



NASIONALE SENIOR CERTIFIKAAT-EKSAMEN
NOVEMBER 2018

INLIGTINGSTEGNOLOGIE: VRAESTEL II

NASIENRIGLYNE

Tyd: 3 uur

120 punte

Hierdie nasienriglyne word voorberei vir gebruik deur eksaminatore en hulpeksaminatore. Daar word van alle nasieners vereis om 'n standaardiseringsvergadering by te woon om te verseker dat die nasienriglyne konsekwent vertolk en toegepas word tydens die nasien van kandidate se skrifte.

Die IEB sal geen gesprek aanknoop of korrespondensie voer oor enige nasienriglyne nie. Daar word toegegee dat verskillende menings rondom sake van beklemtoning of detail in sodanige riglyne mag voorkom. Dit is ook voor die hand liggend dat, sonder die voordeel van bywoning van 'n standaardiseringsvergadering, daar verskillende vertolkings mag wees oor die toepassing van die nasienriglyne.

AFDELING A**VRAAG 1**Vraag 1.1

```
SELECT *
FROM Toeris
WHERE Epos LIKE '%seattletimes.com'    -- JavaDB en MySQL

WHERE Epos LIKE '*seattletimes.com'    -- Access
```

ALTERNATIEF:

JavaDB: WHERE SUBSTR(Epos, LENGTH(Epos)-16+1, 16) = 'seattletimes.com'**Access:** WHERE RIGHT(Epos, 16) = 'seattletimes.com'Vraag 1.2

```
UPDATE Toeris
SET Hotel = 'Three Seasons Hotel'
WHERE Hotel = 'Lunar Hotel'
```

Vraag 1.3

```
SELECT EkskursieNaam,
EindUur - BeginUur
AS Duur
FROM Ekskursie
WHERE EindUur <= 11          -- alternatief < 12
AND EindUur - BeginUur <= 3
```

Vraag 1.4

```
SELECT Hotel,
COUNT(*)
FROM Toeris
GROUP BY Hotel
HAVING COUNT(*)>=3
```

Vraag 1.5

```
SELECT ToerisNaam
FROM Toeris
WHERE ToerisID NOT IN
(SELECT ToerisID
  FROM Bespreking)
ORDER BY ToerisNaam
```

ALTERNATIEF:

```
SELECT ToerisNaam
FROM Toeris LEFT JOIN Bespreking
ON Toeris.ToerisID = Bespreking.ToerisID
WHERE EkskursieID IS NULL
ORDER BY ToerisNaam
```

Vraag 1.6**JavaDB**

```
SELECT EkskursieNaam,
       SUBSTR (EkskursieNaam, 1, 3)
       ||
       CHAR (INT(RANDOM()*90)+10 )   korrekte formule
FROM Ekskursie
```

MySQL

```
SELECT EkskursieNaam,
       CONCAT(
         SUBSTR( EkskursieNaam, 1, 3 ) ,
         FLOOR ( RAND ( ) *90 ) +10 )   korrekte formule
FROM Ekskursie
```

Access

```
SELECT EkskursieNaam,
       LEFT (EkskursieNaam, 3)
       &
       INT (Rnd(EkskursieID) *90 + 10)   korrekte formule
                                     -- Access Rnd moet ander saad hê
FROM Ekskursie
```

Vraag 1.7

```
INSERT INTO Bespreking (ToerisID, EkskursieID, Koste, EkskursieDatum)
                                     -- 2 korrekte tabel en alle velde aanwesig
SELECT ToerisID, EkskursieID, -- 1 korrekte volgorde (stem ooreen met insert)
       HuidigeKoste+Bybetaling,
       CURRENT_DATE -- 1 huidige datum of NOW() vir Access (nie vaste kode nie)
FROM Ekskursie , Toeris -- trek 1 af indien join aanwesig is
WHERE EkskursieID IN ( 1, 7, 13, 5 ) -- of 4 voorwaardes met OR
AND Hotel = 'President Hotel'
```

AFDELING B**JAVA-OPLOSSING****VRAAG 2: Stop-klas**

// V2.1

public class Stop {

// V2.2

```
private String stopNaam;
private String roeteKodes;
private int stopTipe;
```

Privaat
Korrekte tipe
Benoem soos gevra

// V2.3

```
public static final int STOPTIPE_KAFEE = 1;
public static final int STOPTIPE_SKUILING = 2;
public static final int STOPTIPE_SNEL = 3;
public static final int STOPTIPE_ANDER = 4;
```

Finaal/konstante
Korrek benoem
Heelgetalwaardes korrek

// V2.4

```
public Stop(String inStopNaam, String inRoeteKodes, int inStopTipe)
{
```

```
    stopNaam = inStopNaam;
    roeteKodes = inRoeteKodes;
```

Eienskappe korrek gestel

```
    if (inStopTipe == STOPTIPE_KAFEE ||
        inStopTipe == STOPTIPE_SKUILING ||
        inStopTipe == STOPTIPE_SNEL)
```

Vergelyking korrek

Kontroleer teen konstantes

Gebruik OR korrek

```
    {
        stopTipe = inStopTipe;
        // aanvaar ook indien elkeen individueel gestel word
    }
```

Stel slegs indien geldig

else

```
{
    stopTipe = STOPTIPE_ANDER;
}
```

Stel andersins op TIPE ander
(nie letterlike heelgetal nie)

}

// V2.5

```
public String kryStopTipeNaam()
{
```

```
    switch (stopTipe) // if/else if/else aanvaarbaar
    {
```

```
        case STOPTIPE_KAFEE:
            return "kafee";
        case STOPTIPE_SNEL:
            return "snel";
        case STOPTIPE_SKUILING:
            return "skuiling";
        default:
            return "ander";
    }
```

Stuur korrekte string terug
op grond van konstantes

```
}
```

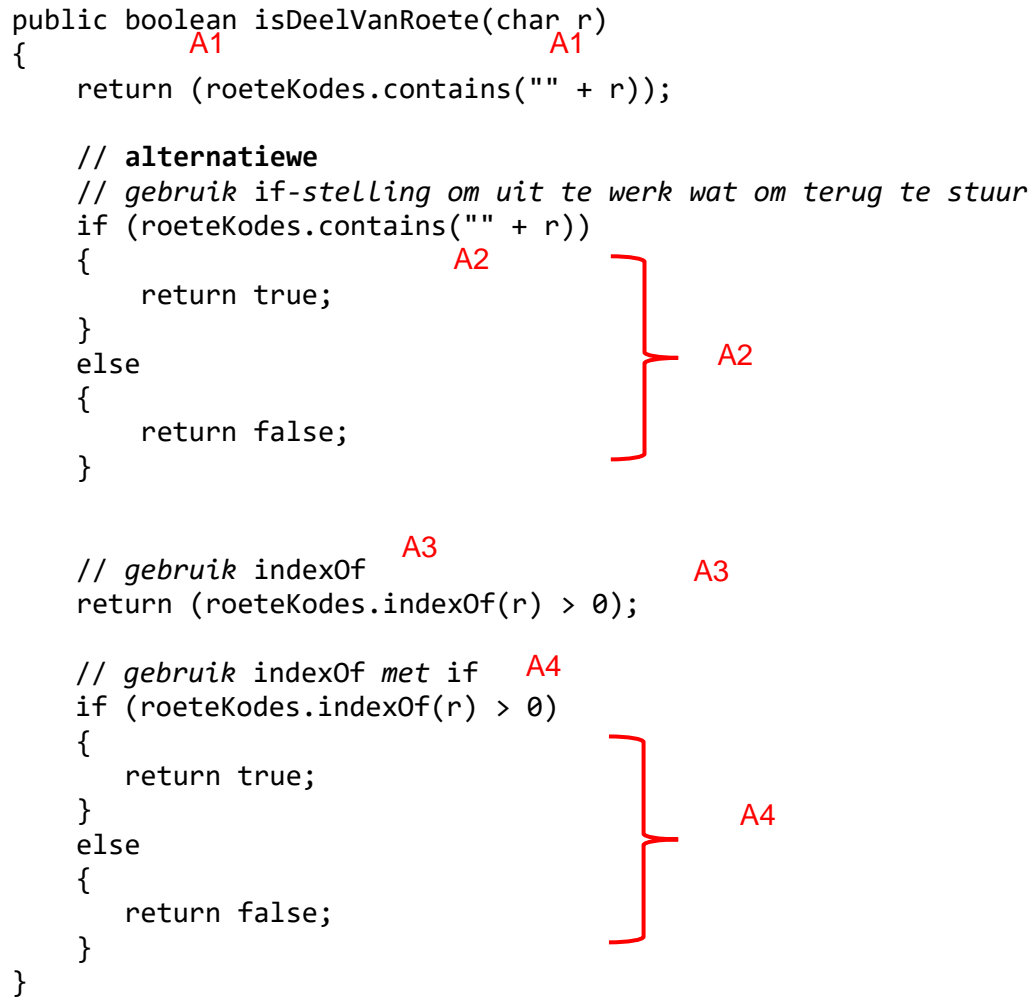
// V2.6

```
public boolean isDeelVanRoete(char r)
{
    return (roeteKodes.contains("" + r));

    // alternatiewe
    // gebruik if-stelling om uit te werk wat om terug te stuur
    if (roeteKodes.contains("" + r))
    {
        return true;
    }
    else
    {
        return false;
    }

    // gebruik indexOf
    return (roeteKodes.indexOf(r) > 0);

    // gebruik indexOf met if
    if (roeteKodes.indexOf(r) > 0)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```



// V2.7

```
public String toString()
{
    return kryStopTipeNaam() + "\t" + stopNaam;
}
```

alle inhoud ingesluit
korrekte formaat

}

VRAAG 3 EN 5: Roete-klas

// V3.1

```
public class Roete {
    private char roeteKode;
    private boolean isSirkel;
    private Stop[] stoppe;
```

Privaat
Korrekte tipe

// V3.2

```
public Roete(char inRoeteKode, boolean inIsSirkel)
{
    roeteKode = inRoeteKode;
    isSirkel = inIsSirkel;
```

opskrif
Korrek gestel

// V3.3

```
public void stelStoppe(Stop[] inStoppe)
{
    stoppe = inStoppe;
```

opskrif
Korrek gestel

// V3.4

```
public char kryRoeteKode()
{
    return roeteKode;
```

// V3.5

```
public Stop kryStopBy(int num)
{
    if (num >= 0 && num < stoppe.length)
    {
        return stoppe[num];
    }
    else return null;

    // aanvaar ook indien argument by 1 begin solank gleufverwerking
    //korrek is en V7.1 ook ooreenstem
    /*
    if (num >= 1 && num <= stoppe.length)
    {
        return stoppe[num-1];
    }
    else return null;
    */
```

A1
objekt
A1 – stuur objek terug
A1 stuur nul terug vir nie geldig nie
A2
objekt
A2 – stuur objek terug
A2 stuur nul terug vir nie geldig nie

```
// V5
public String toString()
{
    String toRet = roeteKode + " - ";
```

```
    if (stoppe.length == 0) // aanvaar ook <= 0
```

```
    {
        toRet += "Ongeldig";
    }
    else
    {
```

Doeltreffendheidspunt:

2 – geen roeteverwerking vir ongeldige stop nie
 1 – ongeldigheidstoets word gedoen, maar daar is onnodige verwerking/kontrolering van roetetipe
 0 – geen ongeldigheidstoets word gedoen nie

```
        if (isSirkel)
        {
            toRet += "Sirkel";
        }
        else
        {
            toRet += "Lineêr";
        }
```

voeg sirkel/lineêr korrek by

```
    for (int i = 0; i < stoppe.length; i++)
    {
        toRet += "\n-> " + (i+1) + "\t" + stoppe[i];
    }
```

sluit stoppe in

Doeltreffendheidspunt – kode om die eerste deel van die skikking te druk (waar skikkings vir sirkel/lineêr gemeenskaplike formaat het), verskyn slegs een keer in die kode, d.w.s. geen deel daarvan verskyn op veelvuldige plekke nie

```
    if (isSirkel)
    {
        toRet += "\n-> 1\t" + stoppe[0];
    }
    else
```

sluit eerste stop in
indien sirkel

```
    {
        for (int i = stoppe.length - 2 ; i >= 0; i--)
        {
            toRet += "\n-> " + (i+1) + "\t" + stoppe[i];
        }
    }
```

lus om stoppe in omgekeerde volgorde in te sluit van tweede laaste gleuf af

```
    }
    return toRet;
}
```

alle stoppe in korrekte formaat teruggestuur

```
}
```

VRAAG 4 EN 7.1: ToerBestuurder-klas

```
import java.io.*;
```

```
// V4.1
```

```
public class ToerBestuurder {
```

```
    // V4.2
```

```
    private Stop[] alleStoppe = new Stop[100];
```

```
    private int stopCount = 0;
```

Privaat

Korrekte tipe

Korrekte aanvangswaardes

```
    // V4.3
```

korrekte metodeopskrif

```
    public ToerBestuurder (String fn)
```

```
    {
```

```
        try
```

```
        {
```

```
            BufferedReader br = new BufferedReader(new FileReader(fn));
```

```
            String line = br.readLine();
```

```
            while (line != null)          lus om tot einde van lêer te lees
```

```
            {
```

```
                String[] tokens = line.split(",");
```

Verdeel in elemente

```
                String sNaam = tokens[0];
```

```
                int sTipe = Integer.parseInt(tokens[1]);
```

```
                String rKodes = tokens[2];
```

```
                alleStoppe[stopCount] = new Stop(sNaam, rKodes, sTipe);
```

```
                // 1 L1 inkrementeer teller
```

```
                stopCount++;
```

```
                line = br.readLine();      lees elke reël korrek in lus
```

```
            }
```

```
        } catch (Exception ex) { // of gee uitsondering
```

```
            System.out.println("Lêer nie gevind nie");
```

```
        }
```

```
    }
```

```
    // V4.4
```

korrekte metodeopskrif

```
    public Roete kryRoeteMetKode(char inKode, boolean inIsSirkel)
```

```
    {
```

```
        Roete r = new Roete(inKode, inIsSirkel);
```

skep Roete-objek korrek

0 – indien alternatiewe konstruktore geskep en gebruik is

Doeltreffendheidspunt: Lêer word nooit herlees nie – slegs Stop-skikking word gebruik. Geen ander openbare metode in Roete/Stop-klas geskep om enige deel van hierdie metode te doen nie.

Enige manier om skikking tot korrekte lengte reg te stel
 (2 punte – sien elke alternatiewe kode vir puntetoekenningsregmerkies)
 alternatief 1: tel eers, skep dan presiese skikking en kopieer
 alternatief 2: skep skikking van 100 en verminder dan grootte na vulling

// Alternatief 1: Skep presiese skikking en laai

```
int count = 0;
for (int i = 0; i < stopCount; i++) {
    if (alleStoppe[i].isDeelVanRoete(inKode))
    {
        count++;
    }
}

Stop[] arr = new Stop[count];

count = 0;

for (int i = 0; i < stopCount; i++) {
    if (alleStoppe[i].isDeelVanRoete(inKode))
    {
        arr[count] = alleStoppe[i];
        count++;
    }
}
// Einde van Alternatief 1
```

A1 Manier om lengte van skikking reg te stel (tel eers)

A1: skep nuwe Stop-skikking

A1: lus om korrek deur stoppe te gaan

A1: kontroleer en voeg by indien stop deel is van roete

// Alternatief 2: Skep dieselfde grootte as
 // alleStoppe – kopieer oor en verkort skikking

```
Stop[] tempArr = new Stop[alleStoppe.length];
int count = 0;
for (int i = 0; i < stopCount; i++) {
    if (alleStoppe[i].isDeelVanRoete(inKode))
    {
        tempArr[count] = alleStoppe[i];
        count++;
    }
}

Stop[] arr = new Stop[count];
System.arraycopy(tempArr, 0, arr, 0, count);
// of gebruik for-lus om na werklike skikking te kopieer

// Einde van Alternatief 2
```

A2: skep nuwe Stop-skikking met voldoende lengte

A2: lus om korrek deur stoppe te gaan

A2: kontroleer en voeg by indien stop deel is van roete

A2: skep nuwe Stop-skikking

A2: Vaste skikking-grootte deur oor skikking van korrekte grootte te kopieer

r.stelStoppe(arr);

return r;

}

// V7.1

```

public String werkStopPunteUit(Roete r1, Roete r2)
{
    // korrekte openbaremetode-opskrif: enige naam, moet openbaar
    // wees, neem twee Roete-objekte in, stuur string korrek terug

    // Punte toegeken volgens doelwitte bereik:
    // 1 - 01: lus om deur roete 1-stoppe te gaan
    // 1 - 02: toets of dit ook aan roete 2 behoort
    // 1 - 03: doeltreffendheid – geen herlees van lêer, geen ander metode geskep nie
    // 1 - 04: stringaaneenskakeling korrek gedoen
    // 1 - 05: korrekte terugsending (0 indien daar terugsending is maar nie korrek nie)

    String toret = "";

    // alternatief 1: werk met die twee roetes

    int count = 0;
    Stop s = r1.kryStopBy(count);
    // s nie nodig nie – kan direk met r1.kryStopBy(count) werk;
    // A1: O1 lus om korrek deur r1-stoppe te gaan
    while (s != null)
    {
        // A1: O2 toets om te sien of stop deel is van r2
        if (s.isDeelVanRoete(r2.kryRoeteKode()))
        {
            // A1: O3 geen herlees van lêer nie, geen nuwe openbare metode gebruik nie
            toret += s + "\n";
            // A1: O4 Stringaaneenskakeling korrek gedoen
        }
        count++;
        s = r1.kryStopBy(count);
    }
    // A1: O5 korrekte terugsending
    return toret;

    // alternatief 2: werk met die alleStoppe-skikking
    for (int i = 0; i < stopCount; i++)
    {
        // A2: O1 lus om deur r1-stoppe te gaan
        if (alleStoppe[i].isDeelVanRoete(r1.kryRoeteKode())
            // A2: O2 toets om te sien of stop deel is van r2
            &&
            alleStoppe[i].isDeelVanRoete(r2.kryRoeteKode()))
        {
            // A2: O4 Stringaaneenskakeling korrek gedoen
            toret += alleStoppe[i] + "\n";
        }
        // A2: O3 geen herlees van lêer nie, geen nuwe openbare metode gebruik nie
    }
    // A2: O5 korrekte terugsending
    return toret;
}
}

```

VRAAG 6 EN 7.2: ToerGK-klas**// V6.1**

public class ToerGK {

public static void main(String[] args) {

// V6.2

ToerBestuurder sm = new ToerBestuurder("data.txt");

// V6.3

Roete roeteR = sm.kryRoeteMetKode('R', true);

Roete roeteY = sm.kryRoeteMetKode('Y', false);

// V6.4 – druk albei roetes

System.out.println(roeteR);

System.out.println(roeteY);

// V7.2

System.out.println(sm.werkStopPunteUit(roeteR, roeteY));

}

}

DELPHI-OPLOSSING**VRAAG 2: Stop-klas**

```

unit uStop;

interface
  uses SysUtils;

  // V2.1
  type TStop = class

    // V2.2
    private
      stopNaam : string;
      roeteKodes : string;
      stopTipe : integer;

    // V2.3
    public
      const
        STOPTIPE_KAFEE = 1;
        STOPTIPE_SKUILING = 2;
        STOPTIPE_SNEL = 3;
        STOPTIPE_ANDER = 4;

      constructor Create(inStopNaam , inRoeteKodes : string;
                        inStopTipe : integer);
      function kryStopTipeNaam() : string ;
      function isDeelVanRoete(r: char) : boolean;
      function toString() : string;
  end;

implementation
  // V2.4
  constructor TStop.Create(inStopNaam , inRoeteKodes : string;
                          inStopTipe : integer);
  begin
    stopNaam := inStopNaam;
    roeteKodes := inRoeteKodes;

    if (inStopTipe = STOPTIPE_KAFEE) or
       (inStopTipe = STOPTIPE_SKUILING) or
       (inStopTipe = STOPTIPE_SNEL) then
    begin
      stopTipe := inStopTipe;
      // aanvaar ook indien elkeen individueel gestel word
    end
    else
    begin
      stopTipe := STOPTIPE_ANDER
    end;
  end;
end;

```

Privaat
Korrekte tipe
Benoem soos gevra

Finaal/konstante
Korrek benoem
Heelgetalwaardes korrek

Eienskappe korrek gestel

Vergelyking korrek
Kontroleer teen konstantes
Gebruik OR korrek

Stel slegs indien geldig

Stel andersins op TIPE ander
(nie letterlike heelgetal nie)

```
// V2.5
function TStop.kryStopTipeNaam() : string ;
begin
  case stopTipe of
    STOPTYPE_KAFEE : Result := 'kafee';
    STOPTYPE_SNEL : Result := 'snel';
    STOPTYPE_SKUILING : Result := 'skuiling';
    else Result := 'ander';
  end;
end;
```

Stuur korrekte string terug
op grond van konstantes

```
// V2.6
function TStop.isDeelVanRoete(r: char) : boolean;
begin
  Result := Pos(r, roeteKodes) > 0;

  // gebruik if-stelling
  if Pos(r, roeteKodes) > 0 then
  begin
    Result := true;
  end
  else
  begin
    Result := false;
  end;
end;
```

```
// V2.7
function TStop.toString() : string;
begin
  Result := kryStopTipeNaam() + #9 + stopNaam;
end;
end.
```

alle inhoud ingesluit
korrekte formaat

VRAAG 3 EN 5: Roete-klas

```
unit uRoete;

interface
  uses uStop, SysUtils;

  type StopSkikking = array of TStop;

// V3.1
  type TRoete = class
  private
    roeteKode : char;
    isSirkel : boolean;
    stoppe : StopSkikking;

  public
    constructor Create(inRoeteKode : char; inIsSirkel : boolean);
```

Privaat
Korrekte tipe

```

    procedure stelStoppe(inStoppe : StopSkikking);
    function kryRoeteKode() : char;
    function kryStopBy(num : integer) : TStop;
    function toString() : string;
end;

```

implementation

```

// V3.2 opskrif
constructor TRoete.Create(inRoeteKode: Char; inIsSirkel: boolean);
begin
    roeteKode := inRoeteKode; Korrek gestel
    isSirkel := inIsSirkel;
end;

```

```

// V3.3 opskrif
procedure TRoete.stelStoppe(inStoppe : StopSkikking);
begin
    stoppe := inStoppe; Korrek gestel
end;

```

```

// V3.4
function TRoete.kryRoeteKode() : char;
begin
    Result := roeteKode;
end;

```

}

```

// V3.5
function TRoete.kryStopBy(num: integer) : TStop;
begin
    A1 A1
    if (num >= 0) and (num < Length(stoppe)) then objekt objekt
    begin
        Result := stoppe[num] A1 – stuur objek terug
    end
    else
    begin
        Result := nil A1 – stuur nul terug vir nie geldig nie
    end;
    // aanvaar ook indien argument by 1 begin solank gleufverwerking
    // korrek is en V7.1 ook ooreenstem
    {
        A2 A2
        if (num >= 1) and (num <= Length(stoppe)) then objekt objekt
        begin
            Result := stoppe[num-1] A2 – stuur objek terug
        end
        else
        begin
            Result := nil A2 – stuur objek terug
        end;
    }
end;

```

```

// V5
function TRoete.toString() : string;
var
    i : integer;
begin
    Result := roeteKode + ' - ';

```

```

if Length(stoppe) = 0 then
begin
  Result := Result + 'Ongeldig';
end
else
begin

```

// aanvaar ook ≤ 0

Doeltreffendheidspunt:

2 – geen roeteverwerking vir ongeldige stop nie
 1 – ongeldigheidstoets word gedoen, maar daar is
 onnodige verwerking/kontrolering van roetetipe
 0 – geen ongeldigheidstoets word gedoen nie

```

if (isSirkel) then
begin
  Result := Result + 'Sirkel';
end
else
begin
  Result := Result + 'Lineêr';
end;

```

voeg sirkel/lineêr korrek by

```

for i:= 0 to Length(stoppe)-1 do
begin
  Result := Result + #13#10 + '-> ' + IntToStr(i+1)
    + #9 + stoppe[i].toString();
end;

```

sluit stoppe in

Doeltreffendheidspunt – kode om die eerste deel van die skikking te druk
 (waar skikkings vir sirkel/lineêr gemeenskaplike formaat het), verskyn slegs een
 keer in die kode, d.w.s. geen deel daarvan verskyn op veelvuldige plekke nie

```

if (isSirkel) then
begin
  Result := Result + #13#10 + '-> 1' + #9 + stoppe[0].toString();
end

```

sluit eerste stop in indien sirkel

```

else
begin
  for i:= Length(stoppe)-2 downto 0 do
  begin
    Result := Result + #13#10 + '-> ' + IntToStr(i+1) + #9
      + stoppe[i].toString();
  end;
end;

```

lus om stoppe in omgekeerde
 volgorde in te sluit van tweede
 laaste gleuf af

alle stoppe in korrekte formaat
 teruggestuur

```

end;
end;
end.

```

VRAAG 4 EN 7.1 : ToerBestuurder-klas

```
unit uToerBestuurder;
```

```
interface
```

```
uses uStop, uRoete, SysUtils;
```

```
// V4.1
```

```
type TStopBestuurder = class
```

```
    // V4.2
```

```
    private
```

```
        alleStoppe : array[1..100] of TStop;
```

```
        stopCount : integer;
```

```
    public
```

```
        constructor Create(inFilename : string);
```

```
        function kryRoeteMetKode(inKode:char; inIsSirkel:boolean):TRoete;
```

```
        function werkStopPunteUit(r1, r2 : TRoete) : string;
```

```
end;
```

Privaat

Korrekte tipe

Korrekte aanvangswaardes

```
implementation
```

```
// V4.3
```

```
constructor TStopBestuurder.Create(inFilename : string);
```

```
var
```

```
    inFile : TextFile;
```

```
    line, sNaam, rKodes: string;
```

```
    sTipe: integer;
```

```
begin
```

```
    AssignFile(inFile, inFilename);
```

```
    Reset(inFile);
```

```
    while NOT EOF(infile) do
```

```
    begin
```

```
        Readln(inFile, line); lees elke reël korrek in lus
```

```
        sNaam := Copy(line, 1, Pos(',', line) -1 );
```

```
        Delete(line, 1, Pos(',', line));
```

Verdeel in elemente

```
        sTipe := StrToInt(Copy(line, 1, Pos(',', line) -1 ));
```

```
        Delete(line, 1, Pos(',', line));
```

```
        rKodes := line;
```

```
        Inc(stopCount);
```

```
        alleStoppe[stopCount] := TStop.Create(sNaam, rKodes, sTipe);
```

```
    end;
```

```
end;
```

```
// V4.4
```

```
function TStopBestuurder.kryRoeteMetKode(inKode: char ;
```

```
        inIsSirkel : boolean) : TRoete;
```

```
var
```

```
    count, i : integer;
```

```
    tStoppe : StopSkikking;
```

```
begin
```

```
    Result := TRoete.Create(inKode, inIsSirkel);
```

Doeltreffendheidspunt: Lêer word nooit herlees nie – slegs Stop-skikking word gebruik. Geen ander openbare metode in Roete/Stop-klas geskep om enige deel van hierdie metode te doen nie.

skep Roete-objek korrek

0 – indien alternatiewe konstruktorskep en gebruik is


```

        skep nuwe Stop-skikking met voldoende lengte
count := 0;
setLength(tStoppe, 0);
setLength(tStoppe, 100);

for i := 1 to stopCount do      lus om korrek deur stoppe te gaan
begin
    if (alleStoppe[i].isDeelVanRoete(inKode)) then
        begin
            tStoppe[count] := alleStoppe[i];      kontroleer en voeg by indien stop deel is van roete
            inc(count);
        end;
    end;
end;

setLength(tStoppe, count);      verminder lengte van skikking korrek

                                stel skikking
Result.stelStoppe(tStoppe);
end;

                                korrekte openbaremetode-opskrif: enige naam, moet openbaar
                                wees, neem twee Roete-objekte in, stuur string korrek terug
// V7.1
function TStopBestuurder.werkStopPunteUit(r1, r2 : TRoete) : string;

// Punte toegeken volgens doelwitte bereik:
// 1 - 01: lus om deur roete 1-stoppe te gaan
// 1 - 02: toets of dit ook aan roete 2 behoort
// 1 - 03: doeltreffendheid - geen herlees van lêer, geen ander metode geskep nie
// 1 - 04: stringaaneenskakeling korrek gedoen
// 1 - 05: korrekte terugsending (0 indien daar terugsending is maar nie korrek nie)

var
    // vir alternatief 1
    count : integer;
    // vir alternatief 2
    i :integer;
begin
    Result := '';
    // alternatief 1: werk met die twee Roete-objekte

    count := 0;                  A1: O1 lus om korrek deur r1-stoppe te gaan

    while (r1.kryStopBy(count) <> nil) do
        begin
            if r1.kryStopBy(count).isDeelVanRoete(r2.kryRoeteKode()) then
                begin
                    Result := Result + r1.kryStopBy(count).toString() + #13#10;
                    A1: O2 toets om te sien of stop deel is van r2
                    A1: O4 Stringaaneenskakeling korrek gedoen
                end;
            inc(count);          A1: O3 geen herlees van lêer nie, geen nuwe openbare metode gebruik nie
        end;                    A1: O5 korrekte terugsending

    // alternatief 2: werk met die alleStoppe-skikking
    for i := 1 to stopCount do
        begin
            A2: O1 lus om deur r1-stoppe te gaan

```

```

    if (alleStoppe[i].isDeelVanRoete(r1.kryRoeteKode()))
        and
        (alleStoppe[i].isDeelVanRoete(r2.kryRoeteKode())) then
    begin
        Result := Result + alleStoppe[i].toString() + #13#10;
    end;
end;
end;
end.

```

A2: O2 toets om te sien of stop deel is van r2
A2: O4 Stringaaneenskakeling korrek gedoen
A2: O5 korrekte terugsending
A2: O3 geen herlees van lêer nie, geen nuwe openbare metode gebruik nie

VRAAG 6 EN 7.2: ToerGK-klas

// V6.1

program ToerGK;

```

{$APPTYPE CONSOLE}           // of skep vorm
{$R *.res}

```

uses

```

    System.SysUtils,
    uStop in 'uStop.pas',
    uRoete in 'uRoete.pas',
    uToerBestuurder in 'uToerBestuurder.pas';

```

var

```

    temp : string;
    sm : TStopBestuurder;
    roeteR, roeteY : TRoete;

```

begin

try

// V6.2

```
sm := TStopBestuurder.Create('data.txt');
```

// V6.3

```
roeteR := sm.kryRoeteMetKode('R', true);
roeteY := sm.kryRoeteMetKode('Y', false);

```

// V6.4 – druk albei roetes

```

WriteLn(roeteR.toString());
WriteLn(roeteY.toString());

```

// V7.2

```
WriteLn(sm.werkStopPunteUit(roeteR, roeteY));
```

```
ReadLn(temp);
```

except

on E: Exception do

```
    Writeln(E.ClassName, ': ', E.Message);
```

end;

end.

AFVOER**AFDELING A****VRAAG 1.1**

ToerisID	ToerisNaam	Epos	Hotel	DatumGeregistreer
8	Jorrie Potten	jpotten@seattletimes.com	Mount Grace Hotel	2016-11-03
10	Johnath Nixon	jnixon@seattletimes.com	Peninsula Hotel	2016-05-13
11	Davis Eginton	degintona@seattletimes.com	Peninsula Hotel	2018-07-30

VRAAG 1.2*GEEN AFVOER NIE***VRAAG 1.3**

EkskursieNaam	Duur
Sunrise Breakfast River Cruise	3
Township Excursion 1	2
National Art Museum Excursion 1	3

VRAAG 1.4

Hotel	AantalToeriste
Mount Grace Hotel	5
Peninsula Hotel	6
President Hotel	3
Village Lodge	5

VRAAG 1.5

ToerisNaam
Arron Haney
Cornall Prout
Darryl Poleykett
Davis Eginton
Eba Gillison
Eleen Yeomans
Irina Gouny
Johnath Nixon
Jorrie Potten
Kort McAndie
Marge Hengoed
Mic MacArd
Paul Buller
Rozalie Kebell
Sharl MacMenamy
Sharlene Bendall

VRAAG 1.6

EkskursieNaam	EkskursieKode <i>Die laaste twee syfers sal anders wees in elke geval omdat hulle ewekansig gegenereer word.</i>
Sunrise Breakfast River Cruise	Sun96
Sunset River Cruise	Sun56
Morning Safari	Mor32
Afternoon Safari	Aft31
Night Safari	Nig25
Township Excursion 1	Tow36
Township Excursion 2	Tow84
Township Excursion 3	Tow82
Township Excursion 4	Tow18
Township Excursion 5	Tow25
National Art Museum Excursion 1	Nat82
National Art Museum Excursion 1	Nat70
National Art Museum Excursion 1	Nat39
Splash Marine Park Morning Excursion	Spl48
Splash Marine Park Afternoon Excursion	Spl76
Splash Marine Park Day Excursion	Spl20
National Park Day Excursion	Nat19

VRAAG 1.7**GEEN AFVOER NIE****AFDELING B****FINALE AFVOER**

R - Sirkel

- > 1 kafee Waterfront
- > 2 snel St Monicas Cathedral
- > 3 skuiling Conference Centre
- > 4 snel CC
- > 5 skuiling Market Square
- > 6 snel Jewel Africa
- > 7 skuiling Taemane National Park
- > 8 skuiling Igugu Marine Park
- > 9 snel President Hotel
- > 10 ander St Johns Road
- > 11 snel Winchester Hotel
- > 12 skuiling Soccer Stadium
- > 1 kafee Waterfront

Y - Lineêr

- > 1 skuiling Market Square
- > 2 skuiling Clock Tower
- > 3 snel Idayimani Museum
- > 4 ander Grande Hotel
- > 5 skuiling SA Heritage Museum
- > 6 skuiling Apartheid Museum
- > 7 snel Fort of Idayimani
- > 6 skuiling Apartheid Museum
- > 5 skuiling SA Heritage Museum
- > 4 ander Grande Hotel
- > 3 snel Idayimani Museum
- > 2 skuiling Clock Tower
- > 1 skuiling Market Square
- skuiling Market Square

BYLAAG A: ALTERNATIEWE OPLOSSING**DELPHI: VRAAG 3.5 EN 5 : Roete-klas met statiese skikking**

```
unit uRoete;
```

```
interface
```

```
  uses uStop, SysUtils;
```

```
  // V3.1
```

```
  type StopSkikking = array[1..100] of TStop;
```

```
  type TRoete = class
```

```
    private
```

```
      roeteKode : char;
```

```
      isSirkel : boolean;
```

```
      stoppe : StopSkikking;
```

```
      numStopsFound : integer; // needed for Q3.5
```

```
  ...
```

```
  // Q3.3
```

```
  procedure TRoete.stelStoppe (inStoppe: StopSkikking);
```

```
  var
```

```
    i : integer;
```

```
  begin
```

```
    stoppe := inStoppe;
```

```
    numStopsFound := 0;
```

```
    for i := 1 to 100 do
```

```
    begin
```

```
      if stoppe[i] <> nil then
```

```
      begin
```

```
        Inc(numStopsFound);
```

```
      end
```

```
    end
```

```
  end;
```

```
  ...
```



Need this loop to work out how many stops there are in this array

```
  // V3.5
```

```
  function TRoete.kryStopBy(num: integer) : TStop;
```

```
  begin
```

```
    if (num >= 1) and (num <= numStopsFound) then
```

```
    begin
```

```
      Result := stoppe[num] A1 – stuur objek terug
```

```
    end
```

```
    else
```

```
    begin
```

```
      Result := nil A1 – stuur nul terug indien nie geldig nie
```

```
    end;
```

```
  end;
```

```

// V5
function TRoete.toString() : string;
var
  i : integer;
begin
  Result := roeteKode + ' - ';

  if Length(stoppe) = 0 then
  begin
    Result := Result + 'Ongeldig';
  end
  else
  begin
    if (isSirkel) then
    begin
      Result := Result + 'Sirkel';
    end
    else
    begin
      Result := Result + 'Lineêr';
    end;
  end;

  for i:= 1 to numStopsFound do
  begin
    Result := Result + #13#10 + '-> ' + IntToStr(i)
      + #9 + stoppe[i].toString();
  end;

  if (isSirkel) then
  begin
    Result := Result + #13#10 + '-> 1' + #9 + stoppe[1].toString();
  end
  else
  begin
    for i:= numStopsFound-1 downto 1 do
    begin
      Result := Result + #13#10 + '-> ' + IntToStr(i) + #9 +
        stoppe[i].toString();
    end;
  end;
end;
end.

```

Doeltreffendheidspunt:
 2 – geen roeteverwerking vir ongeldige stop nie
 1 – ongeldigheidstoets word gedoen, maar daar is onnodige verwerking/kontrolering van roetetype
 0 – geen ongeldigheidstoets word gedoen nie

voeg sirkel/lineêr korrek by

sluit stoppe in

Doeltreffendheidspunt – kode om die eerste deel van die skikking te druk (waar skikkings vir sirkel/lineêr gemeenskaplike formaat het), verskyn slegs een keer in die kode, d.w.s. geen deel daarvan verskyn op veelvuldige plekke nie

sluit eerste stop in indien sirkel

lus om stoppe in omgekeerde volgorde in te sluit van tweede laaste stop af

alle stoppe in korrekte formaat teruggestuur

VRAAG 4.4 EN 7.1: ToerBestuurder-klas met statiese skikking

```
unit uToerBestuurder;
```

```
interface
  uses uStop, uRoete, SysUtils;
  ...
  // V4.4
  function TStopBestuurder.kryRoeteMetKode(inKode: char ;
      inIsSirkel : boolean) : TRoete;
var
  count, i : integer;
  tStoppe : StopSkikking;
begin
  Result := TRoete.Create(inKode, inIsSirkel);
  count := 1;

  for i := 1 to Length(tStop) do
  begin
    tStoppe[i] := nil;
  end;

  for i := 1 to stopCount do
  begin
    if (alleStoppe[i].isDeelVanRoete(inKode)) then
    begin
      tStoppe[count] := alleStoppe[i];
      inc(count);
    end;
  end;

  Result.stelStoppe(tStoppe);
end;
```

skep nuwe Stop-skikking met voldoende lengte – toegeken volgens definisie van StopSkikking = array[1..100] in die Roete-definisie

korrekte metodeopskrif

skep Roete-objek korrek
0 – indien alternatiewe konstruktore geskep en gebruik word

Doeltreffendheidspunt: Lêer word nooit herlees nie – slegs Stop-skikking word gebruik. Geen ander openbare metode in Roete/Stop-klas geskep om enige deel van hierdie metode te doen nie.

om enige ongebruikte gleuwe op nul te stel
(kan ook na soektog wees)

lus om korrek deur stoppe te gaan

kontroleer en voeg by indien stop deel is van roete

stel skikking

korrekte openbaremetode-opskrif: enige naam, moet openbaar wees,
neem twee Roete-objekte in, stuur string korrek terug

```
// V7.1
function TStopBestuurder.werkStopPunteUit(r1, r2 : TRoete) : string;

// Punte toegeken volgens doelwitte bereik:
// 1 - 01: lus om deur roete 1-stoppe te gaan
// 1 - 02: toets of dit ook aan roete 2 behoort
// 1 - 03: doeltreffendheid - geen herlees van lêer, geen ander metode geskep nie
// 1 - 04: stringaaneenskakeling korrek gedoen
// 1 - 05: korrekte terugsending (0 indien daar terugsending is maar nie korrek nie)

var
  // vir alternatief 1
  count : integer;
  // vir alternatief 2
  i :integer;
begin
  Result := '';
  // alternatief 1: werk met die twee Roete-objekte

  count := 1;           A1: 01 lus om korrek deur r1-stoppe te gaan

  while (r1.kryStopBy(count) <> nil) do
    begin
      A1: 02 toets om te sien of stop deel is van r2
      if r1.kryStopBy(count).isDeelVanRoete(r2.kryRoeteKode()) then
        begin
          A1: 04 Stringaaneenskakeling korrek gedoen
          Result := Result + r1.kryStopBy(count).toString() + #13#10;
        end;
        inc(count);    A1: 03 geen herlees van lêer nie, geen nuwe openbare metode gebruik nie
      end;              A1: 05 korrekte terugsending

  // alternatief 2: werk met die alleStoppe-skikking
  for i := 1 to stopCount do
    begin
      A2: 01 lus om deur r1-stoppe te gaan
      if (alleStoppe[i].isDeelVanRoete(r1.kryRoeteKode()))
        and
          A2: 02 toets om te sien of stop deel is van r2
          (alleStoppe[i].isDeelVanRoete(r2.kryRoeteKode())) then
        begin
          A2: 04 Stringaaneenskakeling korrek gedoen
          Result := Result + alleStoppe[i].toString() + #13#10;
          A2: 05 korrekte terugsending
        end;
      end;
    end;
    A2: 03 geen herlees van lêer nie, geen nuwe openbare metode gebruik nie
  end.
end.
```