

T.H. Rijswijk

KLAS(SEN) : EH38 (C&D)	BLAD : 1 van 4 BLADEN
OND. DEEL : Software Ontwikkeling en Programmeren 3	DOCENT : Harry Broeders
AFK. O.D. : SOPX3C1	DATUM : 27 oktober 2003
KWART. : 1	TIJD : 8 + 9
TYPE : toets	

Tijdens dit tentamen mogen **alle** boeken, dictaten, aantekeningen enz. worden gebruikt.

Bij elke opgave staat tussen haakjes het maximale aantal te behalen punten vermeld.

Eindcijfer = (aantal behaalde punten + 10) / 10.

- 1.** In een C++ programma is de class `TcpClient` gedefinieerd. Met behulp van deze class kan een TCP/IP verbinding worden opgebouwd met een server.

```
class URL {
// ... rest niet gegeven.
};

class TcpError {
// ... rest niet gegeven.
};

class TcpClient {
public:
    connect(const URL& url, int port);
    disconnect();
    send(const string& s) const;
    receive(string& s) const;
// ... rest niet gegeven.
};
```

Verder is gegeven dat alle memberfuncties, behalve `disconnect`, van `TcpClient` de exception `TcpError` kunnen gooien. Een functie om de pagina `index.htm` van een http server op te vragen is als volgt geïmplementeerd:

```
string getIndex(const URL& url) {
    string response;
    try {
        TcpClient client;
        client.connect(URL, 80);
        client.send("GET index.html HTTP/1.0\n\n");
        client.receive(response);
    }
    catch (TcpError& e) {
        client.disconnect();
        throw;
    }
    client.disconnect();
    return response;
}
```

- A. (5) Kan de functie `getIndex` een exception gooien? Zo ja, welke? Zo nee, waarom niet?

Zie volgende blad ⇨

T.H. Rijswijk

KLAS(SEN) : EH38 (C&D)	BLAD : 2 van 4 BLADEN
OND. DEEL : Software Ontwikkeling en Programmeren 3	DOCENT : Harry Broeders
AFK. O.D. : SOPX3C1	DATUM : 27 oktober 2003
KWART. : 1	TYPE : toets
	TIJD : 8 + 9

Tijdens dit tentamen mogen **alle** boeken, dictaten, aantekeningen enz. worden gebruikt.

- B. (5) Wat is het belangrijkste voordeel van het gebruik van `throw;` in plaats van `throw TcpError();`
- C. (15) Het opvangen van de `TcpError` exception in de functie `getIndex` alleen maar om de `disconnect` memberfunctie van `TcpClient` aan te roepen is onhandig. Door gebruik te maken van “resource acquisition is initialisation” is een handigere oplossing mogelijk. Het is dan niet meer nodig om een `catch` blok in de functie op te nemen en toch kan gegarandeerd worden dat de `disconnect` memberfunctie van `TcpClient` voor het verlaten van de functie **altijd** aangeroepen wordt. **Geef deze “handige” oplossing en herschrijf de functie `getIndex`.**

2. Een docent heeft de resultaten van een tentamen in een C++ programma in een list opgeslagen. Een deel van dit programma is hieronder gegeven.

```
#include <list>
#include <vector>
#include <algorithm>
#include <functional>
#include <iostream>

using namespace std;

void main() {
    list<double> resultaat;
    resultaat.push_back(1.2);
    resultaat.push_back(6.6);
    resultaat.push_back(6.7);
    resultaat.push_back(4.1);
    resultaat.push_back(9.0);
    //...
    vector<double> aangepast;
    //...
}
```

- A. (10) Schrijf een stukje C++ code waarmee de inhoud van de list `resultaat` op het scherm wordt afgedrukt. De getallen moeten zijn gescheiden door een spatie. **Maak indien mogelijk gebruik van een standaard algoritme en van een stream iterator.**

Een student dient een klacht in tegen dit tentamen (de vragen hebben niets te maken met de behandelde theorie). De examencommissie, die deze klacht behandelt, stelt de student in het gelijk en beslist dat alle studenten die een cijfer < 6.0 hebben gehaald als vervangend resultaat een 6.0 krijgen. Een student die een cijfer >= 6.0 heeft gehaald, behoudt zijn cijfer. De docent

Zie volgende blad ⇨

T.H. Rijswijk

KLAS(SEN) : EH38 (C&D)	BLAD : 3 van 4 BLADEN
OND. DEEL : Software Ontwikkeling en Programmeren 3	DOCENT : Harry Broeders
AFK. O.D. : SOPX3C1	DATUM : 27 oktober 2003
KWART. : 1	TIJD : 8 + 9
TYPE : toets	

Tijdens dit tentamen mogen **alle** boeken, dictaten, aantekeningen enz. worden gebruikt.

heeft in zijn C++ programma een vector `aangepast` gedefinieerd waarin de aangepaste resultaten moeten komen. De docent komt na het bestuderen van de standaard library tot de conclusie dat hij gebruik kan maken van het standaard `replace_copy_if` algoritme. Een deel van de documentatie van dit algoritme kun je vinden op pagina 4 van dit tentamen.

- B. (5) Wat is de orde van dit algoritme? **Verklaar je antwoord!**
- C. (5) Hoe kun je zien dat je dit algoritme kunt toepassen op een list?
- D. (15) Schrijf een stukje C++ code waarmee de getallen uit de list `resultaat` naar de vector `aangepast` worden gekopieerd waarbij een getal < 6.0 vervangen wordt door het getal `6.0`. Maak gebruik van het `replace_copy_if` algoritme en van een standaard comparator functie object.

3. Gevraagd wordt een beveiligingssysteem te ontwikkelen. Het systeem meldt en voorkomt ongeautoriseerde toegang tot het gebouw of tot een gedeelte van het gebouw. Het systeem bewaakt brandmelders en slaat alarm bij brand. De gedetailleerde functies van het systeem zijn:

- Bewaken en besturen van het betreden en verlaten van het gebouw door de gebruikers. Deze gebruikers zijn in het bezit van een persoonlijke kaart met machineleesbare strip. Het bewaken betreft de normale in- en uitgangen van het gebouw en speciale zones in het gebouw. Het systeem bestuurt de deursloten en zal een deur alleen openen als een goede kaart door een bij de deur aangebrachte kaartlezer is gehaald.
- Detecteren van inbraak via ramen en deuren. Zowel ramen en deuren aan de buitenzijde van het gebouw, als ramen en deuren van de veilige zones worden bewaakt. Het systeem moet in geval van inbraak de betreffende zone of een omliggende zone afsluiten om te verhinderen dat de binnendringers het gebouw of de zone verlaten.
- Op strategische plaatsen in het gebouw staan branddetectoren. Als een brand wordt gedetecteerd moet een passend alarm afgaan. In geval van brand zal het systeem noodevacuatie uit de betreffende zones mogelijk maken. Noodevacuatie betekent onbeperkt verlaten van de betreffende zone maar geen toegang tot die zone. In de betreffende zones loeien sirenes om de bewoners aan te geven dat de zone verlaten moet worden.

- A. (20) Bij een eerste brainstormsessie zijn de volgende classes gevonden: `Gebouw`, `Zone`, `Deur`, `KaartLezer`, `Kaart`, `Sirene`, `BrandDetector` en `InbraakDetector`. Teken een UML klassediagram waarin de associaties tussen deze classes worden aangegeven. Neem alle informatie die beschikbaar is betreffende de associaties in het klassediagram op. **Het is dus niet nodig om operaties en attributen in de classes op te nemen.**
- B. (10) Teken een UML sequencediagram dat modelleert hoe het afgaan van een branddetector wordt afgehandeld.

Succes!

T.H. Rijswijk

KLAS(SEN) : EH38 (C&D)	BLAD : 4 van 4 BLADEN
OND. DEEL : Software Ontwikkeling en Programmeren 3	DOCENT : Harry Broeders
AFK. O.D. : SOPX3C1	DATUM : 27 oktober 2003
KWART. : 1	TIJD : 8 + 9
TYPE : toets	

Tijdens dit tentamen mogen **alle** boeken, dictaten, aantekeningen enz. worden gebruikt.

Informatie over het `replace_copy_if` algoritme. Copyright © 1993-2003 Silicon Graphics, Inc.

Prototype

```
template <class InputIterator, class OutputIterator,  
         class Predicate, class T>  
OutputIterator replace_copy_if(  
    InputIterator first, InputIterator last,  
    OutputIterator result, Predicate pred,  
    const T& new_value  
)
```

Description

`replace_copy_if` copies elements from the range `[first, last)` to the range `[result, result + (last-first))`, except that any element for which `pred` is true is not copied; `new_value` is copied instead. More precisely, for every integer `n` such that $0 \leq n < \text{last} - \text{first}$, `replace_copy_if` performs the assignment `*(result+n) = new_value` if `pred(*(first+n))`, and `*(result+n) = *(first+n)` otherwise.

Complexity

Linear.