



NASIONALE SENIOR CERTIFIKAAT-EKSAMEN  
NOVEMBER 2020

## INLIGTINGSTEGNOLOGIE: VRAESTEL II

### NASIENRIGLYNE

Tyd: 3 uur

120 punte

---

Hierdie nasienriglyne is opgestel vir gebruik deur eksaminators en hulpeksaminators van wie verwag word om almal 'n standaardiseringsvergadering by te woon om te verseker dat die riglyne konsekwent vertolk en toegepas word by die nasien van kandidate se skrifte.

Die IEB sal geen bespreking of korrespondensie oor enige nasienriglyne voer nie. Ons erken dat daar verskillende standpunte oor sommige aangeleenthede van beklemtoning of detail in die riglyne kan wees. Ons erken ook dat daar sonder die voordeel van die bywoning van 'n standaardiseringsvergadering verskillende vertolkings van die toepassing van die nasienriglyne kan wees.

---

**AFDELING A SQL****VRAAG 1.1 [3]**

```
SELECT * FROM Klient -- 1
WHERE Ouderdom between 18 and 30 -- 1
ORDER BY Ouderdom -- 1
```

**ALTERNATIEF:**

```
SELECT * FROM Klient -- 1
WHERE Ouderdom >= 18 AND Ouderdom <=30 -- 1
ORDER BY Ouderdom -- 1
```

**VRAAG 1.2 [3]**

```
SELECT * FROM Klient -- 1 vir SELECT en FROM
WHERE Poskode LIKE -- 1
'012%' -- 1 (ACCESS: '012*')
```

**ALTERNATIEF:****ACCESS en MYSQL**

```
SELECT * FROM Klient -- 1 vir SELECT en FROM
WHERE LEFT(Poskode, 3) -- 1
= '012' -- 1
```

**JAVADB**

```
SELECT * FROM Klient -- 1 vir SELECT en FROM
WHERE SUBSTR(Poskode,1, 3) -- 1
= '012' -- 1
```

**VRAAG 1.3 [5]****MYSQL:**

```
SELECT KlientNaam, CONCAT( -- 1 Saamvoeging
    RIGHT(KlientNaam, 2) -- 1 RIGHT korrek
    , FLOOR(11 + -- 1 Begin by 11 en FLOOR
        RAND()* -- 1 Ewekansig
        (17-11+1)) -- 1 Korrekte reeks
    ) FROM Klient
```

**ACCESS:**

```
SELECT KlientNaam,
    RIGHT(KlientNaam, 2) -- 1 RIGHT korrek
    & -- 1 Saamvoeging
    Int(11+ -- 1 Begin by 11 en INT
        (17-11+1)* -- 1 Korrekte reeks
        Rnd(KlientID)) -- 1 Ewekansig met saad wat verskil
FROM Klient
```

**JAVADB:**

```

SELECT KlientNaam,
      SUBSTR(KlientNaam, LENGTH(KlientNaam)-1, 2) -- 1 regs korrek
      ||                                           -- 1 Saamvoeging
      CHAR(INT(11 +                               -- 1 Begin by 11 en INT
            (RANDOM()                             -- 1 Ewekansig
            *(17-11+1))))                       -- 1 Korrekte reeks
FROM Klient

```

**VRAAG 1.4 [3]**

```

SELECT KlientID, Plek FROM Afspraak           -- 1
WHERE Plek NOT IN                             -- 1
('Bergsig', 'Panorama', 'Highlands')         -- 1 Formaat

```

**ALTERNATIEF:**

```

SELECT KlientID, Plek FROM Afspraak           -- 1
WHERE Plek <> 'Bergsig'                       -- 1 <> korrek gebruik (aanvaar != JavaDB)
      AND Plek <> 'Panorama'                 -- 1 Al 3 korrek gekontroleer
      AND Plek <> 'Highland'

```

**VRAAG 1.5 [5]**

```

SELECT beraderNaam,
Tarief
FROM Berader                                -- 1 vir korrekte SELECT en FROM
WHERE Tarief =                              -- 1
( SELECT MIN(Tarief)                         -- 1
  FROM Berader                               -- 1
  WHERE Tarief <> 0                           -- 1 (aanvaar != JavaDB)
)

```

**VRAAG 1.6 [6]**

```

SELECT Plek, COUNT(*)                       -- 1
FROM Afspraak
WHERE AfspraakDatum > CURRENT_DATE          -- (ACCESS: NOW())
                                           -- 1 huidige datum (aanvaar >=)

GROUP BY Plek                               -- 2
HAVING COUNT(*) > 15                         -- 2

```

**VRAAG 1.7 [5]**

```
SELECT AfspraakID, Afspraak.KlientID, KlientNaam, BeraderNaam,  
       Tarief * 0.75                                -- 1  
FROM Klient, Berader, Afspraak                    -- 1  
WHERE Klient.KlientID = Afspraak.KlientID         -- 1  
      AND Berader.BeraderID = Afspraak.BeraderID  -- 1  
      AND Ouderdom < 15                           -- 1
```

**VRAAG 1.8 [4]**

```
UPDATE Afspraak                                -- 1  
SET Plek = 'Middelburg'                       -- 1  
WHERE Plek = 'Bergsig'                         -- 1  
      AND AfspraakDatum >= '2020-10-24'        -- 1 korrekte datum  
ACCESS: AND AfspraakDatum >= #10/24/2020#
```

**VRAAG 1.9 [6]**

```
INSERT INTO Afspraak                          -- 1  
(KlientID, BeraderID, AfspraakDatum, Plek)  
-- 1 lys sluit alle velde in wat gevra is  
SELECT 27, BeraderID, '2020-10-20', Plek  
-- 1 volgorde stem ooreen  
-- 1 velde beraderID, Plek ingesluit  
-- 1 formaat korrek vir ander  
-- ACCESS #10/20/2020# vir datum  
  
FROM Afspraak  
WHERE AfspraakID = 18                          -- 1
```

**AFDELING B      OBJEKGEORIËNTEERDE PROGRAMMERING****JAVA****VRAAG 2      Klient.java**

```
// Vraag 2.1 - 4
// klas geskep
public class Klient {
    private String klientNaam;           // privaat
    private String voorkeurBerader;      // korrekte tipes
    private int vroegsteUur;             // korrekte name

    // Vraag 2.2 - 4
    // korrekte opskrif en name/tipe/volgorde soos gevra
    public Klient(String inKN, String inVBR, int inVU) {
        klientNaam = inKN;               // stringvelde as parameters gestel
        voorkeurBerader = inVBR;

        if (inVU > 16)                    // vroegsteUur op 16 gestel indien inVU > 16
        {
            vroegsteUur = 16;
        }
        else                             // stel andersins op inVU
        {
            vroegsteUur = inVU;
        }
    }

    // Vraag 2.3 - 2
    // korrekte opskrifte vir al drie metodes
    // korrekte terugsendings vir al drie metodes
    public String kryKlientNaam() {
        return klientNaam;
    }

    public String kryVoorkeurBerader() {
        return voorkeurBerader;
    }

    public int kryVroegsteUur() {
        return vroegsteUur;
    }

    @Override
    // Vraag 2.4 - 4
    // korrekte opskrif
    public String toString() {
        return klientNaam + "\t" + voorkeurBerader + "\tVroegste "
            + vroegsteUur + ":00";
        // korrekte terugsending
        // alle elemente ingesluit
        // formaat korrek met inbegrip van inkeping en :00
    }
}
```

**VRAAG 3****TydGleuf.java**

```
// Vraag 3.1 - 3
// klas korrek geskep
public class TydGleuf {
    private String berader;           // privaat
    private int beginUur;             // tipes/name korrek en soos gegee
    private boolean isBeskikbaar = true;

    // Vraag 3.2 - 4
    // opskrif korrek
    // neem slegs string en heelgetal in - nie Boole nie
    // Boole by verstek op true (in konstruktor of veldverklaring)
    public TydGleuf(String inBR, int inBU) {
        berader = inBR;
        beginUur = inBU;             // stel ander velde
    }

    // Vraag 3.3 - 2
    // stel opskrifte korrek
    // stel toegekende waardes korrek
    public void stelIsBeskikbaar(boolean isBeskikbaar) {
        this.isBeskikbaar = isBeskikbaar;
    }

    // Vraag 3.4 - 1
    // al drie kry-metodes korrek
    public String kryBerader() {
        return berader;
    }

    public int kryBeginUur() {
        return beginUur;
    }

    public boolean kryIsBeskikbaar() {
        return isBeskikbaar;
    }

    // Vraag 3.5 - 2
    // korrekte opskrif met privaat
    private int kryEindUur() {
        // stuur beginUur + 1 terug
        return beginUur + 1;
    }
}
```

```

// Vraag 3.6 - 4
@Override
public String toString() {
    // gebruik kryEindUur om eindtyd te kry
    // sluit ander velde in
    // formaat korrek
    return berader + ": " + beginUur + ":00 - " + kryEindUur()
        + ":00";
}
}

```

## VRAAG 4, 5, 6 EN 7

### GleufBestuurder.java

```

import java.io.*;

// Vraag 4.1 - 4
public class GleufBestuurder {
    private Klient[] kSkik = new Klient[20];
    private TydGleuf[] tSkik = new TydGleuf[40];

    // klas korrek geskep
    // kliëntskikking geskep
    // tydgleufskikking geskep
    // beide privaat met korrekte
    // groottes

    // Vraag 4.2 - 8
    public GleufBestuurder() {
        try {
            // lees lêer korrek
            BufferedReader br = new BufferedReader(new FileReader("kliente.txt"));

            // korrekte lusstruktuur
            // korrekte waarde om korrekte lees van 20 kliënte in 60 reëls moontlik te
            // maak
            for (int i = 0; i < kSkik.length; i++) {
                String klientnaam = br.readLine();
                String voorkeurbr = br.readLine();
                int vroegste = Integer.parseInt(br.readLine());

                // lees drie reëls
                // korrekte omskakeling

                // skep kliënt
                Klient k = new Klient(klientnaam, voorkeurbr, vroegste);
                // voeg nuwe kliënt by die korrekte gleuf by
                kSkik[i] = k;
                // korrekte bywerking (kan deel wees van for-opskrif)
            }
        } catch (FileNotFoundException ex) {
            System.out.println("Lêer nie gevind nie");
        } catch (IOException ex) {
            System.out.println("TA-fout");
        }
    }
}

```

```
// Vraag 4.3 - 4
// korrekte opskrif
public String vertoonAlleKliente() {
    String temp = "";
    // lus om deur korrekte skikking en korrekte lengte te loop
    for (int i = 0; i < kSkik.length; i++) {
        // bou 'n string korrek op
        temp += kSkik[i] + "\n";
    }
    // stuur die opgeboude string terug
    return temp;
}

// Vraag 6.1 - 11
// korrekte opskrif
public void genereerTydGleuwe() {
    try {
        // maak lêer korrek oop om te lees
        BufferedReader br = new BufferedReader(new
            FileReader("beraders.txt"));

        String line = br.readLine();    // lees een reël

        // verdeel beradername korrek
        String[] brSkik = line.split(",");

        int counter = 0;    // skep en inisialiseer teller korrek

        // gebruik lusse om tydgleuwe suksesvol te genereer
        // tydgleuwe gelaai
        for (int i = 0; i < brSkik.length; i++) {
            for (int j = 8; j <= 16; j++) {
                // sluit etensuur uit
                if (j!= 12) {
                    // tydgleuf met korrekte parameters geskep
                    TydGleuf t = new TydGleuf(brSkik[i], j);
                    // by korrekte skikking in korrekte posisie bygevoeg
                    tSkik[counter] = t;
                    counter++;    // inkrementeer teller laaste
                }
            }
        }
    } catch (FileNotFoundException ex)
    {
        System.out.println("Lêer nie gevind nie");
    } catch (IOException ex)
    {
        System.out.println("TA-fout");
    }
}
```



```

// Vraag 6.2 - 4
// korrekte opskrif
public String vertoonAlleBeskikbareTydGleuwe() {
    String temp = "";
    // lus om deur korrekte skikkinglengte te loop
    for (int i = 0; i < tSkik.length; i++) {
        // kontroleer of gleuf beskikbaar is
        if (tSkik[i].kryIsBeskikbaar()) {
            // kombineer elke tydgleuf in korrekte formaat
            temp += tSkik[i] + "\n";
        }
    }

    return temp;
}

// Vraag 7.1 - 13
// opskrif korrek
public String genereerBespreekteGleuwe() {
    // skep opskrif met nuwe reël
    String temp = "Afsprake: \n";

    // lus om deur kliëntskikking te loop
    for (int k = 0; k < kSkik.length; k++) {
        // lus om deur tydgleufskikking te loop
        for (int t = 0; t < tSkik.length; t++) {
            // kontroleer tydgleufberader teen kliënt se voorkeur
            if (tSkik[t].kryBerader().equals(
                kSkik[k].kryVoorkeurBerader())) {
                // kontroleer beginuur teen kliënt
                if (tSkik[t].kryBeginUur() >= kSkik[k].kryVroegsteUur()) {
                    // kontroleer of tydgleuf beskikbaar is
                    // korrekte voorwaardelike skakel (aanvaar ook and)
                    if (tSkik[t].kryIsBeskikbaar()) {
                        // stel ooreenstemmende tydgleuf se beskikbaarheid op false
                        tSkik[t].stelIsBeskikbaar(false);

                        // sluit slegs die inligting in wat gevra is (en niks meer nie)
                        // formatering korrek
                        temp = temp + kSkik[k].kryKlientNaam() + " ("
                            + kSkik[k].kryVroegsteUur() + ") sien "
                            + tSkik[t] + "\n";
                        break; // onderbreking wanneer gevind (of gebruik vlaggie)
                    }
                }
            }
        }
    }

    return temp; // stuur korrekte string terug
}
}

```

**ALTERNATIEF MET *WHILE* EN *VLAGGIE***

```

// Vraag 7.1 - 13
// opskrif korrek
public String genereerBespreekteGleuwe() {
    // skep opskrif met nuwe reël
    String temp = "Afsprake: \n";

    // lus om deur kliëntskikking te loop
    for (int k = 0; k < kSkik.length; k++) {

        int t = 0;
        boolean found = false;
        // lus om deur tydgleufskikking te loop
        while (t < tSkik.length && found == false) {
            // kontroleer tydgleufberader teen kliënt se voorkeur
            if (tSkik[t].kryBerader().equals(
                kSkik[k].kryVoorkeurBerader())) {
                // kontroleer beginuur teen kliënt
                if (tSkik[t].kryBeginUur() >= kSkik[k].kryVroegsteUur()) {
                    // kontroleer of tydgleuf beskikbaar is
                    // korrekte voorwaardelike skakel (aanvaar ook and)
                    if (tSkik[t].kryIsBeskikbaar()) {
                        // stel ooreenstemmende tydgleuf se beskikbaarheid op false
                        tSkik[t].stelIsBeskikbaar(false);

                        // sluit slegs die inligting in wat gevra is (en niks meer nie)
                        // formatering korrek
                        temp = temp + kSkik[k].kryKlientNaam() + " ("
                            + kSkik[k].kryVroegsteUur() + ") sien "
                            + tSkik[t] + "\n";
                        found = true; // gebruik vlaggie
                    }
                }
            }
            t++;
        }
    }

    return temp; // stuur korrekte string terug
}

```

**BeradingGK.java**

```
import java.io.IOException;

public class BeradingGK {
    // Vraag 5.1 - 1
    // teksgebaseerde koppelvlak vir toevoer/afvoer

    public static void main(String[] args) throws IOException {
        // Vraag 5.2 - 1
        // GleufBestuurder geskep
        GleufBestuurder gb = new GleufBestuurder();

        // Vraag 5.3 - 1
        // vertoon kliëntskikking korrek
        System.out.println(gb. vertoonAlleKliente());

        // Vraag 6.3 - 2
        // roep genereerTydGleuwe() korrek
        gb.genereerTydGleuwe();
        // roep vertoonAlleBeskikbareTydGleuwe
        System.out.println(gb. vertoonAlleBeskikbareTydGleuwe());

        // Vraag 7.2 - 1
        // metode geroep en gedrukte string teruggestuur
        System.out.println(gb.genereerBespreekteGleuwe());
    }
}
```

**DELPHI****VRAAG 2****uKlient.pas**

```
unit uKlient;

interface
    uses SysUtils;

    // Vraag 2.1 - 4
    // klas geskep
    type TKlient = class
    private
        klientNaam : string;           // privaat
        voorkeurBerader : string;      // korrekte tipes
        vroegsteUur : integer;         // korrekte name
    public
        constructor Create(inKN: string; inVBR: string; inVU: integer);
        function kryKlientNaam() : string ;
        function kryVoorkeurBerader(): string;
        function kryVroegsteUur() : integer;
        function toString() : string;
    end;

implementation
```

```
// Vraag 2.2 - 4
// korrekte opskrif en name/tipe/volgorde soos gevra
constructor TKlient.Create(inKN: string; inVBR: string; inVU: integer);
begin
    klientNaam := inKN;           // stringvelde op parameters gestel
    voorkeurBerader := inVBR;

    if (inVU > 16) then           // vroegsteUur op 16 gestel indien inVU > 16
    begin
        vroegsteUur := 16;
    end
    else                         // stel andersins op inVU
    begin
        vroegsteUur := inVU;
    end;
end;

// Vraag 2.3 - 2
// korrekte opskrifte vir al drie metodes
// korrekte terugsendings vir al drie metodes
function TKlient.kryKlientNaam() : string;
begin
    Result := klientNaam;
end;

function TKlient.kryVoorkeurBerader() : string;
begin
    Result := voorkeurBerader;
end;

function TKlient.kryVroegsteUur() : integer;
begin
    Result := vroegsteUur;
end;

// Vraag 2.4 - 4
function TKlient.toString() : string;    // korrekte opskrif
begin
    Result := klientNaam + #9 + voorkeurBerader + #9 + 'Vroegste '
        + IntToStr(vroegsteUur) + ':00';
    // korrekte terugsending
    // alle elemente ingesluit
    // formaat korrek met inbegrip van inkeping en :00
end;
end.
```

**VRAAG 3****uTydGleuf.pas**

```
unit uTydGleuf;

interface
    uses SysUtils;

    // Vraag 3.1 - 3
    // klas korrek geskep
    type TTydGleuf = class
        private
            berader : string;      // privaat
            beginUur : integer;    // tipes/name korrek en soos gegee
            isBeskikbaar : boolean;
            function kryEindUur(): integer;
        public
            constructor Create(inBR: string; inBU: integer);
            procedure stelIsBeskikbaar(inIB: boolean);
            function kryBerader() : string ;
            function kryBeginUur(): integer;
            function kryIsBeskikbaar(): boolean;
            function toString() : string;
        end;

implementation
    // Vraag 3.2 - 4
    // opskrif korrek
    // neem slegs string en heelgetal in - nie Boole nie
    // Boole by verstek op true (in konstruktor of veldverklaring)
    constructor TTydgleuf.Create(inBR: string; inBU: integer);
    begin
        berader := inBR;    // stel ander velde
        beginUur := inBU;
        isBeskikbaar := true;
    end;

    // Vraag 3.3 - 2
    // stel opskrifte korrek
    // stel toegekende waardes korrek

    procedure TTydGleuf.stelIsBeskikbaar(inIB: boolean);
    begin
        isBeskikbaar := inIB;
    end;
```

```

// Vraag 3.4 - 1
// al drie kry-metodes korrek
function TTydGleuf.kryBerader() : string;
begin
    Result := berader;
end;

function TTydGleuf.kryBeginUur() : integer;
begin
    Result := beginUur;
end;

function TTydGleuf.kryIsBeskikbaar() : boolean;
begin
    Result := isBeskikbaar;
end;

// Vraag 3.5 - 2
// korrekte opskrif met privaat (kontroleer hierbo)
function TTydGleuf.kryEindUur() : integer;
begin
    // stuur beginUur + 1 terug
    Result := beginUur + 1;
end;

// Vraag 3.6 - 4
function TTydGleuf.toString() : string; // korrekte opskrif
begin
    Result := berader + ': ' + IntToStr(beginUur) + ':00 - '
        + IntToStr(kryEindUur()) + ':00';
    // gebruik kryEindUur om eindtyd te kry
    // sluit ander velde in
    // formaat korrek
end;
end.

```

## VRAAG 4, 5, 6 EN 7

### uGleufBestuurder.pas

```

unit uGleufBestuurder;

interface
uses
    SysUtils, uKlient, uTydGleuf;

// Vraag 4.1 - 4
type TGleufBestuurder = class // klas korrek geskep
private
    kSkik : array[1..20] of TKlient; // kliëntskikking geskep
    tSkik : array[1..40] of TTydGleuf; // tydgleufskikking geskep
    // beide privaat met korrekte groottes
public
    constructor Create();

```

```

    function vertoonAlleKliente() : string;
    procedure genereerTydGleuwe();
    function vertoonAlleBeskikbareTydGleuwe() : string;
    function genereerBespreekteGleuwe() : string;
end;
implementation

// Vraag 4.2 - 8
constructor TGleufBestuurder.Create();
var
    inFile : TextFile;
    klientnaam, voorkeurbr, vroegstebg : string;
    vroegste, i : integer;
begin
    AssignFile(inFile, 'kliente.txt'); // lees lêer korrek
    Reset(inFile);

    // korrekte lusstruktuur
    // korrekte waarde om korrekte lees van 20 kliënte in 60 reëls moontlik te
maak
    for i :=1 to length(kSkik) do
    begin
        Readln(inFile, klientnaam); // lees drie reëls
        Readln(inFile, voorkeurbr);
        Readln(inFile, vroegstebg);
        vroegste := StrToInt(vroegstebg); // korrekte omskakeling

        // skep kliënt
        // voeg nuwe kliënt by die korrekte gleuf by
        kSkik[i] := TKlient.Create(klientnaam, voorkeurbr, vroegste);
        // korrekte bywerking (kan deel wees van opskrif)
    end
end;

// Vraag 4.3 - 4
// korrekte opskrif
function TGleufBestuurder.verteonAlleKliente() : string;
var
    i : integer;
begin
    Result := '';
    // lus om deur korrekte skikking en korrekte lengte te loop
    for i:=1 to length(kSkik) do
        // bou 'n string korrek op
        Result := Result + kSkik[i].toString() + #10#13;
        // stuur die opgeboude string terug
    end;
end;

```

```
// Vraag 6.1 - 11
// korrekte opskrif
procedure TGleufBestuurder.genereerTydGleuwe();
var
    inFile : TextFile;
    line, berader : string;
    i, h, tcount : integer;
begin
    // maak lêer oop om korrek te lees
    AssignFile(inFile, 'beraders.txt');
    Reset(inFile);

    tcount := 1;           // skep en inisialiseer teller korrek
    Readln(inFile, line);  // lees een reël

    // gebruik lus om tydgleuwe suksesvol te genereer
    // tydgleuwe gelaai
    for i:=1 to 5 do
    begin
        if (i < 5) then
        begin
            // verdeel beradername korrek
            berader := Copy(line, 1, Pos(',', line) -1 );
            Delete(line, 1, Pos(',', line));
        end
        else
            berader := line;

        for h:=8 to 16 do
        begin
            // sluit etensuur uit
            if (h <> 12) then
            begin
                // tydgleuf geskep met korrekte parameters
                // by korrekte skikking in korrekte posisie bygevoeg
                tSkik[tcount] := TTydGleuf.Create(berader, h);
                tcount := tcount + 1; // inkrementeer teller laaste
            end;
        end;
    end;
end;

// Vraag 6.2 - 4
// korrekte opskrif
function TGleufBestuurder.verteonAlleBeskikbareTydGleuwe() : string;
var
    i : integer;
begin
    Result := '';
    // lus om deur korrekte skikkinglengte te loop
    for i:=1 to length(tSkik) do
        // kontroleer of gleuf beskikbaar is
        if (tSkik[i].kryIsBeskikbaar()) then
            // sluit elke tydgleuf in korrekte formaat in
            Result := Result + tSkik[i].toString();
    end;
end;
```



```

// Vraag 7.1 - 13
// opskrif korrek
function T GleufBestuurder.genereerBespreekteGleuwe() : string;
var
    k, t : integer;
begin
    // skep opskrif met nuwe reël
    Result := 'Afsprake: ' + #10#13;

    // lus om deur kliëntskikking te loop
    for k:=1 to length(kSkik) do
        // lus om deur tydgleufskikking te loop
        for t:=1 to length(tSkik) do
            // kontroleer tydgleufberader teen kliënt se voorkeur
            if (tSkik[t].kryBerader()
                = kSkik[k].kryVoorkeurBerader()) then
                // kontroleer beginuur teen kliënt
                if (tSkik[t].kryBeginUur() >= kSkik[k].kryVroegsteUur()) then
                    // kontroleer of tydgleuf beskikbaar is
                    // korrekte voorwaardelike skakel (aanvaar ook and)
                    if (tSkik[t].kryIsBeskikbaar()) then
                        begin
                            // stel ooreenstemmende tydgleuf se beskikbaarheid op false
                            tSkik[t].stelIsBeskikbaar(false);

                            // sluit slegs die inligting in wat gevra is (en niks meer nie)
                            // formatering korrek
                            Result := Result + kSkik[k].kryKlientNaam() + ' ('
                                + IntToStr(kSkik[k].kryVroegsteUur()) + ') sien '
                                + tSkik[t].toString() + #10#13;
                            break; // onderbreking wanneer gevind (of gebruik vlaggie)
                        end
                    end;

                    // stuur korrekte string terug
                end;
            end;
        end;
    end;
end.

```

## ALTERNATIEF MET WHILE EN VLAGGIE

```

// Vraag 7.1 - 13
// opskrif korrek
function T GleufBestuurder.genereerBespreekteGleuwe() : string;
var
    k, t : integer;
    found : boolean;
begin
    // skep opskrif met nuwe reël
    Result := 'Afsprake: ' + #10#13;

    // lus om deur kliëntskikking te loop
    for k:=1 to length(kSkik) do
        begin
            // lus om deur tydgleufskikking te loop
            t := 0;
            found := false;
            while (t < length(tSkik)) and (found = false) do
                begin

```

```

    // kontroleer tydgleufberader teen kliënt se voorkeur
    if (tSkik[t].kryBerader()
        = kSkik[k].kryVoorkeurBerader()) then
    // kontroleer beginuur teen kliënt
    if (tSkik[t].kryBeginUur() >= kSkik[k].kryVroegsteUur()) then
    // kontroleer of tydgleuf beskikbaar is
    // korrekte voorwaardelike skakel (aanvaar ook and)
    if (tSkik[t].kryIsBeskikbaar()) then
    begin
        // stel ooreenstemmende tydgleuf se beskikbaarheid op false
        tSkik[t].stelIsBeskikbaar(false);

        // sluit slegs die inligting in wat gevra is (en niks meer nie)
        // formatering korrek
        Result := Result + kSkik[k].kryKlientNaam() + ' ('
            + IntToStr(kSkik[k].kryVroegsteUur()) + ') sien '
            + tSkik[t].toString() + #10#13;
        found := true;    // gebruik vlaggie
    end;
    t := t+1;
end;
end;
    // stuur korrekte string terug
end;
end.

```

## BeradingGK

```

program HoopProjek;

{$APPTYPE CONSOLE}

{$R *.res}

uses
    System.SysUtils,
    uKlient in 'uKlient.pas',
    uTydGleuf in 'uTydGleuf.pas',
    uGleufBestuurder in 'uGleufBestuurder.pas';

var
    input : string;
    gb : TGleufBestuurder;
begin
    // Vraag 5.1 - 1
    // teksgebaseerde koppelvlak vir toevoer/afvoer
    try
        // Vraag 5.2 - 1
        // GleufBestuurder geskep
        gb := TGleufBestuurder.Create();

        // Vraag 5.3 - 1
        // roep drukAlleKliente() korrek
        WriteLn(gb.drukAlleKliente());
    end;
end.

```

```
// Vraag 6.3 - 2
// roep genereerTydGleuwe() korrek
gb.genereerTydGleuwe();
// roep drukAlleBeskikbareTydGleuwe
WriteLn(gb.drukAlleBeskikbareTydGleuwe());

// Vraag 7.2 - 1
// metode geroep en gedrukte string teruggestuur
WriteLn(gb.genereerBespreekteGleuwe());
```

```
ReadLn(input);
except
on E: Exception do
    Writeln(E.ClassName, ': ', E.Message);
end;
end.
```

## AFVOER

## AFDELING A

### VRAAG 1.1

KlientID	KlientNaam	Ouderdom	Poskode
28	Cynthia Fourie	18	1500
8	Lindewe Khoza	18	1500
16	Maria Nkosi	21	0152
3	Steve Jacobs	21	0125
33	Bianca Abrahams	22	0160
26	Zandile Methembu	22	1600
2	Leo Sithole	22	1240
31	Al Naidoo	23	0127
23	Willem du Plessis	23	1600
4	Pat Khumalo	23	1251
30	Bongiwe Mokoena	25	0160
7	Sibongile Ngcobo	25	1240
37	Hendrick van Zyl	27	0180
15	Peter Zwane	30	1240

### VRAAG 1.2

KlientID	KlientNaam	Ouderdom	Poskode
3	Steve Jacobs	21	0125
6	Musi Mahlangu	38	0120
18	Patricia Williams	17	0120
31	Al Naidoo	23	0127
35	Moses Sibisi	31	0121
38	Lucky Shabangu	16	0123

**VRAAG 1.3**

*Let daarop dat die laaste twee syfers ewekansig gegenereer word*

KlientNaam	Expr1001
John Dlamini	ni16
Leo Sithole	le15
Steve Jacobs	bs16
Pat Khumalo	lo11
Sipho Nkosi	si12
Musi Mahlangu	gu11
Sibongile Ngcobo	bo11
Lindewe Khoza	za13
Johannes Botha	ha11
Bongani Radebe	be11
Caleb Pillay	ay14
Anna Mthembu	bu15
Thabo Baloyi	yi14
Mpho Sithole	le16
Peter Zwane	ne11
Maria Nkosi	si12
Samuel Naidoo	oo15
Patricia Williams	ms14
Joseph Mkhize	ze13
Zenele Ndlovu	vu12
Daniel Dlamini	ni15
Jabu Chauke	ke17
Willem du Plessis	is14
Jan Venter	er11
Linda Gumede	de16
Zandile Methembu	bu13
Robert Maseko	ko14
Cynthia Fourie	ie14
Nomsa Baloyi	yi12
Bongiwe Mokoena	na13
Al Naidoo	oo11
Phindile Ntuli	li15
Bianca Abrahams	ms12
Freddy Chetty	ty17
Moses Sibisi	si11
Vusi Khumalo	lo14
Hendrick van Zyl	yl12
Lucky Shabangu	gu17

**VRAAG 1.4**

KlientID	Plek
2	Greenside
5	Greenside
24	Greenside
29	Greenside
36	Greenside
4	Greenside
15	Greenside
27	Greenside
2	Greenside
24	Greenside
29	Greenside
14	Greenside
4	Greenside
2	Greenside
24	Greenside
29	Greenside
4	Greenside
15	Greenside
27	Greenside
14	Greenside
29	Greenside
4	Greenside
15	Greenside

**VRAAG 1.5**

BeraderNaam	Tarief
Siyanda Mabuza	120
Joshua Hendricks	120
Linda September	120

**VRAAG 1.6**

Korrekte afvoer toe dit op 16 Oktober 2020 geloop is.  
'n Ander huidige datum sal die afvoer verander.

Plek	GetalAfsprake
Bergsig	22
Panorama	17

**VRAAG 1.7**

AfspraakID	KlientID	KlientNaam	BeraderNaam	Expr1004
6	5	Sipho Nkosi	Joshua Hendricks	90
35	14	Mpho Sithole	Thabo Matlala	112.5
69	14	Mpho Sithole	Thabo Matlala	112.5
24	25	Linda Gumede	Matthew Kunene	135
44	25	Linda Gumede	Matthew Kunene	135
8	29	Nomsa Baloyi	Joshua Hendricks	90
34	29	Nomsa Baloyi	Joshua Hendricks	90
57	29	Nomsa Baloyi	Joshua Hendricks	90
70	29	Nomsa Baloyi	Thabo Matlala	112.5
9	36	Vusi Khumalo	Joshua Hendricks	90

**AFDELING B****FINALE AFVOER**

John Dlamini	Vernon Booyesen	Vroegste 10:00
Leo Sithole	Matthew Kunene	Vroegste 13:00
Steve Jacobs	Vernon Booyesen	Vroegste 8:00
Pat Khumalo	Heather Modise	Vroegste 6:00
Sipho Nkosi	Vernon Booyesen	Vroegste 16:00
Musi Mahlangu	Joshua Hendricks	Vroegste 9:00
Sibongile Ngcobo	Joshua Hendricks	Vroegste 8:00
Lindewe Khoza	Siyanda Mabuza	Vroegste 10:00
Johannes Botha	Vernon Booyesen	Vroegste 10:00
Bongani Radebe	Heather Modise	Vroegste 9:00
Caleb Pillay	Vernon Booyesen	Vroegste 9:00
Anna Mthembu	Siyanda Mabuza	Vroegste 11:00
Thabo Baloyi	Heather Modise	Vroegste 12:00
Mpho Sithole	Matthew Kunene	Vroegste 12:00
Peter Zwane	Vernon Booyesen	Vroegste 9:00
Maria Nkosi	Joshua Hendricks	Vroegste 14:00
Samuel Naidoo	Vernon Booyesen	Vroegste 9:00
Patricia Williams	Heather Modise	Vroegste 8:00
Joseph Mkhize	Matthew Kunene	Vroegste 7:00
Zenele Ndlovu	Siyanda Mabuza	Vroegste 14:00

Vernon Booyesen: 8:00 - 9:00  
 Vernon Booyesen: 9:00 - 10:00  
 Vernon Booyesen: 10:00 - 11:00  
 Vernon Booyesen: 11:00 - 12:00  
 Vernon Booyesen: 13:00 - 14:00  
 Vernon Booyesen: 14:00 - 15:00  
 Vernon Booyesen: 15:00 - 16:00  
 Vernon Booyesen: 16:00 - 17:00  
 Matthew Kunene: 8:00 - 9:00  
 Matthew Kunene: 9:00 - 10:00  
 Matthew Kunene: 10:00 - 11:00  
 Matthew Kunene: 11:00 - 12:00  
 Matthew Kunene: 13:00 - 14:00  
 Matthew Kunene: 14:00 - 15:00  
 Matthew Kunene: 15:00 - 16:00

Matthew Kunene: 16:00 - 17:00  
Heather Modise: 8:00 - 9:00  
Heather Modise: 9:00 - 10:00  
Heather Modise: 10:00 - 11:00  
Heather Modise: 11:00 - 12:00  
Heather Modise: 13:00 - 14:00  
Heather Modise: 14:00 - 15:00  
Heather Modise: 15:00 - 16:00  
Heather Modise: 16:00 - 17:00  
Siyanda Mabuza: 8:00 - 9:00  
Siyanda Mabuza: 9:00 - 10:00  
Siyanda Mabuza: 10:00 - 11:00  
Siyanda Mabuza: 11:00 - 12:00  
Siyanda Mabuza: 13:00 - 14:00  
Siyanda Mabuza: 14:00 - 15:00  
Siyanda Mabuza: 15:00 - 16:00  
Siyanda Mabuza: 16:00 - 17:00  
Joshua Hendricks: 8:00 - 9:00  
Joshua Hendricks: 9:00 - 10:00  
Joshua Hendricks: 10:00 - 11:00  
Joshua Hendricks: 11:00 - 12:00  
Joshua Hendricks: 13:00 - 14:00  
Joshua Hendricks: 14:00 - 15:00  
Joshua Hendricks: 15:00 - 16:00  
Joshua Hendricks: 16:00 - 17:00

// Hierdie lys kan ander tye hê na gelang van die algoritme  
// wat deur die kandidaat gebruik is. Dit moet egter in ooreenstemming wees met  
// die vroegste tyd vir elke kliënt (sien hierbo).

Afsprake:

John Dlamini (10) sien Vernon Booyesen: 10:00 - 11:00  
Leo Sithole (13) sien Matthew Kunene: 13:00 - 14:00  
Steve Jacobs (8) sien Vernon Booyesen: 8:00 - 9:00  
Pat Khumalo (6) sien Heather Modise: 8:00 - 9:00  
Sipho Nkosi (16) sien Vernon Booyesen: 16:00 - 17:00  
Musi Mahlangu (9) sien Joshua Hendricks: 9:00 - 10:00  
Sibongile Ngcobo (8) sien Joshua Hendricks: 8:00 - 9:00  
Lindewe Khoza (10) sien Siyanda Mabuza: 10:00 - 11:00  
Johannes Botha (10) sien Vernon Booyesen: 11:00 - 12:00  
Bongani Radebe (9) sien Heather Modise: 9:00 - 10:00  
Caleb Pillay (9) sien Vernon Booyesen: 9:00 - 10:00  
Anna Mthembu (11) sien Siyanda Mabuza: 11:00 - 12:00  
Thabo Baloyi (12) sien Heather Modise: 13:00 - 14:00  
Mpho Sithole (12) sien Matthew Kunene: 14:00 - 15:00  
Peter Zwane (9) sien Vernon Booyesen: 13:00 - 14:00  
Maria Nkosi (14) sien Joshua Hendricks: 14:00 - 15:00  
Samuel Naidoo (9) sien Vernon Booyesen: 14:00 - 15:00  
Patricia Williams (8) sien Heather Modise: 10:00 - 11:00  
Joseph Mkhize (7) sien Matthew Kunene: 8:00 - 9:00  
Zenele Ndlovu (14) sien Siyanda Mabuza: 14:00 - 15:00