



Provinsie van die
OOS-KAAP
ONDERWYS

Steve Vukile Tshwete Onderwyskompleks • Sone 6 Zwelitsha 5608 • Privaatsak X0032 • Bhisho 5605
REPUBLIEK VAN SUID-AFRIKA

HOOFDIREKTORAAT – KURRIKULUMBESTUUR

**GRAAD 12 LEERDER
ONDERSTEUNINGSPROGRAM**

**HERSIENING EN REMEDIËRENDE ONDERRIG
INSTRUMENT:
VRAE EN ANTWOORDE**

VAK: INLIGTINGSTEGNOLOGIE – EERSTE VRAESTEL

Junie 2009

Hierdie dokument bestaan uit 14 bladsye.

Streng gesproke nie vir toets/eksamen doeleindes nie.

INSTRUKSIES EN INLIGTING

1. Hierdie is 'n drie-uur-eksamen. As gevolg van die aard van hierdie eksamen is dit belangrik om kennis te neem dat jy nie toegelaat sal word om die eksamenlokaal voor die einde van die eksamensessie te verlaat nie.
2. Jy het die onderstaande lys lêers nodig om die vrae te beantwoord. Dit sal op 'n stiftie of 'n CD aan jou verskaf word of die toesighouer/opvoeder sal vir jou sê waar om dit op die hardeskyf van die werkstasie wat jy gebruik, te vind of in watter netwerklêergids ('network folder') dit is:

<i>P1 Data-lêergids</i>		
<i>Q1-lêergids</i>	<i>Q2-lêergids</i>	<i>Q3-lêergids</i>
SQL_P.dpr	TestGames_P.dpr	Simulation_P.dpr
SQL_U.dfm	TestGames_U.dfm	Simulation_U.dfm
SQL_U.pas	TestGames_U.pas	Simulation_U.pas
SuperRugby.mdb	Games_U.pas	Spectators.txt
	Results.txt	

3. Indien jy die lêers op 'n CD of 'n stiftie ontvang het, skryf jou naam en eksamennommer op die etiket.
4. Stoor jou werk met gereelde tussenposes as 'n voorsorgmaatreël teen kragonderbrekings.
5. Verander die naam van die *P1 Data-lêergids* na jou eksamennommer (of jou naam indien jou eksamennommer nie beskikbaar is nie), byvoorbeeld 3020160012.
6. Tik jou eksamennommer (of jou naam indien jou eksamennommer nie beskikbaar is nie) in as 'n kommentaar in die eerste reël van elke program.
7. Lees AL die vrae aandagtig deur. Doen slegs wat vereis word.
8. Aan die einde van hierdie eksamensessie moet jy die stiftie of CD met al die lêers met die werk wat jy daarop gedoen het, inlewer of jy moet seker maak dat AL die lêers met jou werk op die netwerk gestoor is soos deur die toesighouer/opvoeder aan jou verduidelik is.
9. Maak seker dat AL die lêers gelees kan word.
10. Jy mag die HELP-funksies van die sagteware gedurende die eksamen gebruik. Jy mag NIE van enige ander hulpbronne gebruik maak nie.

SCENARIO:

Die vrae in hierdie vraestel handel oor verskeie aspekte van die Super 12/14 rugby kompetisie waaraan spanne van Suid Afrika, Australië en Nieu-Seeland deelneem.

VRAAG 1: DELPHI-PROGRAMMERING EN DATABASIS

Die databasis, *SuperRugby.mdb*, wat in die Q1-lêergids aan jou voorsien is, bevat besonderhede van spanne en resultate van vorige seisoene van die Super 12/14 rugby kompetisie. Die *SuperRugby-databasis* bevat twee tabelle genaamd *tblTeams* en *tblData*.

Die *tblTeams*-tabel bevat data aangaande spanne wat aan die kompetisie deelneem. Die velde in hierdie tabel is soos volg gedefinieer:

Naam van Veld	Tipe	Beskrywing	Voorbeeld
Team	Text	Die kode van die span wat uit twee letters bestaan wat die land identifiseer (AU vir Australië, SA vir Suid-Afrika en NZ vir Nieu-Seeland) gevolg deur die nommer van die span.	AU1, SA5 of NZ4
Name	Text	Die naam van die span.	Blues, Sharks of Reds
Ground	Text	Die tuisveld van die span.	Newlands, Cape Town
AKA	Text	Ander name waaronder die span bekend staan.	Cats, Transvaal, Gauteng Lions, Golden Cats

Die volgende tabel is 'n voorbeeld van die data wat in die tabel met die naam *tblTeams* in die databasis met die naam *SuperRugby* voorkom.

Team	Name	Ground	AKA
NZ1	Blues	Eden Park, Auckland	Auckland Blues
AU1	Brumbies	Canberra Stadium, Canberra	ACT Brumbies
SA1	Bulls	Loftus Versfeld, Pretoria	Northern Transvaal, Blue Bulls
SA2	Cheetahs	Vodacom Park, Bloemfontein	Orange Free State Cheetahs
NZ2	Chiefs	Waikato Stadium, Hamilton	Waikato Chiefs
NZ3	Crusaders	Jade Stadium, Christchurch	Canterbury Crusaders
AU4	Waratahs	Aussie Stadium, Sydney	NSW Waratahs

Data aangaande resultate van sommige van die vorige wedstryde wat in die kompetisie gespeel is, word in die tabel *tblData* gestoor. Die velde in hierdie tabel is soos volg gedefinieer:

Naam van Veld	Tipe	Beskrywing
Season	Number	Die jaar of seisoen waarin die wedstryd gespeel is.
Placing	Number	Die finale posisie van die span op die puntelys vir die seisoen.
Team	Number	Die kode van die span wat uit twee letters bestaan wat die land identifiseer (AU vir Australië, SA vir Suid-Afrika en NZ vir Nieu-Seeland) gevolg deur die nommer van die span
Won	Number	Die aantal wedstryde wat die span gewen het in die seisoen.
Drawn	Number	Die aantal wedstryde waar die span gelykop gespeel het in die seisoen.
Lost	Number	Die aantal wedstryde wat die span verloor het in die seisoen.
PF	Number	Die aantal punte wat die span aangeteken het in die seisoen.
PA	Number	Die aantal punte wat teen die span aangeteken is in die seisoen.
TF	Number	Die aantal drieë wat die span aangeteken het in die seisoen.
TA	Number	Die aantal drieë wat teen die span aangeteken is in die seisoen.
Points	Number	Die aantal punte wat die span op die puntelys behaal het vir die seisoen.

Die volgende tabel is 'n voorbeeld van die data wat in die tabel met die naam *tblData* in die databasis met die naam *SuperRugby* voorkom.

Season	Placing	Team	Won	Drawn	Lost	PF	PA	TF	TA	Points
1998	2	NZ3	8	0	3	340	260	37	31	41
1999	11	SA3	4	0	7	312	341	37	38	22
2000	6	NZ1	6	0	5	300	262	37	27	30
2001	2	SA4	8	0	3	322	246	34	24	38
2002	1	NZ3	11	0	0	469	264	55	29	51
2004	9	NZ4	4	1	6	299	347	36	46	26
2006	9	NZ4	6	0	7	228	276	21	31	27
2006	11	SA5	4	1	8	263	334	25	37	23
2007	4	NZ1	9	0	4	355	235	41	20	42

Let op die volgende:

- 'n Een-tot-baie-verwantskap is reeds tussen die twee tabelle geskep.
- Jy is ook voorsien van 'n onvolledige Delphi-program in die lêergids ('folder') met die naam *Q1* waarin 'n eenheid ('unit') met die naam *SQL_U* en 'n projek met die naam *SQL_P* voorkom.
- Die program vertoon agt knoppies ('buttons') sowel as 'n DBGrid wat as 'n afvoerkomponent gebruik gaan word.
- Die program koppel outomaties met die databasis *SuperRugby* in die *Q1*-lêergids.
- Punte sal slegs toegeken word vir die programmeringskode in die eenheid ('unit') met die naam *SQL_U* wat die SQL-stellings bevat.
- Voltooi die program deur die nodige SQL-stellings in die knoppies met die name *btn1_1*, *btn1_2* ens. onderskeidelik te formuleer soos aangedui in VRAE 1.1 tot 1.8.

- 1.1 Voltooi die kode in die knoppie ('button') met die naam *btn1_1* deur 'n SQL-stelling te formuleer wat al die velde in die *tblData*-tabel sal vertoon, gesorteer in dalende volgorde volgens die seisoen (*Season*) en dan in stygende volgorde volgens die span (*Team*). Plaas die stelling in die toepaslike reël in die programkode. Voorbeeld van die afvoer:

	Season	Placing	Team	Won	Drawn	Lost	PF	PA	TF	TA	Points
	2008	9	AU1	6	0	7	277	317	36	38	30
	2008	8	AU2	7	0	6	247	278	26	34	33
	2008	12	AU3	3	1	9	258	323	32	37	18
	2008	2	AU4	9	1	3	255	186	33	21	43
	2008	6	NZ1	8	0	5	354	267	41	33	40
	2008	7	NZ2	7	0	6	348	349	44	47	34
	2008	1	NZ3	11	0	2	369	176	49	20	52
	2008	11	NZ4	3	0	10	257	338	29	41	19
	2008	4	NZ5	8	1	4	310	204	40	23	41
	2008	10	SA1	6	0	7	324	347	34	44	28
	2008	13	SA2	1	0	12	255	428	34	57	13
	2008	14	SA3	2	1	10	206	367	23	44	12
	2008	3	SA4	9	1	3	271	209	32	23	42
	2008	5	SA5	8	1	4	269	211	30	21	41
	2007	5	AU1	9	0	4	234	173	23	16	40
	2007	7	AU2	6	1	6	276	292	28	33	32
▶	2007	14	AU3	2	0	11	201	438	15	52	11
	2007	13	AU4	3	1	9	266	317	24	37	21

(5)

- 1.2 Voltooi die kode in die knoppie met die naam *btn1_2*, deur 'n SQL-stelling te formuleer wat die *Season*, *Team* en *Placing* velde sal vertoon vir spanne wat nie Suid-Afrikaans is nie, vir die 2006-seisoen en later. Plaas die stelling in die toepaslike reël in die programkode. Voorbeeld van die afvoer:

	Season	Team	Placing
	2008	NZ3	1
	2008	AU4	2
	2008	NZ5	4
	2008	NZ1	6
	2008	NZ2	7
	2008	AU2	8
	2008	AU1	9
	2008	NZ4	11
	2008	AU3	12
	2007	NZ3	3

(5)

- 1.3 Voltooi die kode in die knoppie ('button') met die naam *btn1_3* deur 'n SQL-stelling te formuleer wat 'n nuwe rekord in die *tblTeams*-tabel met die waardes 'SA6', 'Eastern Province', 'Telkom Park' en 'Mighty Elephants' sal invoeg. Plaas die stelling in die toepaslike reël in die programkode. (3)
- 1.4 Sommige van die inskrywings in die veld met die naam *Team* in die *tblData*-tabel is verkeerdelik as 'AU5' ingetik. Voltooi die kode in die knoppie met die naam *btn1_4* deur 'n SQL-stelling te formuleer om al die inskrywings van 'AU5' na 'AU4' te verander. Die kode vir die inlees van die twee parameters is vir jou gegee. Plaas die stelling in die toepaslike reël in die programkode. (4)
- 1.5 Voltooi die kode in die knoppie met die naam *btn1_5* wat die gebruiker vra om die span (byvoorbeeld SA1) en minimum posisie van die span op die puntelys (byvoorbeeld 4) as toevoer in te tik. Formuleer 'n SQL-stelling om die *Season* en *Placing*-velde van die betrokke rekords te vertoon. Plaas die stelling in die toepaslike reël in die programkode. Voorbeeld van die afvoer:

(SA1 as die span en 4 as die minimum posisie)

	Season	Placing
▶	2007	2
	2006	4
	2005	3

(4)

- 1.6 Voltooi die kode in die knoppie met die naam *btn1_6*, deur 'n SQL-stelling te formuleer wat die *Season*, *Placing*, *TF*, *Drawn*, *Team* en *Name*-velde van die rekords van spanne wat tussen 15 en 30 (inklusief) drieë in 'n seisoen aangeteken het en wat in meer as een wedstryd in die seisoen gelykop gespeel het.

LW: Jy sal die tabelle met 'n geskikte 'Where'-klousule moet koppel om hierdie opdrag te kan uitvoer. Voorbeeld van die afvoer:

	Season	Placing	TF	Drawn	team	Name
	2000	11	19	2	SA1	Bulls
	2006	14	26	2	AU2	Force

(8)

1.7 Voltooi die kode in die knoppie met die naam *btn1_7*, deur 'n SQL-stelling te formuleer wat die *Season*, *Team*, *Won*, *Drawn*, *Lost* en *Win Percentage*-velde vertoon.

Win Percentage is 'n berekende veld wat die persentasie van die wedstryde wat 'n span in 'n seisoen gewen het, aantoon. Die waardes in die *Win Percentage*-veld moet tot een desimale plek afgerond word. Voorbeeld van die afvoer:

Season	Team	Won	Drawn	Lost	Win Percentage
2008	NZ3	11	0	2	84.6
2008	AU4	9	1	3	69.2
2008	SA4	9	1	3	69.2
2008	NZ5	8	1	4	61.5

(7)

1.8 Voltooi die kode in die knoppie met die naam *btn1_8*, deur 'n SQL-stelling te formuleer wat 'n tabel van die meeste aantal drieë wat deur enige span aangeteken is per seisoen, sal vertoon. Voorbeeld van die afvoer:

Season	Max Tries
1998	49
1999	37
2000	48
2001	39
2002	55
2003	53
2004	55
2005	61
2006	47
2007	47
2008	49

(4)
[40]

- 2.2 Doen die volgende in die *TestGames_U*-lêer (die hoof-eenheid) in die gegewe program:
- 2.2.1 Skep 'n skikking met die naam *arrGames* wat 600 objekte van *TGame* kan stoor. (2)
- 2.2.2 Voltooi die kode in die *ReadTextFile*-prosedure wat die inligting uit die tekslêer *Results.txt* (gestoor in die Q2-lêergids) volgens die volgende stappe sal lees:
- (a) Toets of die tekslêer *Results.txt* bestaan. As die lêer nie bestaan nie, moet 'n toepaslike foutboodskap vertoon word en die program moet beëindig word. (4)
 - (b) Maak die tekslêer oop en inisieer 'n lus om die data te lees. (3)
 - (c) Lees 'n lyn van teks uit die tekslêer. (1)
 - (d) Verdeel die teks in die seisoen, die twee spanne en die twee spanne se tellings vir die wedstryd. (12)
 - (e) Gebruik 'n globale teller-veranderlike *iNumber* om boek te hou van die aantal items wat in die skikking is. (2)
 - (f) Gebruik hierdie inligting om 'n nuwe objek in die skikking *arrGames* te plaas deur van die konstruktorklasse wat jy in VRAAG 2.1.3 geskep het, gebruik te maak. (3)
- 2.3 Voeg die volgende metodes by die *TGame*-klas wat jy in die *Games_U*-eenheid gedefinieer het:
- 2.3.1 Skryf 'n string-tipe metode ('n string-funksie) met die naam *toString* wat die besonderhede van 'n wedstryd as een string soos volg uiteengesit, terugstuur:
- Season: Team1 vs Team2 Score1 - Score2
- Voorbeeld van die stringe vir die eerste twee wedstryde in die tekslêer:
- 1998: NZ4 vs AU3 26 – 19
1998: SA5 vs NZ5 31 – 45 (6)
- 2.3.2 Skryf 'n metode met die naam *CalcPointsDiff* om die punteverskil tussen die twee spanne in 'n wedstryd te bereken. Die verskil moet altyd as 'n positiewe getal vertoon word. (3)

2.3.3 Skryf 'n metode met die naam *CalcPointsTeam1* om die punte wat aan 'Team1' toegeken word per wedstryd te bereken, gebaseer op die volgende parameters:

- Vier punte word vir 'n wen toegeken.
- Twee punte word vir 'n gelykopuitslag toegeken.
- Een punt word toegeken vir 'n verlies van 7 of minder punte.

Die volgende tabel gee voorbeelde van hoe hierdie puntstelsel toegepas word:

Team1	Team2	Score1	Score2	Punte toegeken aan Team 1
SA1	AU3	20	8	4 vir 'n wen
NZ2	SA4	12	12	2 punte vir 'n gelykopuitslag
AU1	NZ5	12	30	0 punte
SA4	AU2	10	16	1 vir 'n verloor van 7 of minder punte

2.4 Gebruik die relevante metodes wat jy in VRAAG 2.3 definieer het om die kode vir die *ListResults*-prosedure wat in die *btnListResults*- gebeurtenishanteerder ('event handler') in die *TestGames_U*-eenheid geroep word.

Die prosedure moet die besonderhede van al die wedstryde wat in die *arrGames* skikking gestoor is, vertoon in die volgende formaat in die memo-kontrole wat voorsien word:

1998: NZ4 vs AU3 26 - 19	PD = 7	Points for Team 1 : 4
1998: SA5 vs NZ5 31 - 45	PD = 14	Points for Team 1 : 0
1998: NZ2 vs NZ3 25 - 25	PD = 0	Points for Team 1 : 2
1998: AU4 vs AU1 32 - 7	PD = 25	Points for Team 1 : 4
1998: SA4 vs NZ1 24 - 8	PD = 16	Points for Team 1 : 4
1998: SA1 vs SA3 32 - 39	PD = 7	Points for Team 1 : 1

Let Wel: Jy kan die prosedure *SetupDummyData* gebruik as jy nie die *ReadTextFile*- prosedure in VRAAG 2.2.2 suksesvol kon skryf nie. Hierdie kode sal vyf uitslae in die skikking *arrGames* laai waarmee jy kan werk. (10)

2.5 Voltooi die kode vir die *WinningAverage*-prosedure wat in die *btnAverageWinningScore*- gebeurtenishanteerder ('event handler') geroep word. Hierdie prosedure moet die gemiddelde telling van die wenspanne in alle wedstryde bereken, en vertoon, korrek tot twee desimale plekke, asook die aantal wedstryde wat betrokke was.

Gebruik die memo-kontrole wat voorsien word. Hierdie berekening sluit alle wedstryde waar die spanne gelykop gespeel het, uit.

Let op die volgende:

- Jy moet eers metodes by die *TGame*-klas in die *Games_U*-eenheid byvoeg om die tellings van die twee spanne te kry. Noem hierdie metodes *GetTeamScore1* en *GetTeamScore2*.
- Jy kan die prosedure *SetupDummyData* gebruik as jy nie die *ReadTextFile*- prosedure in VRAAG 2.2.2 suksesvol kon skryf nie. Hierdie kode sal vyf uitslae in die skikking *arrGames* laai waarmee jy kan werk.

Voorbeeld van die afvoer:

Gemiddelde telling van wenspanne in 497 wedstryde = 33.98

(18)

2.6 Die datavasleggers het per ongeluk van die spanne omgeruil.

2.6.1 Voeg 'n metode *SwopTeams*(*pTeam1* , *pTeam2* : *String*) by die *TGame*-klas in die *Games_U*-eenheid wat die twee spanne sal omruil indien hulle met enige van die twee parameters ooreenstem (*pTeam1* met *PTeam2* en omgekeerd). As die metode byvoorbeeld met die parameters 'SA1' en 'SA2' geroep word, sal die volgende veranderinge aangebring moet word:

(11)

Voor die omruiling		Na die omruiling	
Team1	Team2	Team1	Team2
SA1	NZ1	SA2	NZ1
AU1	SA2	AU1	SA1
SA1	SA2	SA2	SA1

2.6.2 Voltooi die kode vir die *SwopTeams*-prosedure wat in die *btnSwopTeams*- gebeurtenishanteerder ('event handler') in die *TestGames_U*-eenheid geroep word. Hierdie prosedure moet die twee spanne wat omgeruil moet word van die gebruiker kry en dan die metode wat jy in VRAAG 2.6.1 geskep het, gebruik om al die gevalle waar die spanne in die skikking *arrGames* voorkom, om te ruil.

Let Wel: Jy kan die prosedure *SetupDummyData* gebruik as jy nie die *ReadTextFile*- prosedure in VRAAG 2.2.2 suksesvol kon skryf nie. Hierdie kode sal vyf uitslae in die skikking *arrGames* laai waarmee jy kan werk.

(5)

2.7 Voltooi die kode vir die *FirstDraw*-prosedure wat in die *btnFirstDraw*-gebeurtenishanteerder ('event handler') in die *TestGames_U*-eenheid geroep word. Hierdie prosedure moet op 'n effektiewe manier die eerste voorkoms van 'n gelykopresultaat tussen spanne van dieselfde land vind. Die soektog moet gestaak word sodra die eerste voorkoms van 'n gelykopresultaat tussen spanne van dieselfde land gevind word.

Vertoon die besonderhede van die wedstryd deur van die *toString*-metode wat jy in VRAAG 2.3.1 geskep het, gebruik te maak. 'n Toepaslike boodskap moet vertoon word as geen gelykopresultaat tussen spanne van dieselfde land in die skikking gevind word nie.

Voorbeeld van afvoer: 1998: NZ2 vs NZ3 25 – 25

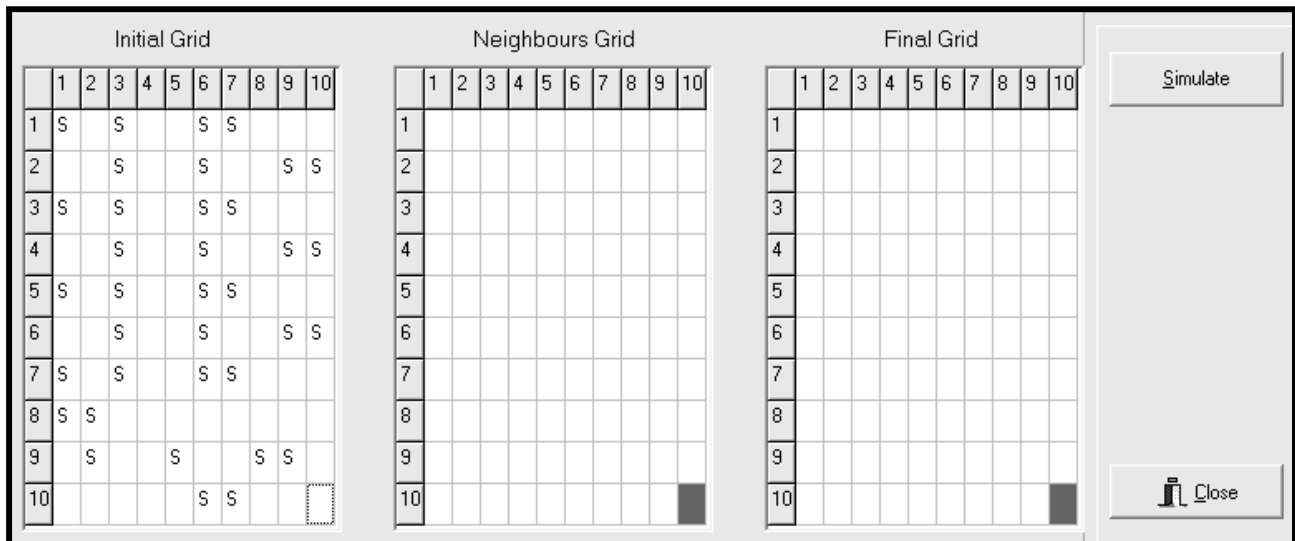
Let Wel: Jy kan die prosedure *SetupDummyData* gebruik as jy nie die *ReadTextFile*- prosedure in VRAAG 2.2.2 suksesvol kon skryf nie. Hierdie kode sal vyf uitslae in die skikking *arrGames* laai waarmee jy kan werk.

(16)

(120 ÷ 2) [60]

VRAAG 3: DELPHI-PROGRAMMERING

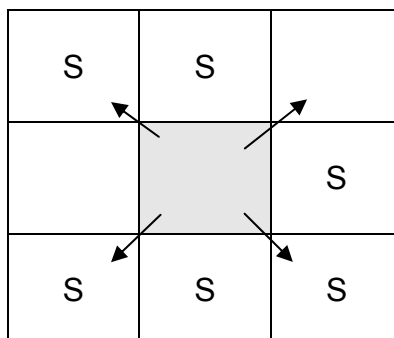
Jy is gevra om 'n program te skryf wat sal demonstree hoe dit moontlik kan wees om die besetting en ontruiming van sitplekke deur rugbytoeskouers in 'n gedeelte van 'n pawiljoen te simuleer. 'n Tweedimensionele skikking met 10 rye en 10 kolomme genaamd *arrSpectators* word gebruik om die sitplekke voor te stel. Die 'S'-karakter stel 'n sitplek voor wat in gebruik is en oop sitplekke word deur die blanko karakter (die ' ' karakter) voorgestel. Jy is voorsien van 'n onvolledige program in die lêergids met die naam Q3. Die program het drie stringroosters ('string grids') en twee knoppies ('buttons') soos hieronder aangetoon.



- 3.1 Die aanvanklike rooster ('grid') op die linkerkant (*sgdInitial*) en die skikking *arrSpectators* word outomaties opgestel as die program begin deur data te lees uit die lêer *Spectators.txt* in die Q3-lêergids. Die onvoltooide funksie *CountNeighbours* met die volgende opskrif word gegee:

Function CountNeighbours (iRow , iCol : Integer ; arrSpectators : t2DArray): Integer;

Hierdie funksie moet tel hoeveel toeskouers diagonaal (skuins) langs aan voor of agter die sitplek sit in die posisie in die skikking *arrSpectators* soos aangedui deur die *iRow* (ry nommer) en *iCol* (kolom nommer) parameters. Dit sluit die vier sitplekke wat reghoekig teenoor die sitplek lê, uit. Die voorbeeld onder vertoon 'n sitplek wat omring word deur drie toeskouers wat *diagonaal* teenoor die sitplek sit:



Onderstaande twee roosters ('grids') dien as verdere voorbeelde. Die rooster op die regterkant vertoon die ooreenstemmende aantal 'aangrensende' toeskouers wat diagonaal teenoor die sitplek sit vir elkeen van die sitplekke in die grid op die linkerkant. Die getalle in die boonste rye en in die linkerkantste kolomme, is bloot vir verwysingsdoeleindes.

	1	2	3	4	5	6	7	8	9	10
1	S		S			S	S			
2			S			S			S	S
3	S		S			S	S			
4			S			S			S	S
5	S		S			S	S			
6			S			S			S	S
7	S		S			S	S			
8	S	S								
9		S			S				S	S
10			S	S		S	S			

	1	2	3	4	5	6	7	8	9	10
1	0	1	0	1	1	0	1	1	1	1
2	0	4	0	2	2	2	2	2	0	0
3	0	2	0	2	2	0	2	2	2	2
4	0	4	0	2	2	2	2	2	0	0
5	0	2	0	2	2	0	2	2	2	2
6	0	4	0	2	2	2	2	2	0	0
7	1	2	1	1	1	0	1	1	1	1
8	1	2	1	2	1	2	2	2	1	1
9	1	2	2	1	2	1	1	1	0	0
10	1	0	1	1	0	1	1	1	1	1

- Voltooi die kode vir die *CountNeighbours*-funksie wat die aantal aangrensende toeskouers bereken wat diagonaal teenoor die sitplek sit, soos deur die ry- en kolomparameters aangedui word.
- Toepaslike foutkontrolering moet plaasvind om te sorg dat daar na geen element buite die perke van die skikking verwys word nie.

(23)

3.2 Voeg kode by die *Simulate*-prosedure wat in die gebeurtenishanteerder ('event handler') van die *btnSimulate*-knoppie geroep word met die volgende opskrif:

Procedure Simulate (arrSpectators: t2DArray);

Die kode moet die aantal aangrensende toeskouers wat diagonaal teenoor elke sitplek in die *arrSpectators*-skikking sit, bereken en in die middel rooster('grid') op die skerm (*sgdNeighbours*) vertoon, deur gebruik te maak van die *CountNeighbours*-funksie. Die diagram in VRAAG 3.1 wys hoe die afvoer moet lyk.

(5)

3.3 Die simulase van die beweging van toeskouers word gebaseer op die aantal toeskouers wat diagonaal teenoor elke sitplek sit. Hierdie 'reëls' kan in twee gevalle ingedeel word – naamlik reëls wat van toepassing is op 'n sitplek wat leeg is en reëls wat van toepassing is op sitplekke wat nie leeg is nie (n toeskouer sit reeds daar).

Vir sitplekke wat oop is:

As die oop sitplek 2 of 3 aangrensende toeskouers het wat diagonaal teenoor die sitplek sit, sal 'n toeskouer daar geplaas word, anders bly die sitplek oop.

Vir sitplekke met toeskouers:

Die toeskouer sal daar bly sit as daar 2 of 3 aangrensende toeskouers is wat diagonaal teenoor die sitplek sit, anders sal die toeskouer die sitplek verlaat en die sitplek sal oopgaan.

Voeg verdere kode by die *Simulate*-prosedure wat deur die gebeurtenishanteerder ('event handler') van die *btnSimulate*-knoppie geroep word. Die kode moet die 'reëls' wat hierbo geskets is en die 'middelste' rooster (*sgdNeighbours*) gebruik om die rooster aan die regterkant (*sgdFinal*) op te stel. 'n Voorbeeld van die verwagte afvoer word hieronder vertoon:

Initial Grid											Neighbours Grid											Final Grid										
	1	2	3	4	5	6	7	8	9	10		1	2	3	4	5	6	7	8	9	10		1	2	3	4	5	6	7	8	9	10
1	S		S			S	S				1	0	1	0	1	1	0	1	1	1	1	1										
2			S			S			S	S	2	0	4	0	2	2	2	2	2	0	0	2			S	S	S	S	S			
3	S		S			S	S				3	0	2	0	2	2	0	2	2	2	2	3		S		S	S		S	S	S	S
4			S			S			S	S	4	0	4	0	2	2	2	2	2	0	0	4			S	S	S	S	S			
5	S		S			S	S				5	0	2	0	2	2	0	2	2	2	2	5		S		S	S		S	S	S	S
6			S			S			S	S	6	0	4	0	2	2	2	2	2	0	0	6			S	S	S	S	S			
7	S		S			S	S				7	1	2	1	1	1	0	1	1	1	1	7		S								
8	S	S									8	1	2	1	2	1	2	2	2	1	1	8		S		S		S	S	S		
9		S			S				S	S	9	1	2	2	1	2	1	1	1	0	0	9		S	S		S					
10			S	S		S	S				10	1	0	1	1	0	1	1	1	1	1	10										

(8)

3.4 Om die simulasieproces te kan herhaal, moet die waardes wat huidiglik in die stringrooster *sgdFinal* gestoor is, aan die *arrSpectators*-skikking toegeken word. Voltooi die kode van die *ResetInitialArray*-prosedure wat deur die gebeurtenishanteerder ('event handler') van die *btnSimulate*-knoppie geroep word. Die *ResetInitialArray*-prosedure het die volgende opskrif:

Procedure ResetInitialArray (Var arrSpectators: t2DArray);

Hierdie prosedure moet die waardes wat in die *sgdFinal*-stringrooster gestoor word lees en die huidige waardes in die *arrSpectators*-skikking vervang.

(8)

(40 ÷ 2) [20]

TOTAAL: 120