



INTERNASIONALE SEKONDÊRE SERTIFIKAAT-EKSAMEN
NOVEMBER 2024

REKENAARWETENSKAP: 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 GESTRUKTUREERDE NAVRAAGTAAL**VRAAG 1**

1.1

```
SELECT *
FROM tblVerskaffers
WHERE Aanslag > 3
```

1.2

```
SELECT RoomysNaam, Beskrywing, Grootte, Afslag
FROM tblRoomyse
WHERE Beskrywing LIKE "*berry*"
AND Afslag = Yes
```

MySQL en JavaDB:

```
SELECT RoomysNaam, Beskrywing, Grootte, Afslag
FROM tblRoomyse
WHERE Beskrywing LIKE '%berry%'
AND Afslag = True
```

1.3

```
SELECT BestelID, RoomysID, Hoeveelheid, Prys
FROM tblItemsBestel
WHERE AflewerDatum BETWEEN #2024/09/13# AND #2024/09/20#
AND BestelID MOD 2 <> 0
```

Alternatief:

```
SELECT BestelID, RoomysID, Hoeveelheid, Prys
FROM tblItemsBestel
WHERE AflewerDatum >= #2024/09/13#
AND AflewerDatum <= #2024/09/20#
AND BestelID MOD 2 <> 0
```

MySQL:

```
SELECT BestelID, RoomysID, Hoeveelheid, Prys
FROM tblItemsBestel
WHERE AflewerDatum BETWEEN '2024-09-13' AND '2024-09-20'
AND MOD(BestelID, 2) <> 0
```

JavaDB:

```
SELECT TBLBESTELLINGS.BESTELID, ROOMYSID, HOEVEELHEID, PRYS
FROM TBLITEMSBESTEL, TBLBESTELLINGS
WHERE TBLITEMSBESTEL.BESTELID = TBLBESTELLINGS.BESTELID
AND BESTELDATUM BETWEEN '2024-09-13' AND '2024-09-20'
AND NOT MOD(TBLBESTELLINGS.BESTELID,2) = 0;
```

1.4

```
SELECT WerknemerNaam, COUNT(*) AS BestellingsGeplaas
FROM tblBestellings
WHERE VerskafferID = 10
GROUP BY WerknemerNaam
ORDER BY COUNT(*) DESC
```

Alternatief:

```
SELECT WerknemerNaam, COUNT(*) AS BestellingsGeplaas
FROM tblBestellings
WHERE VerskafferID = 10
GROUP BY WerknemerNaam
ORDER BY 2 DESC
```

MySQL en JavaDB:

```
SELECT WerknemerNaam, COUNT(*) AS BestellingsGeplaas
FROM tblBestellings
WHERE VerskafferID = 10
GROUP BY WerknemerNaam
ORDER BY BestellingsGeplaas DESC
```

1.5

```
SELECT RoomysID, SUM(Hoeveelheid)
FROM tblItemsBestel
GROUP BY RoomysID
HAVING SUM(Hoeveelheid) > 250
```

1.6

```
SELECT RoomysNaam, BestelID, ROUND (Prys * (1 + Winsgrens)
,2) AS VerkoopPrys
FROM tblRoomyse, tblItemsBestel
WHERE tblRoomyse.RoomysID = tblItemsBestel.RoomysID
AND Prys * (1 + Winsgrens) > 16
AND LEN (RoomysNaam) <= 20
ORDER BY RoomysNaam, tblBestellings.BestelID
```

Alternatief:

```
SELECT RoomysNaam, tblBestellings.BestelID, ROUND (Prys *
(1 + Winsgrens) ,2) AS VerkoopPrys
FROM tblRoomyse, tblItemsBestel
WHERE tblRoomyse.RoomysID = tblItemsBestel.RoomysID
AND Prys * (1 + Winsgrens) > 16
AND LEN (RoomysNaam) <= 20
ORDER BY 1, 2
```

MySQL:

```
SELECT RoomysNaam, tblBestellings.BestelID, ROUND (Prys *
(1 + Winsgrens) ,2) AS VerkoopPrys
FROM tblRoomyse, tblItemsBestel
WHERE tblRoomyse.RoomysID = tblItemsBestel.RoomysID
AND Prys * (1 + Winsgrens) > 16
AND LENGTH(RoomysNaam) <= 20
ORDER BY RoomysNaam, tblBestellings.BestelID
```

JavaDB:

```
SELECT ROOMYSNAAM, BESTELID,
FLOOR((PRYS*(1+WINSGRENS))*100+0.5)/100 AS VERKOOPPRYS
FROM TBLROOMYSE, TBLITEMSBESTEL
WHERE TBLROOMYSE.ROOMYSID = TBLITEMSBESTEL.ROOMYSID
AND PRYS*(1+WINSGRENS)>16
AND LENGTH(ROOMYSNAAM)<=20
ORDER BY RoomysNaam, BestelID;
```

1.7

```
INSERT INTO tblItemsBestel (BestelID, RoomysID, Prys,
Hoeveelheid)
SELECT 20, RoomysID, Prys, INT (RND(RoomysID) * 51 + 125)
vir RoomysID- en Prys-veld
FROM tblItemsBestel
WHERE BestelID = 12
```

MySQL:

```
INSERT INTO tblItemsBestel (BestelID, RoomysID, Prys,
Hoeveelheid)
SELECT 20, RoomysID, Prys, FLOOR(RAND()* 51 + 125)
FROM tblItemsBestel
WHERE BestelID = 12
```

JavaDB:

```
INSERT INTO TBLITEMSBESTEL
(BESTELID,ROOMYSID,PRYS,HOEVEELHEID)
(SELECT 20,ROOMYSID, PRYS, INT(RANDOM()*51+125)
FROM TBLITEMSBESTEL
WHERE BESTELID=12);
```

1.8

```
DELETE
FROM tblRoomyse
WHERE RoomysID IN (1, 2, 6, 10)
```

Alternatief:

```
DELETE
FROM tblRoomyse
WHERE RoomysID = 1
OR RoomysID = 2
OR RoomysID = 6
OR RoomysID = 10
```

AFDELING B OBJEKGEORIËNTEERDE PROGRAMMERING**VRAAG 2, VRAAG 6.1**

```
//Vraag 2.1 - 3 punte
//klasopskrif
//alle velde privaat
//alle velde korrek getipeer met korrekte name
public class Roomys
{
    private String naam;
    private int kategorieNom;
    private double winsgrens;
    private int hoeveelheid;
    private LocalDate verval;

    //Vraag 2.2 - 4 punte
    //korrekte metodenamaam
    //korrekte parametername en -tipes
    //velde as parameters gestel, trek 1 af per fout, maks van 2
    //aftrekkings
    public Roomys (String inNaam, int inKategorieNom,
                  double inWinsgrens, int inHoeveelheid, LocalDate
                  inVerval)
    {
        naam = inNaam;
        kategorieNom = inKategorieNom;
        winsgrens = inWinsgrens;
        hoeveelheid = inHoeveelheid;
        verval = inVerval;
    }

    //Vraag 2.3 - 2 punte
    //korrekte opskrifte vir toegangers
    //korrekte terugsendings vir toegangers
    public String getNaam()
    {
        return naam;
    }

    public LocalDate getVerval()
    {
        return verval;
    }
}
```

```
//Vraag 2.4 - 5 punte
//korrekte opskrif
//korrekte koste per kategorie - 2 punte
//korrekte berekening
//stuur berekening terug
public double getVerkoopPrys()
{
    double katKoste = 10.15;
    switch (kategorieNom)
    {
        case 2 -> katKoste = 11.09;
        case 3 -> katKoste = 11.26;
        case 4 -> katKoste = 11.65;
    }
    return (katKoste + katKoste * winsgrens / 100);
}
```

```
//Vraag 2.5 - 5 punte
//korrekte opskrif
//korrekte teks vir velde
//bevat alle velde
//gebruik van getVerkoopPrys()
//stuur geformateerde string met nuwe reëls terug
public String toString()
{
    return "Roomysnaam: " + naam + "\n" +
           "Hoeveelheid: " + hoeveelheid + "\n" +
           "Vervaldatum: " + verval + "\n" +
           "Verkoopsprys: " + getVerkoopPrys() + "\n";
}
```

```
//Vraag 6.1 - 4 punte
//metodeopskrif korrek
//vergelyk roomys se vervaldatum met huidige datum
//bepaal regte Boole-waarde
//stuur Boole-waarde terug
public boolean kontroleerVerval()
{
    if(verval.isBefore(LocalDate.parse("2024-10-18")))
    //if(verval.isBefore(LocalDate.now()))
    {
        return true;
    }
    return false;
}
}
```

VRAAG 3

```
//Vraag 3.1 - 4 punte
//klasopskrif
//klas erf van Roomys-klas
//alle velde privaat
//alle velde korrek getipeer met korrekte name
public class EkstraRoomys extends Roomys
{
    private int ekstras;
    private String allergene;

    //Vraag 3.2 - 2 punte
    //konstante verklaar met final
    //konstante en korrek benoem en getipeer met korrekte waarde
    public final double PRYSPEREKSTRA = 12.00;

    //Vraag 3.3 - 5 punte
    //korrekte metodenaam
    //korrekte parametername en -tipes
    //supermetode geroep
    //korrekte parameters vir supermetode
    //velde as parameters gestel
    public EkstraRoomys (String inNaam, int inKategorieNom,
                        double inWinsgrens, int inHoeveelheid,
                        LocalDate inVerval, int inEkstras,
                        String inAllergene)
    {
        super (inNaam, inKategorieNom, inWinsgrens, inHoeveelheid,
              inVerval);
        ekstras = inEkstras;
        allergene = inAllergene;
    }

    //Vraag 3.4 - 1 punt
    //korrekte opskrif en terugstuurstelling
    public String getAllergene()
    {
        return allergene;
    }
}
```

```
//Vraag 3.5 - 4 punte
//korrekte opskrif om getVerkoopPrys-metode te oorheers,
//@Override-vertalerdirektief is opsioneel
//gebruik van die getVerkoopPrys-metode uit Roomys-klas
//gebruik van klaskonstante
//terugsending van nuwe prys
@Override
public double getVerkoopPrys()
{
    return super.getVerkoopPrys() + ekstras * PRYSPEREKSTRA;
}

//Vraag 3.6 - 4 punte
//korrekte opskrif om toString-metode te oorheers,
//@Override-vertalerdirektief is opsioneel
//gebruik van die toString-metode uit Roomys-klas
//bevat alle velde
//stuur geformateerde string terug
@Override
public String toString()
{
    return super.toString() +
        "Ekstras: " + ekstras + "\n" +
        "Allergene: " + allergene + "\n";
}
}
```

VRAAG 4, VRAAG 6.2, VRAAG 7.1

```

//Vraag 4.1 - 1 punt
//klas korrek geskep
public class RoomysBestuurder
{
    //Vraag 4.2 - 4 punte
    //beide eienskappe privaat
    //Roomys-skikking verklaar met korrekte naam
    //skikkinggrootte op 100 gestel
    //grootte-eienskap geskep
    private Roomys [] rSkik = new Roomys[100];
    private int grootte = 0;

    //Vraag 4.3 - 13 punte
    //konstruktoropskrif korrek
    //try catch korrek geïmplementeer om uitsondering te hanteer
    //maak lêer oop om te lees
    //lus deur alle reëls
    //lees volgende reël uit lêer
    //verdeel reël in verlangde dele
    //parseer datum na LocalDate
    //indien reël bykomende data vir EkstraRoomys bevat
    //onttrek bykomende data vir EkstraRoomys
    //skep EkstraRoomys-objek wanneer nodig
    //skep Roomys-objek wanneer nodig
    //voeg geskepte objek by skikking by
    //inkrementeer grootte
    public RoomysBestuurder()
    {
        try
        {
            Scanner is = new Scanner(
                new FileInputStream("Roomyse.txt"));
            while (is.hasNextLine())
            {
                String l = is.nextLine();
                Scanner lr = new Scanner(l).useDelimiter(";");
                String naam = lr.next();
                int kategorieNom = lr.nextInt();
                double winsgrens = lr.nextDouble();
                int hoeveelheid = lr.nextInt();
                LocalDate verval = LocalDate.parse (lr.next(),
                    DateTimeFormatter.ofPattern("yyyy-MM-dd"));
                if (lr.hasNext())
                {
                    int ekstras = lr.nextInt();
                    String allergene = lr.next();
                    rSkik[grootte] = new EkstraRoomys (naam,
                        kategorieNom, winsgrens, hoeveelheid, verval, ekstras,

```

```

        allergene) ;
        }
        else
        {
            rSkik[grootte] = new Roomys(naam, kategorieNom,
                winsgrens, hoeveelheid, verval) ;
        }
        grootte++;
        lr.close();
    }
    is.close();
}
catch (FileNotFoundException e)
{
    System.out.println("Error, file not found");
}
}

//Vraag 4.4 - 5 punte
//metodeopskrif korrek
//string geïnisialiseer
//lus deur Roomys-skikking
//las by string aan
//stuur string terug
public String toString()
{
    String besonderhede = "";
    for (int i = 0; i < grootte; i++)
    {
        besonderhede += rSkik[i].toString() + "\n";
    }
    return besonderhede;
}

```

```

//Vraag 4.5 - 8 punte
//metodeopskrif korrek
//skep en inisialiseer 'n teller en 'n totaalveranderlike
//lus deur Roomys-skikking
//if binne lus om te kontroleer of objek EkstraRoomys-objek is
//tipeomskakeling van EkstraRoomys-objek
//roep getAllergene-metode en kontroleer of nie een nie
//voeg objek se prys by totaal en inkrementeer teller
//stuur korrekte berekeningtotaal / teller terug
public double getGemiddeldeEkstra()
{
    int s = 0;
    double somPrys = 0;
    for (int i = 0; i < grootte; i++)
    {
        if (rSkik[i] instanceof EkstraRoomys)
        {
            if (!((EkstraRoomys) rSkik[i]).getAllergene().
                equalsIgnoreCase("None"))
            {
                somPrys += rSkik[i].getVerkoopPrys();
                s++;
            }
        }
    }
    return Math.round(somPrys / s);
}

```

```
//Vraag 6.2 - 8 punte
//metodeopskrif korrek
//try catch korrek geïmplementeer om uitsondering te hanteer
//maak lêer oop om te skryf
//lus deur Roomys-skikking
//if binne lus
//kontroleerVerval-metode geroep op objek in toestand van if
//skryf besonderhede van roomys na tekslêer binne if
//maak verbinding met tekslêer toe om skryfwerk te stoor
public void vervaldeRoomyse()
{
    try
    {
        PrintWriter pw = new PrintWriter(new
        FileOutputStream("VervaldeRoomyse.txt"));
        for (int i = 0; i < grootte; i++)
        {
            if (rSkik[i].kontroleerVerval())
            {
                pw.println(rSkik[i].toString());
                pw.println();
            }
        }
        pw.close();
    } catch (Exception e)
    {
        System.out.println("Error writing to file");
    }
}
```

```

//Vraag 7.1 - 11 punte

//metodeopskrif korrek

//METODE 1 gebruik tydelike skikking:
//skep 'n tydelike skikking
//stel 'n teller op 0
//lus deur Roomys-skikking
//if binne lus met kontrolering nie King Cone
    deur getNaam te gebruik: 2 punte
//kopieer objek na tydelike skikking
//inkrementeer teller
//verander grootte na teller
//stel skikking gelyk aan tydelike skikking: 2 punte

//METODE 2 gebruik slegs bestaande skikking:
//lus deur Roomys-skikking
//while binne lus met kontrolering van King Kone deur getNaam
    te gebruik om te verseker geskuifde objek is nie King Kone
    nie: 2 punte
//binneste lus begin by i en gaan tot grootte - 1: 2 punte
//objekte 1 na links gekopieer: 2 punte
//vee laaste objek uit: 2 punte
//verminder grootte met 1

public void verwyderRoomyse()
{

    //METODE 1:
    Roomys [] rSkikTemp = new Roomys[100];
    int j = 0;
    for (int i = 0; i < grootte; i++)
    {
        if(!rSkik[i].getNaam().contains("King Kone"))
        {
            rSkikTemp[j] = rSkik[i];
            j++;
        }
    }
    grootte = j;
    rSkik = rSkikTemp;
}

```

```
//METODE 2:  
for (int i = 0; i < grootte; i++)  
{  
    while(rSkik[i].getNaam().contains("King Kone"))  
    {  
        for (int j = i; j < grootte - 1; j++)  
        {  
            rSkik[j] = rSkik[j + 1];  
        }  
        rSkik[grootte - 1] = null;  
        grootte--;  
    }  
}  
}
```

VRAAG 5, VRAAG 6.3, VRAAG 7.3

```
//Vraag 5.1 - 1 punt
//klas met hoofmetode geskep
public class RoomysGK
{
    public static void main(String[] args)
    {
        //Vraag 5.2 - 1 punt
        //RoomysBestuurder in korrekte posisie geskep
        RoomysBestuurder rb = new RoomysBestuurder();

        //Vraag 5.3 - 1 punt
        //vertoon RoomysBestuurder-objek
        System.out.println(rb.toString());
        System.out.println();

        //Vraag 5.4 - 1 punt
        //vertoon gemiddelde prys vir ekstra roomyse met allergene
        System.out.println("Ekstra Roomyse met Allergene se
            Gemiddelde Prys: " + rb.getGemiddeldeEkstra());
        System.out.println();

        //Vraag 6.3 - 1 punt
        //roep die vervaldeRoomyse-metode
        rb.vervaldeRoomyse();

        //Vraag 7.2 - 2 punte
        //roep die verwyderRoomyse-metode
        //vertoon RoomysBestuurder-objek
        rb.verwyderRoomyse();
        System.out.println(rb.toString());
    }
}
```

Totaal: 150 punte