



NASIONALE SENIOR CERTIFIKAAAT-EKSAMEN
NOVEMBER 2021

INLIGTINGSTEGNOLOGIE: VRAESTEL I
NASIENRIGLYNE

Tyd: 3 uur

150 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**VRAAG 1****Vraag 1.1 (4)**

```
SELECT *  
FROM tblDomeine  
WHERE VPN = TRUE  
ORDER BY DomeinNaam
```

Vraag 1.2 (5)

```
SELECT *  
FROM tblDiensAgente  
WHERE Ondervinding BETWEEN 2 AND 5 AND  
Departement IN ("Maintenance" , "Admin")
```

Alternatief

```
SELECT *  
FROM tblDiensAgente  
WHERE Ondervinding BETWEEN 2 AND 5 AND  
(Departement = "Maintenance" or Departement = "Admin")
```

Vraag 1.3 (4)

```
SELECT COUNT(*) AS RSA_Maatskappye  
FROM tblDomeine  
WHERE DomeinNaam LIKE "*.co.za" aanvaar % vir mysql / javadb
```

Vraag 1.4 (7)

```
SELECT DomeinNaam, Pakket, Koste * 1.5 AS WinskoopPrys  
FROM tblDomeine, tblPakkette  
WHERE MONTH(DatumIngeteken) = MONTH(NOW() ) AND Pakket <> 'platinum' AND  
tblDomeine.PakketID = tblPakkette.PakketID
```

MySQL/JavaDB

```
SELECT DomeinNaam, Pakket, Koste * 1.5 AS WinskoopPrys  
FROM tblDomeine, tblPakkette  
WHERE MONTH(DatumIngeteken) = MONTH(Current_Time) AND Pakket <>  
'platinum' AND tblDomeine.PakketID = tblPakkette.PakketID
```

Vraag 1.5 (7)

```
SELECT Departement, AVG (Ondervinding) as GemOndervinding  
FROM tblDiensAgente  
GROUP BY Departement  
HAVING GemOndervinding > 6
```

Vraag 1.6 (6)

```
UPDATE tblDomeine  
SET DomeinNaam = LEFT(DomeinNaam, LEN(DomeinNaam) - 3) & ".co.ind"  
WHERE DomeinNaam LIKE "*.in"
```

Vraag 1.7 (8)

```
SELECT DomeinNaam, Voornaam, Van, tblAgent.AgentID, PrioriteitVlak,
DatumAangemeld
FROM tblDomeine, tblDiensAgente, tblKaartjies
WHERE tblDomeine.DomeinID = tblKaartjies.DomeinID AND
tblDiensAgente.AgentID = tblKaartjies.AgentID AND
DatumVoltooi IS NULL
ORDER BY tblKaartjies.AgentID, PrioriteitVlak aanvaar inner joins
```

Vraag 1.8 (9)

```
INSERT INTO tblDomeine (DomeinNaam, DatumIngeteken, VPN, PakketID)
SELECT INT (RND(DomeinID) * 5 + 1 ) & DomeinNaam, NOW() , TRUE, PakketID
(pas by die volgorde van insert fields)
FROM tblDomeine (korrekte insert met select)
WHERE DomeinNaam LIKE "*.ru"
```

MySQL

```
INSERT INTO tblDomeine (DomeinNaam, DatumIngeteken, VPN, PakketID)
SELECT CONCAT(floor(RAND(DomeinID) * 5 + 1 ), DomeinNaam ) ,
CURRENT_TIME , TRUE, PakketID(pas by die volgorde van insert fields)
FROM tblDomeine (korrekte insert met select)
WHERE DomeinNaam LIKE '.ru'
```

JavaDB

```
INSERT INTO tblDomeine (DomeinNaam, DatumIngeteken, VPN, PakketID)
SELECT SUBSTR( '12345' , INTEGER (RANDOM() * 5) + 1 , 1) || DomeinNaam ,
CURRENT_TIME , TRUE, PakketID(pas by die volgorde van insert fields)
FROM tblDomeine (korrekte insert met select)
WHERE DomeinNaam LIKE '%.ru'
```

JAVA-OPLOSSING**VRAAG 2 TEGNIKUS-KLAS**

//Vraag 2.1 - 3

```
public class Tegnikus { klasopskrif

    private String tegID; alle velde privaat
    private String naam; almal korrek getipeer met korrekte name
    private int ondervinding;
    private String rolSpesialiteit;
```

//Vraag 2.2 - 4

korrekte opskrif

```
public Tegnikus(String inTID, String inN, int inE, String inR)
    korrekte parametername en tipes
{
    tegID = inTID; velde op parameters gestel
    naam = inN;
    ondervinding = inE;
    rolSpesialiteit = inR;
}
```

//Vraag 2.3 - 2

korrekte opskrif en stuur tipe terug vir al vier krymetodes

```
public String kryTegID()
{
    return tegID;
}

public String kryNaam()
{
    return naam;
}

public int kryOndervinding()
{
    return ondervinding;
}

public String kryRolSpesialiteit()
{
    return rolSpesialiteit;
}
```

//Vraag 2.4 - 4

korrekte opskrif

```
public String naString()
{
    bevat alle velde
    veld in korrekte formaat
    korrek geformateerde string
    return naam + ", " + tegID + ", " + ondervinding + " jaar" [" +
roleSpeciality + "];
}
}
```

VRAAG 3 BEDIENER-KLAS

//Vraag 3.1 – 5

klasopskrif korrek

public class Bediener {

private String bedienerID; Stringeienskappe privaaf gemaak

private String ligging; korrek getipeer

private String rol; korrek benoem

private String fout;

Tegnikus-eienskap benoem en korrek getipeer

private Tegnikus toegekendeTeg;

//Vraag 3.2 – 4

Konstante verklaar met eindwaarde / konstante

korrek benoem

korrek getipeer

korrekte waardes toegeken

public static final String ROLTIPE_EMAIL = "Email";

public static final String ROLTIPE_FILE = "File";

public static final String ROLTIPE_PRINT = "Print";

public static final String ROLTIPE_CUSTOM = "Custom";

//Vraag 3.3 – 6

konstruktor korrek benoem

public Bediener (String inSID, String inLo, String inRo , String
inFa)

parameters korrek met tegnikus uitgesluit

{

bediener, ligging en fout korrek toegeken

bedienerID = inSID;

ligging = inLo;

fout = inFa;

if-stelling om rol-parameter teen Konstantes te kontroleer

Kontroleer vir kassensitiwiteit

genes – ken verstekwaarde "custom" korrek toe

if(inRo.equalsIgnoreCase(ROLTIPE_EMAIL))

{

rol = inRo;

}

else if (inRo.equalsIgnoreCase(ROLTIPE_FILE))

{

rol = inRo;

}

else if (inRo.equalsIgnoreCase(ROLTIPE_PRINT))

{

rol = inRo;

}

else

{

```
        rol = ROLTIPE_CUSTOM;
    }

}

//Vraag 3.4 - 2
metode-opskrifte en stuur korrek terug
public String kryBedienerID() {
    return bedienerID;
}

public String kryLigging() {
    return ligging;
}

public String kryRol() {
    return rol;
}

public String kryFout() {
    return fout;
}

//Vraag 3.5 - 2

metodeopskrif en stuur korrek terug
public Tegnikus kryToegekendeTeg() {
    return toegekendeTeg;
}

metode-opskrif korrek, aanvaar Tegnikus-parameter

public void stelToegekendeTeg(Tegnikus inTeg) {
    toegekendeTeg = inTeg;
}

//Vraag 3.6 - 6
metode-opskrif korrek
public String naString()
{

    String r = "";

    velde by string bygevoeg
    korrekte formaat
    r = r + "Bediener: " + bedienerID + "(Rol: " + rol + ")\n";
    r = r + "Fout: " + fout + "@" + ligging + "\n";
}
```

```

    kontroleer of daar 'n tegnikus toegeken is en korrek aangelas
    if(toegekendeTeg != null)
    {

        r = r + "Toegeken aan: " + toegekendeTeg.toString();
        korrek aangelas

    }
    else
    {
        r = r + "Toegeken aan: niemand toegeken nie";
    }

    stuur opgeboude string terug
    return r;
}
}

```

VRAAG 4, 6.1, 7.1 BEDIENERBESTUURDER-KLAS

```

//Vraag 4.1 - 1
    korrekte klasopskrif
public class BedienerBestuurder {

    //Vraag 4.2 - 4
    Albei eienskappe privaat
    Bediener-skikking verklaar met korrekte naam
    Skikkinggrootte op 50 gestel
    private Bediener bSkik[] = new Bediener[50];
    grootte korrek geïnisialiseer
    private int grootte = 0;

    //Vraag 4.3 - 9
    konstruktoropskrif korrek
    public BedienerBestuurder()
    {
        try
        {
            maak die lêer oop om te lees
            Scanner sc = new Scanner(new File("bedieners.txt"));

            lus deur al die reëls
            while(sc.hasNextLine())
            {
                lees die volgende reël uit die lêer
                String line = sc.nextLine();

                verdeel die reël in die verlangde dele
                String tokens[] = line.split("#");

                String sid = tokens[0];
                String ligging = tokens[1];
                String fout = tokens[2];
            }
        }
        catch (FileNotFoundException e) {}
    }
}

```

```

        String rol = tokens[3];

        skep bediener-objek

        Bediener s = new Bediener(sid, ligging, fout, rol);
        voeg bediener by skikking by
        bSkik[grootte] = s;
        inkrementeer grootte
        grootte++;

    }
    sc.close();
}
catch(FileNotFoundException e)
{

    System.out.println("Lêer ontbreek"); hanteer uitsondering
}
}

```

//Vraag 4.4 - 5

metode-opskrif korrek en stuur String terug

```

public String alleBedieners()
{
    String r = ""; string geïnisialiseer
    lus deur bediener-skikking
    for (int i = 0; i < grootte; i++) {
        las by string aan met ekstra blanko reël
        r = r + bSkik[i].naString() + "\n\n";
    }

    return r;
}

```

//Vraag 4.5 - 5

metode-opskrif korrek en stuur heelgetal terug

```

public int telBedieners(String fout, String roltipe)
{
    int telling = 0;
    lus deur skikking
    for (int i = 0; i < grootte; i++) {
        kontroleer of bedienerfout
        rol stem ooreen met parameters, ignoreer kas
        if(bSkik[i].kryFout().equalsIgnoreCase(fout) &&
        bSkik[i].kryRol().equalsIgnoreCase(roltipe))
        {
            telling = telling + 1; werk telling korrek by
        }
    }

    return telling;
}

```



```

//Vraag 6
//Vraag 6.1 - 11
    opskrif korrek
public void toegekendeTegnici()
{
    try
    {
        maak lêer oop
        Scanner sc = new Scanner(new File("tegnici.txt"));
        lus deur lêer
        while(sc.hasNextLine()) {

            kry reëls en verdeel
            String line = sc.nextLine();
            String tokens[] = line.split("#");

            String tid = tokens[0];
            String naam = tokens[1];
            int exp = Integer.parseInt(tokens[2]);
            String rs = tokens[3];
            skep tegnikus-objek
            Tegnikus t = new Tegnikus(tid, naam , exp , rs);

            lus deur al die bedieners

                gebruik terwyl lusstruktuur

                kontroleer dat die perk van 4 bedieners per tegnikus
                nie oorskry word nie

                int perk = 0;
            int k = 0;

            while (k < grootte & perk < 4) {
                kontroleer om te sien of die bedienerrol met die tegnikusrol
                ooreenstem en dat geen tegnici toegeken is nie
                if(bSkik[k].kryRol().equalsIgnoreCase(t.kryRolSpesialiteit()) &
                bSkik[k].kryToegekendeTeg() == null)
                {
                    ken tegnikus aan bediener toe deur metode te gebruik van
                    die bediener-klas
                    bSkik[k].stelToegekendeTeg(t);
                    vermeerder getal bedieners toegeken
                    perk++;
                }
                k++;
            }
        }
    }
    catch(FileNotFoundException e)
    {
        System.out.println("Lêer ontbreek " + e.kryBoodskap());
    }
}

```

```
//Vraag 7
//Vraag 7.1 - 17
```

```
private boolean vindBediener(String loc, String tID)
{
    lus deur bedieners
    for (int i = 0; i < grootte; i++)
    {
        kontroleer of tegID met bediener se tegid ooreenstem
        if (bSkik[i].kryToegekendeTeg() != null &&
bSkik[i].kryToegekendeTeg().kryTegID().equals(tID) &&
bSkik[i].kryLigging().equals(loc))
        {
            return true;
        }
    }
    return false;
}

public String drukKaart(String tegID)
{
    skep string vir die kaart
    String kaart = "";

    skep datumstempel
    korrekte formaat
    DateTimeFormatter formateerDatum =
DateTimeFormatter.ofPattern("YYYY/MM/dd HH:mm:ss");

    voeg datum by kaart by
    kaart = kaart + formateerDatum.format(LocalDateTime.now()) + "\n";

    skep en las kolomnommers aan
    for (int i = 1; i <= 15; i++)
    {
        kaart += "\t" + i;
    }
    kaart += "\n";

    skep lus vir ryletters
    for (char row = 'A'; row <= 'J'; row++)
    {
        kaart += row;
        skep lus vir kolomme
        for (int col = 1; col <= 15; col++)
        {
            kontroleer of 'n bedienerligging gevind word
            String loc = (row + "" + col);
            if (vindBediener(loc, tegID))
            {
                kaart += "\tX"; las x aan
            } else
            {
                kaart += "\t*"; las * aan
            }
        }
    }
}
```

```

    }
  }
  kaart += "\n";
}

try
{
  skep lêer om kaartdata te skryf en te stoor met tegID as lêernaam
  PrintWriter out = new PrintWriter(new FileWriter(tegID + ".txt"));

  skryf kaartdata na lêer
  out.println(kaart);
  maak lêer toe
  out.close();

} catch (Exception e)
{
  System.out.println("Kon nie na lêer skryf nie");
}

  stuur kaart terug
  return kaart;
}
}

```

VRAAG 5, 6.2, 7.2 BEDIENERGK-KLAS

```

//Vraag 5
//Vraag 5.1 - 1
  toepassingsklas geskep met hoofmetode
public class BedienerGK {

  public static void main (String args[])
  {
    //Vraag 5.2 - 1
    BedienerBestuurder-objek op toepaslike plek in die kode geskep
    BedienerBestuurder sm = new BedienerBestuurder();

    //Vraag 5.3 - 1
    alleBedieners geroep en korrek vertoon
    System.out.println(sm.alleBedieners());

    //Vraag 5.4 - 3
    telBedieners geroep
    gebruik Konstante se waarde en korrek geroep
    System.out.println("Getal bedieners met 'n temperatuurfout en
doelrol" + sm.telBedieners("Temp", Bediener.ROLTIPE_CUSTOM));

    //Vraag 6.2 - 2
    toegekendeTegnici korrek geroep en hervertoon alleBedieners
    sm.toegekendeTegnici();
    System.out.println(sm.alleBedieners());

    //Vraag 7.2 - 2
    drukKaart-metode geroep
    Korrekte tegID gebruik
    System.out.println(sm.drukKaart("T-D1"));
  }
}

```

DELPHI-OPLOSSING**VRAAG 2 TEGNIKUS-KLAS**

```
unit uTegnikus;

interface
uses SysUtils;
//Vraag 2.1 - 3
  klasopskrif
type TTegnikus = class
  private
    alle velde privaat
    tegID, naam, rolSpesialiteit : string;
    korrek benoem met korrekte tipe
    ondervinding : integer;
  public
    constructor Create( inTID : string; inN : string; inE : integer,
inR : string);
    function kryTegID : string;
    function kryNaam : string;
    function kryOndervinding : integer;
    function kryRolSpesialiteit : integer;
    function naString : string;

end;

implementation

{ TTegnikus }
//Vraag 2.2 - 4
  opskrif korrek
constructor TTegnikus.Create(inTID, inN: string; inE: integer ,
inR : string);
  korrekte parametername en tipes
begin
  tegID := inTID;    velde op parameters gestel
  naam := inN;
  ondervinding := inE;
  rolSpesialiteit := inR;
end;

//Vraag 2.3 - 2
korrekte opskrif en stuur tipe terug vir al vier krymetodes
function TTegnikus.kryOndervinding: integer;
begin
  Result:= ondervinding;
end;

function TTegnikus.kryNaam: string;
begin
  Result:= naam;
end;
```

```

function TTegnikus.kryRolSpesialiteit: string;
begin
    Result:= rolSpesialiteit;
end;
function TTegnikus.kryTegID: string;
begin
    Result := tegID;
end;

```

//Vraag 2.4 - 4

korrekte opskrif

```

function TTegnikus.naString: string;
begin
    bevat alle velde
    veld in korrekte formaat
    korrek teruggestuur
    Result:= naam + ', ' + tegID + ', ' + IntToStr(ondervinding) + '
jaar';
end;

end.

```

VRAAG 3 BEDIENER-KLAS

```

unit uBediener;

```

```

interface
uses SysUtils, uTegnikus;
//Vraag 3.1 - 5
klasopskrif korrek
type TBediener = class

private
    Stringeienskappe privaat gemaak
    korrek getipeer
    korrek benoem
    bedienerID, ligging, rol, fout: string;
    Tegnikus-eienskap benoem en korrek getipeer
    toegekendeTeg : TTegnikus;

public
//Vraag 3.2 - 4
const   Konstante verklaar
    ROLTIPE_EMAIL = 'Email';   korrek benoem
    ROLTIPE_FILE = 'File';   korrek getipeer
    ROLTIPE_PRINT = 'Print';   waardes korrek toegeken
    ROLTIPE_CUSTOM = 'Custom';

    constructor Create (inSID, inLo, inFa, inRo : string);
    function kryBedienerID() : string;
    function kryLigging() : string;
    function kryRol() : string;
    function kryFout() : string;

```

```
procedure stelToegekendeTeg(inTeg : TTegnikus);
function kryToegekendeTeg() : TTegnikus;
function naString() : string;

end;

implementation

{ TBediener }
//Vraag 3.3 - 6
opskrif korrek
parameters korrek met tegnikus uitgesluit
constructor TBediener.Create(inSID, inLo, inFa, inRo: string);
//korrekte parameters met Tegnikus uitgesluit
begin

    bediener, ligging en fout korrek toegeken
    bedienerID:= inSID;
    ligging:= inLo;
    fout:= inFa;
    if-stelling om rol-parameter teen Konstantes te kontroleer
    Kontroleer vir kassensitiwiteit
    genes - ken verstekwaarde "custom" korrek toe
    if (CompareText(inRo,ROLTIPE_EMAIL) = 0) or
        (CompareText(inRo,ROLTIPE_FILE) = 0) or
        (CompareText(inRo,ROLTIPE_PRINT) = 0) then
    begin
        rol := inRo;
    end
    else
    begin
        rol := ROLTIPE_CUSTOM
    end;

end;

end;

//Vraag 3.4 - 2
metode-opskrifte korrek en stuur korrek terug
function TBediener.kryFout: string;
begin
    Result:= fout;
end;

function TBediener.kryLigging: string;
begin
    Result:= ligging;
end;

function TBediener.kryRol: string;
begin
    Result:= rol;
end;
```

```
function TBediener.kryBedienerID: string;
begin
    Result:= bedienerID;
end;

//Vraag 3.5 - 2
    metode-opskrif en stuur korrek terug
function TBediener.kryToegekendeTeg: TTegnikus;
begin
    Result:= toegekendeTeg; //stuur tegnikus-tipe terug
end;

    metode-opskrif korrek en toekenning korrek
procedure TBediener.stelToegekendeTeg(inTeg: TTegnikus);
begin
    toegekendeTeg := inTeg; //ken korrek toe
end;

//Vraag 3.6 - 6
    metode-opskrif korrek
function TBediener.naString: string;
begin
    las in Result aan
    veld by Result bygevoeg
    korrekte formaat
    Result:= 'Bediener: ' + bedienerID + '(Rol: ' + rol + ')' +
#13#10;
    Result:= Result + 'Fout: ' + fout + ' @ ' + ligging + #13#10;

    kontroleer of daar 'n tegnikus is en las naString of "niemand
toegeken nie" aan
    if(toegekendeTeg <> nil) then
        begin
            Result:= Result + 'Toegeken aan: ' + toegekendeTeg.naString();
        end
    else
        begin
            Result:= Result + 'Toegeken aan: niemand toegeken nie';
        end;

    Result korrek opgebou
end;

end.
```

VRAAG 4, 6.1, 7.1 BEDIENERBESTUURDER-KLAS

```

unit uBedienerBestuurder;

interface
  uses SysUtils, uTegnikus, uBediener;
  //Vraag 4.1 - 1
  opskrif korrek
type TBedienerBestuurder = class
  private
    //Vraag 4.2 - 4
    Beide eienskappe privaat
    Skikking van tipe bedieners met korrekte naam
    Skikkinggrootte op 50 gestel
    bSkik : array[1..50] of TBediener;
    grootte geskep
    grootte : integer;
  public
    constructor Create;
    function alleBedieners : string;
    function telBedieners(fout, rol : string) : integer;
    procedure toegekendeTegnici();
    function vindBediener(loc , tid : string) : Boolean;
    function drukKaart(tegID : string) : string;
end;

implementation

{ TBedienerBestuurder }
//Vraag 4.3 - 9
konstruktoropskrif korrek
constructor TBedienerBestuurder.Create;
var
  infile : textfile;
  line, bedienerID, ligging, fout, rol : string;
begin
  if FileExists('bedieners.txt') <> true then hanteer uitsondering
  begin
    WriteLn('Lêer ontbreek');

  end
  else
  begin
    maak lêer oop om te lees
    AssignFile(infile, 'bedieners.txt');
    Reset(infile);

    grootte:=0;
    lus deur al die reëls

    while NOT EOF(infile) do
      begin
        lees die volgende reël uit die lêer
        ReadLN(infile, line);

```



```

    inkrementeer grootte
    Inc(grootte);
    verdeel die reël in die verlangde dele
    bedienerID := Copy(line, 1, Pos('#', line) - 1);
    Delete(line, 1, Pos('#', line));

    ligging:= Copy(line, 1, Pos('#', line) - 1);
    Delete(line, 1, Pos('#', line));

    fout := Copy(line, 1, Pos('#', line) - 1);
    Delete(line, 1, Pos('#', line));

    rol:= line;
    skep Bediener-objek
    voeg by skikking by
    bSkik[grootte] := TBediener.Create(bedienerID, ligging, fout,
rol);
    end;
    end;
end;

```

//Vraag 4.4 - 5

```

metode-opskrif korrek en stuur string terug
function TBedienerBestuurder.alleBedieners: string;
var
    i : integer ;
    output : string;
begin
    output := ''; string teruggestuur geïnisialiseer
    lus deur skikking
    for i := 1 to grootte do
    begin
        las by string aan met ekstra blanko reël
        output := output + bSkik[i].naString + #13#10 + #13#10;

    end;
    Result:=output;
end;

```

//Vraag 4.5 - 5

```

opskrif korrek, stuur heelgetal terug
function TBedienerBestuurder.telBedieners(fout, rol: string):
integer;
var
    telling : integer;
    i : integer;
begin
    telling:=0;
    lus deur skikking
    for i := 1 to grootte do
    begin
        kontroleer of bedienerfout
        rol stem ooreen met parameters ignoreer kas
        if ( CompareText(fout, bSkik[i].kryFout) = 0 ) AND

```

```

    (CompareText(rol, bSkik[i].kryRol) = 0 ) then
    begin
        telling:= telling + 1;    werk telling korrek by
    end;
end;
Result:=telling;
end;

```

//Vraag 6

//Vraag 6.1 - 11

opskrif korrek

```

procedure TBedienerBestuurder.toegekekendeTegnici;

```

```

var

```

```

    infile : textfile;

```

```

    line, tegID, naam, rolSpesialiteit: string;

```

```

    ondervinding : integer;

```

```

    teg : TTegnikus;

```

```

    assigned: TArray<string>;

```

```

    j,k,perk: integer;

```

```

begin

```

```

    if FileExists('bedieners.txt') <> true then

```

```

    begin

```

```

        WriteLn('Lêer ontbreek');

```

```

    end

```

```

else

```

```

    begin

```

maak lêer oop

```

    AssignFile(infile, 'tegnici.txt');

```

```

    Reset(infile);

```

lus deur lêer

```

    while NOT EOF(infile) do

```

```

    begin

```

kry reëls en verdeel

```

        ReadLN(infile, line);

```

```

        tegID := Copy(line, 1, Pos('#', line) - 1);

```

```

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

```

```

        naam:= Copy(line, 1, Pos('#', line) - 1);

```

```

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

```

```

        ondervinding := StrToInt(Copy(line, 1, Pos('#', line) - 1));

```

```

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

```

```

        rolSpesialiteit:= line;

```

skep Tegnikus-objek

```

        teg := TTegnikus.Create(tegID, naam, ondervinding,

```

```

        rolSpesialiteit);

```

lus deur al die bedieners

kontroleer dat die perk van 4 bedieners per tegnikus nie oorskry word nie

gebruik terwyl lusstruktuur

```

    while (perk < 4) and (k <= grootte) do
    begin
        kontroleer om te sien of die bedienerrol met die tegnikusrol
ooreenstem en dat geen tegnici toegeken is nie
        if (CompareText(bSkik[k].kryRol(), teg.kryRolSpesialiteit())
= 0) and (bSkik[k].kryToegekendeTeg() = nil) then
            begin
                ken tegnikus aan bediener toe deur metode te gebruik deur
bediener-klas te gebruik
                bSkik[k].stelToegekendeTeg(teg);
                    vermeerder getal bedieners toegeken
                perk := perk + 1;
            end;
            Inc(k);
        end;
    end;
end;
//Vraag 7

```

//Vraag 7.1 – 17

```

function TBedienerBestuurder.findBediener(loc , tid : string): Boolean;
var
    i : integer;
begin
    Result:=false;
    lus deur bedieners
    for i := 1 to grootte do
        begin
            kontroleer of tegid met bediener se tegid ooreenstem
            if (bSkik[i].kryToegekendeTeg() <> nil) and
(bSkik[i].kryToegekendeTeg().kryTegID() = tid) and
(bSkik[i].kryLigging() = loc) then
                begin
                    Result:=true;
                end;
            end;
        end;
    end;
end;

```

```

function TBedienerBestuurder.drukKaart(tegID: string): string;
var
    f : TextFile;
    kaart ,loc: string;
    i, col:integer;
    row : char;

begin
    skep 'n string vir die kaart
    kaart:= '';
    skep datumstempel
    korrekte formaat
    by kaart bygevoeg
    kaart:= kaart + (FormatDateTime('YYYY/MM/dd HH:mm:ss', Now)) + #13#10;
;

```

skep en las kolomnommers aan

```

for i := 1 to 15 do
begin
    kaart:= kaart + #9 + IntToStr(i);
end;
kaart:= kaart + #13#10;
skep lus vir ryletters
for row := 'A' to 'J' do
begin
    kaart:= kaart + row;
    skep lus vir kolomme
    for col := 1 to 15 do
    begin
        kontroleer of 'n bedienerligging gevind word
        loc:= row + ' ' + IntToStr(col);
        if vindBediener(loc, tegID) then
        begin
            kaart:= kaart + ' ' + #9 + 'X'; las x aan
        end
        else
        begin
            kaart:= kaart + ' ' + #9 + '*'; las * aan
        end;
    end;
    kaart:= kaart + #13#10;
end;
skep lêer om kaartdata te skryf en te stoor met tegID as lêernaam
AssignFile(f, tegID + '.txt');
ReWrite(f);
skryf kaartdata na lêer
Writeln(f, kaart);
maak lêer toe
CloseFile(f);
stuur kaart terug
Result:= kaart;
end;

end.

```

VRAAG 5, 6.2, 7.2**BEDIENERGK-KLAS****//Vraag 5.1 - 1****toepassingsklas geskep**

```

program BedienerGK;
{$APPTYPE CONSOLE}
{$R *.res}
uses
    System.SysUtils,
    uTegnikus in 'uTegnikus.pas',
    uBediener in 'uBediener.pas',
    uBedienerBestuurder in 'uBedienerBestuurder.pas';

```

```
var
    sm : TBedienerBestuurder;

begin
    try
        { TODO -oUser -cConsole Main : Insert code here }

        //Vraag 5.2 - 1
        skep BedienerBestuurder-objek
        sm:= TBedienerBestuurder.Create();
        //Vraag 5.3 - 1
        alleBedieners-metode geroep en korrek vertoon
        WriteLn(sm.alleBedieners());

        //Vraag 5.4 - 3
        //telBediener-metode geroep deur Konstante se waarde te gebruik en
        korrek vertoon
        telBediener-metode geroep
        gebruik Konstante se waarde en korrek geroep
        WriteLn(sm.telBedieners('Temp' , TBediener.ROLTIPE_CUSTOM)) ;

        //Vraag 6.2 - 2
        toegekendeTegnici korrek geroep en hervertoon alleBedieners
        sm.toegekendeTegnici();

        WriteLn(sm.alleBedieners);
        //Vraag 7.2 - 2
        drukKaart korrek geroep
        korrekte tegID gebruik
        WriteLn(sm.drukKaart('T-D1'));
        ReadLn;
    except
        on E: Exception do
            Writeln(E.ClassName, ': ', E.Message);
    end;
end.
```