat is 'mooi'? etc. Houd vanaf het begin een roostertje bij wie wanneer aanwezig is. Je dient dit op het eind van het project bij het projectrapport in te leveren.

## 3 De taakverdeling is in een planning opgenomen

Om alles in goede banen te leiden en te zorgen dat alles afkomt dien je een planning te maken. En belangrijker nog..... je er aan te houden! Hierin staat tevens de taakverdeling. Je mag zelf weten hoe je dit aanpakt, als het maar een planning is waar je goed mee kunt werken. Je zou het bijv. in Excel kunnen maken. Een professioneel programma is MS-Project wat je kunt downloaden van RCmsdn.

We geven hier al wel mee dat je ongeveer in week 6 met alles klaar moet zijn. Je gaat dan de acceptatietest doen. Ga er vanuit dat de opdrachtgever nog wat wensen of aanpassingen wil en dat je die ook uit gaat voeren. Ga ook nu weer professioneel te werk. Denk van te voren na over zaken als: doe je het?, doe je het gratis?, Vraag je er geld voor?, Jouw fout, zijn fout?

Op deze manier gaat het verder. Dus:

- de kleine afwijking t.o.v. andere groepen wordt genoteerd,
- functioneel ontwerp wordt indien nodig aangepast, (Je kunt dit van MySite downloaden)
- je gaat je spullen klaarzetten,
- gaat (netjes) programmeren,
- laat Visual Studio een klassediagram genereren,
- laat een ander groepje een functionele test uitvoeren,
- weer een ander groepje de technische test (Wordt behandeld tijdens een VHT),
- het (positieve) resultaat hiervan wordt gemeld bij de opdrachtgever,
- er wordt een contract opgesteld,
- er wordt een factuur verzonden,

Ga er vanuit dat er voor het programma betaald moet worden.

- Hoe hoog is dit bedrag?
- Hoe kom je tot dit bedrag? Ga er bij je uurtarief vanuit dat je een eerste jaars leerling Applicatieontwikkelaar bent. Je mag dus zeker niet het tarief hanteren wat een ervaren programmeur zou hanteren.
- Hoe dient dit bedrag betaald te worden? Ineens of in delen? Handje contantje of via overschrijving?
- Hoe breng je je opdrachtgever hiervan op de hoogte? Doe dit dus ook.

- acceptatietest wordt uitgevoerd, Je moet proberen alles zo reëel mogelijk uit te voeren. We bedoelen hiermee dat je je programma zogenaamd bij een echte opdrachtgever op gaat leveren. Behandel hem dan ook als zodanig. Probeer professioneel over te komen.

Maak een acceptatietest en laat deze door (een gebruiker van) de opdrachtgever uitvoeren. Maak hiervoor (ruim op tijd) een afspraak. Doe dit volgens de eisen zoals gesteld bij de VHT.

Ga er hierbij vanuit dat de opdrachtgever meer dan 100 km van je vandaan zit. Zorg er dus voor dat je alles op orde hebt om je product netjes af te leveren. Je wordt er niet vrolijk van als je een nieuwe afspraak moet maken.

De opdrachtgever komt zeer hoogstwaarschijnlijk met een of meer wensen/aanpassingen etc. (Dit met het doel dat jij goed leert programmeren.) De bedoeling van dit 'spelletje' is dat je hierin meegaat. (Dit met het doel om te leren goed om te gaan met een klant.)

Normaal gesproken zou jij als ontwikkelaar/leverancier een rapportage opleveren. Nu draaien we de rollen net om en krijg je deze van de opdrachtgever zodat je ongeveer weet hoe dat er uit hoort te zien. Bij een volgend project zul je die zelf op moeten stellen.

**Houd er rekening mee dat de opdrachtgever het product mogelijk niet accepteert en jij nog aanpassingen moet doen. Je zult dus een tweede afspraak moeten maken. Houd rekening met de deadline. Deze is 4 april 2017.**

Als je applicatie de acceptatietest passeert krijg je een testrapport daarop de stempel Accepted:



mee met

- specifieke wensen die daarbij naar voren kwamen worden genoteerd.

Het testen verloopt volgens dit schema:

Functionele test		Technische test	
Groep	Laat testen door	Groep	Laat testen door
1	3	1	2
2	4	2	3
3	5	3	4
4	6	4	5
5	7	5	6
6	8	6	7
7	9	7	8
8	10	8	9
9	11	9	10
10	12	10	11
11	13	11	12
12	14	12	13
13	15	13	14
14	16	14	15
15	17	15	16
16	18	16	17
17	19	17	18
18	20	18	19
19	21	19	20
20	1	20	21
21	2	21	1

## 4 Afronding van het project

De afronding van het project bestaat uit twee gedeelten, zoals je het kent van het vorige project "De torens van Hanoi". Uiterlijk dinsdag 4 april 2017 kun je het Projectrapport en Teamrapport inleveren. Bij het projectrapport dien je een rooster toe te voegen met daarop aangegeven wie er wanneer aanwezig was.

De afronding vindt plaats in de toetsweek.

- Bij de Technische oplevering worden jullie als groep getoetst op jullie individuele kennis van het project.  
Dit telt voor 55% mee in het eindresultaat.  
Ook jullie peilinglijst wordt nu administratief verwerkt.  
Deze lijst telt voor 30% mee in het eindresultaat.
- Bij de Presentatie verwachten we een 15-minuten-durende presentatie. De facetten waar bij de beoordeling op gelet wordt, vind je als bijlage. (Actuele versie staat op MySite.)  
De presentatie telt voor 15% mee in het eindresultaat.

## Bijlage 2, Protocol presentatie

<u>Stemgebruik en tempo</u>
Articulatie
Volume
Snelheid
<u>Contact met publiek</u>
Houding
Opvangen signalen publiek
Iedereen van de groep is aan het woord geweest.
Er is met powerpoint gewerkt.
Er is veel met plaatjes gewerkt.
Er is aandacht besteed aan 'storende zaken'. Denk hierbij aan verduistering, beschreven bord, onduidelijk spreken, open deur (storende geluiden op de gang), collega's die op een of andere manier aandacht vragen, te veel tekst op dia's etc. etc.
Er is een korte inleiding.
Elk item wordt ingeleid en afgesloten voordat naar het volgende item wordt gegaan.
Er is aan het begin aangegeven of er tussendoor vragen gesteld mogen worden of juist niet.
De presentatie wordt ingeleid door o.a. iedereen voor te stellen.
Aan het begin wordt het 'programma' aangegeven
Er wordt aangegeven:
Tegen welke problemen men is aangelopen.
Hoe de problemen zijn opgelost.
Hoe de Technische test is verlopen
Hoe de Functionele test is verlopen.
Hoe de Acceptatietest is verlopen.
Hoe de samenwerking is verlopen.
Hoe de aanwezigheid van de verschillende groepsleden was.
Welke specifieke eisen zijn opgelegd.
Hoe de specifieke eisen zijn aangepakt.
Hoe het project ervaren is.
Wat er geleerd is.
Het geheel komt professioneel over
De presentatie is boeiend
Het verhaal is afgestemd op publiek (opdrachtgever)