



Benodigdhede vir hierdie vraestel:

Multikeusekaarte: ☐
Grafiekpapier: ☐

Nie-programmeerbare sakrekenaar: ☒
Draagbare rekenaar: ☐

Oopboek eksamen: ☒

EKSAMEN:

EXAM:

MODULEKODE:

MODULE CODE:

MODULEBESKRYWING:

MODULE DESCRIPTION:

EKSAMEN 1E GEL

EXAM 1ST OPP

REII313

**OBJEKGEORIËNTEERDE
SAGTEWAREONTWIKKELING**

**OBJECT ORIENTED SOFTWARE
DEVELOPMENT**

KWALIFIKASIE: B.Ing.

QUALIFICATION:

DUUR: 180 minute

DURATION: 180 minutes

MAKS: 65 punte

MAX: 65 marks

EKSAMINATOR:

EXAMINER:

MODERATOR:

AJ Alberts

PJ Joubert

DATUM:

DATE:

TYD:

TIME:

14-06-2022

14:00

Hierdie is 'n praktiese, oopboek, maar toe bewussein toets. Gebruik C++ en die Qt sagte-wareeraamwerk om die praktiese vraag te beantwoord. Lees eers al die vrae om die geheelbeeld te kry, aangesien hulle op mekaar bou. Antwoord die teorievraag elektronies en handig dit saam met die praktiese vraag elektronies in.

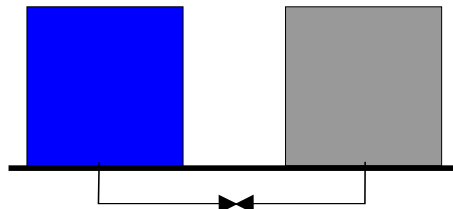
This is a practical, open book, but closed consciousness test. Use C++ and the Qt software framework to answer the practical question. First read all the questions to get the big picture, as they build on one another. Answer the theory question electronically and hand it in electronically with the practical question.

Vraag 1 / Question 1

Die doelwit van hierdie vraag is om 'n eenvoudige visuele simulatie te skep wat die watervlak in tenks wat aan mekaar verbind is te vertoon soos in die diagram, terwyl die stelsel balanseer en ewewig bereik. Ignoreer turbulensie en aanvaar lineêre vloeiweerstand.

The purpose of this question is to create a simple visual simulation that displays the water level in tanks connected to each other as shown in the diagram, while the system is balancing and reaching equilibrium. Ignore turbulence and assume linear flow resistance.

Beantwoord nou die volgende vrae deur kode te skryf / *Now answer the following questions by writing code*



- (a) Skep 'n grafiese program wat 'n nuwe venster oopmaak wat plek het vir die voorstelling van drie tenks en knoppies bevat om die krane tussen die tenks oop en toe te maak.

Create a graphical program that opens a new window that is large enough to contain the representation of three tanks and buttons to open and close the valves between the tanks.

(5)

- (b) Skep 'n grafiese element wat 'n tenk met sy watervlak visueel kan voorstel.
Create a graphical element that can visually represent a tank and its water level. (5)
- (c) Skryf kode wat twee tenks skep soos in die diagram, die eerste vol en die tweede leeg. Die kode moet die vlak in die tenks in 20ms intervalle uitwerk, opdateer en grafies ten toon stel, gebaseer op die watervlakke van die tenks. Kies jou konstantes so dat die eerste kwart van die inhoud van die vol tenk na die leë tenk oorvloeï binne vyf sekondes. Wenk: die vloeitempo in die pyp moet lineêr verband hou met die drukverskil oor die pyp.
Write code that will create two tanks as in the diagram, the first full and the second empty. The code must calculate, update and graphically show the water level in the tanks in 20ms intervals. Choose your constants so that the first quarter of the contents of the full tank flows to the empty tank within 5 seconds. Hint: the flow rate in the pipe must be linearly related to the pressure difference over the pipe. (8)
- (d) Brei die simulatie uit na drie tenks. Koppel die derde leë tenk met sy eie pyp en kraan aan die tweede tenk aan die regterkant, wat aanvanklik leeg is. Die pype tussen die tenks moet dieselfde vloei-eienskappe hê. Die tenks moet ewe groot wees en op dieselfde vlak staan.
Expand the simulation to three tanks. Connect the third empty tank with its own pipe and valve to the second tank on the right, that is initially empty. The pipes between the tanks must have the same flow properties. The tanks must be the same size and stand on the same level. (5)
- (e) Voorsien die eerste tenk aan die linkerkant van 'n inlooppyp met 'n kraan wat kan oop- en toemaak, wat konstant kan vloei teen 5% van die tenk se volume per sekonde. Voorsien die derde tenk aan die regterkant van 'n dreinkraan wat kan oop- en toemaak en vloei teen 5% van die tenk se volume per sekonde wanneer die tenk half is. Al die tenks moet kan oorloop wanneer hulle vol is.
Provide the first tank on the left with an inflow pipe and a valve that can open and close, that can constantly flow at a rate of 5% of the tank's volume per second. Provide the third tank on the right with a drain valve that can open and close and flows at 5% of the tank's volume per second when the tank is half. All the tanks must be able to overflow when they are full. (10)
- (f) Demonstreer jou program met 'n skermopname video sonder klank. Maak seker alles wat punte werd is word gedemonstreer. As jy net twee tenks het, demonstreer dit. As jy die simulatie uitgebrei het na drie tenks toe, demonstreer net dit, aangesien dit reeds die twee tenks geval bevat.
Demonstrate your program with a screen recording video without sound. Make sure everything that is worth marks is demonstrated. If you only have two tanks, demonstrate that. If you expanded the simulation to three tanks, demonstrate only that, since it already contains the two tank case. (2)

Totaal vir Vraag 1/ Total for Question 1: **35**

Vraag 2 / Question 2

Bespreek hoe jy te werk sal gaan om 'n objekgeoriënteerde ontwerp te doen van 'n netwerkgebaseerde rekenaarspeletjie waar 'n hoeveelheid spelers, insluitend 'n moontlike algoritmiese speler, teen mekaar kaart speel. Spelers moet eie identiteite hê en die stelsel moet rekord hou van wie het hoeveel keer gewen. Druk die argitektuur van jou objekgeoriënteerde ontwerp duidelik uit in 'n UML diagram en bespreek kern ontwerpskeuses, asook hoe die netwerk kommunikasie gaan werk.

Verduidelik hoe die lewenssiklus van die sagtewareprodukt sal verloop van aanvang tot af-trede en beskryf hoe jy die ontwikkelingsproses gaan benader.

Lewer dan kritiese kommentaar oor die ontwerp en bespreek watter dele van die ontwerp die belangrikste is om te optimeer vir maksimum verwerkingseffektiwiteit en of en hoekom objekgeoriënteerde ontwerp 'n goeie manier is om die probleem aan te spreek.

Discuss how you will proceed with an object oriented design of a network based computer game where a number of players, including a possible computer based player, play cards against each other. Players must have own identities and the system must keep track of who has won how many times. Express the architecture of your object oriented design clearly in a UML diagram and discuss core design decisions, as well as how the network communication is going to work.

Explain how the life cycle of the software product will proceed from inception to retirement, and describe how you will approach the development process.

Critically comment on the design and discuss which parts of the design are the most important to optimize for maximum processing efficiency and if and why object oriented design is a good way to address the problem.

(30)

Totaal vir Vraag 2/ Total for Question 2: **30**

TOTAAL / TOTAL : 65

MAKS / MAX : 65