NASIONALE SENIOR SERTIFIKAAT-EKSAMEN
NOVEMBER 2019

INLIGTINGSTEGNOLOGIE: VRAESTEL II

Tyd: 3 uur — 120 punte

LEES ASSEBLIEF DIE VOLGENDE INSTRUKSIES NOUKEURIG DEUR

1. Hierdie vraestel bestaan uit 16 bladsye. Maak asseblief seker dat jou vraestel volledig is.

2. Hierdie vraestel moet beantwoord word deur objekgeoriënteerde programmerings-beginsels te gebruik. Jou program moet sinvol gebruik maak van metodes en parameters.

3. Hierdie vraestel is in twee afdelings verdeel. Alle kandidate moet albei afdelings beantwoord.

4. Hierdie vraestel is opgestel in programmeerterme wat nie eie is aan enige spesifieke programmeertaal (Java/Delphi) of databasis (Access/MySQL/JavaDB) nie.

5. Maak seker dat jy die vrae beantwoord op die wyse wat beskryf word, aangesien punte vir jou oplossing toegeken sal word ooreenkomstig die spesifikasies wat in die vraag gegee word.

6. Beantwoord slegs wat in elke vraag gevra word. As die vraag byvoorbeeld nie vra vir datageldigheidstoetsing nie, word geen punte daarvoor toegeken nie en hoef geen kode dus vir datageldigheidstoetsing geskryf te word nie.

7. As jy 'n gedeelte van die kode nie kan laat werk nie, moet jy dit as kommentaar merk sodat dit nie uitgevoer word nie en sodat jy met die eksamen kan voortgaan. Probeer indien moontlik om die fout te verduidelik om die nasiener te help.

8. Wanneer jy in jou kode toegang tot lêers verkry, MOENIE die name van die volledige pad van die lêers gebruik nie, aangesien dit probleme sal skep wanneer die program nagesien word op 'n ander rekenaar as die een waarop jy werk. Verwys bloot na die lêers deur die name en uitbreidings daarvan te gebruik waar nodig.

9. Jou programme moet op so 'n manier gekodeer word dat hulle met enige data sal werk en nie slegs die voorbeelddata wat voorsien is of enige data-uittreksels wat in die vraestel voorkom nie. Jy word aangeraai om die datalêers wat voorsien word, noukeurig te bestudeer.

IEB Copyright © 2019

10. Maak seker dat roetines soos soektogte, sorterings en selekterings vir skikkings uit eerste beginsels ontwikkel word en dat jy nie die ingeboude eienskappe van 'n programmeertaal vir enigeen van hierdie roetines gebruik nie.

11. Alle datastrukture moet deur jou, die programmeerder, gedefinieer en verklaar word. Jy mag nie komponente wat binne die koppelvlak voorsien word, gebruik om data te stoor en later te herwin nie.

12. Lees die hele vraestel voordat jy 'n datastruktuur kies. Jy kan vind dat daar 'n alternatiewe metode is om die data voor te stel wat dalk meer doeltreffend sal wees met inagneming van die vrae wat in die vraestel gevra word.

13. Jy moet al jou werk gereeld stoor op die skyf wat jy ontvang het of die skyfspasie wat vir hierdie eksamen aan jou toegeken is. Jy moet voor jy begin ook 'n rugsteunkopie (*backup*) van die oorspronklike lêers skep ingeval die oorspronklike weergawe per ongeluk deur jou oplossing gewysig word.

14. Indien jou eksamen deur 'n tegniese probleem soos 'n kragonderbreking onderbreek word, sal jy wanneer jy weer begin skryf, slegs die tyd gegee word wat oor was toe die onderbreking begin het, om jou eksamen te voltooi. Geen ekstra tyd sal gegee word om werk wat nie gestoor is nie weer oor te doen nie.

15. Maak seker jou eksamennommer verskyn as kommentaar in elke program wat jy kodeer, sowel as op elke bladsy sigkopie (*hard copy*) wat jy inlewer.

16. Druk 'n kodelys van al die programme/klasse wat jy kodeer. Drukwerk moet ná die eksamen gedoen word. Jy sal na afloop van die eksamen 'n halfuur kry om te druk. Jou onderwyser sal jou inlig oor die reëlings vir die druk van jou werk.

17. Jy moet van die volgende twee gidse (*folders*) (in vetdruk) en lêers (*files*) voorsien word. Hierdie lêers moet as data vir hierdie eksamen gebruik word. Let daarop dat die databasislêers in MS Access-, JavaDB- en MySQL-formaat voorsien word. Maak seker dat jy die lêers kan oopmaak met die pakkette wat jy gaan gebruik om jou oplossings vir hierdie eksamen te kodeer.

    Afdeling A:
    SosialeUitreik_Access.mdb
    SosialeUitreik_JavaDB.sql
    SosialeUitreik_MySQL.sql
    SQLAntwoordblad.rtf
    SQLBlaaier.exe

    Afdeling B:
    Studente.txt

IEB Copyright © 2019

SCENARIO

Jy is betrokke by die skep van 'n sagteware-oplossing vir 'n plaaslike skool se Sosiale Uitreik-program. Hierdie program word deur studente tot voordeel van gemeenskappe rondom die skool gelei en bedryf.

AFDELING A SQL

VRAAG 1

Die Sosiale Uitreik-program bestaan uit veelvuldige projekte. Elke projek word deur een van die studente gelei en het 'n weeklikse uitreik-aktiwiteit op 'n bepaalde dag. Studente wat aan hierdie program deelneem, kan by een of meer projekte betrokke raak en die ure wat hulle aan elke projek bestee, word opgeteken.

STUDENT-tabel bevat die besonderhede van al die studente wat aan die skool se Sosiale Uitreik-program deelneem.

VELD	DATATIPE	BESKRYWING
StudentID	HEELGETAL	'n Unieke identifikasienommer vir elke student.
StudentNaam	TEKS	Die naam van die student.
IsSenior	BOOLE	Of die geregistreerde student 'n senior student by die skool is of nie.

Student-tabel – voorbeelddata (eerste 5 rekords)
Data kan na gelang van die instellings op jou rekenaar anders geformateer wees.

StudentID	StudentNaam	IsSenior
1	Jacob Ncube	TRUE
2	Samantha Marais	FALSE
3	Karabo Mlangeni	TRUE
4	Thapelo Mabena	FALSE
5	Nic de Klerk	FALSE

PROJEK-tabel bevat die besonderhede van die projekte wat deel vorm van die Sosiale Uitreik-program. Elke projek het 'n weeklikse uitreik-aktiwiteit op 'n bepaalde dag. Hierdie inligting word in die **ProjekNaam**-veld ingesluit.

VELD	DATATIPE	BESKRYWING
ProjekID	HEELGETAL	Die unieke identifikasienommer vir elke span.
ProjekNaam	TEKS	Die naam van die projek. Die eerste drie karakters van die naam dui die dag aan waarop die weeklikse aktiwiteit vir hierdie projek plaasvind.
StudentLeierID	HEELGETAL	Die StudentID van die leier van hierdie projek. Dit is 'n vreemde sleutel vir die Student-tabel.
DatumBegin	DATUM	Die datum waarop hierdie projek begin is.

Projek-tabel – voorbeelddata (eerste 5 rekords)
Data kan na gelang van die instellings op jou rekenaar anders geformateer wees.

ProjekID	ProjekNaam	StudentLeierID	DatumBegin
1	THU Sandwich feeding scheme	1	2019-01-17
2	SAT English tutoring programme	3	2019-01-19
3	SAT River clean up	16	2019-02-02
4	FRI Mathematics tutorials	12	2019-01-18
5	TUE Knitting for moms	9	2019-03-21

REGISTRASIE-tabel bevat die besonderhede van al die studente wat vir minstens een projek geregistreer het. Die tabel bevat ook die projekidentifikasienommers. Studente mag vir meer as een projek registreer. Die getal kere wat 'n student 'n aktiwiteit vir 'n sekere projek bywoon, word in die **GetalBygewoon**-veld opgeteken. Die getal ure wat 'n student aan elke projek bestee het, word in the **OpgehoopteUre**-veld opgeteken.

VELD	DATATIPE	BESKRYWING
RegistrasieID	HEELGETAL	Die unieke identifikasienommer vir elke registrasie.
StudentID	HEELGETAL	Die StudentID van die student wat vir 'n projek geregistreer het. Dit is 'n vreemde sleutel vir die Student-tabel.
ProjekID	HEELGETAL	Die ProjekID van die projek waarvoor die student geregistreer het. Dit is 'n vreemde sleutel vir die Projek-tabel.
GetalBygewoon	HEELGETAL	Die getal kere wat die student 'n aktiwiteit vir hierdie projek bygewoon het.
OpgehoopteUre	DUBBEL	Die opgehoopte getal ure wat deur hierdie student aan hierdie projek bestee is.

Registrasie-tabel – voorbeelddata – eerste 5 rekords
Data kan na gelang van die instellings op jou rekenaar anders geformateer wees.

RegistrasieID	StudentID	ProjekID	GetalBygewoon	OpgehoopteUre
1	2	1	15	16.25
2	22	1	14	12.5
3	35	1	15	12
4	30	2	14	11
5	1	2	16	13.5

1.1 Skryf 'n navraag (*query*) wat alle inligting oor alle senior studente in die STUDENT-tabel sal vertoon. Orden hierdie lys alfabeties volgens studentnaam. 'n Voorbeeld van die korrekte afvoer word hieronder getoon.

Let daarop dat slegs die eerste vyf rekords getoon word.

StudentID	StudentNaam	**Is Senior** (Data kan anders geformateer wees op jou rekenaar.)
53	Amy Radebe	TRUE
30	Andrea Badenhorst	TRUE
54	Brendan Smit	TRUE
1	Jacob Ncube	TRUE
25	Joshua Jacobs	TRUE
...	...	...

(4)

1.2 Skryf 'n navraag wat die name van alle projekte met "tutor" in die naam sal vertoon. Die korrekte afvoer word hieronder getoon.

ProjekNaam
SAT English tutoring program
FRI Mathematics tutorials

(4)

IEB Copyright © 2019

1.3 Skryf 'n navraag wat die gemiddelde getal ure sal vertoon wat per aktiwiteit bestee word vir elke inskrywing in die REGISTRASIE-tabel. Dit kan bereken word deur die opgehoopte ure te deel deur die getal kere wat 'n aktiwiteit bygewoon is. Vertoon hierdie inligting langs elke **RegistrasieID**. Dit is nie nodig om 'n naam vir die berekende veld te verskaf nie. 'n Voorbeeld van die korrekte afvoer word hieronder getoon.

Let daarop dat slegs die eerste vyf rekords getoon word.

RegistrasieID	**GemUre** Data kan anders geformateer wees op jou rekenaar
1	1.0833333333333333
2	0.8928571428571429
3	0.8
4	0.7857142857142857
5	0.84375
...	...

(3)

1.4 Die eerste drie karakters van 'n projek se naam dui die dag aan waarop die weeklikse aktiwiteit plaasvind. Skryf 'n navraag wat alle dae van die week waarop aktiwiteite plaasvind, sal vertoon. Moenie duplikate vertoon nie. Die korrekte afvoer word hieronder getoon.

Dag
FRI
MON
SAT
THU
TUE
WED

(4)

1.5 Skryf 'n navraag wat die naam van die studenteleier vir elkeen van die projekte sal vertoon. Die korrekte afvoer word hieronder getoon.

ProjekNaam	**StudentNaam**
THU Sandwich feeding scheme	Jacob Ncube
SAT English tutoring programme	Karabo Mlangeni
SAT River clean up	Penny Mbele
FRI Mathematics tutorials	Kenneth Motala
TUE Knitting for moms	Patricia Davids
MON Old Age Home Visit	Steven Govender
WED Recycling Programme	Wiseman Legodi
FRI Textbook Collection	Joshua Jacobs

(4)

1.6 Skryf 'n navraag wat die totale opgehoopte ure vir alle projekte per student sal vertoon **gebaseer op hul StudentID** – hernoem die berekende veld **TotaalUre**. Die korrekte afvoer word hieronder getoon.

StudentID	**TotaalUre**
1	13.5
2	16.25
3	8.5
4	17.5
5	9.25

(5)

IEB Copyright © 2019

1.7 Skryf 'n navraag wat die name sal vertoon van die studente wie se ID nie in die REGISTRASIE-tabel verskyn nie. Die korrekte afvoer word hieronder getoon.

StudentNaam
Khaya Mokoena
Mishka Hassen

(4)

1.8 Skryf 'n navraag wat die name van die onlangsste projek(te), d.w.s. projekte wat die jongste datum het, sal vertoon. Die korrekte afvoer word hieronder getoon.

ProjekNaam
TUE Knitting for moms
FRI Textbook Collection

(4)

1.9 Skryf 'n navraag wat 'n lys van die name van die studente wat vir minstens twee projekte geregistreer het, sal vertoon. Die getal projekte waarvoor hulle registreer het, moet ook vertoon word. Hernoem die berekende veld **GetalProjekte**. 'n Voorbeeld van die korrekte afvoer word hieronder getoon.

Let daarop dat slegs die eerste nege rekords getoon word.

StudentNaam	GetalProjekte
Bhule Mbasa	2
Blessing Mkhize	2
Conrad Snyman	2
Frans Theron	2
Heinriche Pretorius	2
Joshua Jacobs	2
Kobus Venter	2
Laetitia Adams	2
Mahmood Chetty	3
...	...

(8)

40 punte

IEB Copyright © 2019

AFDELING B OBJEKGEORIËNTEERDE PROGRAMMERING

SCENARIO

Die Sosiale Uitreik-span het 'n nuwe projek waaraan studente kan deelneem. Hulle het 'n lys van alle studente in die skool wat bereid is om by enige nuwe projekte betrokke te raak. Hierdie studente word in 'n tekslêer genaamd **Studente.txt** gestoor. Die tekslêer stoor die student se volle naam, graad (8 tot 12), belangstelling (binnenshuis, buitenshuis of albei) en die getal ure wat die student gedurende die termyn in die uitreikprogram opgehoop het.

Die tekslêer bevat elke student se besonderhede, geskei deur kommas, elkeen op 'n nuwe reël. 'n Voorbeeld van die eerste 15 reëls word hieronder getoon:

Studente.txt
volle naam,graad,belangstelling, ure

```
Annie Morris,11,Buitenshuis,2
Lonnie Heriot,11,Binnenshuis,3.5
Morris Scotty,10,Binnenshuis,1
Lucius McNabb,10,Buitenshuis,8.5
Gerhard Leaby,10,Binnenshuis,3
Hayden Haley,10,Buitenshuis,1.5
Carl Rabey,10,Binnenshuis,4
Mary Gibbs,8,Buitenshuis,0
Myron Rhoan,8,Buitenshuis,3.5
Francisco Pettie,8,Binnenshuis,2.5
Bertrand Delaney,9,Binnenshuis,3
Mattie Rules,11,Buitenshuis,10
Seth Maler,8,Albei,0
Justin Doyle,9,Albei,4
Bessie Monahan,12,Binnenshuis,5
```

Die Sosiale Uitreik-span moet 'n program kodeer om die besonderhede van elke student te stoor en 'n projek op grond van hul belangstelling aan hulle toe te ken. 'n Projek het 'n maksimum getal moontlike studente wat toegeken kan word en word as óf "Binnenshuis" óf "Buitenshuis" geklassifiseer. Studente wie se belangstelling as "Albei" gelys word, kan aan óf binnenshuise óf buitenshuise projekte toegeken word.

Indien daar meer studente is as wat die projek nodig het, sal oortollige studente op 'n ewekansige (lukraak) basis van die projek verwyder word.

Laastens het die skoolhoof die totale ure vir elke graad gedurende die jaar as 'n string `"11;90;8;72;12;13;10;34;9;53"` opgeteken. Die string is in die formaat graad gevolg deur totale ure, en elke waarde word van die volgende geskei deur 'n kommapunt (;). Ongelukkig word die grade nie in enige volgorde gelys nie. Hierdie string sal in **Vraag 7** gebruik word. Om studente aan te moedig om aan projekte deel te neem, sal die skoolhoof 10 ekstra ure toeken aan enige graad wat meer as 20 uur vir 'n nuwe projek opgehoop het.

IEB Copyright © 2019

Die program benodig die volgende basisklasse:

Student

'n Klas sal geskep moet word om elke student te verteenwoordig. 'n **Student** het die volgende inligting:

- **volleNaam** – die naam van die student
- **graad** – die graad van die student
- **belangstelling** – die voorkeur van die student vir binnenshuis, buitenshuis of albei
- **ure** – die getal ure wat die student in die termyn opgehoop het deur aan projekte te werk

Graad

'n Klas sal ook geskep moet word om elke graad te verteenwoordig. 'n **Graad** het die volgende inligting:

- **graad** – 'n waarde van 8 tot 12
- **totaal** – die som van die uitreikure van al die studente in daardie graad vir die jaar
- **PERK = 20** – die waarde van 20 uur waarbo 'n BONUS bygevoeg sal word
- **BONUS = 10** – die waarde van 10 uur wat by die totaal bygevoeg sal word indien dit bo 20 is

VRAAG 2

Gebruik die klasdiagram hieronder om 'n nuwe klas genaamd **Student** te skep. Hierdie klas sal gebruik word om die besonderhede van 'n student te stoor. Die diagram hieronder dui die eienskappe en metodes van die klas aan. Nie al die metodes sal in latere klasse benodig word nie.

Let op dat die tipe *double* reële getalle in alle klasdiagramme aandui.

Student
Eienskappe: - string volleNaam - integer graad - string belangstelling - double ure
Metodes: + Constructor(string inFn, integer inG, string inI, double inH) + kryGraad() : integer + kryBelangstelling() : string + kryUre() : double + toString() : string - veranderNaam() : string

IEB Copyright © 2019

2.1 Skep 'n nuwe klas genaamd **Student** met die eienskappe ***volleNaam***, ***graad***, ***belangstelling*** en ***ure*** soos hierbo aangedui. (4)

2.2 Skep 'n konstruktormetode wat die volgende as parameters aanvaar: 'n string ***inFn*** om die **volleNaam**-eienskap te verteenwoordig, 'n heelgetal ***inG*** om die **graad**-eienskap te verteenwoordig, 'n string ***inI*** om die **belangstelling**-eienskap te verteenwoordig en 'n dubbel ***inH*** om die **ure**-eienskap te verteenwoordig. Gebruik hierdie parameters om waardes aan die eienskappe toe te ken. (3)

2.3 Skep toeganger-/krymetodes vir die **graad**-, **belangstelling**- en **ure**-eienskappe. (2)

2.4 Kodeer 'n metode genaamd **veranderNaam** om die formaat van die student se naam te verander na: van en voorletter:

```
van,<spasie>voorletter
```

`Annie Morris` sal byvoorbeeld wees:

```
Morris, A
```

(4)

2.5 Kodeer 'n ***toString()***-metode om al die eienskappe, gekombineer as 'n string en geskei deur tabelkarakters, terug te stuur.

```
van,<spasie>voorletter<tab>graad<tab>belangstelling<tab>
ure
```

Byvoorbeeld: `Heriot, L    11    Binnenshuis    3.5` (3)

[16]

VRAAG 3

Gebruik die klasdiagram hieronder om 'n nuwe klas genaamd **Graad** te skep. Hierdie klas sal gebruik word om die besonderhede van 'n graad te stoor.

Die diagram hieronder dui die eienskappe en metodes van die klas aan. Nie al die metodes sal in latere klasse benodig word nie.

Hierdie klas sal in **Vraag 7** gebruik word.

Graad
Eienskappe: - integer graad - double totaal - (*final/constant*) integer PERK = 20; - (*final/constant*) integer BONUS = 10;
Metodes: + Constructor(integer inG, double inT) + kryTotaal() : integer + stelTotaal(double inT) + toString() : string

3.1 Skep 'n nuwe klas genaamd **Graad** met die eienskappe ***graad*** en ***totaal*** soos hierbo aangedui. (2)

3.2 Skep die twee finale/konstantes met die name en waardes soos in die klasdiagram getoon. (2)

3.3 Skep 'n konstruktormetode wat die volgende as parameters aanvaar: 'n heelgetal ***inG*** om die **graad**-eienskap te verteenwoordig en 'n dubbel ***inT*** om die **totaal** (totale ure)-eienskap vir 'n graad te verteenwoordig. Gebruik hierdie parameters om waardes aan die eienskappe toe te ken. Indien die totale ure (verteenwoordig deur die parameter ***inT***) groter as die **PERK** is, moet 'n **BONUS** by die totaal bygevoeg word. Gebruik die konstantes **PERK** en **BONUS** om die totaal te vermeerder indien dit nodig is. (4)

3.4 Skep 'n toeganger-/kry- en mutator-/stelmetode vir die **totaal**-eienskap. (2)

3.5 Kodeer 'n ***toString()***-metode om die **graad**- en **totaal**-eienskap as 'n string in die volgende formaat terug te stuur:

`"Graad:"graad<4 spasies>"totale ure"<spasie>totaal`

Byvoorbeeld: `Graad:8    totale ure 82.0`

(2)

[12]

IEB Copyright © 2019

VRAAG 4

Gebruik die klasdiagram hieronder om 'n nuwe klas genaamd **Projek** te skep. Die **Projek**-klas sal die besonderhede van 'n enkele projek stoor, met inbegrip van die naam van die projek, die maksimum getal studente wat aan die projek mag deelneem en 'n skikking van **Student**-objekte wat aan die projek toegeken is.

Die besonderhede van die projek sal as parameters na die konstruktormetode van die **Projek** gestuur word. Die studente wat gekies is om aan die projek deel te neem, sal uit die tekslêer **Students.txt** onttrek word.

Hierbenewens sal die klas 'n skikking van **Graad**-objekte stoor. Hierdie skikking sal in **Vraag 7** gebruik word om die totale ure wat elke graad bygedra het, te stoor deur 'n stringparameter te gebruik.

Die **korrekteGetalle**- en **skepGraadSkikking-**metode sal onderskeidelik in **Vraag 6** en **Vraag 7** geskep word.

```
Projek
--------------------------------------------------------
Eienskappe:
- string naam
- integer maks
- Student [] sSkik
- integer sTelling
- Graad [] gSkik
--------------------------------------------------------
Metodes:
+ Constructor(string inN, string inC , integer inM)
+ toString() : string
+ sort()

+ korrekteGetalle() : string
+ skepGraadSkikking (string inH) : string
```

4.1 Skep 'n nuwe klas genaamd **Projek** met die eienskappe **naam** en **maks** om die naam van die projek en die maksimum getal studente wat aan hierdie projek toegeken kan word, te stoor. (2)

4.2 Verklaar die volgende TWEE skikkings:
- 'n Skikking genaamd **sSkik** om tot 50 **Student**-objekte te stoor.
- 'n Tellerveranderlike genaamd **sTelling** om op hoogte te bly van die getal **Student**-objekte wat gestoor is.
- 'n Skikking genaamd **gSkik** om vyf **Graad**-objekte (vir Graad 8, 9, 10, 11 en 12) te stoor. (4)

4.3 Skryf kode vir die konstruktormetode wat die volgende parameters aanvaar: 'n string ***inN*** wat die **naam**-eienskap verteenwoordig*,* 'n string ***inC*** wat die klassifikasie "Binnenshuis", "Buitenshuis" of "Albei" verteenwoordig en heelgetal ***inM*** wat die **maks**-eienskap verteenwoordig. Die konstruktor sal hierdie waardes ontvang om die tipe projek waaraan studente toegeken is, te definieer. Hierdie waardes sal verskaf word wanneer die **Projek** in **Vraag 5.2** geïnstansieer word.

Die konstruktormetode sal die inhoud van 'n tekslêer genaamd **Studente.txt** lees om studente wat vir die projek geskik is, te vind. Die lêernaam moet vasgekodeer (*hardcoded*) word in die konstruktormetode. Elke reël van die lêer stoor die besonderhede van een student. Doen die volgende:

- Maak die tekslêer, **Students.txt**, oop om te lees.
- Voer die lus deur die lêer uit totdat daar geen reëls oor is nie. In elke herhaling van die lus:
  - Lees elke reël in en verdeel die student se data in afsonderlike items.
  - Bepaal of die student se belangstelling ooreenstem met die klassifikasie van die projek. 'n Student kan belangstelling as binnenshuis, buitenshuis of albei aandui. Indien die projek as binnenshuis geklassifiseer word, kan studente wat hul belangstelling as binnenshuis of albei aangedui het, by die skikking gevoeg word. Indien 'n projek as buitenshuis geklassifiseer word, kan studente wat hul belangstelling as buitenshuis of albei aangedui het, by die skikking gevoeg word.
  - Indien die student se belangstelling met die klassifikasie ooreenstem, skep en voeg 'n **Student**-objek by die skikking **sSkik** en inkrementeer die teller **sTelling**.
  - LET WEL – nie al die studente in die tekslêer sal by die skikking gevoeg word nie. (8)

4.4 Kodeer die **toString**-metode om 'n string wat die inligting van al die **Student**-objekte in die skikking genaamd **sSkik** bevat, terug te stuur. Gebruik die **toString**-metode van die **Student**-klas wat jy vroeër in **Vraag 2.5** geskep het. Elke student se besonderhede moet op 'n nuwe reël wees. Sluit 'n opskrif in om die naam en die maksimum getal van die projek voor die studentbesonderhede te vertoon. Die opskrif moet in die volgende formaat wees:

```
"Naam:"<tab>naam
"Maksimum:"<tab>maks
```

(4)

4.5 Kodeer 'n metode genaamd **sorteer** om die studente volgens graad te rangskik. Die Graad 8's moet eerste en die Graad 12's laaste vertoon word. (6)

[24]

IEB Copyright © 2019

VRAAG 5

5.1 Skryf kode om 'n teksgebaseerde gebruikerskoppelvlak genaamd **ProjekGK** te skep wat eenvoudige afvoer sal toelaat. (1)

5.2 Die Sosiale Uitreik-span het 'n nuwe projek om ou klere vir die inwoners van 'n ouetehuis in te samel. Die projek kan hoogstens 12 lede hê en word as binnenshuis geklassifiseer. Verklaar en instansieer 'n **Projek**-objek deur hierdie inligting as parameters op die toepaslike plek in die kode te gebruik. (1)

5.3 Voeg kode by wat al die **Student**-objekte sal sorteer en dan sal vertoon deur die toepaslike metodes in die **Projek**-klas te roep. Jou afvoer moet vertoon word as:

```
Naam:     Samel ou klere in
Maksimum: 12
Pettie, F      8     Binnenshuis    2.5
Maler, S       8     Albei          0.0
Honiford, E    8     Binnenshuis    5.5
Doyle, J       9     Albei          4.0
Boyder, L      9     Albei          9.5
Delaney, B     9     Binnenshuis    3.0
Leaby, T       9     Albei          7.0
Shorts, H      9     Albei          12.0
Rabey, C       10    Binnenshuis    4.0
Janson, C      10    Albei          7.0
Scotty, M      10    Binnenshuis    1.0
Leaby, G       10    Binnenshuis    3.0
Morvel, M      11    Albei          1.0
Heriot, L      11    Binnenshuis    3.5
McCalum, A     11    Albei          5.0
Monahan, B     12    Binnenshuis    5.0
```

(2)

[4]

VRAAG 6

Keer terug na die **Projek**-klas en voeg kode by om enige ekstra studente ewekansig (lukraak) van die projek te verwyder op grond van die maksimum getal studente wat in 'n projek toegelaat word soos gestoor in die eienskap **maks**.

Kodeer 'n getipeerde metode genaamd **korrekteGetalle** wat enige ekstra **Student**-objekte uit die skikking verwyder. Die studente wat verwyder moet word, moet ewekansig geselekteer word. Die maksimum getal studente wat aan hierdie projek toegeken kan word, word in die eienskap genaamd **maks** gestoor. Die metode moet 'n string terugstuur wat die studente wat geskrap is, gevolg deur die oorblywende studente, lys.

Jy kan enige korrekte metode gebruik om die ewekansig geselekteerde studente uit die skikking te verwyder.

Jy kan aanneem dat daar meer studente is as wat die projek nodig het.

Jy kan bykomende helpermetodes in die **Projek**-klas kodeer indien nodig.

Die formaat van die string wat teruggestuur word, word in **Vraag 8.1** getoon.

[13]

IEB Copyright © 2019

VRAAG 7

Keer terug na die **Projek**-klas om die skikking van grade genaamd **sSkik** te skep wat gebaseer is op waardes wat as 'n stringparameter gestuur is.

Kodeer 'n getipeerde metode genaamd **skepGraadSkikking** om die getal ure vir elke graad te stoor, gestuur as 'n stringparameter ***inH***. Die volgende waarde sal aan die stringparameter gestuur word:

```
"11;90;8;72;12;13;10;34;9;53"
```

Die data sal in vyf pare wees, geskei deur kommapunte (;). Die eerste waarde verteenwoordig die graad en die tweede waarde die totale ure. Die string hierbo verteenwoordig Graad 11 met 'n totaal van 90, Graad 8 met 'n totaal van 72, Graad 12 met 'n totaal van 13, Graad 10 met 'n totaal van 34 en Graad 9 met 'n totaal van 53.

Die grade word nie in enige volgorde gelys nie, maar daar sal altyd die vyf pare waardes wees. Jou kode moet die grade in enige volgorde kan hanteer. Die data hierbo, wat ekwivalent is aan die vorige string, kan byvoorbeeld in die volgende volgorde wees:

```
"9;53;12;13;8;72;11;90;10;34"
```

Jou oplossing moet die volgende doen:
- Onttrek die graadnommer en ure vir elke graad uit die parameter ***inH***.
- Skep 'n **Graad**-skikkingsobjek vir elke graad deur die graadnommer en ure wat onttrek is, te gebruik.
- Skep en stuur 'n string terug om elke graad saam met sy totaal te lys. Die lys moet voorafgegaan word deur die opskrif `"Graadtotale:"`.

Jy kan bykomende helpermetodes in die **Projek**-klas kodeer indien nodig.

Die formaat van die string wat teruggestuur word, word in **Vraag 8.2** getoon.

[9]

VRAAG 8

Voeg kode in die **ProjekGK**-klas by om die metodes wat jy in **Vraag 6** en **7** gekodeer het, te roep.

8.1 Voeg kode by die **ProjekGK**-klas om die **korrekteGetalle**-metode te roep. Jou afvoer moet soortgelyk aan die volgende vertoon, na gelang van die studente wat geskrap is:

```
Studente verwyder:
Maler, S  8          Albei          0.0
Doyle, J  9          Albei          4.0
Leaby, T  9          Albei          7.0
Pettie, F 8          Binnenshuis    2.5

Oorblywende studente:
Naam:         Samel ou klere in
Maksimum:     12
Honiford, E    8      Binnenshuis    5.5
Boyder, L      9      Albei          9.5
Delaney, B     9      Binnenshuis    3.0
Shorts, H      9      Albei          12.0
Rabey, C       10     Binnenshuis    4.0
Janson, C      10     Albei          7.0
Scotty, M      10     Binnenshuis    1.0
Leaby, G       10     Binnenshuis    3.0
Morvel, M      11     Albei          1.0
Heriot, L      11     Binnenshuis    3.5
McCalum, A     11     Albei          5.0
Monahan, B     12     Binnenshuis    5.0
```

(1)

8.2 Voeg kode by **ProjekGK** by om die **skepGraadSkikking**-metode te roep. Jou afvoer moet soos volg lyk:

```
Graadtotale:
Graad:8    totale ure 82.0
Graad:9    totale ure 63.0
Graad:10    totale ure 44.0
Graad:11    totale ure 100.0
Graad:12    totale ure 13.0
```

Let daarop dat daar 10 ure by die totaal van die grade wat 20 uur oorskry, gevoeg sal word. (1)

[2]

80 punte

Totaal: 120 punte